Universidad de Huelva

Grado en Ingeniería Informática

Tema 3. Máquina de Soporte Vectorial

Resumen

Autor: Alberto Fernández Merchán
Asignatura: Aprendizaje Automático

1. Introducción

Antes de 1980, casi todos los métodos de aprendizaje de máquina se basaban en superficies de decisión lineal. Las propiedades teóricas del **aprendizaje lineal** funcionaban bien.

Sin embargo, en la **década de los 80's**, los árboles de decisión y los métodos de redes neuronales (NN) permitieron un aprendizaje eficiente sobre superficies de decisión no lineales. Estos métodos carecían de bases teóricas y, además, se veían afectados negativamente por los mínimos locales.

En la década de los 90's, se desarrollan algoritmos de aprendizaje eficientes para funciones no lineales (teoría del aprendizaje computacional). En **1992** se introduce el concepto de **Máquinas de Soporte Vectorial** ideadas por Boser, Guyon y Vapnik. Es un algoritmo bien fundamentado que se venía desarrollando mediante la **Teoría del Aprendizaje Estadístico** desde los años 60's. Además, tiene un buen desempeño empírico.

2. Conceptos Básicos

Una máquina de soporte vectorial (SVM) se puede utilizar para **clasificación** y regresión. Se aplica un **aprendizaje supervisado** donde los datasets incluyen la clase a la que pertenece cada instancia. El aprendizaje consistirá en encontrar la división entre las clases.

Se utiliza para realizar clasificación binaria, sin embargo, también se puede usar para clasificar más de dos clases utilizando un clasificador binario por cada una de las clases que se quiera clasificar. Una máquina de soporte vectorial aprende la **superficie de decisión** de dos clases diferentes sobre los puntos de entrada y forma una **frontera de decisión** que divide los elementos que forman cada clase.

Clasificar significa aprender la función de mapeo: $X \rightarrow Y$. Donde X es un objeto e Y la etiqueta de la clase que le asigna el clasificador. Las clases las definiremos como $+1$ (clase positiva) y -1 (clase negativa).

La máquina de soporte vectorial trabaja sobre un **espacio convexo**, esto quiere decir que no existen máximos locales en los que se pueda quedar estancada.

Para saber si una nueva instancia pertenece o no a una clase seguiremos los siguientes pasos:

- Construimos un **hiperplano de separación** que divida las muestras positivas$(+1)$ de las negativas(-1).
- Los puntos de dicho hiperplano satisfacen la ecuación: $wx + b = 0$
- Los puntos que se encuentren por encima o por debajo del hiperplano corresponderán con:
  - $wx + b > 0 \rightarrow y = +1$
  - $wx + b < 0 \rightarrow y = -1$
- La **matriz de confusión** nos permite visualizar la calidad del clasificador.

3. Separación de Datos Inicial

Podemos definir infinitos hiperplanos de separación entre dos clases (o ninguno). Necesitamos añadir más características a dicho hiperplano de separación para que sea único, dichas características dependerán de los datos.

Existen tres tipos de conjuntos de datos:

- Linealmente Separables
- Quasi-Separables
- No separables linealmente

4. Datos Linealmente Separables

Establecemos que la ecuación del hiperplano es $wx + b = 0$. (Aprender significa encontrar los coeficientes w y b. Definimos una **distancia de margen** (la distancia del hiperplano al punto más cercano de la clase). Nos proponemos encontrar una ecuación del hiperplano que haga que el margen sea lo más grande posible.

La idea es una recta o plano que pase justo por el medio de las clases, haciendo al clasificador lo más tolerante posible. Los puntos de cada clase más cercanos al hiperplano se denominan **vectores soporte**. La distancia de todos los vectores soporte al hiperplano es la misma. Hacemos dos hiperplanos paralelos al que

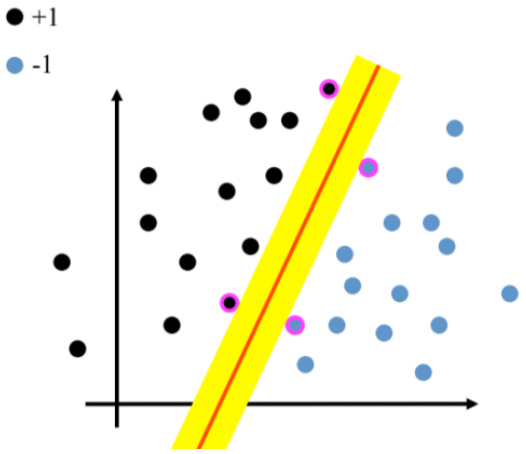

Figura 1: Vectores Soporte

buscamos y que pasen por los vectores soporte. Expresaremos las ecuaciones de estos dos nuevos planos de la siguiente forma:

- Hiperplano positivo: $xw + b = +1$
- Hiperplano negativo: $wx + b = -1$

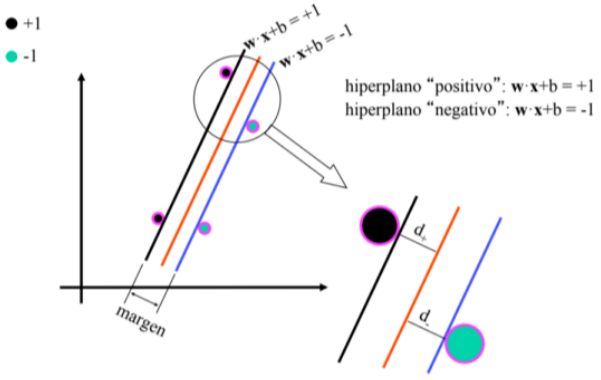

Figura 2: Hiperplanos que pasan por los vectores soporte

Queremos construir un clasificador que nos diga a qué clase pertenece un punto en función de sus coordenadas.

- $wx + b > 0$ para $y_i = +1$
- $wx + b < 0$ para $y_i = -1$

4.1. Notación

- $\vec{w}$ es un vector normal al hiperplano.
- $\frac{|b|}{||w||}$ es la distancia perpendicular del hiperplano al origen.
- $||w||$ es la norma euclídea de w.
- $< w, x_i >$ es el producto escalar de w y x_i

4.2. Conceptos Matemáticos

Por tanto, podemos expresar el problema de forma que debemos **minimizar** $||\mathbf{w}||$ sujeto a la siguiente ecuación: $< w, x_i > +b = 0$. Sin embargo, podemos transformar el problema utilizando los **multiplicadores de Lagrange** (α_i).

$$L_p = \frac{1}{2}||w|| - \sum_{i-1}^{m} \alpha_i y_i (w x_i + b) + \sum_{i=1}^{m} \alpha_i$$

Pasando el problema "al dual", se convierte en:

- Maximizar: $L_D = \sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i x_j$
- Sujeto a: $w = \sum_{i=1}^{m} \alpha_i y_i x_i$ y $\sum_{i=1}^{m} \alpha_i y_i = 0$

5. Datos Quasi-Separables

Los problemas reales se caracterizan por tener ejemplos ruidosos y no ser linealmente separable. La estrategia que se sigue en este tipo de problemas es permitiendo que haya errores de clasificación en algunos de los ejemplos del conjunto de entrenamiento. Sin embargo, sigue siendo **el objetivo principal** encontrar un hiperplano óptimo con el **menor error posible**.

En la formulación anterior, un ejemplo es no separable si no cumple la condición:
$y_i(< w_i, x_i > +b) \geq 1 \quad i = 1, \ldots, n.$
Introduciremos un factor de relajación (ξ), por lo que los hiperplanos quedarán:

- $< w, x_i > +b \geq 1 - \xi_i$ para $y = +1$
- $< w, x_i > +b \leq 1 + \xi_i$ para $y = -1$

6. Datos No Separables Linealmente

Los datos originales no pueden ser separados por una superficie lineal (pérdida de convexidad). Podemos mapear los datos por medio de una **función Kernel** a un espacio de características en un espacio dimensional más alto. Buscando, así, la máxima separación entre las clases.

Dicha función Kernel (K) será el producto escalar de una función ($K = \phi \cdot \phi$)

Las funciones Kernel estándar son:

- Lineal: $K(a, b) = a \cdot b$
- Polinómicas: $K(a, b) = (\gamma a \cdot b + c)^d$
- Funciones de Base Radial (RBF): $K(a, b) = exp(-\frac{(a-b)^2}{2a^2})$
- Sigmoide (o NN): $K(a, b) = tanh(\gamma a \cdot b + c)$

No todas las funciones pueden ser Kernel, es necesario que sean factorizables y deben cumplir **la condición de Mercer**.

Condición de Mercer: Existe una transformación Φ y una expresión en series $K(x_i, x_j) = \sum \Phi(x_i)\Phi(x_j)$ si, y solo sí, para cualquier $g(x)$ para que la integral $\int g(x)^2$ sea finita, se tiene que $\int K(xy)g(x)g(y)dxdy \geq 0$

Utilizando esta propiedad, podemos reducir el problema a minimizar:

$$L_p = \frac{1}{2}||w^2|| + C(\sum_{i=1}^{n} \xi_i) \tag{1}$$

El problema está sujeto a las restricciones: $z_i(w \cdot \phi(x_i) + b) - 1 + \xi_i \geq 0$ y $\xi_i \geq 0$

7. Implementación

El cálculo de parámetros del hiperplano es un problema de optimización cuadrática. Existen varios algoritmos especializados para resolver rápidamente el problema QP (Quadratic Programming), sobre todo basados en heurísticas para dividir el problema en trozos más pequeños y manejables.

Uno de los métodos más comunes es el algoritmo de optimización secuencial mínima (SMO) de Plaat[2]. Dicho algoritmo consiste en descomponer el problema es sub-problemas de 2 dimensiones que se pueden resolver analíticamente, eliminando la necesidad de un algoritmo de optimización numérica.