Curso de especialización en Inteligencia Artificial y Big Data

UD04. DESARROLLO DE APLICACIONES DE IA: ALGORITMOS

MAQUINAS DE SOPORTE DE VECTORES (RESUMEN)

Carlos Sáenz Adán

1. Máquinas de Soporte de Vectores (Support Vector Machines)

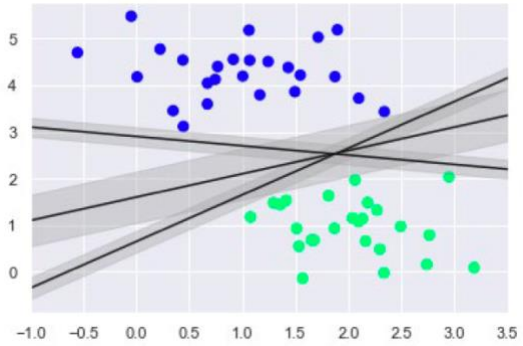
Antes de comenzar con la explicación, te recomiendo que eches un ojo al siguiente vídeo. En este video se realiza una explicación muy bien ilustrada de la regresión lineal.

https://www.youtube.com/watch?v=kl6tyEi5eso

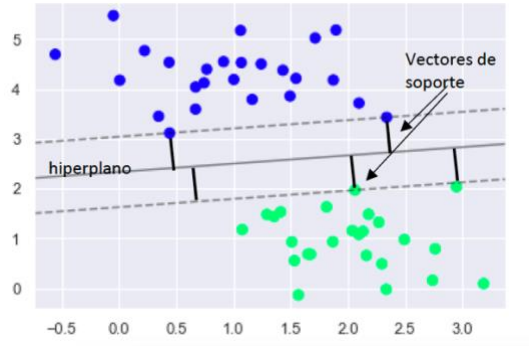
1.1. Introducción matemática

Este es uno de los algoritmos clásicos basados en razonamientos supervisado. Primero, se partirá de una colección de datos etiquetados, por ejemplo, uno conjunto de emails que serán la entrada X del sistema, con una etiqueta, "spam" o "no spam", que será la salida del sistema. En líneas generales, la clasificación mediante SVMs está basada en la búsqueda de un hiperplano (o varios) que permita separar, con la mayor precisión posible, los elementos de una colección de datos según la clase a la que pertenecen.

A modo de ejemplo, en la siguiente imagen que contiene varios puntos, los cuales representan datos de la colección que se pueden categorizar en dos clases, azul y verde. Para separar ambas clases se pueden usar varios hiperplanos como se puede ver; sin embargo, algunos son óptimos que los otros.

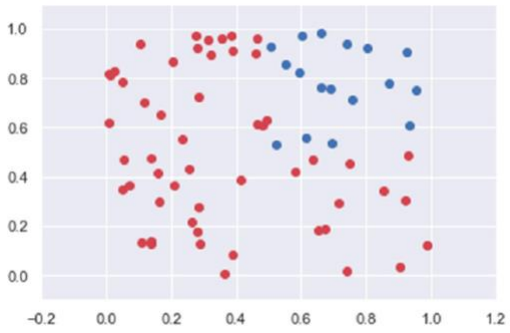


Para medir esa optimalidad, se usan los vectores de soporte, que son los puntos que definen el margen máximo de separación entre el hiperplano y las clases que separan, que vienen determinadas por la línea punteada que se puede ver en la siguiente figura, que se denomina margen máximo del hiperplano.



El rendimiento de los SVM es especialmente bueno en aquellas colecciones de datos donde la separación entre clases es relativamente clara; sin embargo, en casos como el de la siguiente figura donde la

separación entre clases es compleja, el rendimiento decae.



Con el fin de ajustar el hiperplano, se recurre a la modificación de los hiperparámetros típicos de este algoritmo. Entre estos destacan:

Kernel (núcleo): La función principal del núcleo (kernel) es transformar los datos de entrada en otra forma más útil. Hay varios tipos de funciones como la lineal, la polinómica o la de base radial (RBF). Las funciones polinómicas y RBF son útiles para los hiperplanos no lineales. Los núcleos polinómicos y RBF calculan la línea de separación en la dimensión superior. En algunas aplicaciones, se sugiere utilizar un kernel más complejo para separar las clases que son curvas o no lineales. Esta transformación puede conducir a clasificadores más precisos.

Regularización: El parámetro C se utiliza para establecer el mecanismo de margen/regularización. C es el parámetro de penalización que representa la mala clasificación o el término de error. El término de error o clasificación errónea establece el nivel de error asumible en el proceso de optimización. De esta forma se controla el equilibrio entre el límite de decisión y el término de clasificación errónea. Un valor menor de C crea un hiperplano de margen pequeño y un valor mayor de C crea un hiperplano de margen mayor.

Gamma: Modela la parametrización del kernel. Un valor más bajo de gamma se ajustará de forma imprecisa al conjunto de datos de entrenamiento, mientras que un valor más alto de gamma se ajustará exactamente al conjunto de datos de entrenamiento, lo que provoca un sobreajuste. Es decir, se puede decir que un valor bajo de gamma considera solo los puntos cercanos en el cálculo de la línea de separación, mientras que un valor alto de gamma considera todos los puntos de datos en el cálculo de la línea de separación.

1.2. Implementación con scikit-lean

La explicación puedes verla en la sección *Máquinas de Soporte de Vectores (SVM) d*el siguiente enlace de Google Colab:

https://colab.research.google.com/drive/15l_wHChYt-JNOTMV3Qtl_4YGqZHW8l1P

En *Scikit Learn* pueden encontrarse tres implementaciones distintas del algoritmo *Suport Vector Machine*:

- Las clases sklearn.svm.**SVC** y sklearn.svm.**NuSVC** permiten crear modelos SVM de clasificación empleando kernel lineal, polinomial, radial o sigmoide. La diferencia es que SVC controla la

regularización a través del hiperparámetro C, mientras que NuSVC lo hace con el número máximo de vectores soporte permitidos.

- La clase sklearn.svm.**LinearSVC** permite ajustar modelos SVM con kernel lineal. Es similar a SVC cuando el parámetro kernel='linear', pero utiliza un algoritmo más rápido.
- Las clases sklearn.svm.**SVCR** y sklearn.svm.**LinearSVR** son semejantes pero para modelos de regresión.

Los parámetros más relevantes a la hora de construir modelos de este tipo serán:

- **kernel**. Puede ser "lineal", "poly", "rbf", "sigmoid", "precomputed". Por defecto será "rbf".
- **C**. Parámetro de regularización.
- **gamma**: en un kernel rbf puede tener los valores "scale", "auto" o un valor de tipo float. Modela la parametrización del kernel.
- **degree**: en un kernel polinomial indica el grado que se aplicará
- **Coef0**: indica cómo está influido el modelo por los polinomios de grado alto.

```
from sklearn.svm import SVC

model = SVC(kernel="poly", gamma='scale', C=5)
model.fit(X_train,y_train)
print('Model score: ', model.score(X_test,y_test))


model = SVC(kernel="rbf", gamma=5, C=0.0001)
model.fit(X_train,y_train)
print('Model score: ', model.score(X_test,y_test))
```