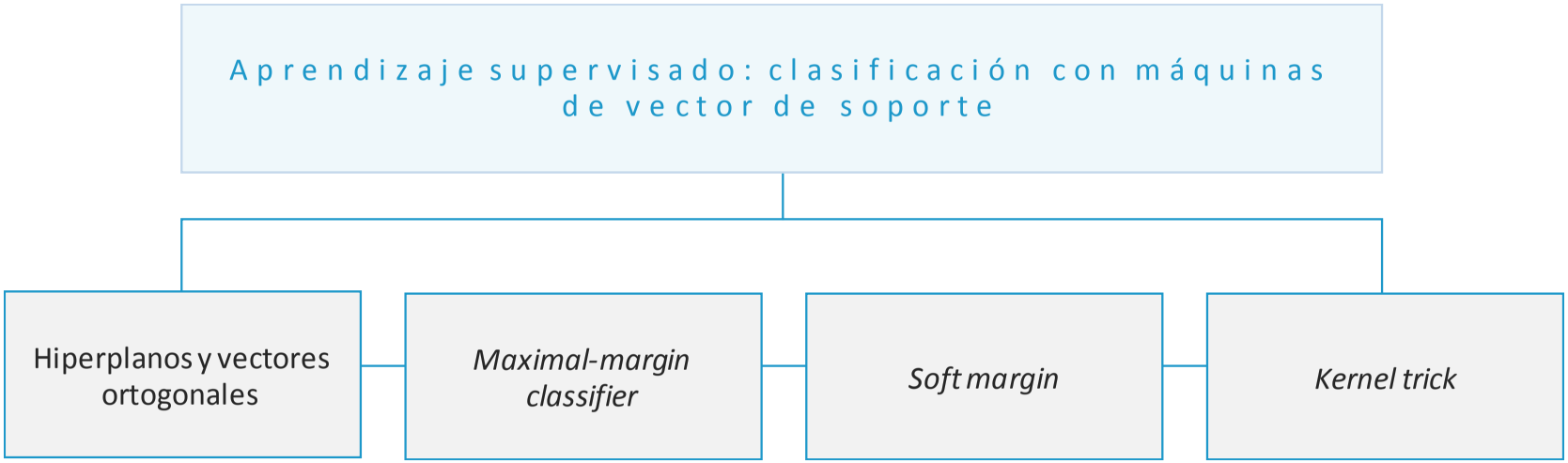
Aprendizaje Automático

Aprendizaje supervisado: clasificación con máquinas vector de soporte

Índice

Esquema



Ideas clave

5.1. ¿Cómo estudiar este tema?

Estudia este tema a través de las **Ideas clave** disponibles a continuación.

En este tema vamos a estudiar las máquinas de vector de soporte y cómo se pueden aplicar en problemas de clasificación dentro del aprendizaje supervisado.

Las **máquinas de vector de soporte** han sido el primer modelo que utilizaba un enfoque no probabilístico para realizar la estimación de datos no observados.

- En primer lugar, vamos a ver los conceptos geométricos necesarios para entender el funcionamiento y la base que hay por detrás de las máquinas de vector de soporte.
- En segundo lugar, veremos los problemas existentes al separar por hiperplanos.
- A continuación, veremos cómo solucionar el problema de separar por hiperplanos.
- Finalmente, las técnicas conocidas como *kernel trick*.

5.2. Introducción a las máquinas de vector de soporte: hiperplanos

Las máquinas vectores de soporte conocidas como *support vector machines* fueron inventadas por el científico de origen ruso Vladimir Vapnik y su equipo en el año 1963. La gran contribución de Vapnik al campo del aprendizaje automático ha consistido precisamente en proponer uno de los **primeros modelos**

de estimación/predicción que no está basado en ningún modelo probabilístico, lo cual es un gran cambio de mentalidad. Puesto que todos los modelos que se usaban hasta ese momento estaban basados en la teoría estadística y de probabilidad.

Sin embargo, las **máquinas de vector de soporte** (SVM) se pueden considerar el primer modelo que utiliza un enfoque completamente distinto y que está **basado en un modelado geométrico** y que se resuelve mediante un problema de optimización con restricciones.

El objetivo de las redes de vectores de soporte es buscar un plano que separe las clases en *feature space*. Por *feature space* se entiende un nuevo espacio de dimensiones diferente (por lo general con un mayor número de dimensiones) al espacio original.

Debido a que este objetivo de buscar un plano que separe las clases por lo general no es posible obtener, en las SVM se proponen las siguientes modificaciones:

1. Relajar la definición de «separar».
2. Mejorar y enriquecer el *feature space* para que la separación sea posible.

Antes de entrar en detalle en el funcionamiento de las máquinas de vector de soporte es necesario definir los conceptos de hiperplanos y vectores ortogonales.

Hiperplanos

En un espacio de dos dimensiones el hiperplano es una recta. Pero, un hiperplano se puede definir para un espacio de *p* dimensiones. La ecuación general del hiperplano para un espacio de p dimensiones es:

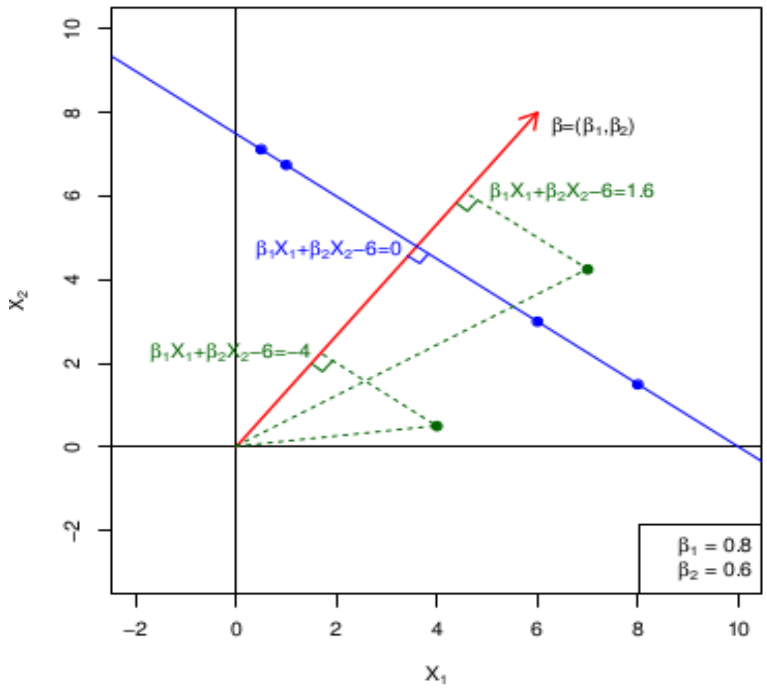
$$\beta_o + \beta_1 X_1 + \beta_2 X_2 + \cdots + B_p X_p = 0$$

donde el vector:

$$\beta = (\beta_1, B_2, \ldots, B_P)$$

Se llama vector normal, y β es un vector unitario en el cual la suma de los cuadrados es 1. Este vector apunta a una dirección ortogonal a la superficie del hiperplano (vector de color rojo de la figura 1.).

Si proyectamos cada uno de los puntos sobre el vector normal, los puntos que caen sobre el hiperplano tienen valor 0 al proyectarse. Los puntos por encima del hiperplano tienen un valor positivo, los puntos por debajo del hiperplano un valor negativo y además estos valores son mayores cuanto más lejanos están del hiperplano. Es decir, el valor que se obtiene al proyectar los puntos sobre el vector normal es proporcional a la distancia de los puntos al hiperplano. Esta característica geométrica hace posible el uso de las máquinas de vector de soporte para buscar los patrones necesarios.



Gráfica 1. Representación de un hiperplano, un vector normal y la distancia de los puntos proyectados en el vector normal. Recuperado de: https://lagunita.stanford.edu/courses/HumanitiesSciences/StatLearning/Winter2016/about

La cuestión es que el **hiperplano** busca separar dos conjuntos de puntos en dos regiones distintas. Por ejemplo, en un problema de clasificación el hiperplano puede

separar aquellos puntos que corresponden a personas con un tumor determinado de aquellas personas sanas.

Sin embargo, para un conjunto de puntos determinados existen múltiples hiperplanos posibles que nos separan los puntos en dos regiones.

El objetivo de las máquinas de vector de soporte es buscar un hiperplano que separe las clases en *feature space.* Por *feature space* se entiende un nuevo espacio de dimensiones diferente al espacio original.

5.3. Separando por hiperplanos

Dado una serie de puntos en un espacio geométrico que se desea clasificar, se puede establecer que aquellos puntos que al proyectar sobre el vector normal a un hiperplano determinado sean *> 0,* correspondan a una clase y aquellos que sean *< 0,* a otra clase.

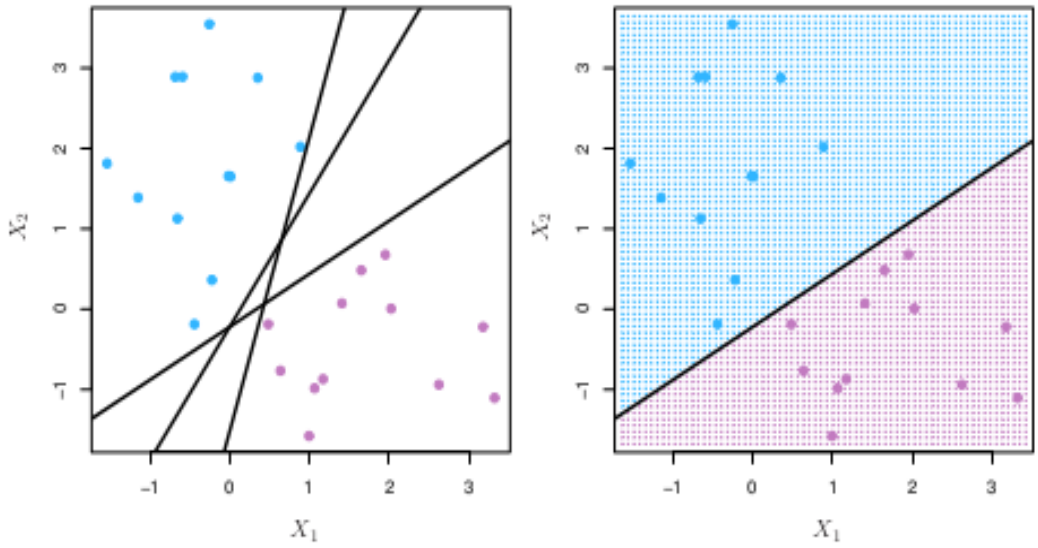
Es decir, de forma matemática:

$$Si\ f(X) = \beta_0 + \beta_1 X_1 + B_2\, X_2 + \cdots + B_p\, X_P\ entonces$$

$f(X) > 0$ establece los puntos en un lado del hiperplano

$f(X) < 0$ en el otro

Sin embargo, lo habitual es encontrarse en la situación en la cual existen múltiples hiperplanos posibles, y por tanto es necesario decidir ¿cuál de ellos establecer?



Gráfica 2. Ejemplo de tres hiperplanos posibles para separar dos conjuntos de clase (izquierda). En el caso de elegir un hiperplano los puntos se clasifican en función de cada una de las regiones sombreadas de rojo y azul (derecha). Fuente: James et.al., 2013.

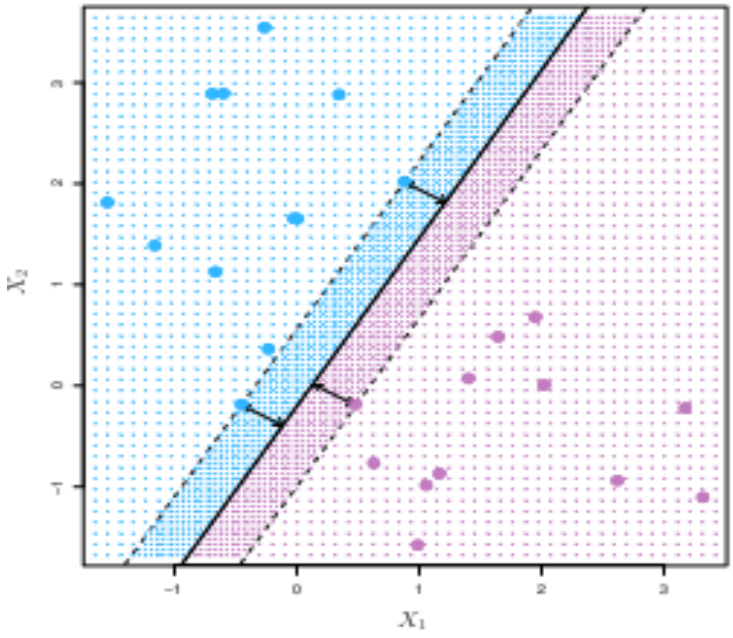
5.4. *Maximal-margin classifier*

En el caso de un problema de clasificación binaria, de todos los hiperplanos posibles es necesario buscar aquel que nos proporciona la mayor diferencia entre las dos clases, lo cual se traduce en la mayor distancia entre los puntos que pertenecen a una clase y a otra. La hipótesis que hay detrás de esto, es porque suponemos que este hiperplano será el que tendrá una mayor distancia en el conjunto de test y en las predicciones futuras.

Esta situación se puede modelar como un problema de optimización con restricciones donde es necesario maximizar el margen y, matemáticamente, se define:

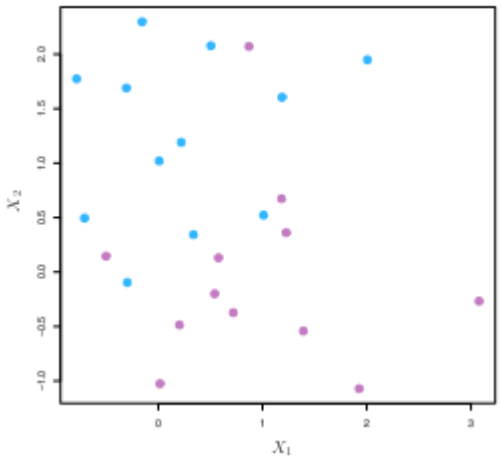
$$maximize\ M$$
$$\beta_0, \beta_1, \dots, \beta_p$$
$$subject\ to \sum_{j=1}^{p} \beta_j^2 = 1$$
$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) \geq M$$
$$for\ all\ i = 1, \dots, N$$

En la siguiente imagen se muestra de forma gráfica el concepto donde la línea continua resaltada es el *maximal-margin classifier,* y se muestran dos bandas con líneas discontinúas que contienen la distancia del hiperplano a los primeros puntos de cada una de las clases, el objetivo es maximizar la distancia de estas dos bandas.



Gráfica 3. Ejemplo del hiperplano con una separación óptima entre dos conjuntos de puntos de clases diferentes. Fuente: James et.al., 2013.

El principal problema con esta separación óptima del hiperplano es que **los datos habitualmente no son linealmente separables con una recta** en el caso de un espacio de dos dimensiones, como se observa en la siguiente gráfica, donde es imposible separar los puntos de una clase de los de otra.



Gráfica 4. Ejemplo de un espacio de dos dimensiones donde los puntos no son linealmente separables. Fuente: James et.al., 2013.

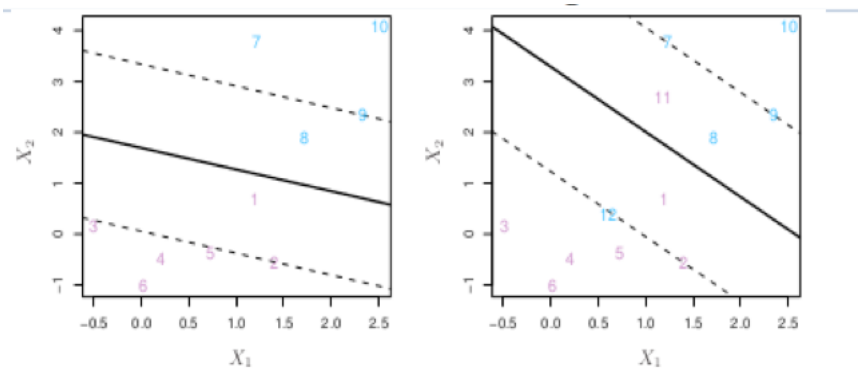
Esta situación produce una solución pobre para el clasificador *maximal-margin.* Por tanto, el clasificador vector de soporte maximiza un *soft margin***.**

5.5. *Soft margin*

La solución al problema anterior en el cual los puntos no son linealmente separables es maximizar un *soft margin.* Matemáticamente el problema se define de la siguiente forma:

$$maximize\ M$$
$$\beta_0, \beta_1, \dots, \beta_p, \epsilon_1, \dots, \epsilon_n$$
$$subject\ to \sum_{j=1}^{p} \beta_j^2 = 1$$
$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) \geq M(1 - \epsilon_i)$$
$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$

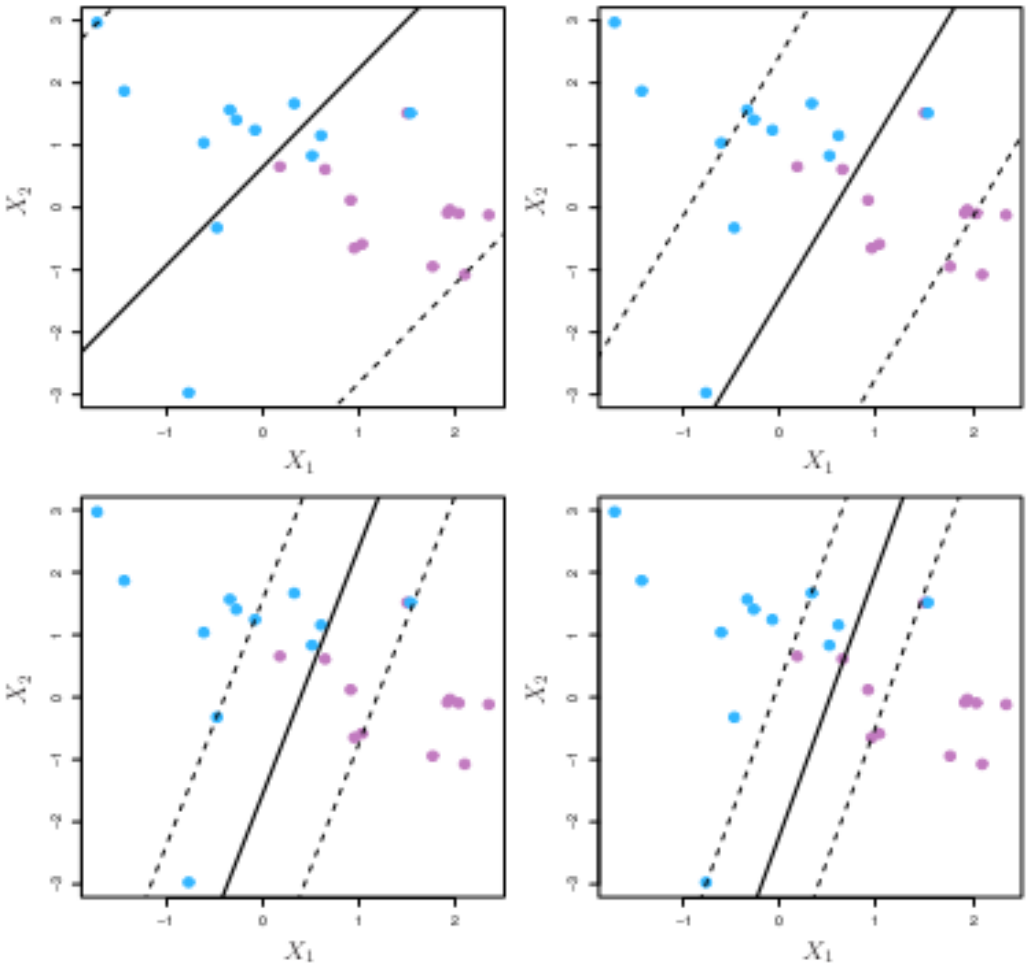
De forma gráfica, en la siguiente figura se observa:



Gráfica 5. Ajuste de un clasificador de vector de soporte a pocos datos. El hiperplano se muestra con una línea sólida y los márgenes con una punteada. La imagen de la derecha muestra el efecto cuando aparecen dos nuevas observaciones (11 y 12) las cuales están en el lado incorrecto del hiperplano y de los márgenes. Fuente: James et.al., 2013.

Modificando el parámetro *C* de la ecuación anterior, el cual se conoce como función de coste se puede hacer el margen más grande o pequeño. En concreto un valor de *C* más grande hace el margen más pequeño y a la inversa un valor *C* más pequeño hace el margen más grande.

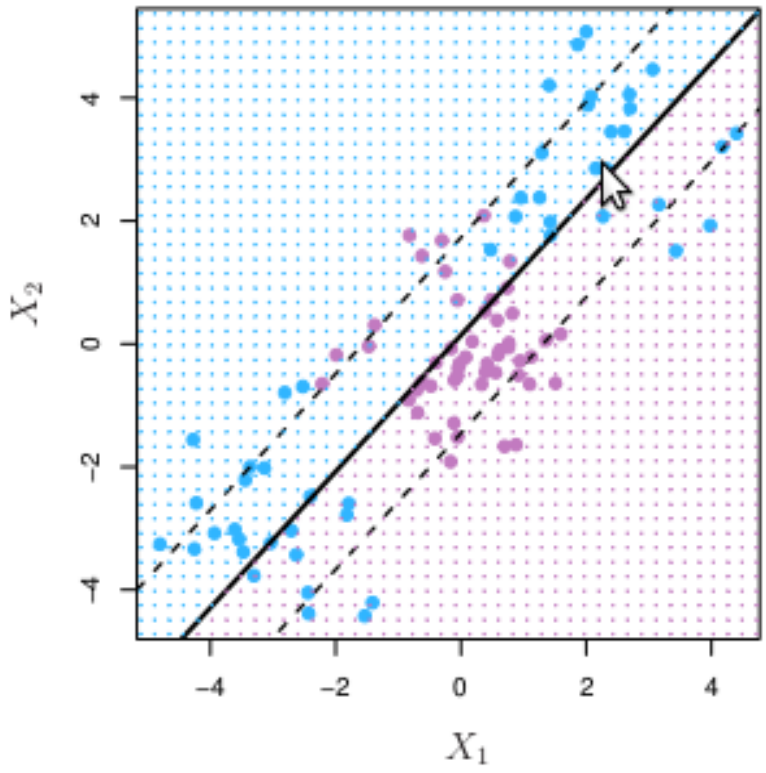
En la siguiente figura se muestra un hiperplano y los márgenes correspondientes en función de diferente valor de *C.*



Gráfica 6. Diferentes fronteras de decisión en los márgenes de una *soft-margin* en función de diferentes valores de C y utilizando los mismos puntos y el mismo hiperplano. El margen más grande es el de la gráfica superior izquierda y el más pequeño el de la inferior derecha. Fuente: James et.al., 2013.

Problema con las fronteras lineales

Muchas veces las fronteras lineales siguen sin funcionar, independientemente del valor de coste *C* que se utilice. Por ejemplo, en la siguiente gráfica se muestra un problema donde la separación necesaria entre los puntos de una clase y la otra es no lineal.



Gráfica 7. Ejemplo donde no es posible una separación lineal entre dos clases. Fuente: James et.al., 2013.

Expansión de variables

Para solucionar este problema donde no es posible realizar una separación entre dos clases, una opción es **incrementar el espacio de las variables por medio de transformaciones,** es decir pasar de un espacio **p-dimensional** a un espacio de D dimensiones $\mathrm{D} > \mathrm{p}$**.** Una vez realizada esta transformación, se debería ajustar un clasificador de vector de soporte en este nuevo espacio. El **objetivo es obtener fronteras de decisión no lineales sobre el espacio original.**

Por ejemplo, si tenemos datos de entrada en dos dimensiones (X_1, X_2) es posible utilizar el siguiente espacio de 6 dimensiones: $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$ siendo por tanto la frontera de decisión:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + B_5 X_1 X_2 = 0$$

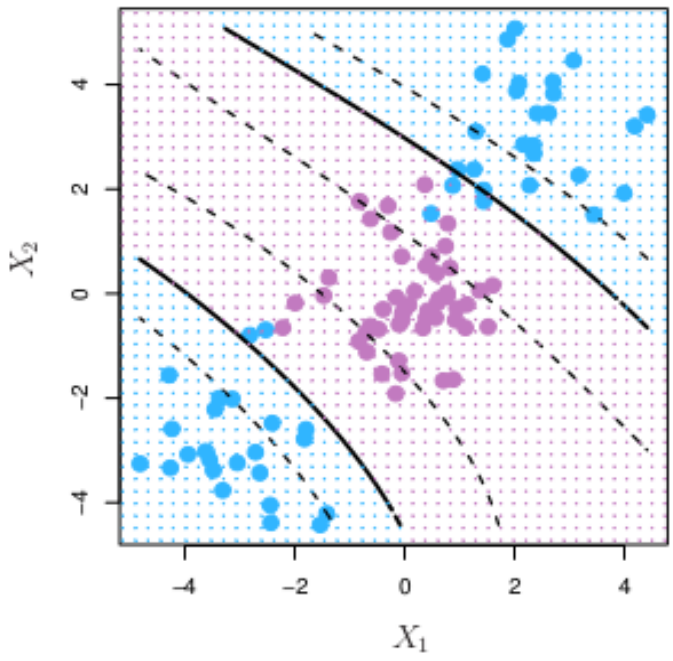
Esta transformación conlleva fronteras de decisión no-lineales en el espacio original.

Polinomios cúbicos

Utilizando una transformación en forma de polinomios cúbicos se puede pasar de 2 a 9 variables. De esta forma el clasificador en el nuevo espacio soluciona el problema de buscar los patrones en el espacio con menor dimensiones.

En la siguiente gráfica se muestra el resultado de las fronteras de separación obtenidas por medio de una transformación en polinomios cúbicos., resultado del cálculo de la siguiente ecuación:

$$B_o + B_1X_1 + B_2X_2 + B_3X_1^2 + B_4X_2^2 + B_5X_1X_2 + B_6X_1^2 + B_7X_2^3 + B_8X_1X_2^2 + B_9X_1^2X_2 = 0$$



Gráfica 8. Ejemplo de las fronteras de decisión no lineales obtenidas por medio de una transformación obtenida con polinomios cúbicos. Fuente: James et.al., 2013.

5.6. *Kernel trick*

Las expansiones de las variables con polinomios, especialmente aquellos con grandes dimensiones, son computacionalmente costosos. Existe una solución más elegante y controlada de introducir no-linealidad que es utilizada en las máquinas vector de soporte por medio del uso de *kernels.* El *kernel* de un operador A denotado por *ker* es el conjunto de todos los vectores cuya imagen sea el vector nulo:

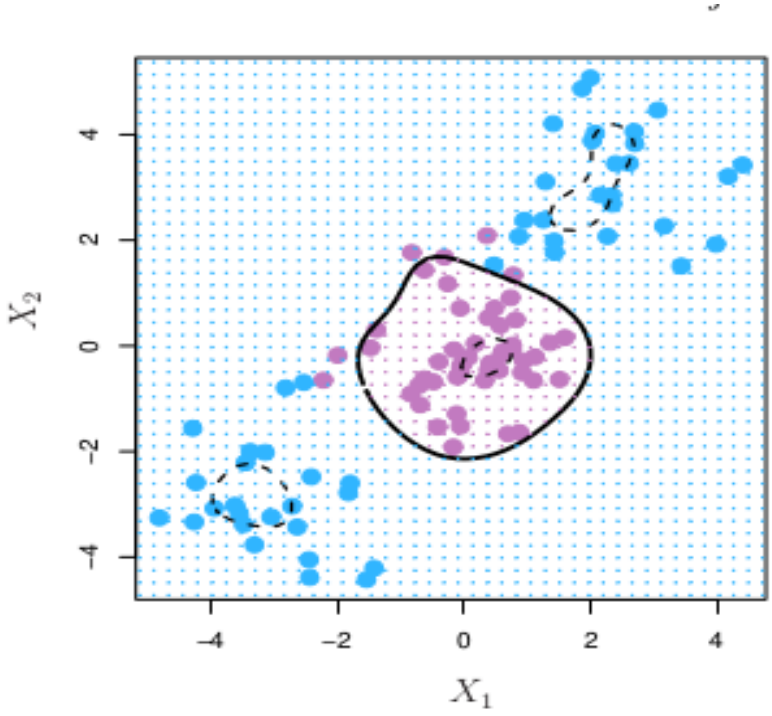
$$\ker A = \{\underset{v}{\rightarrow} \in V : A \underset{v}{\rightarrow} = 0\}$$

Los *kernels* se apoyan en concepto del producto vectorial de los vectores de soporte. Se trata de funciones que reciben dos vectores como parámetros. Uno de los *kernels* más utilizados es el de base radial que asume que el *feature space* es de altas dimensiones y se define con la siguiente función matemática:

$$K(x_i, x_i') = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p X_{ip}) \geq M$$

El uso de *kernels* permite obtener fronteras de decisión no lineales por medio de transformaciones matemáticas sin necesidad de tener que realizar transformaciones con polinomios. En la siguiente figura se muestran las fronteras de decisión no lineales obtenidas por medio del uso de un *kerne*l de base radial.



Gráfica. Ejemplo de las fronteras de decisión no lineales obtenidas por medio del uso de un *kernel* de base radial. Fuente: James et.al., 2013.

5.7. Referencias bibliográficas

Hastie, T., Tibshirani R., Friedman, J. (2011) *The Elements of Statistical Learning.* Second edition. Springer.

James, G., Witten, D., Hastie, T and Tibshirani, R. (2013) *An Introduction to Statistical Leaerning with Applications in R.* Springer.

Lo + recomendado

No dejes de leer

Support Vector Machines

Scikit learn. (s. f.). Support vector Machines.

Documentación del uso de las máquinas de vector de soporte utilizando la librería scikit-learn en Python

Accede a la página a través del aula virtual o desde la siguiente dirección web:
http://scikit-learn.org/stable/modules/svm.html

Ejemplo de SVM en Python utilizando LibSVM, SVMLight y Pegasos

Jupyter nbviewer. (s. f.) Data Preprocessing.

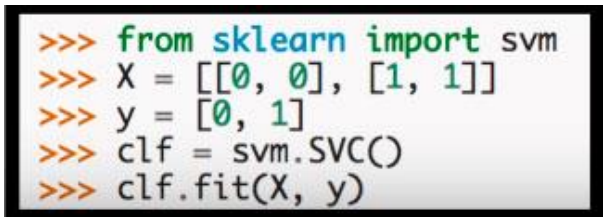
Ejemplo de rendimiento y precisión de las librerías LibSVM, SVMLight y Pegasos en Python.

Accede a la página a través del aula virtual o desde la siguiente dirección web:
http://nbviewer.jupyter.org/gist/nipunreddevil/5153583

No dejes de ver

How Support Vector Machines work/how to open a black box

Explicación del funcionamiento de los máquinas de vector de soporte por Brandon Rohrer, *data scientist* en Facebook.



Accede al vídeo a través del aula virtual o desde la siguiente dirección web:

https://www.youtube.com/watch?v=-Z4aojJ-pdg

+ Información

A fondo

Support-Vector Networks

Cortes, C. Y Vapnik, V. (1995). Support-Vector Networks. *Machine Learning,* 20, 273-297.

Artículo de Vladimir Vapnik sobre las máquinas de vector de soporte.

Accede al artículo a través del aula virtual o desde la siguiente dirección web:

http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf

Test

1. Cuáles de las siguientes afirmaciones sobre las máquinas de vector de soporte son ciertas:

   A. Su objetivo es buscar un plano que separe las clases en *feature space.*

   B. Modifican la definición de separación.

   C. Mejoran y enriquecen el *feature space* original.

2. Cuáles de las siguientes afirmaciones sobre los hiperplanos son ciertas:

   A. Un hiperplano solo se define para 2 dimensiones.

   B. En un espacio de 2 dimensiones el hiperplano es una recta.

   C. Un hiperplano está definido para *n* dimensiones.

3. El vector normal:

   A. Puede apuntar a una dirección ortogonal al hiperplano.

   B. Debe apuntar a una dirección ortogonal al hiperplano.

   C. No requiere ningún tipo de ortogonalidad.

4. Cuáles de las siguientes afirmaciones son ciertas:

   A. Para una serie de datos de entrenamiento de un problema binario existen múltiples hiperplanos posibles.

   B. Solo existe un hiperplano posible para cada uno de los conjuntos de datos.

   C. Los hiperplanos hay que crearlos únicamente para realizar predicciones.

5. Indique las afirmaciones ciertas sobre un *maximal classifier* en una clasificación binaria:

A. Proporciona la mayor diferencia *(gap)* entre las instancias de cada uno de los ejemplos.

B. Proporciona información sobre cuanto margen de mejora existe.

C. Habitualmente los datos no son fácilmente separables por una recta y el modelo falla.

6. De las siguientes afirmaciones *soft-margin* indique cuáles son correctas:

A. Reduce de forma significativa el margen disponible para trabajar.

B. Utiliza desarrollo software para poder ejecutarse.

C. Permite separar puntos que no son linealmente separables.

7. Cuáles de las siguientes afirmaciones sobre la función de coste son apropiadas:

A. Cuanto más grande es el parámetro de coste mayor es el margen.

B. Cuanto más pequeño sea el parámetro de coste mayor es el margen.

C. La función de coste sirve para determinar el balance coste-beneficio de utilizar un modelo de clasificación.

8. La expansión en forma de polinomios:

A. Tiene un alto coste computacional.

B. Permite obtener fronteras de decisión no lineales sobre el espacio original.

C. Ninguna de las anteriores es correcta.

9. Cuáles de las siguientes afirmaciones son una ventaja de los *kernels:*

A. Permiten utilizar un menor volumen de información.

B. El coste computacional es menor al utilizar *kernels.*

C. Los *kernels* permiten obtener fronteras de decisión no lineales.

10. Cuáles de las siguientes afirmaciones sobre los *kernels* son ciertas:

A. Se apoyan en el concepto de producto vectorial.

B. Se trata de funciones que reciben dos vectores como parámetro.

C. Se trata de funciones que reducen la información y pueden recibir cualquier parámetro.