Introducción

El método de clasificación-regresión Support Vector Machines (SVMs) fue desarrollado en la década de los 90, dentro del campo de la ciencia computacional.

Si bien originariamente se desarrolló como un método de clasificación binaria, su aplicación se ha extendido a problemas de clasificación múltiple y regresión. SVMs ha resultado ser uno de los mejores clasificadores para un amplio abanico de situaciones, por lo que se considera uno de los referentes dentro del ámbito de aprendizaje estadístico y machine learning.

El algoritmo SVM se fundamenta en el *Maximal Margin Classifier*, que a su vez, se basa en el concepto de *hiperplano*.

Hiperplano y Maximal Margin Classifier

- En un espacio p-dimensional, un hiperplano se define como un subespacio plano y afín de dimensiones $p-1$. El término afín significa que el subespacio no tiene por qué pasar por el origen.
- En un espacio de dos dimensiones, el hiperplano es un subespacio de 1 dimensión, es decir, una recta.
- En un espacio tridimensional, un hiperplano es un subespacio de dos dimensiones, un plano convencional. Para dimensiones $p>3$ no es intuitivo visualizar un hiperplano, pero el concepto de subespacio con $p-1$ dimensiones se mantiene.

- La definición matemática de un hiperplano es bastante simple. En el caso de dos dimensiones, el hiperplano se describe acorde a la ecuación de una recta:

$$\beta_0+\beta_1x_1+\beta_2x_2=0$$

Dados los parámetros β_0, β_1 y β_2, todos los pares de valores $x=(x_1,x_2)$ para los que se cumple la igualdad son puntos del hiperplano. Esta ecuación puede generalizarse para p-dimensiones:

$$\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_px_p=0$$

y de igual manera, todos los puntos definidos por el vector $(x=x_1,x_2,...,x_p)$ que cumplen la ecuación pertenecen al hiperplano.

Cuando x no satisface la ecuación:

$$\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_px_p<0$$

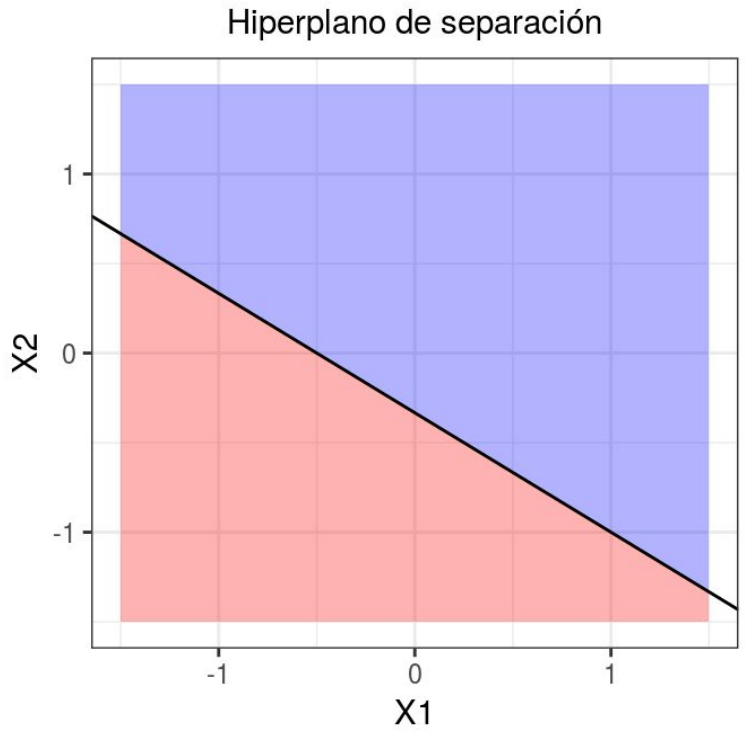
o bien

$$\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_px_p>0$$

el punto x cae a un lado o al otro del hiperplano. Así pues, se puede entender que un hiperplano divide un espacio p-dimensional en dos mitades. Para saber en qué lado del hiperplano se encuentra un determinado punto x, solo hay que calcular el signo de la ecuación.

La siguiente imagen muestra el hiperplano de un espacio bidimensional. La ecuación que describe el hiperplano (una recta) es $1+2x_1+3x_2=0$.

La región azul representa el espacio en el que se encuentran todos los puntos para los que 1+2x1+3x2>0 y la región roja el de los puntos para los que 1+2x1+3x2<0.



Clasificación binaria empleando un hiperplano

Cuando se dispone de n observaciones, cada una con p predictores y cuya variable respuesta tiene dos niveles (de aquí en adelante identificados como +1 y −1), se pueden emplear hiperplanos para construir un clasificador que permita predecir a qué grupo pertenece una observación en función de sus predictores. Este mismo problema puede abordarse también con otros métodos (regresión logística, LDA, árboles de clasificación…) cada uno con ventajas y desventajas.

Para facilitar la comprensión, las siguientes explicaciones se basan en un espacio de dos dimensiones, donde un hiperplano es una recta. Sin embargo, los mismos conceptos son aplicables a dimensiones superiores.

CASOS PERFECTAMENTE SEPARABLES LINEALMENTE

Si la distribución de las observaciones es tal que se pueden separar linealmente de forma perfecta en las dos clases (+1 y −1), entonces, un hiperplano de separación cumple que:

$$\beta 0+\beta 1x1+\beta 2x2+...+\beta pxp > 0, \text{ si } yi=1$$

$$\beta 0+\beta 1x1+\beta 2x2+...+\beta pxp < 0, \text{ si } yi=-1$$

Al identificar cada clase como +1 o −1, y dado que multiplicar dos valores negativos resultan en un valor positivo, las dos condiciones anteriores pueden simplificarse en una única:

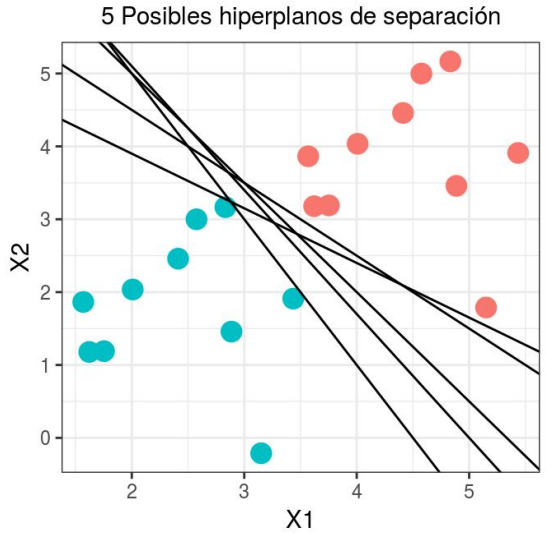
$$yi\ (\beta 0+\beta 1x1+\beta 2x2+...+\beta pxp) > 0, \text{ para } i=1...n$$

Bajo este escenario, el clasificador más sencillo consiste en asignar cada observación a una clase dependiendo del lado del hiperplano en el que se encuentre.

Es decir, la observación x^* se clasifica acorde al signo de la función : $f(x^*)=\beta 0+\beta 1x*1+\beta 2x*2+...+\beta px*p$.

Si $f(x*)$ es positiva, la observación se asigna a la clase +1, si es negativa, a la clase −1. Además, la magnitud de $f(x*)$ permite saber cómo de lejos está la observación del hiperplano y con ello la confianza de la clasificación.

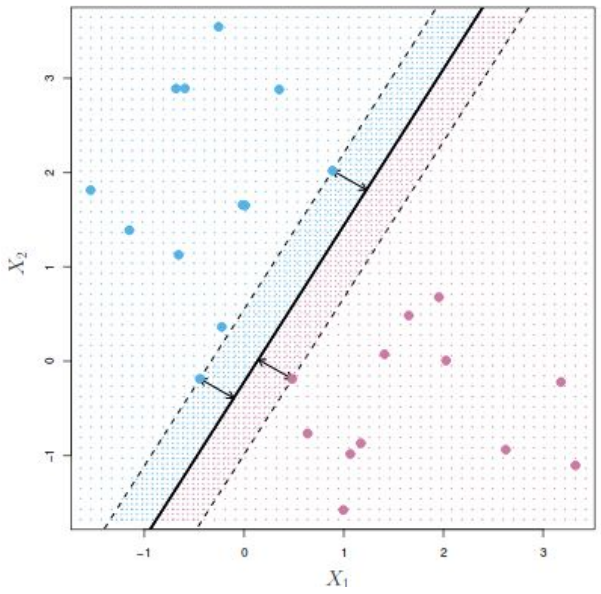
La definición de hiperplano para casos perfectamente separables linealmente resulta en un número infinito de posibles hiperplanos, lo que hace necesario un método que permita seleccionar uno de ellos como clasificador óptimo.



- La solución a este problema consiste en seleccionar como clasificador óptimo al que se conoce como maximal margin hyperplane o hiperplano óptimo de separación, que se corresponde con el hiperplano que se encuentra más alejado de todas las observaciones de entrenamiento.

- Para obtenerlo, se tiene que calcular la distancia perpendicular de cada observación a un determinado hiperplano. La menor de estas distancias (conocida como margen) determina como de alejado está el hiperplano de las observaciones de entrenamiento.

- El maximal margin hyperplane se define como el hiperplano que consigue un mayor margen, es decir, que la distancia mínima entre el hiperplano y las observaciones es lo más grande posible. Aunque esta idea suena razonable, no es posible aplicarla, ya que habría infinitos hiperplanos contra los que medir las distancias. En su lugar, se recurre a métodos de optimización.



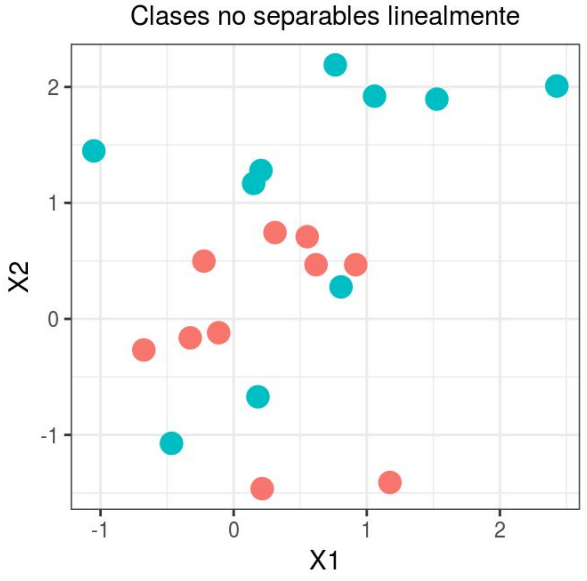
- La imagen anterior muestra el maximal margin hyperplane para un conjunto de datos de entrenamiento. Las tres observaciones equidistantes respecto al maximal margin hyperplane se encuentran a lo largo de las líneas discontinuas que indican la anchura del margen.

- A estas observaciones se les conoce como ***vectores soporte***, ya que son vectores en un espacio p-dimensional y soportan (definen) el maximal margin hyperplane. Cualquier modificación en estas observaciones (vectores soporte) conlleva cambios en el maximal margin hyperplane. Sin embargo, modificaciones en observaciones que no son vector soporte no tienen impacto alguno en el hiperplano.

CASOS CUASI-SEPARABLES LINEALMENTE

- El maximal margin hyperplane descrito en el apartado anterior es una forma muy simple y natural de clasificación siempre y cuando exista un hiperplano de separación.

- En la gran mayoría de casos reales, los datos no se pueden separar linealmente de forma perfecta, por lo que no existe un hiperplano de separación y no puede obtenerse un maximal margin hyperplane.



Para solucionar estas situaciones, se puede extender el concepto de maximal margin hyperplane para obtener un hiperplano que casi separe las clases, pero permitiendo que cometa unos pocos errores. **A este tipo de hiperplano se le conoce como Support Vector Classifier o Soft Margin.**

Support Vector Classifier o Soft Margin SVM

- El Maximal Margin Classifier descrito en la sección anterior tiene poca aplicación práctica, ya que rara vez se encuentran casos en los que las clases sean perfectas y linealmente separables. De hecho, incluso cumpliéndose estas condiciones ideales, en las que exista un hiperplano capaz de separar perfectamente las observaciones en dos clases, esta aproximación sigue presentando dos inconvenientes:

    - Dado que el hiperplano tiene que separar perfectamente las observaciones, es muy sensible a variaciones en los datos. Incluir una nueva observación puede suponer cambios muy grandes en el hiperplano de separación (poca robustez).

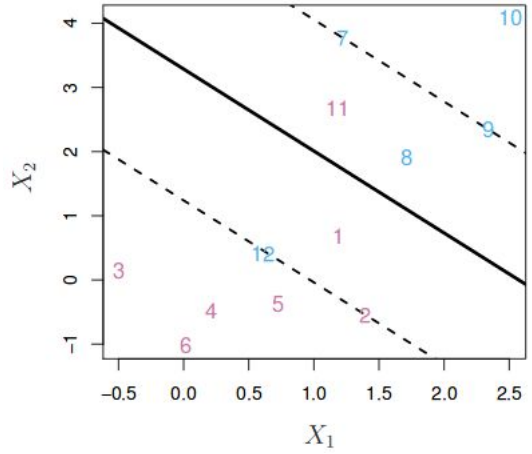
    - Que el maximal margin hyperplane se ajuste perfectamente a las observaciones de entrenamiento para separarlas todas correctamente suele conllevar problemas de overfitting.

Por estas razones, es preferible crear un clasificador basado en un hiperplano que, aunque no separe perfectamente las dos clases, sea más robusto y tenga mayor capacidad predictiva al aplicarlo a nuevas observaciones (menos problemas de overfitting).

Esto es exactamente lo que consiguen los clasificadores de vector soporte, también conocidos como soft margin classifiers o Support Vector Classifiers. Para lograrlo, en lugar de buscar el margen de clasificación más ancho posible que consigue que las observaciones estén en el lado correcto del margen; se permite que ciertas observaciones estén en el lado incorrecto del margen o incluso del hiperplano.

La siguiente imagen muestra un clasificador de vector soporte ajustado a un pequeño set de observaciones. La línea continua representa el hiperplano y las líneas discontinuas el margen a cada lado.

- Las observaciones 2, 3, 4, 5, 6, 7 y 10 se encuentran en el lado correcto del margen (también del hiperplano) por lo que están bien clasificadas. Las observaciones 1 y 8, a pesar de que se encuentran dentro del margen, están en el lado correcto del hiperplano, por lo que también están bien clasificadas.
- Las observaciones 11 y 12, se encuentran en el lado erróneo del hiperplano, su clasificación es incorrecta. Todas aquellas observaciones que, estando dentro o fuera del margen, se encuentren en el lado incorrecto del hiperplano, se corresponden con observaciones de entrenamiento mal clasificadas.



- La identificación del hiperplano de un clasificador de vector soporte, que clasifique correctamente la mayoría de las observaciones a excepción de unas pocas, es un problema de optimización convexa.

- Es importante mencionar que el proceso incluye un hiperparámetro de tuning C. C controla el número y severidad de las violaciones del margen (y del hiperplano) que se toleran en el proceso de ajuste.

- Si C=∞, no se permite ninguna violación del margen y por lo tanto, el resultado es equivalente al Maximal Margin Classifier (teniendo en cuenta que esta solución solo es posible si las clases son perfectamente separables).
- Cuando más se aproxima C a cero, menos se penalizan los errores y más observaciones pueden estar en el lado incorrecto del margen o incluso del hiperplano. C es a fin de cuentas el hiperparámetro encargado de controlar el balance entre bias y varianza del modelo. En la práctica, su valor óptimo se identifica mediante cross-validation.

El proceso de optimización tiene la peculiaridad de que solo las observaciones que se encuentran justo en el margen o que lo violan influyen sobre el hiperplano. A estas observaciones se les conoce como vectores soporte y son las que definen el clasificador obtenido. Esta es la razón por la que el parámetro C controla el balance entre bias y varianza.

Conclusiones:

Cuando el valor de C es pequeño, el margen es más ancho, y más observaciones violan el margen, convirtiéndose en vectores soporte. El hiperplano está, por lo tanto, sustentado por más observaciones, lo que aumenta el bias pero reduce la varianza.

Cuando mayor es el valor de C, menor el margen, menos observaciones serán vectores soporte y el clasificador resultante tendrá menor bias pero mayor varianza.

Otra propiedad importante que deriva de que el hiperplano dependa únicamente de una pequeña proporción de observaciones (vectores soporte), es su robustez frente a observaciones muy alejadas del hiperplano. Esto hace al método de clasificación vector soporte distinto a otros métodos tales como Linear Discriminant Analysis (LDA), donde la regla de clasificación depende de la media de todas las observaciones.

El Support Vector Classifier descrito en los apartados anteriores consigue buenos resultados cuando el límite de separación entre clases es aproximadamente lineal.

Si no lo es, su capacidad decae drásticamente. **Una estrategia para enfrentarse a escenarios en los que la separación de los grupos es de tipo no lineal consiste en expandir las dimensiones del espacio original.**

El hecho de que los grupos no sean linealmente separables en el espacio original no significa que no lo sean en un espacio de mayores dimensiones.

Aumento de la dimensión, kernels

Una vez definido que las Máquinas de Vector Soporte siguen la misma estrategia que el Support Vector Classifier, pero aumentando la dimensión de los datos antes de aplicar el algoritmo, la pregunta inmediata es ¿Cómo se aumenta la dimensión y qué dimensión es la correcta?

La dimensión de un conjunto de datos puede transformarse combinando o modificando cualquiera de sus dimensiones. Por ejemplo, se puede transformar un espacio de dos dimensiones en uno de tres aplicando la siguiente función:

$$f(x_1, x_2) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)$$

Esta es solo una de las infinitas trasformaciones posibles, ¿Cómo saber cuál es la adecuada? Es aquí donde los kernel entran en juego.

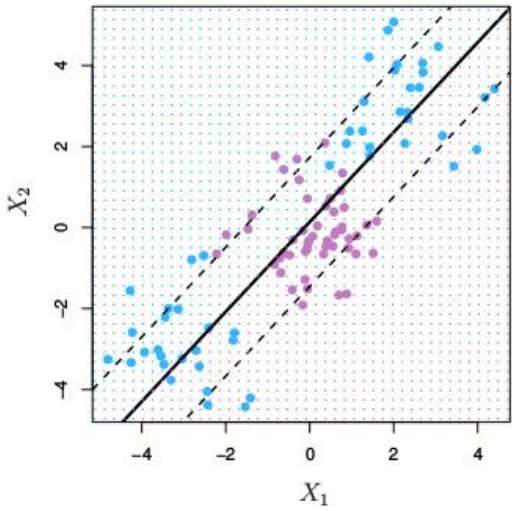
- Un kernel (K) es una función que devuelve el resultado del dot product entre dos vectores realizado en un nuevo espacio dimensional distinto al espacio original en el que se encuentran los vectores.
- Esta contiene un dot product. Si se sustituye este dot product por un kernel, se obtienen directamente los vectores soporte (y el hiperplano) en la dimensión correspondiente al kernel.

- Existen multitud de kernels distintos, algunos de los más utilizados son:

Kernel lineal

$$\mathbf{K(x,x')=x \cdot x'}$$

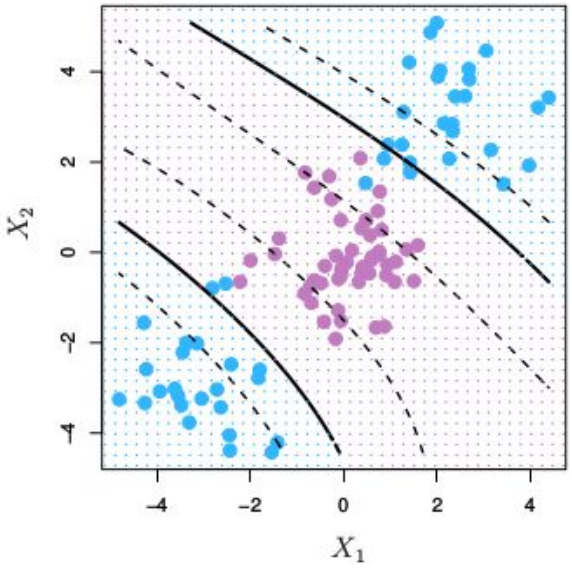
Si se emplea un Kernel lineal, el clasificador Support Vector Machine obtenido es equivalente al Support Vector Classifier.



Kernel polinómico

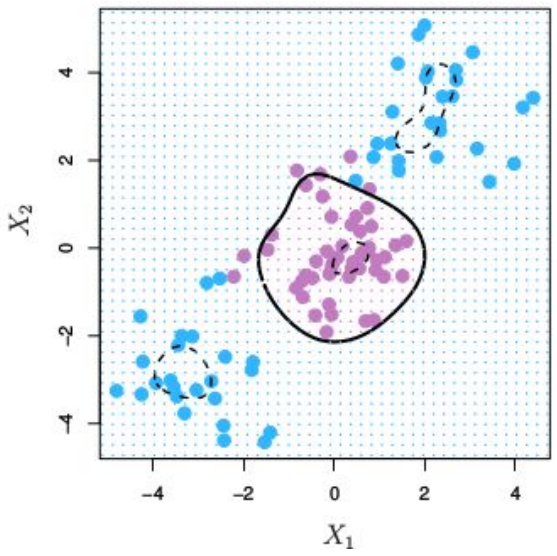
$$K(x,x')=(x \cdot x'+c)d$$

Cuando se emplea d=1 y c=0, el resultado es el mismo que el de un kernel lineal. Si d>1, se generan límites de decisión no lineales, aumentando la no linealidad a medida que aumenta d. No suele ser recomendable emplear valores de d mayores 5 por problemas de overfitting.



Gaussian Kernel (RBF)

$$K(x,x')=\exp(-\gamma||x-x'||2)$$



El valor de γ controla el comportamiento del kernel, cuando es muy pequeño, el modelo final es equivalente al obtenido con un kernel lineal, a medida que aumenta su valor, también lo hace la flexibilidad del modelo.

No puede decirse que haya un kernel que supere al resto, depende en gran medida de la naturaleza del problema que se esté tratando.

Ahora bien, es muy recomendable probar el kernel RBF. Este kernel tiene dos ventajas: que solo tiene dos hiperparámetros que optimizar (γ y la penalización C común a todos los SVM) y que su flexibilidad puede ir desde un clasificador lineal a uno muy complejo.

Support Vector Machines para más de dos clases

El concepto de hiperplano de separación en el que se basan los SVMs no se generaliza de forma natural para más de dos clases. Se han desarrollado numerosas estrategias con el fin de aplicar este método de clasificación a situaciones con k>2-clases, de entre ellos, los más empleados son: one-versus-one, one-versus-all y DAGSVM.

One-versus-one

Supóngase un escenario en el que hay K > 2 clases y que se quiere aplicar el método de clasificación basado en SVMs. La estrategia de one-versus-one consiste en generar un total de K(K-1)/2 SVMs, comparando todos los posibles pares de clases.

Para generar una predicción se emplean cada uno de los K(K-1)/2 clasificadores, registrando el número de veces que la observación es asignada a cada una de las clases.

Finalmente, se considera que la observación pertenece a la clase a la que ha sido asignada con más frecuencia.

La principal desventaja de esta estrategia es que el número de modelos necesarios se dispara a medida que aumenta el número de clases, por lo que no es aplicable en todos los escenarios.

One-versus-all

Esta estrategia consiste en ajustar K SVMs distintos, cada uno comparando una de las K clases frente a las restantes K-1 clases. Como resultado, se obtiene un hiperplano de clasificación para cada clase.

Para obtener una predicción, se emplean cada uno de los K clasificadores y se asigna la observación a la clase para la que la predicción resulte positiva. Esta aproximación, aunque sencilla, puede causar inconsistencias, ya que puede ocurrir que más de un clasificador resulte positivo, asignando así una misma observación a diferentes clases. Otro inconveniente adicional es que cada clasificador se entrena de forma no balanceada. Por ejemplo, si el set de datos contiene 100 clases con 10 observaciones por clase, cada clasificador se ajusta con 10 observaciones positivas y 990 negativas.

DAGSVM

DAGSVM (Directed Acyclic Graph SVM) es una mejora del método one-versus-one. La estrategia seguida es la misma, pero consiguen reducir su tiempo de ejecución eliminando comparaciones innecesarias gracias al empelo de una directed acyclic graph (DAG).

Supóngase un set de datos con cuatro clases (A, B, C, D) y 6 clasificadores entrenados con cada posible par de clases (A-B, A-C, A-D, B-C B-D, C-D). Se inician las comparaciones con el clasificador (A-D) y se obtiene como resultado que la observación pertenece a la clase A, o lo que es equivalente, que no pertenece a la clase D. Con esta información se pueden excluir todas las comparaciones que contengan la clase D, puesto que se sabe que no pertenece a este grupo.

En la siguiente comparación se emplea el clasificador (A-C) y se predice que es A. Con esta nueva información se excluyen todas las comparaciones que contengan C. Finalmente solo queda emplear el clasificador (A-B) y asignar la observación al resultado devuelto.

Siguiendo esta estrategia, en lugar de emplear los 6 clasificadores, solo ha sido necesario emplear 3. DAGSVM tiene las mismas ventajas que el método one-versus-one pero mejorando mucho el rendimiento.

Bibliografía:

https://www.youtube.com/watch?v=_PwhiWxHK8o

Introduction to Statistical Learning Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani

Support Vector Machines Succinctly by Alexandre Kowalczyk

A Practical Guide to Support Vector Classification by Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin