Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones

Supervised Learning: Methods, Properties and Applications

Trabajo Fin de Grado en Matemáticas
Universidad de Málaga

Autor: Gema Valenzuela González

Área de conocimiento y/o departamento: Área de Estadística e Investigación Operativa. Departamento de Análisis Matemático, Estadística e Investigación Operativa y Matemática Aplicada.

Fecha de presentación: Septiembre, 2022.

Tema: Aprendizaje estadístico. Aprendizaje supervisado. Modelos con vectores soporte.

Tipo: Trabajo de revisión bibliográfica

Modalidad: Individual

Número de páginas (sin incluir introducción, bibliografía ni anexos): 47

DECLARACIÓN DE ORIGINALIDAD DEL TFG

D./Dña. *Gema Valenzuela González*, con DNI 26054658M *(DNI, NIE o pasaporte)*, estudiante del Grado en *Matemáticas* de la Facultad de Ciencias de la Universidad de Málaga,
DECLARO:

Que he realizado el Trabajo Fin de Grado titulado "*Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones*" y que lo presento para su evaluación. Dicho trabajo es original y todas las fuentes bibliográficas utilizadas para su realización han sido debidamente citadas en el mismo.

De no cumplir con este compromiso, soy consciente de que, de acuerdo con la normativa reguladora de los procesos de evaluación de los aprendizajes del estudiantado de la Universidad de Málaga de 23 de julio de 2019, esto podrá conllevar la calificación de suspenso en la asignatura, sin perjuicio de las responsabilidades disciplinarias en las que pudiera incurrir en caso de plagio.

Para que así conste, firmo la presente en Málaga, el *12/09/2022*

Fdo:..... Fecha: 2022.09.12 12:37:47 +02'00'

Índice general

Aprendizaje Supervisado: Métodos, Propiedades y Aplicaciones

Resumen

El aprendizaje supervisado se centra en el desarrollo, a partir de un conjunto de datos conocido, de un modelo matemático capaz de predecir el valor correspondiente de un nuevo dato. En este trabajo, estudiaremos la construcción de dichos modelos utilizando el concepto de vector soporte.

En el Capítulo 1, introduciremos los conceptos básicos del aprendizaje supervisado, y la notación que utilizaremos a lo largo del trabajo. En particular, analizamos las tareas de clasificación, regresión y detección de *outliers* o valores atípicos.

El Capítulo 2 se centra en la clasificación binaria. En particular, analizaremos el desarrollo teórico de la metodología conocida como Máquinas de Vectores Soporte, usualmente llamada Support Vector Machines (SVM) por su nombre en inglés. El método SVM consiste en encontrar un hiperplano lineal de separación que maximice la distancia de los datos más cercanos de cada clase. Si los datos son perfectamente separables a través de una función lineal, entonces hablamos del SVM con margen duro. Sin embargo, normalmente se permite que haya algunos errores en la clasificación debido a que los datos no se han podido separar perfectamente por una función lineal. En ese caso, hablaremos de SVM con margen blando. Por último, tenemos el caso en el que los datos no pueden ser separados por una función lineal. Por ello, transformamos los datos de entrada para trasladarlos a un espacio de dimensión superior donde sí pueden ser separados linealmente. Para ello utilizaremos las denominadas funciones kernels.

En el Capítulo 3, tratamos dos extensiones del método SVM: el algoritmo SVR (del inglés *Support Vector Regression*) para resolver problemas de regresión, y la metodología SVDD (del inglés *Support Vector Data Description*) para detectar *outliers*. Análogamente al algoritmo de SVM, en ambos casos analizaremos la versión lineal y no lineal.

En el Capítulo 4, realizaremos varios experimentos computacionales en los que analizaremos cada uno de los métodos vistos en los capítulos anteriores.

En el Capítulo 5, resumiremos las conclusiones de este trabajo.

Palabras clave:

Clasificación, regresión, detección de outliers, vectores soporte, aprendizaje supervisado

Supervised Learning: Methods, Properties and Applications

Abstract

Supervised learning aims to develop a mathematical model from a known data set that can predict the value of a new unseen point. In this work, we will study how to build such models based on the concept of support vectors.

In Chapter 1, the basic concepts of supervised learning, and the notation we will use throughout the paper, are introduced. In particular, we will analyse the tasks of classification, regression and outlier detection.

Chapter 2 focuses on binary classification. More specifically, we will analyse the theoretical development of the methodology known as Support Vector Machines (SVM). The objective of SVM is to find a linear hyperplane that maximises the distance of the closest data of each category. If the data can be perfectly separated through a linear function, then we talk about hard margin SVM. However, it is usually assumed that the data cannot be perfectly classified with a linear function. Consequently, some errors in the classification are allowed. In this case, we assess the soft margin SVM. Finally, we have to consider the case where the data cannot be separated by a linear function. If this situation occurs, then we transform the original input data in order to translate them to a higher dimensional space where they can be linearly separated. For this case, we use so-called kernel functions.

Chapter 3 discusses two extensions of the SVM method: the Support Vector Regression (SVR) algorithm for solving regression problems and the Support Vector Data Description (SVDD) methodology for detecting outliers. Analogously to the SVM algorithm, we analyse the linear and non-linear versions in both cases.

In Chapter 4, we perform several computational experiments to analyse each of the methods seen in the previous chapters.

Finally, in Chapter 5, we draw some conclusions on the work carried out.

key words:

CLASSIFICATION, REGRESSION, OUTLIERS DETECTION, SUPPORT VECTORS, SUPERVISED LEARNING

Capítulo 1

Introducción

El aprendizaje automático estudia la construcción de modelos capaces de aprender ciertas estructuras a partir de la información proporcionada por los datos. Es decir, el aprendizaje automático se centra en encontrar patrones en los datos de tal forma que podamos usar dichos patrones en puntos que no han sido observados previamente. Existen dos tipos principales de modelos de aprendizaje automático, a saber, los modelos de aprendizaje supervisado y los modelos de aprendizaje no supervisado [Hastie et al., 2009].

En el aprendizaje supervisado asumimos conocido el llamado conjunto de entrenamiento (*training sample*). Cada dato de dicho conjunto está formado por las variables explicativas o conjunto de características (*features*) y la variable respuesta. El objetivo del aprendizaje supervisado [Hastie et al., 2009] es construir, a partir de los datos de entrenamiento, un modelo de predicción. Usando este modelo, podremos predecir la variable respuesta de un nuevo conjunto de datos no vistos (*test sample*) únicamente conociendo sus variables explicativas.

Por otro lado, en el aprendizaje no supervisado [Hastie et al., 2009] no existe variable respuesta. Es decir, el conjunto de entrenamiento solo está formado por variables explicativas. Debido a esto, el objetivo del aprendizaje no supervisado es encontrar modelos que permitan describir como se organizan y agrupan los datos. Estos modelos se utilizarán para explicar los patrones existentes en los nuevos datos no observados.

En este trabajo nos centraremos únicamente en el estudio del aprendizaje supervisado, debido a su gran importancia dentro de las Matemáticas, así como el gran número de aplicaciones en la vida real [Carrizosa and Romero Morales, 2013; Benítez-Peña et al., 2021]. Por ejemplo, en muchas áreas de la ciencia [Hastie et al., 2009], tales como en la medicina para el diagnóstico de enfermedades, o en la química para el desarrollo de fármacos. Cabe destacar también su importancia en el mundo de las finanzas [Baesens et al., 2003; Fawcett and Provost, 1997], para la detección de fraudes; para la puntuación crediticia, negociación algorítmica y clasificación de bonos. Por último, podemos encontrar aplicaciones del aprendizaje supervisado en la industria [Goh et al., 2018; Boodhun and Jayabalan, 2018], por ejemplo, en la detección de valores atípicos o en la estimación de la vida útil de los equipos industriales. En [Hastie et al., 2009] podemos encontrar algunos ejemplos del aprendizaje supervisado en la vida real, que detallaremos a continuación:

Ejemplo 1.1. Detección de correos no deseados (spam):

En este ejemplo consideramos un conjunto de datos de entrenamiento formado por 20 correos electrónicos. De estos correos conocemos sus variables explicativas y su variable

respuesta. Las variables explicativas son las palabras y los signos de puntuación más usuales en los correos. Por otro lado, la variable respuesta toma dos valores: "spam" y "no spam". La Tabla 1.1 muestra el porcentaje medio de variables explicativas (palabras y signos de puntuación) que aparecen tanto en los correos spam como no spam contenidos en los datos de entrenamiento. Es importante recordar que se han tomado como variables explicativas sólo las palabras y signos de puntuación más usuales en un correo. Es por ello que la suma de los porcentajes que aparecen en la Tabla 1.1 no es igual a 100 %.

		Variables explicativas									
		george	you	your	hp	free	hpl	!	our	re	edu
Variable respuesta	spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01
	no spam	1.27	1.27	0.44	0.90	0.07	0.43	0.18	0.42	0.29	0.01

Tabla 1.1: Porcentaje medio de palabras y signos más usuales en un correo electrónico (spam o no spam) incluido en el conjunto de datos de entrenamiento.

Una vez observadas las variables explicativas y la variable respuesta del conjunto de datos de entrenamiento, el objetivo de este ejemplo es construir un modelo de aprendizaje supervisado que nos permita predecir si un nuevo correo es spam o no, a través del porcentaje de veces que aparecen las palabras y los signos de puntuación consideradas como variables explicativas. Por ejemplo, una posible regla de predicción podría ser:

$$\begin{aligned} &\text{if} \quad (\,\%\text{george} < 0.6) \quad \& \quad (\,\%\text{you} > 1.3) \quad \text{then} \\ &\qquad \text{spam} \\ &\text{else} \\ &\qquad \text{no spam} \end{aligned} \tag{1.1}$$

Es decir, si tenemos un nuevo correo donde el porcentaje de veces que aparece la palabra "george" es menor del 0.6 % y el porcentaje de veces que aparece la palabra "you" es mayor del 1.3 %, entonces el nuevo correo será identificado como "spam". En caso contrario, el correo será "no spam".

Ejemplo 1.2. Detección de cáncer de próstata:

El conjunto de datos de entrenamiento de este ejemplo está formado por un grupo de pacientes a punto de someterse a una cirugía para extirparle la próstata. De cada paciente conocemos la cantidad de sangre en la orina y el nivel de glóbulos rojos, que serán las variables explicativas. Además, también se conoce la variable respuesta que viene dada por el nivel de antígeno prostático en el paciente.

El objetivo de este ejemplo es construir un modelo de aprendizaje supervisado con el que poder predecir el nivel de antígeno prostático de un nuevo paciente. Se analiza la relación entre las variables explicativas y la variable respuesta del conjunto de datos de entrenamiento. Esto se podría hacer a través de una ecuación del tipo:

$$y = 2 + 3x_1 - x_2$$

donde y es la variable respuesta dada por el antígeno prostático, x_1 y x_2 son las variables explicativas dadas respectivamente por la cantidad de sangre en la orina y el nivel de glóbulos rojos.

Luego, usando el modelo anterior, y sabiendo que un nuevo paciente tiene una cantidad de sangre en la orina igual a 0.9 y un nivel de glóbulos rojos de 2.3, podemos predecir que su nivel de antígeno prostático, y, de la siguiente forma:

$$y = 2 + 3 \cdot 0.9 - 2.3 = 2.4$$

Ejemplo 1.3. Detección de fiebre en un paciente:

En este ejemplo, se ha medido la temperatura corporal de un paciente en ciertos días. Se sabe además que, usualmente, ese paciente no tiene fiebre y, por lo tanto, disponemos de más días en los que el paciente no tiene fiebre. Es por ello que podemos considerar como datos atípicos o *outliers*, aquellos datos (días) en los que el paciente tenga fiebre. El objetivo es encontrar un modelo de aprendizaje supervisado capaz de predecir a partir de qué temperatura el paciente tiene fiebre o no.

Más concretamente, consideramos la temperatura como la variable explicativa, mientras que tener fiebre o no será la variable respuesta.

Con esta información de los datos de entrenamiento, podemos generar un modelo de la forma:

$$\begin{array}{ll} \text{if} & \text{la temperatura es mayor que } 37.2°\text{C} \quad \text{then} \\ & \text{tiene fiebre} \\ \text{else} & \\ & \text{no tiene fiebre} \end{array} \tag{1.2}$$

Por lo tanto, si medimos la temperatura del paciente otro día y ésta es de 38°C, podremos concluir que el paciente tiene fiebre.

1.1. Aprendizaje Supervisado

El aprendizaje supervisado se basa en la teoría del aprendizaje estadístico [Vapnik, 1998, 1999]. Tal y como se ha comentado al principio del Capítulo 1, en el aprendizaje supervisado asumimos conocido un conjunto de datos de entrenamiento formado por las variables explicativas y las variables respuesta. En particular, a lo largo de este trabajo, asumiremos que nuestro conjunto de entrenamiento S está formado por N individuos de la forma $(\mathbf{x}^i, y^i)$ para $i = 1, ..., N$, donde $\mathbf{x}^i = (x_1^i, ..., x_d^i)^T \in \mathbb{R}^d$ son las variables explicativas del individuo i e $y^i \in \mathbb{R}$ es la variable respuesta del dato i. Volviendo al ejemplo de la detección de correos electrónicos spam (Ejemplo 1.1), tenemos que nuestro conjunto de entrenamiento está formado por $N = 20$ datos. En dicho conjunto, tenemos $d = 10$ variables explicativas. Por ejemplo, en el caso del dato i, $x_1^i =$ "*george*" y $x_{10}^i =$ "*edu*". Finalmente, la variable respuesta en este ejemplo sería "spam" ó "no spam", que podemos asociar a los valores -1 y 1, respectivamente. Es decir, la variable respuesta del dato i es tal que $y^i \in \{-1, 1\}$.

A partir de la información proporcionada por $\{(\mathbf{x}^i, y^i)\}$ para $i = 1, ..., N$, construimos un modelo de aprendizaje supervisado dado por una función $f : \mathbb{R}^d \longrightarrow \mathbb{R}$. Así podemos usar la función f para predecir el valor de la variable respuesta $\hat{y}$ de un nuevo punto no observado $\mathbf{x}^{test} \in \mathbb{R}^d$ de la siguiente forma:

$$\hat{y} = f(\mathbf{x}^{test})$$

Volviendo al ejemplo de la detección de correos spam, tenemos que, matemáticamente, la función de predicción en (1.1) puede escribirse como:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{si } x_1 < 0.6 \text{ y } x_2 > 1.3 \\ -1, & \text{en caso contrario} \end{cases} \tag{1.3}$$

Luego, de acuerdo con la función de predicción f en (1.3), y asumiendo que tenemos un nuevo correo electrónico (no observado anteriormente) con variables explicativas $\mathbf{x}^{test} = (0.4, 2.00, 0.98, 0.08, 1.02, 0.00, 2.67, 0.50, 0.14, 0.26)$, tenemos que $f(\mathbf{x}^{test}) = 1$ puesto que en este caso $x_1^{test} = 0.4 < 0.6$ y $x_2^{test} = 2 > 1.3$. Es decir, la predicción del nuevo correo electrónico es spam.

Se puede observar que en el ejemplo de los correos spam, hemos codificado una variable respuesta cualitativa (spam ó no spam) a tráves de una variable tipo $\{-1, 1\}$. Sin embargo, en el ejemplo del cáncer de próstata, la variable respuesta es el nivel de antígeno prostático. Es decir, en este caso, estamos antes una variable respuesta cuantitativa. Esta diferencia en cuanto al tipo de variable respuesta da lugar a dos tareas diferentes en el aprendizaje supervisado. Si la variable respuesta es cualitativa, hablaremos de clasificación. Por otro lado, si la variable respuesta es cuantitativa, estaremos ante una tarea de regresión. Además de estas dos tareas también nos encontramos con la tarea de la detección de *outliers*, que puede ser utilizada para la clasificación y la regresión, aunque en este trabajo sólo la estudiaremos para la tarea de clasificación. Más información sobre la detección de *outliers* en regresión puede ser consultada en [Krawczyk, 2016]. La Sección 1.2 incluye información más detallada sobre estos temas.

1.2. Clasificación, Regresión y Detección de Outliers

Como hemos visto en la sección anterior, en el aprendizaje supervisado hay diferentes tipos de tareas: clasificación, regresión y detección de *outliers*.

En la clasificación supervisada, la variable respuesta y^i es de tipo cualitativo. Estas variables suelen tomar valores en un conjunto de clases $\mathcal{C}$ contenido en el conjunto de los números enteros. Es decir, $y^i \in \mathcal{C} \in \mathbb{Z}$.

Además, a cada uno de los valores que puede tomar y^i se les llama etiquetas. Normalmente, si el conjunto $\mathcal{C}$ solo esta formado por dos valores hablaremos de clasificación binaria. Por otro lado, si el cardinal de $\mathcal{C}$ es mayor que tres, entonces nos encontraremos ante un problema de clasificación multiclase, [Tewari and Bartlett, 2007; Huang et al., 2011; Aiolli et al., 2005]. Por ejemplo, la variable respuesta del problema de detección de spam sólo puede tomar dos valores: 1 y -1. Es por ello, que nos encontramos ante un problema de clasificación binaria. Un ejemplo de clasificación multiclase, podría ser aquel en el que queremos predecir si un individuo se considera alto, medio o bajo a partir de la información proporcionada por su altura. En este caso, la variable respuesta tomaría tres valores y se podría codificar a través del conjunto $\mathcal{C} = \{1, 2, 3\}$.

Existen muchos métodos en la literatura capaces de construir reglas de clasificación supervisada. La mayoría de estos métodos se diseñaron originalmente para la clasificación binaria, aunque actualmente muchos se han generalizado para la clasificación multiclase. Un algoritmo de clasificación muy popular son las llamadas Máquinas de Vectores Soporte, más conocidas como SVM por sus siglas en inglés, *Support Vector Machine*. En el Capítulo 2 se dará una visión detallada de este clasificador lineal. Además, existen otros métodos de clasificación muy conocidos y potentes, como por ejemplo, el método de los k vecinos más cercanos (normalmente conocido como $k$nn por su nombre en inglés k

nearest neighbors), [Gou et al., 2019; Cunningham and Delany; Abu Alfeilat et al., 2019]. Este método asocia a un nuevo individuo la etiqueta que más se repite entre k puntos que se encuentran más próximos a él. Dicha proximidad o cercanía se mide a través de una matriz de distancias previamente definida. Una estrategia alternativa son los árboles de clasificación, [Bertsimas and Dunn, 2017; Carrizosa et al., 2021], los cuales se basan en reglas de tipo *if-then* como la que aparece en el Ejemplo 1.1. Éstos se caracterizan por su fácil interpretación. Por último, podemos destacar metodologías como los bosques aleatorios, [Breiman, 2001; Biau and Scornet, 2016], que se puede entender como un caso general de los árboles de clasificación, o las redes neuronales, [LeCun et al., 2015; Zhang et al., 2021], muy conocidas por su aplicación en problemas de la vida real.

Por otro lado, en la regresión supervisada el objetivo es hacer una predicción de una variable respuesta de tipo cuantitativo a través de un modelo construido a partir de los datos de entrenamiento. Por ejemplo, el problema de la detección del antígeno prostático visto anteriormente en el Ejemplo 1.2 es un caso particular de regresión supervisada.

La regresión lineal [Montgomery et al., 2021], es una de las estrategias más populares en la literatura para la predicción de una función de regresión. Es un método de aprendizaje muy simple pero muy útil y ampliamente utilizado. El algoritmo de regresión establecerá un modelo para ajustar la relación de dependencia entre las variables explicativas y la variable respuesta.

Existen otros métodos de regresión supervisada. Cabe destacar la Regresión de Vectores Soporte (SVR, por su siglas en inglés *Support Vector Regression*). En la Sección 3.1, haremos un estudio detallado sobre esta metodología.

En la clasificación y en la regresión asumimos que el número de datos en las clases consideradas es similar. Sin embargo, muchas ocasiones la distribución de los datos está sesgada pues los representantes de una clase aparecen con más frecuencia. Lo que supone una dificultad para los algoritmos de aprendizaje debido a que estarán sesgados hacia el grupo mayoritario. Sin embargo, hay ocasiones en las que la clase minoritaria es la más importante desde el punto de vista de la minería de datos ya que puede contener conocimientos importantes y útiles. Es por ello que junto con los problemas de clasificación y regresión en la práctica tenemos otro más, el problema de la descripción de datos.

Normalmente, los datos son creados por uno o más procesos generadores, que reflejan las observaciones recogidas sobre determinadas aplicaciones. Cuando el proceso generador se comporta de forma inusual se produce la creación de datos atípicos o *outliers*, es decir, los *outliers* son datos con características significativamente diferentes al resto de los datos de un conjunto. Por tanto, los *outliers* suele contener información útil sobre las características anormales de las aplicaciones que afectan al proceso de generación de datos.

Por lo tanto, a través del problema la descripción de datos llevaremos acabo la detección de *outliers* con el método SVDD, del inglés *Support Vector Data Description*, que veremos con más detalle en la Sección 3.2. Otros métodos para la detección de *outliers* basado en el aprendizaje supervisado, pueden consultarse en [Fernández et al., 2022].

Capítulo 2

Support Vector Machine

Las máquinas de vectores soporte (SVM, del inglés *Suppor Vector Machine*) tiene su origen en los trabajos sobre la teoría del aprendizaje estadístico, introducidos en los años 90 por Vapnik y sus colaboradores [Vapnik, 1998, 1999]. En un principio, el método SVM fue diseñado para resolver problemas de clasificación binaria, pero actualmente se han adaptado para la resolución de otro tipo de problemas como la regresión o la clasificación multiclase [Cortes and Vapnik, 1995]. Además, se han utilizado con éxito en diversos campos, entre ellos la visión artificial, el procesamiento de lenguaje natural o el análisis de series temporales [Suárez, 2014].

El objetivo principal del método SVM es encontrar un hiperplano de separación que esté a igual distancia de los datos de entrenamiento más cercanos de cada clase. Así, se conseguirá el llamado margen máximo a cada lado del hiperplano. Además, cabe destacar que, para definir dicho hiperplano, solo se tendrán en cuenta, los datos que están justo encima del margen máximo. A estos puntos se les llama vectores soporte. Veremos en las siguientes secciones más detalles sobre esta metodología.

2.1. Caso lineal

Vamos a comenzar estudiando el caso lineal, es decir, aquel en el que buscamos un hiperplano lineal que separe ambas clases. Si esta función lineal es capaz de separar perfectamente los datos de los dos grupos, entonces hablamos del problema con margen duro (normalmente conocido por su nombre en inglés *hard margin*). Por otro lado, si se permiten errores en la clasificación hablaremos del problema de margen suave (ó *soft margin*). Estos dos casos serán estudiados en profundidad en las Secciones 2.1.1 y 2.1.2, respectivamente.

2.1.1. SVM de margen duro (*hard margin*)

Dado un conjunto de datos $S = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), ..., (\mathbf{x}^N, y^N)\}$, donde $\mathbf{x}^i \in \mathbb{R}^d$ e $y^i \in \{-1, 1\}$ $\forall i$, diremos que los datos de S son linealmente separables si podemos encontrar un hiperplano definido a través de una función lineal de manera que todos los datos $\mathbf{x}^i$ con etiqueta $y^i = 1$ se encuentren a un lado del hiperplano. En consecuencia, todos los datos tal que $y^i = -1$ se encontrarán al otro lado del hiperplano. Este hiperplano es conocido como hiperplano de separación, $D(\mathbf{x})$, y su expresión viene dada por:

$$D(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = (w_1 x_1 + w_2 x_2 + ... + w_d x_d) + b = 0 \tag{2.1}$$

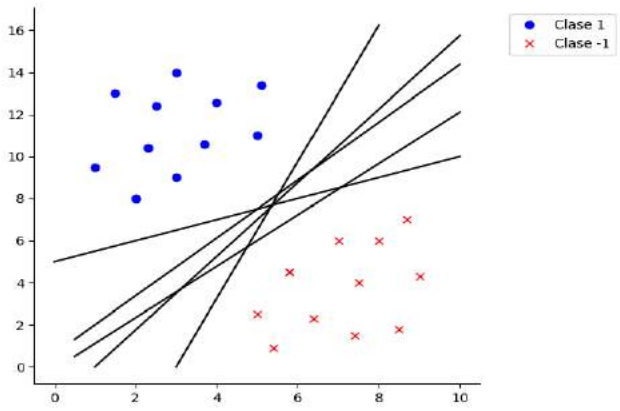


Figura 2.1: Algunos de los posibles hiperplanos de separación que separan perfectamente el conjunto de datos en dos clases.

donde $\mathbf{w} = (w_1, ..., w_d) \in \mathbb{R}^d$, $b \in \mathbb{R}$ y $\langle \cdot, \cdot \rangle$ denota el producto escalar.

Formalmente, el hiperplano de separación debe verificar las siguientes condiciones:

$$\langle \mathbf{w}, \mathbf{x}^i \rangle \ + \ b \ \geq \ 0 \quad \text{si } i \text{ es tal que } y^i = 1 \tag{2.2}$$

$$\langle \mathbf{w}, \mathbf{x}^i \rangle \ + \ b \ < \ 0 \quad \text{si } i \text{ es tal que } y^i = -1 \tag{2.3}$$

ó equivalentemente

$$y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle \ + \ b) \ \geq \ 0, \quad i = 1, ..., N \tag{2.4}$$

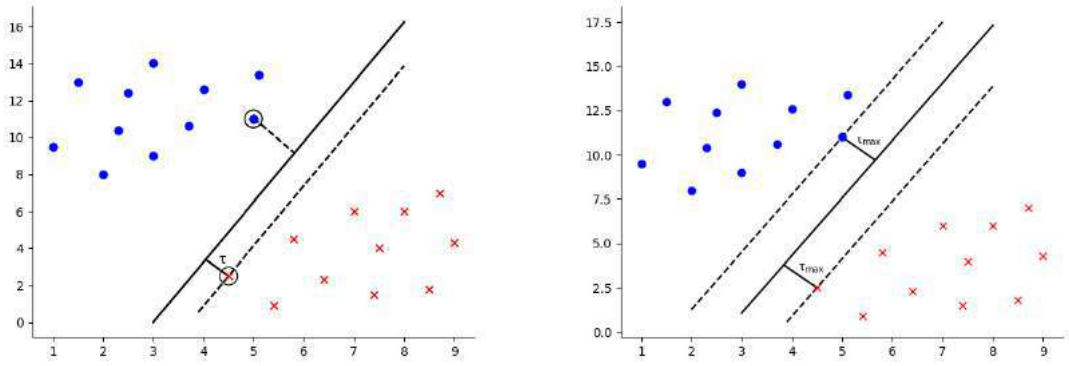
Dicho de otra forma, y usando la notación de hiperplano, tenemos

$$y^i D(\mathbf{x}^i) \geq 0, \quad i = 1, ..., N \tag{2.5}$$

De hecho, en la Figura 2.1 podemos observar algunos de los posibles hiperplanos que separan perfectamente a los puntos de las clase 1 (círculos azules) con los puntos de la clase -1 (cruces rojas). Es por lo tanto necesario encontrar una regla de decisión sobre la base de datos que nos permita hallar un hiperplano único y óptimo, de acuerdo a un criterio por determinar. Para ello, necesitamos definir el concepto de margen de un hiperplano de separación.

Definición 2.1. Sea $D(\mathbf{x})$ el hiperplano dado en (2.1). Definimos el margen de un hiperplano de separación como la mínima distancia entre dicho hiperplano y el dato más cercano a cualquiera de las dos clases. El margen de separación se denotará por τ.

Para ilustrar este concepto vamos a centrarnos en la Figura 2.2a. Observamos que se ha dibujado un hiperplano de la forma (2.1) que separa perfectamente ambas clases. Hemos marcado con un círculo el punto de cada clase más cercano a este hiperplano. Dado que la distancia de la cruz roja (clase -1) al hiperplano es menor que la distancia del círculo azul (clase 1) al hiperplano, entonces de acuerdo a la Definición 2.1, tomamos el margen τ como la primera distancia mencionada.



(a) (b)

Figura 2.2: Margen de un hiperplano de separación: (a) hiperplano de separación no óptimo, donde hemos marcado con un círculo el punto de cada clase más cercano a este hiperplano y (b) hiperplano de separación óptimo.

Veamos ahora como definimos el hiperplano de separación óptimo.

Definición 2.2. Sea $D(\mathbf{x})$ el hiperplano dado en (2.1). Definimos el hiperplano de separación óptimo como aquel que maximiza el margen, denominado éste como margen máximo.

Por tanto, el hiperplano de separación óptimo queda caracterizado por el siguiente resultado:

Proposición 2.1. [Suárez, 2014] Un hiperplano de separación se dirá óptimo si y solo si equidista del dato más cercano de cada clase.

Si observamos la Figura 2.2b, vemos que si el hiperplano es tal que τ se maximiza, debe estar justo en medio de las dos clases. Es decir, τ debe ser la distancia al punto más cercano de ambas clases. De lo contrario el hiperplano se podría mover hacia los puntos de la clase que estén más lejos y aumentar τ, lo que contradice que τ se maximice.

Veamos lo que implica la proposición 2.1 de forma matemática. La distancia entre un hiperplano de separación $D(\mathbf{x})$ y un dato de entrenamiento cualquiera $\mathbf{x}^i$ viene dada por la siguiente fórmula:

$$\frac{|D(\mathbf{x}^i)|}{\|\mathbf{w}\|} \tag{2.6}$$

siendo $|\cdot|$ el valor absoluto, $\|\cdot\|$ la norma de un vector y $\mathbf{w}$ el vector que junto con el parámetro b, definen el hiperplano $D(\mathbf{x})$ en la expresión (2.1).

Sea τ el margen del hiperplano de separación óptimo y haciendo uso de las expresiones (2.5) y (2.6), la distancia de cada uno de los datos al hiperplano de separación óptimo debe ser mayor o igual que el margen de dicho hiperplano, es decir,

$$\frac{y^i D(\mathbf{x}^i)}{\|\mathbf{w}\|} \geq \tau, \quad i = 1, ..., N \tag{2.7}$$

Se da la igualdad para los datos situados justo en la frontera que delimita el margen a cada lado del hiperplano de separación. A estos puntos los denominados vectores soporte, y verifican:

$$\frac{y^i D(\mathbf{x}^i)}{\|\mathbf{w}\|} = \tau, \quad \forall i \in VS \tag{2.8}$$

donde VS es el conjunto formado por todos los vectores soporte. Debido a que estos vectores son los más cercanos al hiperplano de separación, serán los datos más difíciles de clasificar. Es por ello que, tal y como veremos más adelante, van a ser los únicos puntos a considerar a la hora de construir el hiperplano.

A partir de la expresión (2.8), veremos que encontrar el hiperplano de separación óptimo, es decir, maximizar el margen, τ, es equivalente a minimizar $\|\mathbf{w}\|$. Para ello, comenzamos observando que existen infinitas formas de definir un mismo hiperplano diferenciándose solo en la escala de $\mathbf{w}$, es decir, dado un vector $\mathbf{w}^1$ que genera un hiperplano y $\mathbf{w}^2 = \lambda \mathbf{w}^1$, para cierto $\lambda \in \mathbb{R}$, tenemos que ambos generan el mismo hiperplano, aunque claramente tienen distinta norma. Dado que estamos interesados en la dirección del vector, y no en su norma, fijamos el producto de la escala de $\mathbf{w}$ y su margen a la unidad de la siguiente forma:

$$\tau \|\mathbf{w}\| = 1 \tag{2.9}$$

Así, limitamos el número de soluciones a una sola. Luego la condición (2.7) se transforma en:

$$y^i D(\mathbf{x}^i) \geq 1, \quad i = 1, ..., N \tag{2.10}$$

equivalentemente:

$$y^i (\langle \mathbf{w}, \mathbf{x}^i \rangle + \ b) \geq 1, \quad i = 1, ..., N \tag{2.11}$$

Además, por la condición (2.9) tenemos que

$$\text{máx} \ \ \tau = \text{máx} \ \ \frac{1}{\|\mathbf{w}\|} = \text{mín} \ \ \|\mathbf{w}\| = \text{mín} \ \ \frac{1}{2}\|\mathbf{w}\|^2 \tag{2.12}$$

Nótese que consideramos la última igualdad para facilitar la resolución del problema de optimización que detallaremos a continuación:

$$\begin{array}{ll} \min\limits_{\mathbf{w}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.a.} & y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) \geq 1, \quad i = 1, ..., N \end{array} \tag{2.13}$$

El problema (2.13) se utiliza para resolver el SVM de margen duro. Es decir, el problema de la búsqueda del hiperplano óptimo para el caso de la clasificación binaria de datos linealmente separables.

Para la resolución de este problema de optimización cuadrática haremos uso de la teoría de optimización dual [Boyd et al., 2004]. Según dicha teoría, un problema de optimización primal, tiene una forma dual asociada si la función a optimizar y las restricciones son estrictamente convexas. En este caso, se puede observar que tanto la función objetivo como las restricciones que forman el problema (2.13) son estrictamente convexas. La función objetivo por tratarse del cuadrado de una norma y las restricciones por ser funciones lineales. Por lo tanto, tenemos las condiciones para obtener el problema dual a partir de

su versión primal en (2.13). De esta forma, resolver de forma óptima el problema dual será equivalente a obtener la solución óptima del problema primal.

Para realizar la transformación del problema primal al dual necesitaremos definir la funcion de Lagrange.

Definición 2.3. Dado un problema de optimización con variables de decisión $\mathbf{w}$, función objetivo $h(\mathbf{w})$ y restricciones $g_i(\mathbf{w}) = 0$, $i = 1, ..., N$, se define la función lagrangiana ó función de Lagrange, $L(\cdot, \cdot)$, como

$$L(\mathbf{w}, \boldsymbol{\alpha}) = h(\mathbf{w}) + \sum_{i=1}^{N} \alpha_i g_i(\mathbf{w}) \tag{2.14}$$

donde $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)$ con $\alpha_i \geq 0$, $\forall i$. Los coeficientes α_i son los llamados multiplicadores de Lagrange.

Utilizando la función lagrangiana de la Definición 2.3, podemos expresar de forma equivalente el problema de optimización primal (2.13) en su versión dual como:

$$\begin{array}{ll} \max\limits_{\boldsymbol{\alpha}} & \inf\limits_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) \\ \text{s.a.} & \alpha_i \geq 0, \quad i = 1, ..., N \end{array} \tag{2.15}$$

A continuación, enunciaremos varios resultados que establecen una relación entre las soluciones óptimas de los problemas en sus versiones primal y dual. Más información sobre estos resultados y sus demostraciones puede encontrarse en [Fletcher, 2013].

Teorema 2.1. Sean $\mathbf{w}$ y $\boldsymbol{\alpha}$ vectores tales que satisfacen las restricciones de los problemas primal y dual, es decir, $y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) \leq 0$ en (2.13) y $\alpha_i \geq 0$ en (2.15), con $i = 1, ..., N$. Entonces la imagen del vector $\boldsymbol{\alpha}$ mediante la función objetivo del problema dual, $p(\boldsymbol{\alpha}) = \inf\limits_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha})$, es menor o igual que la imagen del vector $\mathbf{w}$ mediante la función objetivo del primal, $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$, es decir, $p(\boldsymbol{\alpha}) \leq h(\mathbf{w})$.

Este teorema nos lleva al siguiente corolario:

Corolario 2.1. Si la imagen del vector $\boldsymbol{\alpha}$ mediante la función objetivo del problema dual es igual que la imagen del vector $\mathbf{w}$ mediante la función objetivo del primal. Es decir, $p(\boldsymbol{\alpha}) = h(\mathbf{w})$, entonces $\boldsymbol{\alpha}$ y $\mathbf{w}$ son soluciones óptimas, respectivamente, del problema dual y primal, siendo $p(\boldsymbol{\alpha})$ y $h(\mathbf{w})$ las funciones definidas en el Teorema 2.1

El siguiente teorema permite establecer los pasos y condiciones necesarias para poder obtener la solución óptima del problema primal a partir del resultado óptimo del problema dual.

Teorema 2.2. (Teorema de Karush-Kuhn-Tucker, KKT). Si en el problema primal, la función objetivo $h : \mathbb{R}^d \to \mathbb{R}$ y las restricciones $g_i : \mathbb{R}^d \to \mathbb{R}$, $i = 1, ..., N$ son convexas y existen constantes $\alpha_i \geq 0$, $i = 1, ..., N$ tales que:

$$\frac{\partial h(\mathbf{w}^*)}{\partial w_j} + \sum_{i=1}^{N} \alpha_i \frac{\partial g_i(\mathbf{w}^*)}{\partial w_j} = 0, \quad j = 1, ..., d \tag{2.16}$$

$$\alpha_i g_i(\mathbf{w}^*) = 0, \quad i = 1, ..., N \tag{2.17}$$

Entonces el punto $\mathbf{w}^*$ es el mínimo global del problema de optimización primal.

Por el Teorema 2.2 podemos hallar el problema dual a partir del problema de optimización primal (2.13). Comenzamos construyendo la función lagrangiana:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) - 1] \tag{2.18}$$

donde $\alpha_i \geq 0$ son los multiplicadores de Lagrange.

Ahora buscamos α_i que verifiquen las hipótesis del Teorema 2.2. Para ello derivamos la función lagrangiana con respecto de las variable $\mathbf{w}$ y b, e igualamos a cero:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i = \mathbf{0} \tag{2.19}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{N} \alpha_i y^i = 0 \tag{2.20}$$

Ahora, multiplicamos los multiplicadores de Lagrange α_i por las restricciones $g_i(\mathbf{w}) = y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) - 1$ e igualamos a cero, resultando:

$$\alpha_i [y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) - 1] = 0 \quad i = 1, ..., N \tag{2.21}$$

De la ecuación (2.19) obtenemos la expresión de $\mathbf{w}$ en términos de los multiplicadores de Lagrange, α_i:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i \tag{2.22}$$

y de la ecuación (2.20), se establece una restricción para los multiplicadores de Lagrange

$$\sum_{i=1}^{N} \alpha_i y^i = 0 \tag{2.23}$$

A continuación, expresamos la función lagrangiana (2.18) sólo en función de los α_i gracias a la igualdad obtenida en (2.22):

$$L(\boldsymbol{\alpha}) = \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i\right)\left(\sum_{\ell=1}^{N} \alpha_\ell y^\ell \mathbf{x}^\ell\right) - \left(\sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i\right)\left(\sum_{\ell=1}^{N} \alpha_\ell y^\ell \mathbf{x}^\ell\right) - b\sum_{i=1}^{N} \alpha_i y^i + \sum_{i=1}^{N} \alpha_i \tag{2.24}$$

Reordenando términos, obtenemos:

$$L(\boldsymbol{\alpha}) = \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i\right)\left(\sum_{\ell=1}^{N} \alpha_\ell y^\ell \mathbf{x}^\ell\right) - \left(\sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i\right)\left(\sum_{\ell=1}^{N} \alpha_\ell y^\ell \mathbf{x}^\ell\right) + \sum_{i=1}^{N} \alpha_i \tag{2.25}$$

$$= -\frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i\right)\left(\sum_{\ell=1}^{N} \alpha_\ell y^\ell \mathbf{x}^\ell\right) + \sum_{i=1}^{N} \alpha_i \tag{2.26}$$

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i,\,\ell=1}^{N} \alpha_i \alpha_\ell y^i y^\ell \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle \tag{2.27}$$

Sustituyendo (2.27) en (2.15) y dado que la función lagrangiana no depende de $\mathbf{w}$, podemos construir el problema dual que quedará de la siguiente forma:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,\ell=1}^{N} \alpha_i \alpha_\ell y^i y^\ell \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle \\ \text{s.a.} \quad & \sum_{i=1}^{N} \alpha_i y^i = 0 \\ & \alpha_i \geq 0, \quad i = 1, ..., N \end{aligned} \tag{2.28}$$

El problema (2.28) es un problema de optimización convexa, puesto que maximiza una función objetivo convexa con restricciones lineales. Al igual que el problema primal, el problema dual se resuelve mediante la programación cuadrática.

Luego, la solución del problema dual, $\boldsymbol{\alpha}^{opt}$, nos permitirá obtener la solución del problema primal. Para ello, bastará con sustituir la solución, $\boldsymbol{\alpha}^{opt}$, en la expresión (2.22), obteniendo:

$$\mathbf{w}^{opt} = \sum_{i=1}^{N} \alpha_i^{opt} y^i \mathbf{x}^i \tag{2.29}$$

Sustituyendo la solución óptima $\mathbf{w}^{opt}$ en (2.1), obtendremos el hiperplano óptimo a partir del problema dual:

$$D(\mathbf{x}) = \langle \mathbf{w}^{opt}, \mathbf{x} \rangle + b = \sum_{i=1}^{N} \alpha_i^{opt} y^i \langle \mathbf{x}^i, \mathbf{x} \rangle + b, \quad \text{para } \mathbf{x} \in \mathbb{R}^d \tag{2.30}$$

Ya hemos visto cómo se obtiene la variable de decisión óptima $\mathbf{w}^{opt}$ a partir del problema dual. Veamos ahora como calcular el valor óptimo de b. Para ello haremos uso de la expresión (2.21) y del hecho de que los multiplicadores de Lagrange son no negativos, es decir, $\alpha_i \geq 0$, $\forall i$.

Comenzaremos suponiendo que no existe ningún i tal que $\alpha_i \neq 0$, es decir, $\alpha_i = 0$ para todo $i = 1, ..., N$. Aplicando esto en la ecuación (2.29) obtenemos que $\mathbf{w}^{opt} = \mathbf{0}$ por lo que el hiperplano de separación (2.30) solo depende de b, $D(\mathbf{x}) = b$. Esto implica que si $b \geq 0$, todos los datos, independientemente de su clase, serán etiquetados con $y^i = 1$. Por el contrario si $b < 0$, todos los datos serán etiquetados con $y^i = -1$. Esto daría lugar a una clasificación degenerada que no queremos que ocurra. Es decir, existe al menos un i tal que $\alpha_i > 0$ con $i = 1, ..., N$.

Como existe al menos un dato con índice i tal que $\alpha_i > 0$, por (2.21) obtenemos que:

$$y_i(\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + b) = 1 \tag{2.31}$$

por lo tanto, se puede afirmar que todos aquellos datos i que tengan asociado un valor $\alpha_i > 0$ cumplen con igualdad la restricción i-ésima dada por (2.11). Como habíamos fijado a la unidad la escala del producto de τ y la norma de $\mathbf{w}$ en (2.9), la expresión (2.8) es equivalente a la expresión (2.31). Por otro lado, los vectores que cumple la expresión (2.8) son los vectores soporte (ver Figura 2.3) y por (2.31), serán los datos que tengan asociado un valor $\alpha_i > 0$.

En consecuencia, el resto de datos del conjunto de entrenamiento tendrán asociado un valor $\alpha_i = 0$, por lo que no intervienen en la construcción del hiperplano de separación, dado que el correspondiente sumando en la expresión (2.29) será nulo.

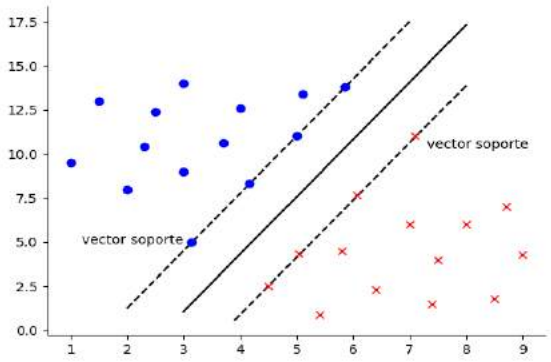


Figura 2.3: Vectores soporte

Utilizando estos resultados, para conocer el valor de b, bastará con despejarlo de la restricción (2.31). Para ello, multiplicamos dicha expresión por y^i, siendo i un vector soporte cualquiera:

$$(y^i)^2(\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + b) = y^i \tag{2.32}$$

Como $y^i \in \{-1, 1\}\ \forall i$, entonces $(y^i)^2 = 1$, $\forall i$. Por tanto, se tiene,

$$\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + b = y^i \tag{2.33}$$

Por último, despejamos para obtener el valor óptimo de b

$$b^{opt} = y^i - \langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle, \quad \forall i \in VS \tag{2.34}$$

La expresión de b puede obtenerse a partir de cualquier vector soporte. Es por ello que en la práctica se suele usar un enfoque más robusto usando todos los vectores soporte. Obtendremos b con la siguiente expresión:

$$b^{opt} = \frac{1}{N_{VS}} \sum_{i \in VS} (y^i - \langle \mathbf{w}^*, \mathbf{x}^i \rangle) \tag{2.35}$$

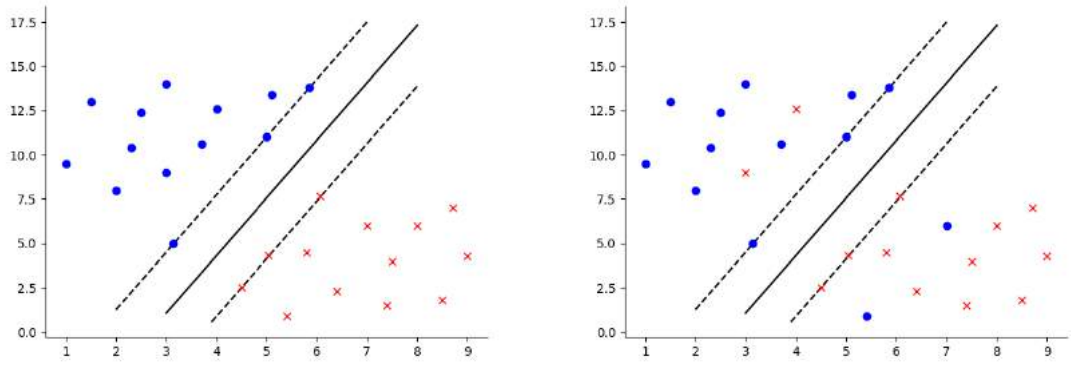
donde N_{VS} es el cardinal del conjunto de vectores soporte.

Luego, para clasificar un nuevo dato no observado anteriormente, $\mathbf{x}^{test}$, bastará con evaluar el punto $\mathbf{x}^{test}$ en el hiperplano de separación óptimo, que se obtiene en las Ecuaciones (2.29), (2.30) y (2.35), resultando que:

- Si $D(\mathbf{x}^{test}) \geq 0$, el nuevo dato $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = 1$.
- Si $D(\mathbf{x}^{test}) < 0$, el nuevo dato $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = -1$.

2.1.2. SVM de margen blando (*soft margin*)

Dado un conjunto de datos $S = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), ..., (\mathbf{x}^N, y^N)\}$, donde $\mathbf{x}^i \in \mathbb{R}^d$ e $y^i \in \{-1, 1\}$, es muy frecuente que los datos del conjunto S no sean separados perfectamente mediante una función lineal, como ocurría en el caso *hard margin*. Es decir, que existan algunos puntos de S que se han clasificado incorrectamente usando un hiperplano



(a) (b)

Figura 2.4: Diferentes tipos de hiperplanos obtenidos con un modelo (a) SVM de margen duro y (b) SVM de margen blando.

lineal. Cortes y Vapnik, [Cortes and Vapnik, 1995], demostraron en 1995 que este tipo de problemas son más fáciles de tratar si permitimos que algunos datos no verifiquen las restricciones sobre el margen formuladas en el problema de optimización primal (2.13). Es decir, si permitimos algunos errores en la clasificación de los puntos. Luego, a cada lado del hiperplano podemos encontrar tanto datos con etiqueta $y^i = 1$ como datos con etiqueta $y^i = -1$. Por eso, ahora vamos a considerar el caso en el que los datos de entrenamiento no pueden separarse sin error mediante un hiperplano, conocido como el hiperplano de separación de margen blando (*soft margin*). En este caso, diremos que el conjunto de datos es no separable linealmente. En la Figura 2.4, podemos ver la diferencia entre el caso *hard margin* SVM (Figura 2.4a) y el caso *soft margin* SVM (Figura 2.4b).

En esta sección, generalizaremos el problema de optimización visto anteriormente para el caso lineal no separable. Para ello introducimos, para cada dato i, las siguientes variables de decisión:

$$\xi_i \geq 0, \quad i = 1, ..., N \tag{2.36}$$

llamadas variables de holgura. Estas variables indican la desviación del punto i con respecto al hiperplano que define su clase.

Las nuevas restricciones que deberán cumplir los datos son:

$$y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) \geq 1 - \xi_i \quad \text{si } i \text{ es tal que } y^i = 1 \tag{2.37}$$

$$y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) \leq -1 + \xi_i \quad \text{si } i \text{ es tal que } y^i = -1 \tag{2.38}$$

De acuerdo con estas restricciones, si $\xi_i = 0$ significa que el punto $\mathbf{x}^i$ está correctamente clasificado de acuerdo a su etiqueta y^i. Por otro lado, si $\xi_i > 0$, se pueden dar dos situaciones para el dato i: si $0 < \xi_i \leq 1$, entonces el dato i se encuentra dentro del margen asociado a la clase correcta por lo que está bien clasificado. En caso contrario, si $\xi_i > 1$, entonces el dato se encuentra al otro lado del hiperplano. Luego el dato está mal clasificado. Ver Figura 2.5 para una información más detallada.

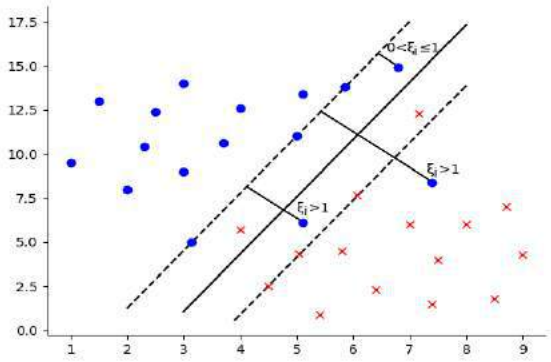


Figura 2.5: Valores de las variables de holgura, ξ_i, para los puntos mal clasificados. Para los datos situados dentro del margen asociado a la clase correcta el valor de la variable de holgura es $0 < \xi_i \leq 1$ y para los datos que están situados al otro lado del hiperplano es $\xi_i > 1$. Aunque en la imagen los valores han sido colocados para los datos representados en azul, para los datos representados en rojo es análago.

El nuevo problema de optimización deberá tener en cuenta, no sólo minimizar la norma de $\mathbf{w}$ (es decir, maximizar el margen) sino también minimizar los errores cometidos al clasificar. Es por ello que la nueva función objetivo es de la forma:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \tag{2.39}$$

donde $C \geq 0$ es una constante, denominada parámetro de regularización, que permite controlar cómo influye la penalización de los errores a la función objetivo. Para C suficientemente grande, el vector $\mathbf{w}$ y b determinan el hiperplano que minimiza el número de errores en el conjunto de entrenamiento con margenes más pequeños. Mientras que para un C pequeño, los hiperplanos tendrán márgenes mayores, a costa de tener más errores de clasificación en el conjunto de entrenamiento.

En consecuencia, el nuevo problema de optimización, dado por la función objetivo (2.39) y las restricciones (2.37), (2.38) y $\xi_i \geq 0$ tiene la siguiente forma:

$$\begin{array}{ll} \underset{\mathbf{w},b,\xi_i,\forall i}{\text{mín}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.a.} & y^i(\langle \mathbf{w}, \mathbf{x}^i\rangle + b) + \xi_i - 1 \geq 0, \quad i = 1, ..., N \\ & \xi_i \geq 0, \quad i = 1, ..., N \end{array} \tag{2.40}$$

El procedimiento para obtener el hiperplano en este caso es muy similar al caso anterior. Al igual que en el caso anterior de la Sección 2.1.1, estamos en condiciones de aplicar la teoría de dualidad ya que estamos ante un problema de programación convexa. Por tanto, obtendremos la solución óptima del problema primal a través de su versión dual. Para ello, construimos la función lagrangiana:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i[y^i(\langle \mathbf{w}, \mathbf{x}^i\rangle + b) + \xi_i - 1] - \sum_{i=1}^{N}\beta_i\xi_i \tag{2.41}$$

donde $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)$ y $\boldsymbol{\beta} = (\beta_1, ..., \beta_N)$. Además, α_i y β_i, para $i = 1, ..., N$, son los multiplicadores de Lagrage asociados a las restricciones del problema (2.40). Por último, $\boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_N)$ son las variables de holgura.

Aplicamos la primera condición (2.17) del Teorema 2.2:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i = \mathbf{0} \tag{2.42}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial b} = \sum_{i=1}^{N} \alpha_i y^i = 0 \tag{2.43}$$

$$\frac{\partial L(\mathbf{w}, b, \xi, \boldsymbol{\alpha}, \beta)}{\partial \xi_i} = C - \alpha_i - \beta_i = 0, \quad i = 1, ..., N \tag{2.44}$$

A continuación, multiplicamos los multiplicadores de Lagrange α_i por las restricciones del problema (2.40), resultando:

$$\alpha_i[y^i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) + \xi_i - 1] = 0, \quad i = 1, ..., N \tag{2.45}$$

$$\beta_i \xi_i = 0, \quad i = 1, ..., N \tag{2.46}$$

De la ecuación (2.42) obtenemos la expresión de $\mathbf{w}$ en términos de los multiplicadores de Lagrange, α_i:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y^i \mathbf{x}^i \tag{2.47}$$

y de las ecuaciones (2.43) y (2.44), se establecen restricciones para los multiplicadores de Lagrange:

$$\sum_{i=1}^{N} \alpha_i y^i = 0 \tag{2.48}$$

$$C = \alpha_i + \beta_i, \quad i = 1, ..., N \tag{2.49}$$

Gracias a la igualdad obtenida en (2.49) podemos expresar las variables β_i en función de α_i. Como consecuencia de esto y por la igualdad (2.47), el problema dual puede escribirse sólo en función de las variables α_i. Las restricciones del problema se obtienen de las expresiones (2.48) y (2.49). Como α_i y β_i son multiplicadores de Lagrange, $\alpha_i \geq 0$ y $\beta_i \geq 0$. De la expresión (2.49), sabemos que $0 \leq \alpha_i = C - \beta_i$ y dado que $\beta_i \geq 0$, $C - \beta_i \leq C$. Por lo tanto $0 \leq \alpha_i \leq C$. Luego, podemos construir el problema dual sólo en función de las variables de decisión α_i de la siguiente forma:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i,\ell=1}^{N} \alpha_i \alpha_\ell y^i y^\ell \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle \\ \text{s.a.} \quad & \sum_{i=1}^{N} \alpha_i y^i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, ..., N \end{aligned} \tag{2.50}$$

El problema (2.50) se trata de un problema de optimización convexo. Al igual que en el problema de margen duro de la Sección 2.1.1, una vez resuelto el problema dual y

obtenida la solución del mismo, $\boldsymbol{\alpha}^{opt}$, podemos obtener la solución del problema primal. Para ello sustituimos la solución del problema dual en la expresión (2.47), obteniendo:

$$\mathbf{w}^{opt} = \sum_{i=1}^{N} \alpha_i^{opt} y^i \mathbf{x}^i \tag{2.51}$$

Sustituyendo la solución del problema primal en (2.1), obtendremos el hiperplano de separación óptimo

$$D(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^{opt} y^i \langle \mathbf{x}^i, \mathbf{x} \rangle + b \tag{2.52}$$

Por último, veremos cuáles son los vectores soporte para el método SVM de margen blando y a partir de esto obtendremos la expresión óptima de b. Para ello fijémonos que por el mismo razonamiento que en el caso *hard margin* SVM debe existir al menos un dato i con $\alpha_i > 0$. Además, recordemos que en el SVM con margen blando, α_i satisface que $0 \leq \alpha_i \leq C$, $\forall i$. Por lo que veamos que existe al menos un dato i con $\alpha_i = C$. Supongamos que todos los datos $\alpha_i \neq C$, es decir, $0 \leq \alpha_i < C$. Entonces C no puede valer cero, porque en ese caso α_i también sería cero. Por la ecuación (2.49) $C - \alpha_i = \beta_i$ entonces $\beta_i > 0$. Por lo tanto, de la ecuación (2.46) obtenemos que $\xi_i = 0$ para todo valor de i. Luego todos los datos están bien clasificados y estaríamos en el caso SVM con margen duro.

Por lo tanto, existe al menos un dato i tal que $0 < \alpha_i \leq C$. Veamos qué ocurre en esta situación distinguiendo dos casos, uno cuando $0 < \alpha_i < C$ y otro cuando $\alpha_i = C > 0$.

Comenzaremos con el caso en el que $\alpha_i = C$. Entonces $C > 0$. Por la expresión (2.49), tenemos que $\beta_i = 0$. Debido a esto y usando la expresión (2.46) obtenemos que $\xi_i > 0$. Como α_i es distinto de cero, de la expresión (2.45) se obtiene:

$$y^i(\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + \ b) + \xi_i - 1 = 0 \tag{2.53}$$

es decir,

$$y^i D(\mathbf{x}^i) = 1 - \xi_i \tag{2.54}$$

En este caso hay dos tipos de datos de entrenamiento. Como vimos al principio de la sección, si $0 < \xi_i \leq 1$ entonces $y^i D(\mathbf{x}^i) \geq 0$, y por tanto, el dato $\mathbf{x}^i$ está dentro del margen pero está bien clasificado. Si $\xi_i > 1$ entonces $y^i D(\mathbf{x}^i) < 0$, luego el dato está mal clasificado.

A continuación, tenemos el caso en que $0 < \alpha_i < C$. De la expresión (2.49) concluimos que $\beta_i \neq 0$ y por tanto, de la restricción (2.46), se deduce que $\xi_i = 0$. Conocido esto y por la expresión (2.45), se obtiene que

$$y^i(\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + \ b) - 1 = 0 \tag{2.55}$$

es decir,

$$y^i D(\mathbf{x}^i) = 1 \tag{2.56}$$

Luego, el dato $\mathbf{x}^i$ pertenece a la frontera del margen. Por tanto, los datos con $0 < \alpha_i < C$ son los datos conocidos como vectores soporte del hiperplano de separación de margen blando.

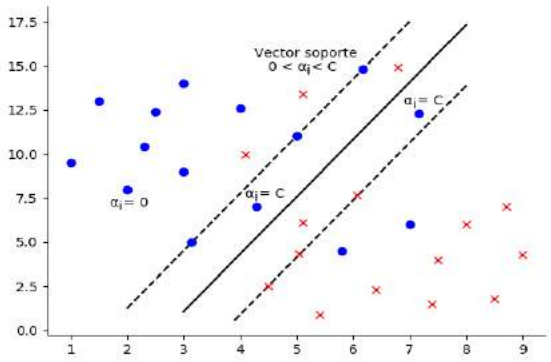


Figura 2.6: Valores de los multiplicadores de Lagrange. El valor de cada α_i depende de la posición en la que se encuentre el dato i respecto de las fronteras del margen y del hiperplano. Para ver la grafica más clara solo se han representado los α_i de los datos representados en azul, para los puntos rojos se obtienen conclusiones análogas.

Aparte de estos dos casos, también se puede dar la situación en la que el dato i satisface que $\alpha_i = 0$. Sin embargo, tal y como vimos en el caso *hard margin* de la Sección 2.1.1, los vectores cuyos $\alpha_i = 0$ no intervienen en la construcción del hiperplano de separación, debido a que el correspondiente sumando en la expresión (2.52) será nulo.

Cada uno los casos que acabamos de estudiar según el valor de α_i podemos verlos gráficamente en la Figura 2.6.

Luego, los vectores que se utilizan para construir el hiperplano son los vectores soporte y los vectores con $\alpha_i = C$, donde $C > 0$. Pues ambos tipos de vectores tienen $\alpha_i > 0$. Además, en este caso, los vectores soporte que tienen $0 < \alpha_i < C$, y $\xi_i = 0$ por lo que de la ecuación (2.45) se deduce

$$y^i(\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + \ b) - 1 = 0 \tag{2.57}$$

Y de aquí podemos obtener una expresión para b, de la misma forma que en el caso *hard margin*:

$$b^{opt} = y^i - \langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle, \quad \forall i \in VS \tag{2.58}$$

Aunque en la práctica se utiliza una expresión más robusta que tiene en cuenta todos los vectores soporte:

$$b^{opt} = \frac{1}{N_{VS}} \sum_{i \in VS} (y^i - \langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle) \tag{2.59}$$

donde VS es el conjunto formado por los vectores soporte y N_{VS} es el cardinal de dicho conjunto.

Luego, al igual que en el caso anterior, para clasificar un nuevo dato no observado anteriormente, $\mathbf{x}^{test}$, bastará con evaluar el punto $\mathbf{x}^{test}$ en el hiperplano de separación óptimo, que se obtiene en las Ecuaciones (2.51), (2.52) y (2.59), resultando que:

- Si $D(\mathbf{x}^{test}) \geq 0$, el nuevo dato $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = 1$.
- Si $D(\mathbf{x}^{test}) < 0$, el nuevo dato $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = -1$.

2.2. Problema no lineal

En las secciones 2.1.1 y 2.1.2 hemos trabajado con datos que se podían clasificar únicamente usando hiperplanos de separación lineales, bien sea a través del llamado margen duro ó margen blando. Sin embargo, existen ejemplos en los que una función lineal no es suficiente para encontrar una buena separación. Por ejemplo, en la Figura 2.7 puede verse una base de datos cuyos puntos son claramente separables a través de una circunferencia, que es una función no lineal, y que por lo tanto no podemos obtener con los resultados vistos hasta el momento.

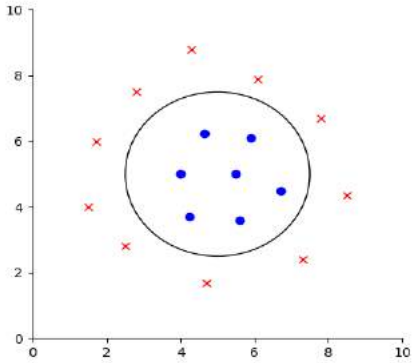


Figura 2.7: Hiperplano no lineal

Debido a que los resultados obtenidos en las secciones anteriores no son directamente aplicables cuando los ejemplos para clasificar no son linealmente separables, en esta sección adaptaremos los conceptos ya estudiados del caso lineal al caso no lineal. Para ello, hacemos una transformación de nuestros datos $\mathbf{x}^i$, $i = 1, ..., N$, en un espacio de dimensión superior F, llamado espacio de características (*feature space*), a través de la función $\phi : \mathbb{R}^d \longrightarrow F$, conocida por su nombre en inglés como *feature map*.

La función $\phi : \mathbb{R}^d \longrightarrow F$ hace corresponder cada punto $\mathbf{x}^i$ para $i = 1, ..., N$ con un punto en el espacio de características F de forma que los puntos transformados $\{\phi(\mathbf{x}^1), \phi(\mathbf{x}^2), ..., \phi(\mathbf{x}^N)\}$ son linealmente separables en el nuevo espacio F. Como podemos observar, en la Figura 2.8, $N = 17$ puntos han sido trasladados al espacio de características $F = \mathbb{R}^3$ a través de una función ϕ.

Como en el espacio de características F, los nuevos puntos $\{\phi(\mathbf{x}^1), \phi(\mathbf{x}^2), ..., \phi(\mathbf{x}^N)\}$ son linealmente separables, podemos aplicar los métodos SVM de *hard margin* y de *soft margin* visto en las secciones 2.1.1 y 2.1.2 respectivamente, para buscar un hiperplano de separación óptimo en el nuevo espacio. Para evitar repeticiones innecesarias, y dada su gran uso en aplicaciones prácticas, en este trabajo sólo nos enfocamos en el problema SVM no lineal de margen blando. Nótese que ahora nuestro conjunto de entrenamiento S es tal que $S = \{(\phi(\mathbf{x}^i), y^i)\}_{i=1,...,N}$.

Así, el hiperplano de separación lineal quedaría:

$$D(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \tag{2.60}$$

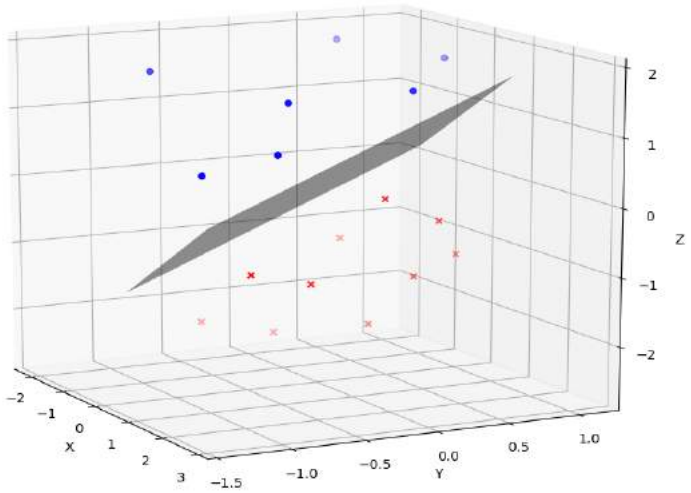


Figura 2.8: Ejemplo de un espacio de características, F. En este caso, tenemos $F = \mathbb{R}^3$. Como vemos en la Figura 2.7, los datos no pueden ser separados linealmente en el espacio original $\mathbb{R}^2$. Por lo que transformamos dichos datos mediante una función ϕ, y los trasladamos a $\mathbb{R}^3$, donde se pueden separar perfectamente con un hiperplano lineal.

Por tanto, siguiendo el mismo procedimiento que en la Sección 2.1.2, el problema SVM primal con *soft margin* sería:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i,\forall i} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \; + \; C\sum_{i=1}^{N}\xi_i \\ \text{s.a.} \quad & y^i(\langle \mathbf{w}, \phi(\mathbf{x}^i)\rangle + b) \leq 1 - \xi_i \;, \quad i = 1, ..., N \\ & \xi_i \geq 0 \;, \quad i = 1, ..., N \end{aligned} \tag{2.61}$$

Y de la misma forma la resolución del problema primal se realiza a través de su problema dual:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N}\alpha_i \; + \; \tfrac{1}{2}\sum_{i,\ell=1}^{N}\alpha_i\alpha_\ell y^i y^\ell \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle \\ \text{s.a.} \quad & \sum_{i=1}^{N}\alpha_i y^i = 0 \\ & 0 \leq \alpha_i \leq C \;, \quad i = 1, ..., N \end{aligned} \tag{2.62}$$

Luego, la expresión el hiperplano sería

$$D(\mathbf{x}) = \sum_{i=1}^{N}\alpha_i^{opt} y^i \langle \phi(\mathbf{x}^i), \phi(\mathbf{x})\rangle + b^{opt} \tag{2.63}$$

siendo $\boldsymbol{\alpha}^{opt}$ la solución óptima de (2.62), y b^{opt} obtenida de forma análoga que en el problema con *soft margin* de la Sección 2.1.2.

Nótese que no es trivial averiguar la expresión de ϕ. Sin embargo, en el problema dual (2.62), vemos que no es necesario saber esta expresión, sino únicamente el valor del producto escalar $\langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle \; \forall i, \ell$. Es por ello, que se recurre al concepto de Kernel ó núcleo que veremos en la siguiente sección.

2.3. Truco del kernel

Como hemos visto en la Sección 2.2, la función objetivo del problema (2.62), y el hiperplano óptimo de (2.63) sólo dependen del producto escalar $\langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle$. El truco del kernel ó nucleo consiste en sustituir dicho producto escalar por una función llamada función kernel $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$. Esta función asigna un valor real a cada par de puntos $\mathbf{x}^i, \mathbf{x}^\ell \ \forall i, \ell$. De esta forma, se realiza de forma implícita a través de la función ϕ una transformación no lineal de los datos a un espacio de características F. En este espacio F, los datos transformados serán linealmente separables. Por lo tanto, el truco kernel nos permitirá resolver el problema SVM no lineal (2.62) sin necesidad de conocer ϕ.

Pasemos ahora a dar una definición formal de la función kernel.

Definición 2.4. Una función kernel es una aplicación $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ tal que para todo par de puntos $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$, se tiene:

$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle$$

donde $\phi : \mathbb{R}^d \longrightarrow F$

Por lo tanto, el problema dual del SVM no lineal (2.62) se puede expresar como:

$$\begin{array}{ll} \max\limits_{\alpha} & \sum\limits_{i=1}^{N} \alpha_i \ - \ \frac{1}{2} \sum\limits_{i,\ell=1}^{N} \alpha_i \alpha_\ell y^i y^\ell k(\mathbf{x}^i, \mathbf{x}^\ell) \\ \text{s.a.} & \sum\limits_{i=1}^{N} \alpha_i y^i = 0 \\ & 0 \leq \alpha_i \leq C \ , \quad i = 1, ..., N \end{array} \tag{2.64}$$

Obteniéndose el siguiente hiperplano de separación

$$D(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i^{opt} y^i k(\mathbf{x}^i, \mathbf{x}) + b^{opt} \tag{2.65}$$

Consecuentemente, un nuevo dato, $\mathbf{x}^{test}$, tendrá etiqueta $y^i = 1$ si y solo si $D(\mathbf{x}^{test}) \geq 0$. En caso contrario tendrá etiqueta $y^i = -1$.

El siguiente resultado nos proporciona las propiedades que debe tener una función para ser función kernel.

Lema 2.1. Una función kernel $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ debe cumplir las siguiente propiedades:

a) Ser simétrica para todo $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$. Es decir:

$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle = \langle \phi(\mathbf{x}^\ell), \phi(\mathbf{x}^i) \rangle = k(\mathbf{x}^\ell, \mathbf{x}^i)$$

b) Satisfacer las desigualdades que derivan de la desigualdad de Cauchy-Schwarz. Formalmente

$$\begin{aligned} k(\mathbf{x}^i, \mathbf{x}^\ell)^2 &= \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle^2 \leq \|\phi(\mathbf{x}^i)\|^2 \|\phi(\mathbf{x}^\ell)\|^2 \\ &= \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^i) \rangle \langle \phi(\mathbf{x}^\ell), \phi(\mathbf{x}^\ell) \rangle = k(\mathbf{x}^i, \mathbf{x}^i) k(\mathbf{x}^\ell, \mathbf{x}^\ell) \end{aligned}$$

Demostración. Sea $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ una función kernel y $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$, veamos que k verifica las propiedades anteriores:

a) Es trivial debido a la propia simetría del producto escalar.

b) Consideremos $\epsilon > 0$ y la norma del vector $(\|\phi(\mathbf{x}^\ell)\| + \epsilon)\phi(\mathbf{x}^i) - (\|\phi(\mathbf{x}^i)\| + \epsilon)\phi(\mathbf{x}^\ell)$

$$0 \ \leq \ \|(\|\phi(\mathbf{x}^\ell)\| + \epsilon)\phi(\mathbf{x}^i) - (\|\phi(\mathbf{x}^i)\| + \epsilon)\phi(\mathbf{x}^\ell)\|^2$$

Reescribimos esta ecuación utilizando la expresión del producto escalar

$$0 \ \leq \ \langle((\|\phi(\mathbf{x}^\ell)\|+\epsilon)\phi(\mathbf{x}^i)-(\|\phi(\mathbf{x}^i)\|+\epsilon)\phi(\mathbf{x}^\ell)), ((\|\phi(\mathbf{x}^\ell)\|+\epsilon)\phi(\mathbf{x}^i)-(\|\phi(\mathbf{x}^i)\|+\epsilon)\phi(\mathbf{x}^\ell))\rangle$$

A continuación, desarrollando el producto escalar obtenemos:

$$0 \ \leq \ (\|\phi(\mathbf{x}^\ell)\|+\epsilon)^2\|\phi(\mathbf{x}^i)\|^2+(\|\phi(\mathbf{x}^i)\|+\epsilon)^2\|\phi(\mathbf{x}^\ell)\|^2-2\langle((\|\phi(\mathbf{x}^\ell)\|+\epsilon)\phi(\mathbf{x}^i), (\|\phi(\mathbf{x}^i)\|+\epsilon)\phi(\mathbf{x}^\ell)\rangle$$

Esto implica que:

$$2(\|\phi(\mathbf{x}^\ell)\|+\epsilon)(\|\phi(\mathbf{x}^i)\|+\epsilon)\langle\phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle \ \leq \ (\|\phi(\mathbf{x}^\ell)\|+\epsilon)^2\|\phi(\mathbf{x}^i)\|^2+(\|\phi(\mathbf{x}^i)\|+\epsilon)^2\|\phi(\mathbf{x}^\ell)\|^2$$

Cuando $\epsilon \longrightarrow 0$, tenemos que

$$2\|\phi(\mathbf{x}^\ell)\|\|\phi(\mathbf{x}^i)\|\langle\phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle \ \leq \ \|\phi(\mathbf{x}^\ell)\|^2\|\phi(\mathbf{x}^i)\|^2 + \|\phi(\mathbf{x}^i)\|^2\|\phi(\mathbf{x}^\ell)\|^2$$

es decir,

$$\langle\phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle \ \leq \ \|\phi(\mathbf{x}^\ell)\|\|\phi(\mathbf{x}^i)\|$$

Luego, elevando al cuadrado y dado que $k(\mathbf{x}^i, \mathbf{x}^\ell) = \langle\phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell)\rangle$ queda demostrado que $k(\mathbf{x}^i, \mathbf{x}^\ell)^2 = k(\mathbf{x}^i, \mathbf{x}^i)k(\mathbf{x}^\ell, \mathbf{x}^\ell)$

□

Estas propiedades son necesarias pero no suficientes para garantizar la existencia de un espacio de características. Por tanto, a continuación enunciaremos el teorema de Mercer, que nos proporciona una caracterización de cuando una función $k(\mathbf{x}^i, \mathbf{x}^\ell)$ es un kernel. Antes demostraremos una proposición para entender mejor dicho teorema.

Proposición 2.2. Sea $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ una función simétrica en $\mathbb{R}^d$. Entonces k es una función kernel si y solo si la matriz

$$K = (k(\mathbf{x}^i, \mathbf{x}^\ell))_{i,\ell=1}^N \quad \forall \mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$$

es semi-definida positiva, es decir, si tiene autovalores no negativos.

Demostración. Consideremos los datos $\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\} \in \mathbb{R}^d$, y $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ una función simétrica. Consideremos también la matriz kernel:

$$K = (k(\mathbf{x}^i, \mathbf{x}^\ell))_{i,\ell=1}^N$$

Como la función kernel es simétrica, entonces la matriz K también lo es. Luego, existe una matriz ortogonal V tal que $K = VAV^T$, donde A es una matriz diagonal que contiene los autovalores, λ_j con $j = 1, ..., N$, de K y sus respectivos autovectores, $\mathbf{v}^j$, son las columnas de V.

En primer lugar, supongamos que la matriz kernel es semi-definida positiva, es decir, todos sus autovalores son no negativos y veamos la función k es una función kernel.

Para ello, consideremos la siguiente aplicación:

$$\mathbf{x}^i \xrightarrow{\phi} \left(\sqrt{\lambda_1}\mathbf{v}_i^1, ..., \sqrt{\lambda_N}\mathbf{v}_i^N\right)^T$$

Entonces tenemos

$$\langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle = \sum_{j=1}^{N} \lambda_j \mathbf{v}_i^j \mathbf{v}_\ell^j = (VAV^T)_{i\ell} = K_{i\ell} = k(\mathbf{x}^i, \mathbf{x}^\ell)$$

Es decir, k puede escribirse como un producto escalar. Por lo tanto k es una función kernel asociada a la función ϕ.

Ahora supongamos que k es una función kernel, veamos que la matriz kernel es semi-definida positiva. Lo probaremos por reducción al absurdo. Supongamos que existe un autovalor negativo, λ_s, con autovector, $\mathbf{v}^s$. Sea $\mathbf{z}$ un punto de $\left\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\right\}$ dado por la siguiente expresión:

$$\mathbf{z} = \sum_{i=1}^{N} \mathbf{v}_i^s \phi(\mathbf{x}^i) = \sqrt{A}V^T\mathbf{v}^s$$

La norma al cuadrado de $\mathbf{z}$ viene dada por:

$$\|\mathbf{z}\|^2 = \langle \mathbf{z}, \mathbf{z} \rangle = (\mathbf{v}^s)^T V\sqrt{A}\sqrt{A}V^T\mathbf{v}^s = (\mathbf{v}^s)^T VAV^T\mathbf{v}^s = (\mathbf{v}^s)^T K\mathbf{v}^s = \lambda_s < 0$$

lo que es una contradicción, dado que la norma toma siempre valores no negativos. Por lo tanto, la matriz kernel es semi-definida positiva. □

Teorema 2.3. (Teorema de Mercer) [Shawe-Taylor et al., 2004] Sea X un conjunto compacto[1] en $\mathbb{R}^d$. Supongamos que k es una función continua y simétrica tal que el operador $T_k : L_2(X) \longrightarrow L_2(X)$[2] que verifica:

$$(T_k f)(\cdot) = \int_X k(\cdot, \mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

es positivo. Es decir,

$$\int_X k(\mathbf{x}^i, \mathbf{x}^\ell) f(\mathbf{x}^i) f(\mathbf{x}^\ell) d\mathbf{x}^i d\mathbf{x}^\ell \geq 0, \quad \forall f \in L_2(X)$$

Entonces podemos desarrollar $k(\mathbf{x}^i, \mathbf{x}^\ell)$ en una serie uniformemente convergente[3], en $X \times X$, en términos de una funciones ϕ_j, normalizada de tal manera que $\|\phi_j\|_{L_2} = 1$, y $\lambda_j \geq 0$

$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}^i) \phi_j(\mathbf{x}^\ell)$$

Observación 2.1. La hipótesis

$$\int_X k(\mathbf{x}^i, \mathbf{x}^\ell) f(\mathbf{x}^i) f(\mathbf{x}^\ell) d\mathbf{x}^i d\mathbf{x}^\ell \geq 0, \quad \forall f \in L_2(X)$$

corresponde a la condición de que una función k es semi-definida positiva en el caso finito.

Esto implica que las hipótesis del teorema de Mercer son equivalentes a exigir que para cualquier subconjunto finito de X, la matriz correspondiente sea semi-definida positiva.

[1] ver Definición A.1 en el Apéndice A
[2] ver Definición A.2 en el Apéndice A
[3] ver Definición A.3 en el Apéndice A

Gracias a las funciones finitamente semi-definidas positivas, podemos dar una nueva caracterización de las funciones kernel.

Definición 2.5. (Funciones finitamente semi-definidas positivas) Una función

$$k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$$

es finitamente semi-definida positiva si es una función simétrica y la matriz $K = (k(\mathbf{x}^i, \mathbf{x}^\ell))_{i,\ell=1}^N$ $\forall \mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$ y cualquier submatriz de la misma son semi-definidas positivas.

Teorema 2.4. [Shawe-Taylor et al., 2004] Sea $k : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ una función continua ó con dominio finito. Sea ϕ una función característica en un espacio de Hilbert $\mathcal{F}$ [4] . La función k se puede descomponer en

$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \langle \phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell) \rangle$$

si y solo si k es finitamente semi-definida positiva.

Las caracterizaciones de las funciones y matrices kernel no solo son útiles para decidir si una determinada función es un kernel válido o no. Una de las principales consecuencias es que puede ser utilizado para justificar una serie de propiedades para manipular y combinar núcleos simples con el fin de obtener otros más complejos y útiles.

A continuación veremos distintas operaciones que se pueden realizar con las funciones kernel.

Teorema 2.5. Dadas $k_1, k_2 : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}$ dos funciones kernel, $a \in \mathbb{R}^+$, $f : \mathbb{R}^d \longrightarrow \mathbb{R}$, $\phi : \mathbb{R}^d \longrightarrow F$, donde F es el *feature space*, con una función kernel $k_3 : F \times F \longrightarrow \mathbb{R}$, B una matriz simétrica semi-definida positiva cuadrada de dimensión d y $\mathbf{x}^i, \mathbf{x}^\ell \in \mathbb{R}^d$

Entonces las siguientes funciones son kernels:

a) $k(\mathbf{x}^i, \mathbf{x}^\ell) = k_1(\mathbf{x}^i, \mathbf{x}^\ell) + k_2(\mathbf{x}^i, \mathbf{x}^\ell)$

b) $k(\mathbf{x}^i, \mathbf{x}^\ell) = ak_1(\mathbf{x}^i, \mathbf{x}^\ell)$

c) $k(\mathbf{x}^i, \mathbf{x}^\ell) = k_1(\mathbf{x}^i, \mathbf{x}^\ell)k_2(\mathbf{x}^i, \mathbf{x}^\ell)$

d) $k(\mathbf{x}^i, \mathbf{x}^\ell) = f(\mathbf{x}^i)f(\mathbf{x}^\ell)$

e) $k(\mathbf{x}^i, \mathbf{x}^\ell) = k_3(\phi(\mathbf{x}^i), \phi(\mathbf{x}^\ell))$

f) $k(\mathbf{x}^i, \mathbf{x}^\ell) = (\mathbf{x}^i)^T B \mathbf{x}^\ell$

Demostración. Dado X un conjunto finito de puntos $\{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$, y dado K_1 y K_2 las correspondientes matrices kernel obtenidas restringiendo las funciones k_1 y k_2 a estos puntos. Consideremos un vector cualesquiera $\alpha \in \mathbb{R}^N$. Recordemos que una matriz es semi-definida positiva si y solo si $\alpha^T K \alpha \geq 0$.

Con estas hipótesis, procedemos a demostrar cada uno de los apartados del teorema.

a) Tenemos que

$$\alpha^T(K_1 + K_2)\alpha = \alpha^T K_1 \alpha + \alpha^T K_2 \alpha \geq 0$$

Entonces por ser K_1 y K_2 matrices semi-definidas positivas $K_1 + K_2$ es una matriz semi-definida positiva y por tanto la función $k_1 + k_2$ es una función kernel.

[4] ver Definición A.7 en el Apéndice A

b) Análogamente tenemos
$$\alpha^T a K_1 \alpha = a\alpha^T K_1 \alpha \geq 0$$
Entonces ak_1 es un función kernel.

c) Sea $K = K_1 \otimes K_2$ el producto tensorial [5] de las matrices K_1 y K_2. El producto tensorial de dos matrices semi-definidas positivas es a su vez semi-definido positivo ya que los valores propios de este producto son todos los pares de productos de los valores propios de las dos componentes. Luego
$$\alpha^T K \alpha \geq 0$$
por lo tanto, k es una función kernel.

d) Definimos ϕ como una función asociada unidimensional tal que:
$$\mathbf{x}^i \xrightarrow{\phi} f(\mathbf{x}^i) \tag{2.66}$$
Entonces $k(\mathbf{x}^i, \mathbf{x}^\ell) = f(\mathbf{x}^i)f(\mathbf{x}^\ell) = \phi(\mathbf{x}^i)\phi(\mathbf{x}^\ell)$. Es decir, k puede expresarse como el producto escalar (unidimensional) de la función de características ϕ definida en (2.66).

e) Dado que k_3 es un kernel, la matriz obtenida al restringir k_3 a los puntos $\{\phi(\mathbf{x}^1), \phi(\mathbf{x}^2), ..., \phi(\mathbf{x}^N)\}$ es semi-definida positiva, por lo que k es una función kernel.

f) Consideramos la diagonalización de $B = V^T AV$ donde V es ortogonal y A es la matriz diagonal que contiene los valores propios no negativos. Sea C la matriz diagonal formada por las raíces cuadradas de los autovalores no negativos $C = \sqrt{A}V$. Entonces
$$k(\mathbf{x}^i, \mathbf{x}^\ell) = (\mathbf{x}^i)^T B \mathbf{x}^\ell = (\mathbf{x}^i)^T V^T AV \mathbf{x}^\ell = (\mathbf{x}^i)^T C^T C \mathbf{x}^\ell = \langle C\mathbf{x}^i, C\mathbf{x}^\ell \rangle$$
Es decir, la función $k(\mathbf{x}^i, \mathbf{x}^\ell) = (\mathbf{x}^i)^T B \mathbf{x}^\ell$ se puede escribir como el producto escalar $\langle C\mathbf{x}^i, C\mathbf{x}^\ell \rangle$. Es decir, $k(\mathbf{x}^i, \mathbf{x}^\ell)$ es un kernel.

□

Podemos encontrar más propiedades en [Shawe-Taylor et al., 2004].

Hasta ahora hemos visto caracterizaciones del kernel y operaciones entre funciones kernel. Pero existen casos particulares de funciones kernels muy usados en la práctica que veremos a continuación:

- Kernel lineal:
$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle = \sum_{j=1}^{d} x_j^i x_j^\ell$$
Utilizar este kernel en (2.62) es equivalente a resolver el problema (2.50). Por tanto, esta función kernel es útil cuando los datos de entrada son linealmente separables.

- Kernel polinomial:
$$k(\mathbf{x}^i, \mathbf{x}^\ell) = (\langle \mathbf{x}^i, \mathbf{x}^\ell \rangle + c)^n$$
Es una forma más generalizada del kernel lineal (proporcionado cuando $c = 0$ y $n = 1$), no solo observa las características dadas por las muestras de entrada para determinar su similitud, sino también las combinaciones de éstas.

[5] ver Definición A.8 en el Apéndice A

Veamos un ejemplo de ϕ para este tipo de kernel con $n = 2$ y $c = 0$, es decir, cuando:

$$k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

En general sabemos que una función kernel cumple:

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

Luego, buscamos una función ϕ que satisface:

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = \langle \mathbf{x}, \mathbf{z} \rangle^2 \tag{2.67}$$

Por simplicidad, vamos a suponer que $\mathbf{x}, \mathbf{z} \in \mathbb{R}^2$. Y consideramos la función

$$\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \quad \forall \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$$

Luego,

$$\begin{aligned} \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = & \quad \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \rangle \\ = & \quad x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \end{aligned} \tag{2.68}$$

Por otro lado,

$$\begin{aligned} \langle \mathbf{x}, \mathbf{z} \rangle^2 = & \quad \langle (x_1, x_2), (z_1, z_2) \rangle^2 = \langle (x_1, x_2), (z_1, z_2) \rangle \langle (x_1, x_2), (z_1, z_2) \rangle \\ = & \quad (x_1z_1 + x_2z_2)(x_1z_1 + x_2z_2) = (x_1z_1 + x_2z_2)^2 \\ = & \quad x_1^2z_1^2 + x_2^2z_2^2 + 2x_1z_1x_2z_2 \end{aligned} \tag{2.69}$$

En conclusión, por (2.68) y (2.69) tenemos que para la función $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ $\forall \mathbf{x} \in \mathbb{R}^2$, se cumple (2.67).

- Kernel Gaussiano:

$$k(\mathbf{x}^i, \mathbf{x}^\ell) = \exp(-\gamma \|\mathbf{x}^i - \mathbf{x}^\ell\|^2)$$

Los kernels gaussianos son las funciones kernels más utilizadas en la práctica. Además, el parámetro γ controla la no linealidad del modelo. También influye en el alcance que los vectores soporte tienen sobre el modelo. Los valores pequeños de γ indicarán un gran alcance y dará lugar a modelos altamente no lineales. Por el contrario, valores grandes de γ producirán modelos con tendencia lineal donde los vectores soporte tienen un radio de influencia pequeño. En el Capítulo 4 veremos más detalles sobre este tema.

Capítulo 3

Extensiones

En este capítulo vamos a extender a la regresión y a la detección de outliers, los modelos de vectores soporte ya aplicados en el Capítulo 2 al campo de la clasificación. Esto dará lugar al método SVR (del inglés *Support Vector Regression*) en la Sección 3.1, y al método SVDD (del inglés *Support Vector Data Description*) en la Sección 3.2.

3.1. Support Vector Regression

En esta sección, vamos a asumir dado un conjunto de entrenamiento de la forma $S = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), ..., (\mathbf{x}^N, y^N)\}$ donde $\mathbf{x}^i \in \mathbb{R}^d$ e $y^i \in \mathbb{R}$ $\forall i$. A diferencia de lo que ocurre en la clasificación binaria donde $y^i \in \{-1, 1\}$, en la regresión la variable respuesta y^i toma valores reales. Además, al igual que en la clasificación binaria buscábamos maximizar el margen para garantizar una buena separación entre las dos clases, en la regresión buscamos que una función f, de forma que $f(\mathbf{x}^i)$ esté lo mas cerca posible a los puntos y^i.

En primer lugar, nos centraremos en el caso lineal del método SVR, y de forma análoga a como lo hacíamos en el método SVM del Capítulo 2, f es de la forma:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad \text{con} \quad \mathbf{w} \in \mathbb{R}^d \quad \text{y} \quad b \in \mathbb{R} \tag{3.1}$$

Luego, el objetivo del SVR será encontrar los mejores $(\mathbf{w}, b)$ para una buena predicción.

Para ello, en el SVR, en lugar de un margen, definimos una banda o tubo de radio $\varepsilon > 0$, de forma que se ignoran los errores asociados a los puntos dentro de esa banda (ver Figura 3.1). Por lo que definimos la función de pérdida ε-insensible, que se anula para los puntos situados dentro de esa banda

$$\mathbb{L}_\varepsilon(y, f(\mathbf{x})) = \begin{cases} 0, & \text{si } |y - f(\mathbf{x})| \leq \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & \text{en otro caso} \end{cases} \tag{3.2}$$

Aunque en este trabajo nos centraremos en el estudio de la metodología del SVR con la función de pérdida (3.2), se podrían considerar muchas otras, como las que podemos encontrar en [Wang et al., 2022; Drucker et al., 1996]

Así, podemos plantear el problema *hard margin* SVR de forma análoga a como lo hacíamos en el Capítulo 2:

$$\begin{array}{ll} \min\limits_{\mathbf{w}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.a.} & y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - b \leq \varepsilon, \quad i = 1, ..., N \\ & \langle \mathbf{w}, \mathbf{x}^i \rangle + b - y^i \leq \varepsilon, \quad i = 1, ..., N \end{array} \tag{3.3}$$

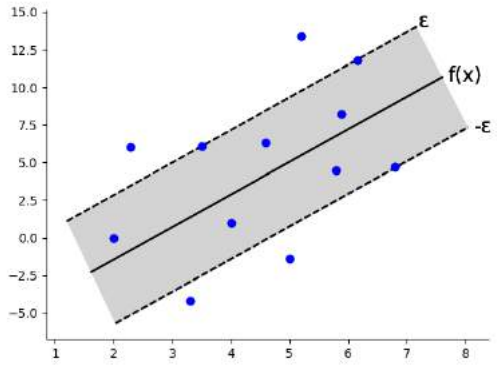


Figura 3.1: En el método SVR definimos una banda de radio ε en torno a la función de predicción, f. Los puntos que se encuentran en el interior de dicha banda se ignoran en la función de pérdida (3.2).

Se trata de un problema de optimización convexo, ya que la función objetivo es convexa y las restricciones son funciones lineales. Es por ello que se puede calcular el problema dual asociado que, a su vez, también será un problema convexo. Ambas formulaciones (primal y dual) pueden resolverse con técnicas de optimización convexas, como las que aparecen en [Boyd et al., 2004].

Por otro lado, al igual que ocurría con el modelo SVM definido en el Capítulo 2, en el caso del SVR, también podemos formular dos versiones de este problema: *hard margin* y *soft margin*. El problema *hard margin* ya ha sido planteado en la ecuación (3.3). En el caso *soft margin* asumimos que los datos no se pueden predecir a través de un modelo lineal con error cero, así que introducimos las variables de holgura ξ_i y ξ_i^* controlan el error cometido por la función de regresión al aproximar un punto i-ésimo, en (3.3). Por tanto, construimos el problema primal SVR con *soft margin* de la siguiente forma:

$$\begin{array}{ll} \underset{\mathbf{w},\xi_i,\xi_i^*,\forall i}{\text{mín}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ \text{s.a.} & y^i - \langle \mathbf{w}, \mathbf{x}^i\rangle - b \leq \varepsilon + \xi_i, \quad i = 1, ..., N \\ & \langle \mathbf{w}, \mathbf{x}^i\rangle + b - y^i \leq \varepsilon + \xi_i^*, \quad i = 1, ..., N \\ & \xi_i,\ \xi_i^* \geq 0, \quad i = 1, ..., N \end{array} \tag{3.4}$$

donde la constante $C \geq 0$ es un parámetro de regulación entre el valor de ε y la cantidad de errores mayores que ε que son tolerados.

Además, si $\xi_i > 0$, entonces la predicción del dato $\mathbf{x}^i$, $f(\mathbf{x}^i)$, será mayor que su valor real, y^i, en una cantidad superior a ε. En otro caso su valor será cero, es decir, si la diferencia entre $f(\mathbf{x}^i)$ e y^i es menor que ε, entonces $\xi_i = 0$. De la misma forma, si $\xi_i^* > 0$, entonces la predicción del dato será menor que el valor real del mismo en una cantidad superior a ε. En otro caso su valor será cero, es decir, cuando la diferencia entre $f(\mathbf{x}^i)$ e y^i sea menor que ε, ξ_i será cero. Además, $\xi_i \cdot \xi_i^* = 0$, ya que la predicción de un dato no puede ser a la vez mayor y menor que su valor real (ver Figura 3.2).

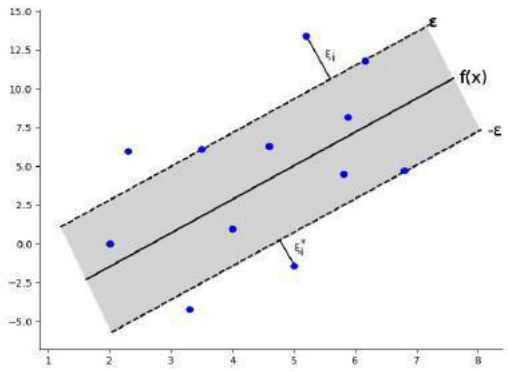


Figura 3.2: Variables de holgura en el método SVR.

El problema (3.4) también se trata de un problema convexo puesto que la función objetivo es una función convexa y las restricciones son funciones lineales. Luego podemos aplicar la teoría de dualidad y resolver del problema de optimización primal mediante su problema dual asociado, siguiendo los mismos pasos que en el método SVM del Capítulo 2. Por lo tanto, los problemas *hard margin* y *soft margin* se pueden resolver mediante la teoría de optimización dual, pero para evitar repeticiones sólo se va a explicar el procedimiento del segundo caso.

Pasemos explicar dicho procedimiento. Construimos la función lagrangiana:

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \; & \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\beta_i\xi_i + \beta_i^*\xi_i^*) \\ & - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y^i + \langle \mathbf{w}, \mathbf{x}^i\rangle + b) \\ & - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y^i - \langle \mathbf{w}, \mathbf{x}^i\rangle - b) \end{aligned} \tag{3.5}$$

donde $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)$ y $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_N^*)$ siendo α_i y α_i^* los multiplicadores de Lagrange asociados a las restricciones $y^i - \langle \mathbf{w}, \mathbf{x}^i\rangle - b \leq \varepsilon + \xi_i$ y $\langle \mathbf{w}, \mathbf{x}^i\rangle + b - y^i \leq \varepsilon + \xi_i^*$, respectivamente. Y $\boldsymbol{\beta} = (\beta_1, ..., \beta_N)$ y $\boldsymbol{\beta}^* = (\beta_1^*, ..., \beta_*)$ con β_i y β_i^* los multiplicadores de Lagrange asociados a las restricciones $\xi_i \geq 0$ y $\xi_i^* \geq 0$, respectivamente. Además, $\boldsymbol{\xi} = (\xi_1, ..., \xi_N)$ y $\boldsymbol{\xi}^* = (\xi_1^*, ..., \xi_N^*)$.

A continuación, aplicamos las condiciones KKT del Teorema 2.2 e igualamos a cero:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\mathbf{x}^i = \mathbf{0} \tag{3.6}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial b} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0 \tag{3.7}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \xi_i} = C - \beta_i - \alpha_i = 0, \quad i = 1, ..., N \tag{3.8}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \xi_i^*} = C - \beta_i^* - \alpha_i^* = 0, \quad i = 1, ..., N \tag{3.9}$$

$$\alpha_i(\varepsilon + \xi_i - y^i + \langle \mathbf{w}, \mathbf{x}^i \rangle + b) = 0, \quad i = 1, ..., N \tag{3.10}$$

$$\alpha_i^*(\varepsilon + \xi_i^* + y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - b) = 0, \quad i = 1, ..., N \tag{3.11}$$

$$\beta_i \xi_i = 0, \quad i = 1, ..., N \tag{3.12}$$

$$\beta_i^* \xi_i^* = 0, \quad i = 1, ..., N \tag{3.13}$$

Para relacionar las variables del problema primal y dual, se hace uso de la ecuación (3.6):

$$\mathbf{w} = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \mathbf{x}^i \tag{3.14}$$

Se trata de la denominada expansión de vectores soporte, es decir, $\mathbf{w}$ se describe como una combinación lineal de los datos de entrenamiento.

De las ecuaciones (3.7)-(3.9) obtenemos las siguientes restricciones de las variables duales:

$$\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0 \tag{3.15}$$

$$\beta_i = C - \alpha_i \quad i = 1, ..., N \tag{3.16}$$

$$\beta_i^* = C - \alpha_i^* \quad i = 1, ..., N \tag{3.17}$$

Sustituyendo las ecuaciones (3.14)-(3.17) en (3.5) tenemos como resultado el problema de optimización dual

$$\begin{aligned} \underset{\alpha_i, \alpha_i^*, \forall i}{\text{máx}} \quad & -\tfrac{1}{2} \sum_{i,\ell=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_\ell - \alpha_\ell^*) \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y^i (\alpha_i - \alpha_i^*) \\ \text{s.a.} \quad & \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \\ & 0 \leq \alpha_i,\, \alpha_i^* \leq C, \quad i = 1, ..., N \end{aligned} \tag{3.18}$$

Por lo tanto, resolviendo de forma óptima el problema dual en (3.18), podemos obtener la solución óptima del problema primal (3.4):

$$\mathbf{w}^{opt} = \sum_{i=1}^{N} (\alpha_i^{opt} - \alpha_i^{*,opt}) \mathbf{x}^i \tag{3.19}$$

donde α_i^{opt} y $\alpha_i^{*,opt}$ son las soluciones óptimas del problema dual.

Luego, la función de regresión asociada de (3.1) quedaría de la siguiente forma:

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i^{opt} - \alpha_i^{*,opt}) \langle \mathbf{x}^i, \mathbf{x} \rangle + b \tag{3.20}$$

A continuación vamos a analizar los distintos tipos de puntos en función de los valores de α_i y α_i^*. Esto se hará utilizando las condiciones KKT, es decir, con las ecuaciones obtenidas en (3.10)-(3.13). Además, por las ecuaciones (3.16) y (3.17) podemos reescribir (3.12) y (3.13) de la siguiente forma:

$$(C - \alpha_i)\xi_i = 0 \tag{3.21}$$

$$(C - \alpha_i^*)\xi_i^* = 0 \tag{3.22}$$

Lo que nos permite obtener varias conclusiones. En primer lugar, obtenemos que $\alpha_i\alpha_i^* = 0$, es decir, α_i y α_i^* no pueden ser distintos de cero a la vez. Para demostrar esto, comenzaremos suponiendo que α_i y α_i^* son distintos cero a la vez, entonces $0 < \alpha_i \leq C$ y $0 < \alpha_i^* \leq C$. En el caso en que $0 < \alpha_i < C$ y $0 < \alpha_i^* < C$, por las ecuaciones (3.21) y (3.22), $\xi_i = 0$ y $\xi_i^* = 0$, respectivamente. Luego, al sustituir esto en las ecuaciones (3.10) y (3.11), nos quedará que

$$y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - b = \varepsilon \tag{3.23}$$

$$-y^i + \langle \mathbf{w}, \mathbf{x}^i \rangle + b = \varepsilon \tag{3.24}$$

y esto no se puede dar para un mismo punto porque la diferencia entre el valor real de un punto y su predicción no puede tener a la misma vez dos valores distintos. En el caso en que $\alpha_i = C$ y $\alpha_i^* = C$ por las ecuaciones (3.21) y (3.22), $\xi_i > 0$ y $\xi_i^* > 0$ pero esto tampoco puede ocurrir ya que hemos visto anteriormente que $\xi_i\xi_i^* = 0$. Por último, tenemos los casos en que $0 < \alpha_i < C$ y $\alpha_i^* = C$, y $\alpha_i = C$ y $0 < \alpha_i^* < C$, estudiaremos el primer caso, el otro será análogo. Si $0 < \alpha_i < C$ y $\alpha_i^* = C$, por las ecuaciones (3.21) y (3.22) tenemos que $\xi_i = 0$ y $\xi_i^* > 0$, respectivamente. Ahora, por la ecuación (3.10)

$$y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - b = \varepsilon \tag{3.25}$$

pero por la ecuación (3.11) obtenemos que

$$y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - b = -\varepsilon - \xi_i^* \tag{3.26}$$

lo que es imposible ya que la diferencia entre el valor real de un punto y su predicción no puede tener a la misma vez dos valores distintos. Luego $\alpha_i\alpha_i^* = 0$.

A continuación, veremos cuáles son los vectores soporte para el método SVR.

Veamos que existe al menos un dato i tal que $\alpha_i = C$. Para ello, supongamos que $\alpha_i \neq C$ para todo $i = 1, ..., N$. Luego, $0 \leq \alpha_i < C$, además $C \neq 0$ pues si $C = 0$ entonces $\alpha_i = 0$, por lo tanto tendríamos que $\alpha_i = C$ y estamos suponiendo que $\alpha_i \neq C$. Por otro lado, de la ecuación (3.21) obtenemos que $\xi_i = 0$, lo que implica que $\xi_i^* > 0$. Y de la ecuación (3.22) deducimos que $\alpha_i^* = C$. Si $\alpha_i \neq 0$, $\forall i$, entonces tenemos que $\alpha_i > 0$ y $\alpha_i^* > 0$ por lo que llegamos a una contracción ya que $\alpha_i\alpha_i^* = 0$. En el caso en que $\alpha_i = 0$, como $\alpha_i^* = C > 0$, sustituimos ambas ecuaciones en (3.15) lo que implica que $\sum_{i=1}^{N} C = 0$ llegando así a una contradicción, ya que $C > 0$.

De forma análoga, vemos que existe al menos un dato i con $\alpha_i^* = C$.

Luego existe algún dato i para el que $\alpha_i = C$, entonces $\alpha_i^* = 0$ y por la ecuación (3.21), $\xi_i > 0$. Lo que implica que $\xi_i^* = 0$. Por otra parte, también existe algún dato i tal que $\alpha_i^* = C$, entonces $\alpha_i = 0$ y por la ecuación (3.22) tenemos que $\xi_i^* > 0$. Luego, $\xi_i = 0$. Los

datos $(\mathbf{x}^i, y^i)$ que tienen estos valores de α_i y α_i^* son los datos situados fuera de la banda ε-insensible.

Por el contrario, los datos $(\mathbf{x}^i, y^i)$ que están situados dentro del ε-insensible verifican que $\xi_i = 0$ y $\xi_i^* = 0$. Además, si para esos datos $\alpha_i = 0$ y $\alpha_i^* = 0$, por la ecuaciones (3.10) y (3.11), $y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle - \ b < \varepsilon$ y $\langle \mathbf{w}, \mathbf{x}^i \rangle + \ b - y^i < \varepsilon$, respectivamente. Luego, estos datos no se encuentran en la frontera de dicha zona.

Por último, tenemos dos casos más: en primer lugar, cuando $0 < \alpha_i < C$, y en segundo lugar, cuando $0 < \alpha_i^* < C$.

En el primer caso, $0 < \alpha_i < C$, como $\alpha_i > 0$, el segundo factor de la parte izquierda de la ecuación (3.10) es nulo, es decir:

$$\varepsilon + \xi_i - y^i + \langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + \ b = 0, \quad \text{si} \quad \alpha_i > 0 \tag{3.27}$$

Por otro lado, como $\alpha_i < C$ entonces por la ecuación (3.21) tenemos:

$$\xi_i = 0, \quad \text{si} \quad \alpha_i < C \tag{3.28}$$

Luego, combinando los dos últimos resultados obtenemos:

$$y^i - \langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle - \ b = \varepsilon, \quad \text{si} \quad 0 < \alpha_i < C \tag{3.29}$$

De forma análoga para el segundo caso, $0 < \alpha_i^* < C$, obtenemos, de las ecuaciones (3.11) y (3.22), que:

$$\langle \mathbf{w}^{opt}, \mathbf{x}^i \rangle + \ b - y^i = \varepsilon, \quad \text{si} \quad 0 < \alpha_i^* < C \tag{3.30}$$

Observemos que los únicos datos que cumplen las restricciones (3.29) y (3.30) son aquellos datos que están situados justo en la frontera de la zona ε-insensible, los llamados vectores soporte.

En la Figura 3.3, podemos observar cada dato i con los distintos valores de α_i y α_i^*.

Finalmente concluimos que los datos que se encuentran en la frontera o fuera de la zona ε-insensible son los únicos que contribuyen a la construcción del hiperplano, ya que $\alpha_i \neq 0$ ó $\alpha_i^* \neq 0$.

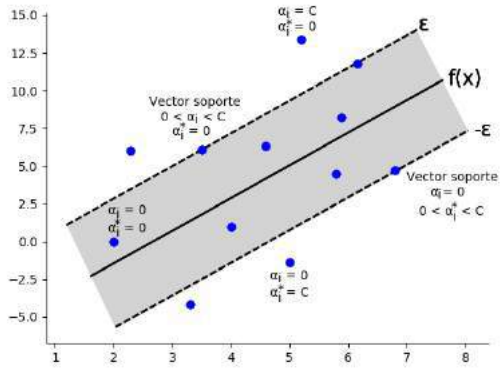


Figura 3.3: Valores de los multiplicadores de Lagrange. El valor de cada α_i y cada α_i^* depende de la posición en la que se encuentre el dato i respecto de las fronteras de la banda y de la función de regresión.

Por último, obtenemos la expresión óptima de b que será de la siguiente forma:

$$\text{máx}\left\{-\varepsilon + y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle | \alpha_i < C \text{ ó } \alpha_i^* > 0\right\} \leq b \leq \tag{3.31}$$

$$\text{mín}\left\{-\varepsilon + y^i - \langle \mathbf{w}, \mathbf{x}^i \rangle | \alpha_i > 0 \text{ ó } \alpha_i^* < C\right\} \tag{3.32}$$

Más informacioń sobre el procedimiento de la obtención de b puede ser obtenida en [Smola and Schölkopf, 2004].

Para obtener la predicción de un nuevo dato, $\mathbf{x}^{test}$, bastará con aplicar al nuevo dato la función de regresión f, es decir,

$$f(\mathbf{x}^{test}) = \sum_{i=1}^{N} (\alpha_i^{opt} - \alpha_i^{*,opt}) \langle \mathbf{x}^i, \mathbf{x}^{test} \rangle + b^{opt}$$

Una vez visto el caso lineal, el siguiente paso será formular la metodología SVR para el caso no lineal. Seguiremos un procedimiento similar al utilizado para los problemas de clasificación en el Capítulo 2, es decir, transformaremos los datos de entrada a un nuevo espacio en el que las transformaciones si puedan ser ajustadas mediante una función de regresión lineal, usando una función kernel. Por lo tanto, para conocer la función de regresión, bastará con conocer la función $k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ que en lugar de calcular explícitamente el producto escalar, nos permite formular el problema de optimización dual obtenido para el caso lineal (3.18) de la siguiente forma:

$$\begin{aligned} \max_{\alpha_i, \alpha_i^*, \forall i} \quad & -\tfrac{1}{2} \sum_{i,\ell=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_\ell - \alpha_\ell^*) k(\mathbf{x}^i, \mathbf{x}^\ell) - \varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y^i (\alpha_i - \alpha_i^*) \\ \text{s.a.} \quad & \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \\ & 0 \leq \alpha_i,\, \alpha_i^* \leq C, \quad i = 1, ..., N \end{aligned} \tag{3.33}$$

Asímismo, la solución óptima del problema primal en función de la soluciones óptimas del problema dual asociado, $\alpha_i^{opt}, \alpha_i^{*,opt}$, es:

$$\mathbf{w}^{opt} = \sum_{i=1}^{n} (\alpha_i^{opt} - \alpha_i^{*,opt}) \phi(\mathbf{x}_i) \tag{3.34}$$

Por lo tanto, la función de regresión óptima es:

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i^{opt} - \alpha_i^{*,opt}) k(\mathbf{x}^i, \mathbf{x}) + b^{opt} \tag{3.35}$$

La diferencia con el caso lineal es que $\mathbf{w}^{opt}$ ya no se da de forma explícita puesto que su expresión depende de ϕ, que apriori es desconocido, y el problema de optimización dual hay que resolverlo en el espacio de características no en el de entrada.

Al igual que en el caso *soft margin* SVR, para obtener la predicción de un nuevo dato, $\mathbf{x}^{test}$, aplicamos al nuevo dato la función de regresión f, es decir,

$$f(\mathbf{x}^{test}) = \sum_{i=1}^{N} (\alpha_i^{opt} - \alpha_i^{*,opt}) k(\mathbf{x}^i, \mathbf{x}^{test}) + b^{opt}$$

3.2. Support Vector Data Description

Hasta ahora hemos estudiado los problemas de clasificación en el Capítulo 2 y de regresión en la Sección 3.1 en los que a partir de conjuntos de datos de entrenamiento se diseña un modelo de predicción entre las variables explicativas y la variable repuesta. Además de las tareas de clasificación y regresión, también es interesante el estudio del problema de la descripción de datos. Este problema consiste en realizar una descripción del conjunto de datos de entrenamiento y detectar qué objetos nuevos se parecen o no a este conjunto de datos.

Una de las aplicaciones más importantes de la descripción de datos es la detección de valores atípicos o *outliers*. El problema de la detección de *outliers* consiste esencialmente en distinguir datos con alguna característica muy diferente respecto al resto de puntos del conjunto de entrenamiento. Por ejemplo, esa característica diferente puede venir dada por un valor muy grande en una de las variables explicativas con respecto al resto de datos. A estos datos con características diferentes los llamaremos datos atípicos ó *outliers*. Al resto de puntos los llamaremos datos objetivo.

En esta sección estudiaremos uno de los métodos más frecuentes para la detección de datos atípicos. Esta metodología se basa en el concepto de vectores soporte, que hemos visto durante el trabajo y se conoce por el nombre (en inglés) de *Support Vector Data Description*, SVDD. Esencialmente, este método construye una frontera esférica alrededor de los datos objetivo. De esta forma todos los puntos dentro de la esfera serán considerados puntos objetivos, mientras que los datos fuera de esta esfera serán *outliers* (ver Figura 3.4). Pasemos ahora a la construcción del modelo.

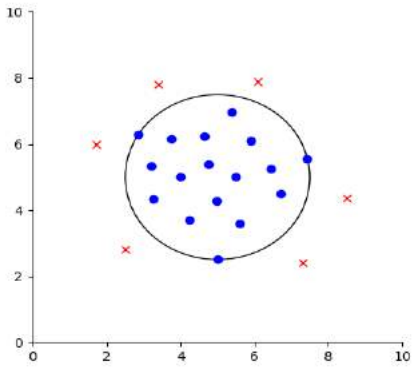


Figura 3.4: Método SVDD. Los datos objetivos están representados con el color azul y los *outliers* con el rojo.

Dado un conjunto de entrenamiento $S = \{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), ..., (\mathbf{x}^N, y^N)\}$ donde $\mathbf{x}^i \in \mathbb{R}^d$ e $y^i \in \{-1, 1\}$ para todo $i = 1, ..., N$. Sin pérdida de generalidad, asumiremos que los datos están ordenados de forma que los M primero puntos serán los datos objetivo y están etiquetados con $y^i = 1$, $i = 1, ..., M$. De igual forma, los $N - M$ últimos puntos serán considerados los datos atípicos y tendrán etiquetas $y^i = -1$, $i = M + 1, ..., N$. Normalmente, hay pocos *outliers* en comparación con los datos objetivo, por lo tanto asumimos que $M > N - M$.

En primer lugar, definimos un modelo que nos proporciona una frontera cerrada de forma esférica alrededor de los datos objetivo. La esfera está caracterizada por su centro

$\mathbf{a} \in \mathbb{R}^d$ y su radio $R > 0$, además buscamos una esfera de volumen mínimo que contenga a todos los datos objetivo en su interior. Para ello, minimizamos la siguiente función:

$$Q(R, \mathbf{a}) = R^2 \tag{3.36}$$

con las siguiente restricciones:

$$\|\mathbf{x}^i - \mathbf{a}\| \leq R^2, \quad i = 1, ..., M \tag{3.37}$$
$$\|\mathbf{x}^i - \mathbf{a}\| \geq R^2, \quad i = M+1, ..., N \tag{3.38}$$

donde la primera restricción está asociada a los datos objetivo y la segunda a los *outliers*. Luego, podemos escribir el problema primal SVDD con *hard margin* de la siguiente forma:

$$\begin{array}{ll} \min\limits_{R} & R^2 \\ \text{s.a.} & \|\mathbf{x}^i - \mathbf{a}\|^2 \leq R^2, \quad i = 1, ..., M \\ & \|\mathbf{x}^i - \mathbf{a}\|^2 \geq R^2, \quad i = M+1, ..., N \end{array} \tag{3.39}$$

En el método SVDD también se pueden aplicar la metodología *hard margin* y *soft margin* pero para evitar repeticiones innecesarias, nos centramos en desarrollar la metodología del caso *soft margin* SVDD. Por lo tanto, permitirémos que se produzcan errores tanto en los datos objetivo como en los datos atípicos, es decir, habrá datos objetivo clasificados como datos atípicos y datos atípicos clasificados como datos objetivo. Es por ello que introducimos la variable de holgura ξ_i para $i = 1, ..., N$. Así, construimos el problema SVDD primal con *soft margin* de la siguiente forma:

$$\begin{array}{ll} \min\limits_{R, \xi_i, \forall i} & R^2 + C_1 \sum\limits_{i=1}^{M} \xi_i + C_2 \sum\limits_{i=M+1}^{N} \xi_i \\ \text{s.a.} & \|\mathbf{x}^i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad i = 1, ..., M \\ & \|\mathbf{x}^i - \mathbf{a}\|^2 \geq R^2 - \xi_i, \quad i = M+1, ..., N \\ & \xi_i \geq 0, \quad i = 1, ..., N \end{array} \tag{3.40}$$

donde C_1 y C_2 son dos constantes que regulan la influencia de los errores en la función objetivo.

Al ser un problema convexo, utilizamos la teoría de optimización dual, y podremos resolver el problema primal a través de su problema dual asociado que también será un problema de optimización convexo. Ambos problemas se resolverán, por tanto, mediante técnicas de optimización convexas. Para la construcción del problema dual seguirémos los mismos pasos que utilizamos en el Capítulo 2 y en la sección 1.2.

Construimos el lagrangiano:

$$\begin{aligned} L(R, \mathbf{a}, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \quad & R^2 + C_1 \sum_{i=1}^{M} \xi_i + C_2 \sum_{i=M+1}^{N} \xi_i - \sum_{i=1}^{N} \beta_i \xi_i \\ & - \sum_{i=1}^{M} \alpha_i [R^2 + \xi_i - \|\mathbf{x}^i - \mathbf{a}\|^2] \\ & - \sum_{i=M+1}^{N} \alpha_i [\|\mathbf{x}^i - \mathbf{a}\|^2 - R^2 + \xi_i] \end{aligned} \tag{3.41}$$

donde $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)$ y $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_N)$ son los multiplicadores de Lagrange. Por lo tanto, $\alpha_i, \beta_i \geq 0$, $\forall i$. Además, $\boldsymbol{\xi} = (\xi_1, \xi_2, ..., \xi_N)$.

Aplicamos las condiciones KKT vistas en el Teorema 2.2. Por la primera condición obtenemos que:

$$\frac{\partial L(R,\mathbf{a},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial R} = 2R - 2R\sum_{i=1}^{M}\alpha_i + 2R\sum_{i=M+1}^{N}\alpha_i = 0 \tag{3.42}$$

$$\frac{\partial L(R,\mathbf{a},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \mathbf{a}} = 2\sum_{i=1}^{M}\alpha_i\mathbf{x}^i - 2\mathbf{a}\sum_{i=1}^{M}\alpha_i - 2\sum_{i=M+1}^{N}\alpha_i\mathbf{x}^i + 2\mathbf{a}\sum_{i=M+1}^{N}\alpha_i = 0 \tag{3.43}$$

$$\frac{\partial L(R,\mathbf{a},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \xi_i} = C_1 - \alpha_i - \beta_i = 0, \quad i = 1,...,M \tag{3.44}$$

$$\frac{\partial L(R,\mathbf{a},\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \xi_i} = C_2 - \alpha_i - \beta_i = 0, \quad i = M+1,...,N \tag{3.45}$$

Y a continuación aplicamos la segunda condición del Teorema 2.2. Para ello, multiplicamos los multiplicadores de Lagrange por sus restricciones asociadas del problema (3.40) e igualamos a cero, es decir:

$$\alpha_i[R^2 + \xi_i - \|\mathbf{x}^i - \mathbf{a}\|^2] = 0, \quad i = 1,...,M \tag{3.46}$$

$$\alpha_i[\|\mathbf{x}^i - \mathbf{a}\|^2 - R^2 + \xi_i] = 0, \quad i = M+1,...,N \tag{3.47}$$

$$\beta_i\xi_i = 0, \quad i = 1,...,N \tag{3.48}$$

Podemos reescribir estas ecuaciones (3.42)-(3.45) de la siguiente forma:

$$\sum_{i=1}^{M}\alpha_i - \sum_{i=M+1}^{N}\alpha_i = 1 \tag{3.49}$$

$$\mathbf{a} = \sum_{i=1}^{M}\alpha_i\mathbf{x}^i - \sum_{i=M+1}^{N}\alpha_i\mathbf{x}^i \tag{3.50}$$

$$\beta_i = C_1 - \alpha_i, \quad i = 1,...,M \tag{3.51}$$

$$\beta_i = C_2 - \alpha_i, \quad i = M+1,...,N \tag{3.52}$$

Expresamos la función lagrangiana (3.41) sólo en función de $\boldsymbol{\alpha}$ gracias a las ecuaciones (3.49)-(3.52):

$$\begin{aligned} L(\boldsymbol{\alpha}) = & \sum_{i=1}^{M}\alpha_i\langle\mathbf{x}^i,\mathbf{x}^i\rangle - \sum_{i=M+1}^{N}\alpha_i\langle\mathbf{x}^i,\mathbf{x}^i\rangle - \sum_{i,\ell=1}^{M}\alpha_i\alpha_\ell\langle\mathbf{x}^i,\mathbf{x}^\ell\rangle \\ & +2\sum_{i=1}^{M}\sum_{\ell=M+1}^{N}\alpha_i\alpha_\ell\langle\mathbf{x}^i,\mathbf{x}^\ell\rangle - \sum_{i,\ell=M+1}^{N}\alpha_i\alpha_\ell\langle\mathbf{x}^i,\mathbf{x}^\ell\rangle \end{aligned} \tag{3.53}$$

Para simplificar la función lagrangiana obtenida en (3.53) definimos una nuevas variables $\alpha_i' = y^i\alpha_i$ para $i = 1,...,N$. Al definir dichas variables, la ecuación (3.49) se convierte en

$$\sum_{i=1}^{N}\alpha_i' = 1 \tag{3.54}$$

y la ecuación (3.50) en

$$\mathbf{a} = \sum_{i=1}^{N} \alpha_i' x_i \tag{3.55}$$

Sustituyendo las ecuaciones (3.54) y (3.55) en (3.41) obtenemos una versión simplificada de la función (3.53) que nos permite construir el problema dual de la siguiente forma:

$$\begin{aligned} \max_{\alpha_i', \forall i} \quad & \sum_{i=1}^{N} \alpha_i' \langle \mathbf{x}^i, \mathbf{x}^i \rangle - \sum_{i,\ell=1}^{N} \alpha_i' \alpha_\ell' \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle \\ \text{s.a.} \quad & \sum_{i=1}^{N} \alpha_i' = 1 \\ & 0 \leq |\alpha_i'| \leq C \quad i = 1, ..., N \end{aligned} \tag{3.56}$$

donde $C = \max\{C_1, C_2\}$. Debido a que por las ecuaciones (3.51) y (3.52) tenemos que

$$\alpha_i = C_1 - \beta_i \leq C_1, \quad i = 1, ..., M \tag{3.57}$$

$$\alpha_i = C_2 - \beta_i \leq C_2, \quad i = M+1, ..., N \tag{3.58}$$

por lo tanto, $\alpha_i \leq \max\{C_1, C_2\}$, $\forall i$.

Con la ecuación (3.50) escrita en función de las nuevas variables tenemos que el centro de la esfera es una combinación lineal de los datos con las variables duales que se obtienen del problema (3.56). Una vez obtenida la solución de dicho problema dual, $\boldsymbol{\alpha}^{opt}$, podemos calcular el centro óptimo de la esfera:

$$\mathbf{a}^{opt} = \sum_{i=1}^{N} \alpha_i'^{,opt} \mathbf{x}^i \tag{3.59}$$

Por último, calcularemos el radio óptimo. Para ello, en primer lugar, demostraremos que existe algún i tal que $\alpha_i > 0$ con $i = 1, ..., N$. Lo haremos por reducción al absurdo, supongamos que $\alpha_i = 0$ para todo $i = 1, ..., N$. Por la ecuación (3.49) tendríamos que cero es igual uno lo que es imposible, es decir, llegamos a una contradicción. Luego, existe al menos un dato i tal que $\alpha_i > 0$. De manera análoga a las secciones anteriores, existe $\alpha_i = C_1$ con $i = 1, ..., M$ y $\alpha_i = C_2$ con $i = M+1, ..., N$.

A continuación, pasemos a calcular los vectores soporte. Supongamos que para algún punto i, $0 < \alpha_i < C_1$ para $i = 1, ..., M$. Por la ecuación (3.51), $\beta_i > 0$ para $i = 1, ..., M$, y sustituyendo esto en (3.48), $\xi_i = 0$ para $i = 1, ..., M$. Luego, por las ecuaciones (3.46) tenemos que:

$$\|\mathbf{x}^i - \mathbf{a}\|^2 = R^2, \quad i = 1, ..., M \tag{3.60}$$

De forma análoga ocurre si para algún punto i, $0 < \alpha_i < C_2$ con $i = M+1, ..., N$. Así que concluimos que:

$$\|\mathbf{x}^i - \mathbf{a}\|^2 = R^2, \quad i = M+1, ..., N \tag{3.61}$$

Luego, los datos objetivo tal que $0 < \alpha_i < C_1$ y los datos atípicos que verifican $0 < \alpha_i < C_2$ se encuentran en la frontera de la esfera, y son los vectores soporte.

Consideremos ahora el caso en el que $\alpha_i = C_1$ para $i = 1, ..., M$. La ecuación (3.51) implica que $\beta_i = 0$ para $i = 1, ..., M$, por ello de la ecuación (3.48) tenemos que $\xi_i > 0$ para $i = 1, ..., M$. Además, de la ecuación (3.46) concluimos que:

$$\|\mathbf{x}^i - \mathbf{a}\|^2 = R^2 + \xi_i, \quad i = 1, ..., M \tag{3.62}$$

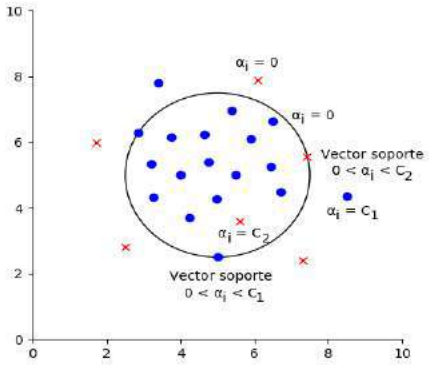


Figura 3.5: Valores de los multiplicadores de Lagrange en función de la posición de cada dato i.

Con el mismo razonamiento para el caso en el que $\alpha_i = C_2$ para $i = M+1, ..., N$. Obtenemos que:

$$\|\mathbf{x}^i - \mathbf{a}\|^2 = R^2 - \xi_i, \quad i = M+1, ..., N \tag{3.63}$$

Las ecuaciones (3.62) y (3.63) nos indican que los datos con $\alpha_i = C_1$ para $i = 1, ..., M$ o con $\alpha_i = C_2$ para $i = M+1, ..., N$ están mal clasificados.

Como último caso tenemos que $\alpha_i = 0$ para $i = 1, ..., N$. Gracias a las ecuaciones (3.51) y (3.52) obtenemos que $\beta_i > 0$ para $i = 1, ..., N$, y sustituyendo esto en (3.48), sabemos que $\xi_i = 0$ para $i = 1, ..., N$. Con estas dos conclusiones y las ecuaciones (3.46) y (3.47) tenemos que

$$\|\mathbf{x}^i - \mathbf{a}\|^2 < R^2, \quad i = 1, ..., M \tag{3.64}$$

$$\|\mathbf{x}^i - \mathbf{a}\|^2 > R^2, \quad i = M+1, ..., N \tag{3.65}$$

Luego, se trata de los datos bien clasificados pero que no se hayan en la frontera de la esfera. En la Figura 3.5, podemos observar cada dato i con los distintos valores de α_i

Como los vectores soporte son los que están en la frontera de la esfera, concluimos que el radio óptimo se obtiene de la siguiente forma:

$$\begin{aligned} R^{2,\,opt} = & \quad \|\mathbf{x}^s - \mathbf{a}^{opt}\|^2 = \langle \mathbf{x}^s - \mathbf{a}^{opt}, \mathbf{x}^s - \mathbf{a}^{opt} \rangle = \\ = & \quad \langle \mathbf{x}^s, \mathbf{x}^s \rangle - 2 \sum_{i=1}^{M} \alpha_i' \langle \mathbf{x}^i, \mathbf{x}^s \rangle + \sum_{i,\,\ell=1}^{M} \alpha_i' \alpha_\ell' \langle \mathbf{x}^i, \mathbf{x}^\ell \rangle \end{aligned} \tag{3.66}$$

donde $\mathbf{x}^s \in VS$.

Dado un nuevo dato, $\mathbf{x}^{test}$, si $\|\mathbf{x}^{test} - \mathbf{a}^{opt}\|^2$ es mayor que el radio óptimo, $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = -1$, es decir, será un dato atípico. En caso contrario, $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = 1$ y será un dato objetivo.

Este método solo calcula una esfera alrededor de los datos en el espacio de entrada, pero, normalmente, los datos no están distribuidos esféricamente. Por eso, al igual en el método SVM del Capítulo 2 y en el método SVR de la sección 1.2, introducimos las funciones kernels. Es decir, transformamos los datos de entrada a un nuevo espacio en el que los nuevos datos si pueden ser ajustados mediante una esfera, donde el producto escalar de los nuevos datos puede sustituirse por una función kernel.

Por lo tanto, el problema de optimización dual SVDD quedaría de la siguiente forma:

$$\begin{aligned} \max_{\alpha'_i, \forall i} \quad & \sum_{i=1}^{N} \alpha'_i k(\mathbf{x}^i, \mathbf{x}^i) - \sum_{i,\ell=1}^{N} \alpha'_i \alpha'_\ell k(\mathbf{x}^i, \mathbf{x}^\ell) \\ \text{s.a.} \quad & \sum_{i=1}^{N} \alpha'_i = 1 \\ & 0 \leq |\alpha'_i| \leq C \quad i = 1, ..., N \end{aligned} \tag{3.67}$$

De la misma forma, el centro óptimo de la esfera en función de la solución óptima del dual, $\alpha_i^{\prime,\, opt}$, es:

$$\mathbf{a}^{opt} = \sum_{i=1}^{N} \alpha_i^{\prime,\, opt} \phi(\mathbf{x}^i) \tag{3.68}$$

Por tanto, el radio óptimo es:

$$R^{2,\, opt} = k(\mathbf{x}^s, \mathbf{x}^s) - 2 \sum_{i=1}^{M} \alpha'_i k(\mathbf{x}^i, \mathbf{x}^s) + \sum_{i,\ell=1}^{M} \alpha'_i \alpha'_\ell k(\mathbf{x}^i, \mathbf{x}^\ell) \tag{3.69}$$

donde $\mathbf{x}^s \in VS$

Como en el caso *soft margin* SVDD, para obtener la predicción de un nuevo dato, $\mathbf{x}^{test}$, comparamos $\|\mathbf{x}^{test} - \mathbf{a}^{opt}\|^2$ con el radio óptimo. Si $\|\mathbf{x}^{test} - \mathbf{a}^{opt}\|^2$ es mayor que el radio óptimo, $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = -1$, es decir, será un dato atípico. En caso contrario, $\mathbf{x}^{test}$ tendrá etiqueta $y^{test} = 1$ y será un dato objetivo.

Capítulo 4

SVM, SVR y SVDD en Python

Este capítulo está dedicado a la aplicación práctica del desarrollo teoríco que se ha visto en los Capítulos 2 y 3. El código fuente utilizado para la realización de estos experimentos computacionales puede descargarse en [Valenzuela-González, 2022].

4.1. SVM

En esta primera sección vamos a ejecutar un experimento sobre el algoritmo visto para el SVM en el Capítulo 2 basandonos en [ScikitLearn, 2022a]. El objetivo de este experimento será comparar los resultados obtenidos al aplicar los distintos modelos de SVM que se obtenien según la función kernel utilizada a un conjunto de datos concreto, es decir, según las distintas funciones kernels vistas en la Sección 2.3.

Para ellos hemos escogido un conjunto de datos de la universidad de Wisconsin sobre el diagnóstico de cáncer de máma en 1995 que los podemos encontrar en [Dua and Graff, 2017]. El conjunto de datos está formado por 569 pacientes de los cuales se tomó una imagen del tejido mamario. De cada paciente conocemos 32 variables, las dos primeras variables son el número de identificación y el diagnóstico del paciente. Se recogieron 10 características de valor real para cada núcleo celular del tejido mamario: el radio (media de las distancia desde el centro a los puntos del perímetro), la textura, el perímetro, el área, la uniformidad (variación en las longitudes del radio), la compacidad ($\text{perímetro}^2/(\text{área}-1)$), la concavidad, el número de puntos cóncavos, la simetría y la dimesión fractal. Para cada una de estas características se calcularon la media, el error estándar y el peor valor de las características, obteniendo así las 30 características restantes.

Para realizar el estudio, comenzamos separando los datos en una matriz X, formada por las variables explicativas, $\mathbf{x}^i$ con $i = 1, ..., 569$, y un vector $\mathbf{y}$, formado por la variable respuesta de cada dato i, y^i. Por simplicidad, y con propósitos ilustrativos, sólo utilizaremos dos variables explicativas de las 32 disponibles. En particular, usaremos las variables "*radius_worst*" (peor valor del radio del tejido mamario) y "*texture_worst*" (peor textura del tejido mamario) debido a que son las que tienen mayor poder de predicción según la Tabla 4 del artículo [Jiménez-Cordero et al., 2021]. Por lo tanto, la matriz X es una matriz de 569 filas y 2 columnas. Por otro lado, el vector $\mathbf{y}$ es un vector de 569 filas que nos proporciona la etiqueta de los datos. Dado que se trata de un problema clasificación binaria, tenemos dos clases. La clase 1 está formada por los pacientes sanos que consta de 357 pacientes. La clase -1 está formada por los pacientes que tienen cáncer de máma.

A continuación, dividimos los datos en dos conjuntos, un conjunto de entrenamiento con el que creamos el modelo SVM y un conjunto test, formado por los nuevos individuos

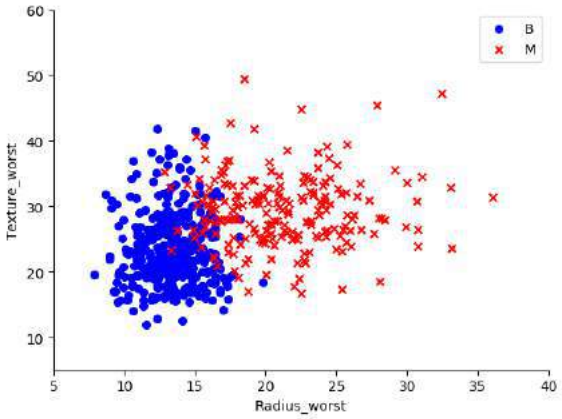


Figura 4.1: Representación de los datos del diagnostico de cáncer de máma donde B representa los pacientes sanos y M los pacientes con cáncer.

que deseamos clasificar correctamente con el modelo SVM anteriormente creado. Esto se realiza usando el comando *train_test_split* contenido en la biblioteca *model_selection*. Tomamos aleatoriamente el 80 % de los datos para el conjunto de entrenamiento y el 20 % para el conjunto test. El código para este procedimiento es el siguiente:

```
data = pd.read_csv('data.csv')
X = pd.DataFrame(data, columns = ['radius_worst','texture_worst'])
y = data['diagnosis']
X_train, X_test, y_train, y_test = train_test_split(X, y,
     test_size = 0.20, random_state = 0)
```

Código Fuente 4.1: Lectura de datos, y división en conjunto de entrenamiento y conjunto test.

Una vez que los datos están separados, comenzamos la construcción del SVM. En primer lugar, entrenamos el clasificador con el conjunto de entrenamiento utilizando el comando SVC contenido en la biblioteca *svm*. Por simplicidad, hemos supuesto en todos los experimentos que el parámetro de regulación C toma el valor 1000. Es decir, tomaremos $C = 1000$. A continuación, predecimos las etiquetas del conjunto test. Por último, visualizamos la distribución de errores cometidos por el clasificador utilizando la matriz de confusión y el rendimiento del SVM con el comando accuracy. Más información sobre la matriz de confusión y el comando accuracy la podemos obtener en [Patro and Patra, 2014].

Pasemos a hacer el estudio de los datos con los diferentes tipos de kernel:

- SVM con kernel lineal

  Como hemos indicado anteriormente, comenzamos con la construción del modelo SVM usando el conjunto de entrenamiento, previamente creado. El resultado puede ser visto gráficamente en la Figura 4.2.

```
from sklearn.svm import SVC
svclassifier = SVC(kernel="linear", C=1000)
svclassifier.fit(X_train, y_train)
print(svclassifier)
```

```
SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,
='ovr', degree=3, gamma='auto_deprecated',
kernel='linear', max_iter=-1, probability=False,
random_state=None, shrinking=True, tol=0.001, verbose=False)
```

Código Fuente 4.2: Construcción del SVM utilizando el kernel lineal.

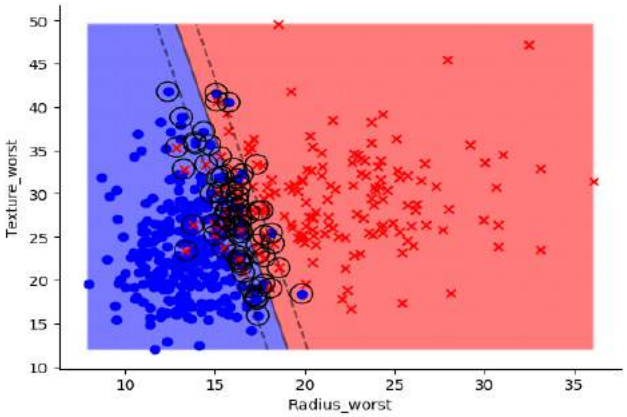


Figura 4.2: Modelo SVM con kernel lineal. Represtación de los datos de entrenamiento clasificados donde los vectores soporte están rodeado por un círculo.

A continuación, predecimos las etiquetas del conjunto test utilizando el modelo SVM que acabamos de construir con el conjunto de entrenamiento. Dichas etiquetas están guardadas en el vector *y_pred*. De esta forma, observando la matriz de confusión de esta predicción obtenemos que de los 67 tumores benignos pertenecientes al conjunto test, 65 de ellos han sido clasificados como tumores benignos y 2 como tumores malignos. Por otro lado, de los 47 tumores malignos que nos encontramos en el conjunto test 45 han sido clasificados como tumores malignos y 2 como tumores benignos. Esto lo podemos ver en el Código Fuente 4.3.

```
from sklearn.metrics import confusion_matrix
confusion_matrix = pd.crosstab(
    y_test.ravel(),
    y_pred,
    rownames=['Real']
    colnames=['Prediccion'])
print(confusion_matrix)
```

```
Prediccion   B   M
Real
B           65   2
M            2  45
```

Código Fuente 4.3: Matriz de confusión.

Además, el comando *accuracy_score* nos proporciona el porcentaje de acierto del SVM que en este caso es del 96.49 % de los individuos pertenecientes al conjunto test.

```
accuracy = accuracy_score(
        y_true    = y_test,
        y_pred    = y_pred,
        normalize = True)
print(f"{100*accuracy}%")
```

```
96.49122807017544%
```

Código Fuente 4.4: Comando accuracy_score para ver el rendimiento del SVM.

- SVM con kernel polinomial

  Para realizar el análisis de los datos con un SVM utilizando un kernel polinomial, asumiremos que el grado del polinomio para dicho kernel es 3. La realización del modelo SVM con kernel polinomial seguirá los mismos pasos que para el SVM con kernel lineal. Por ello, en primer lugar, construimos el SVM usando el conjunto de entrenamiento. Podemos visualizar el modelo gráficamente en la Figura 4.3.

```
from sklearn.svm import SVC
svclassifier = SVC(kernel="poly", C=1000)
svclassifier.fit(X_train, y_train)
print(svclassifier)
```

```
SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3,
gamma='auto_deprecated', kernel='poly', max_iter=-1,
probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)
```

Código Fuente 4.5: Construcción del SVM con kernel polinomial.

A continuación, predecimos las etiquetas del conjunto test utilizando el modelo SVM que acabamos de construir. Utilizando la matriz de confusión de esta predicción (ver Código Fuente 4.6) obtenemos que de los 67 tumores benignos que hay en el conjunto test, 63 de ellos han sido clasificados como tumores benignos y 4 como

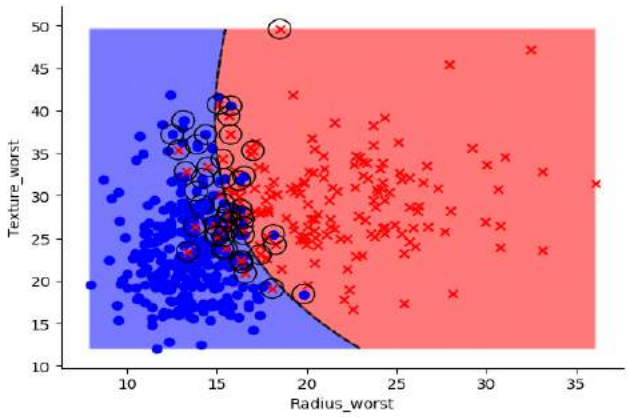


Figura 4.3: Representación del conjunto de entrenamiento clasificado por el modelo SVM con kernel polinomial. Para diferenciar los vectores soporte del resto de datos han sido marcados por un círculo.

tumores malignos. Por otro lado, de los 47 tumores malignos que nos encontramos en el conjunto test 44 han sido clasificados como tumores malignos y 3 como tumores benignos.

```
from sklearn.metrics import confusion_matrix
confusion_matrix = pd.crosstab(
    y_test.ravel(),
    y_pred,
    rownames=['Real'],
    colnames=['Prediccion'])
print(confusion_matrix)
```

```
Prediccion   B   M
Real
B           63   4
M            3  44
```

Código Fuente 4.6: Matriz de confusión.

Además, el comando *accuracy_score* en este caso nos indica que un 93.86 % de los individuos del conjunto test han sido clasificados correctamente.

```
accuracy = accuracy_score(
        y_true    = y_test,
        y_pred    = y_pred,
        normalize = True)
print("")
print(f"{100*accuracy}%")
```

```
93.85964912280701%
```

Código Fuente 4.7: Rendimiento del SVM con kernel polinomial.

- SVM con kernel gaussiano

  De la misma forma que en los SVM con kernel lineal y polinomial, para realizar el análisis de los datos con un SVM utilizando un kernel gaussiano, comenzamos construyendo el modelo SVM usando el conjunto de entrenamiento. Para este modelo tomaremos $\gamma = 0.3$. Además, podemos ver el modelo gráficamente en la Figura 4.4.

```
from sklearn.svm import SVC
svclassifier = SVC(kernel="rbf", C=1000, gamma=0.3)
svclassifier.fit(X_train, y_train)
print(svclassifier)
```

```
SVC(C=1000, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma=0.3,
kernel='rbf', max_iter=-1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
```

Código Fuente 4.8: Construcción del SVM con kernel gaussiano.

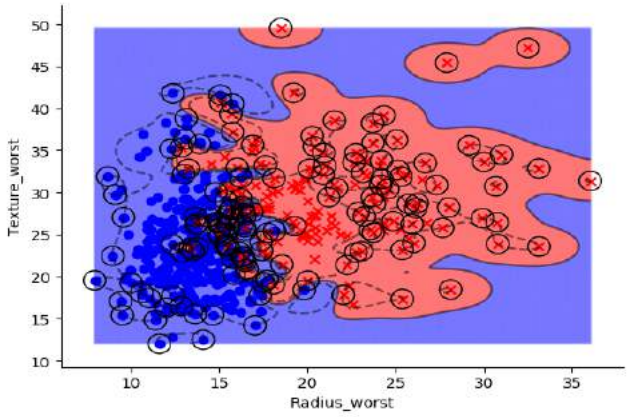


Figura 4.4: Modelo SVM con kernel gaussiano utilizando el conjunto de entrenamiento. Los datos marcados con un círculo representan los vectores soporte del modelo.

A continuación, predecimos la clase y del conjunto test utilizando el SVM que acabamos de construir. Utilizando la matriz de confusión de esta predicción obtenemos que de los 67 tumores benignos que hay en el conjunto test 57 de ellos han sido clasificados como tumores benignos y 10 como tumores malignos. Por otro lado, de los 47 tumores malignos que nos encontramos en el conjunto test 42 han sido clasificados como tumores malignos y 5 como tumores benignos. Podemos verlo en el Código Fuente 4.9.

```
from sklearn.metrics import confusion_matrix
confusion_matrix = pd.crosstab(
    y_test.ravel(),
    y_pred,
    rownames=['Real'],
    colnames=['Prediccion'])
print(confusion_matrix)
```

```
Prediccion   B   M
Real
B           57  10
M            5  42
```

Código Fuente 4.9: Matriz de confusión.

Además, el comando *accuracy_score* en este caso nos indica que un 86.84 % de los individuos del conjunto test han sido clasificados correctamente.

```
accuracy = accuracy_score(
        y_true    = y_test,
        y_pred    = y_pred,
        normalize = True
       )
print("")
print(f"{100*accuracy}%")
```

```
86.8421052631579%
```

Código Fuente 4.10: Comando acurracy_score para obtener la precisón del SVM.

Para evaluar el mejor clasificador de entre los tres expuestos anteriormente, vamos a utilizar el porcentaje de acierto de predicción de cada método SVM con $C = 1000$ sobre el conjunto test: para el SVM con kernel lineal el porcentaje de acierto ha sido 96.49 %, para el SVM con kernel polinomial un 93.86 %, y para el SVM con kernel gaussiano, 86.84 %.

Por lo tanto, el SVM utilizando un kernel lineal tiene mayor porcentaje de acierto que con cualquiera de los otros dos tipos de kernel.

4.2. SVR

Similarmente al procedimiento en la sección anterior, en esta sección, vamos a desarrollar varios ejemplos prácticos del algoritmo SVR visto en la Sección 3.1 basandonos en [ScikitLearn, 2022b]. El primer ejemplo consistirá en generar unos datos a los cuales le aplicaremos el método SVR con diferentes funciones kernels. Para la construcción de este método supondremos que el parámetro de regulación C toma el valor 100. Finalizaremos el ejemplo comparando la tasa de error cometida por cada uno de los modelos, viendo así qué modelo de SVR se ajusta mejor a los datos. El error cometido se calculará con

el error cuadrático medio que mide el promedio de los errores elevado al cuadrado. Más información sobre el error cuadrático medio la podemos obtener en [Hastie et al., 2009].

Comezaremos generando los datos de la siguiente forma:

```
np.random.seed(1)
X = np.sort(5 * np.random.rand(400, 1), axis=0)
y = np.sin(X).ravel()
y[::5] += 3 * (0.5 - np.random.rand(80))
```

Código Fuente 4.11: Obtención de los datos.

A partir de estos datos, realizamos tres modelos de SVR de forma análoga a la Sección 4.1. Para ello, utilizamos los kernels gaussiano, lineal y polinomial.

```
svr_rbf = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1)
svr_lin = SVR(kernel="linear", C=100, gamma="auto")
svr_poly = SVR(kernel="poly", C=100, gamma="auto", degree=3,
               epsilon=0.1, coef0=1)
```

Código Fuente 4.12: Construcción de los modelos SVR.

Cada uno de los modelos SVR que acabamos de realizar pueden ser observados en la Figura 4.5

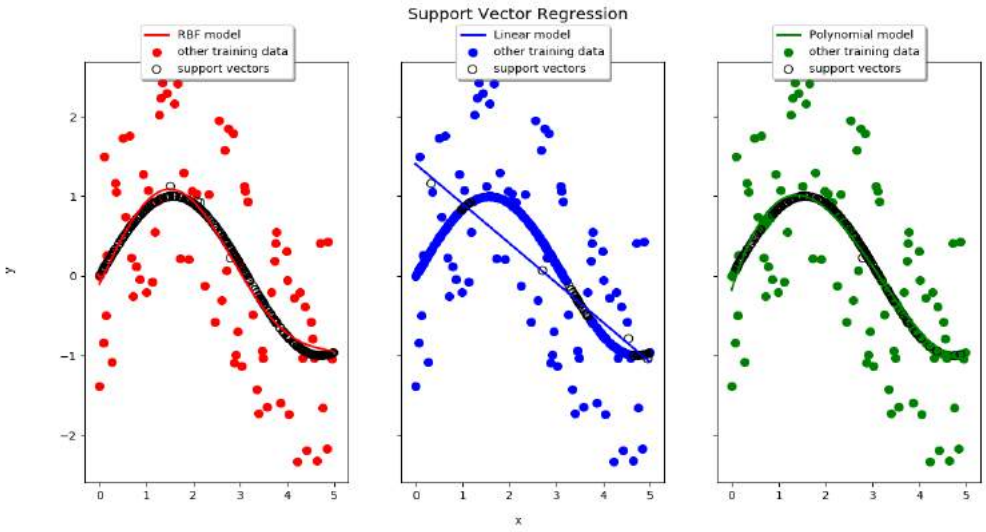


Figura 4.5: Representación de los modelos SVR con kernel gaussiano, lineal y polinomial.

En dicha figura, observamos que el modelo SVR con kernel lineal es el que peor se ajusta a los datos mientras que los modelos SVR con kernel gaussiano o polinomial se ajustan mejor. Sin embargo, a simple vista, no podemos decidir cual será el que mejor se ajusta a los datos. Por lo tanto, vamos a calcular el error cuadrático medio de cada modelo SVR como se indica a continuación:

```
    y_predrbf = svr_rbf.predict(X)
    mse_rbf = (np.square(y - y_predrbf)).mean()
    print(mse_rbf)

    y_predlin = svr_lin.predict(X)
    mse_lin = (np.square(y - y_predlin)).mean()
    print(mse_lin)

    y_predpoly = svr_poly.predict(X)
    mse_poly = (np.square(y - y_predpoly)).mean()
    print(mse_poly)
```

```
    0.17957605760826234
    0.39501255265130264
    0.1801029854676299
```

Código Fuente 4.13: Error cuadrático medio del modelo SVR con kernel gaussiano, lineal y polinomial, respectivamente.

Como evidencian los resultados, el modelo que mejor se ajusta a los datos estudiados es el SVR con kernel gaussiano. Debido a esto, vamos a realizar un segundo experimento, de manera análoga al anterior, en el que estudiaremos diferentes modelos de SVR con kernel gaussiano en función del parámetro ε utilizando los siguientes valores 0.05, 0.1 y 0.3.

```
    svr_rbf1 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.05)
    svr_rbf2 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1)
    svr_rbf3 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.3)
```

Código Fuente 4.14: Construcción de tres modelos de SVR con kernel gaussiano en función de ε.

Las funciones de regresión obtenidas en cada uno de los modelos de SVR con kernel gaussiano que acabamos de realizar podemos observarlas gráficamente en la Figura 4.6

Al igual que antes, el error cuadrático medio nos proporciona el modelo que mejor se ajusta a los datos. El SVR con kernel gaussiano y $\varepsilon = 0.05$ tiene un error de 0.177334 mientras que con $\varepsilon = 0.1$ el error es 0.179576 y con $\varepsilon = 0.3$ es 0.190278. Por lo tanto el SVR con kernel gaussiano y $\varepsilon = 0.05$ es el mejor modelo para estos datos.

Por último, vamos realizar un estudio del SVR con kernel gaussiano dependiendo del valor de ε y del valor de γ. Esto lo veremos a través de un mapa de calor.

En primer lugar, construimos 9 modelos de SVR con kernel gaussiano, fijando $C = 100$, con ε tomando los valores 0.05, 0.1 y 0.3, y γ tomando los valores 0.01, 0.1 y 0.9.El código puede verse a continuación:

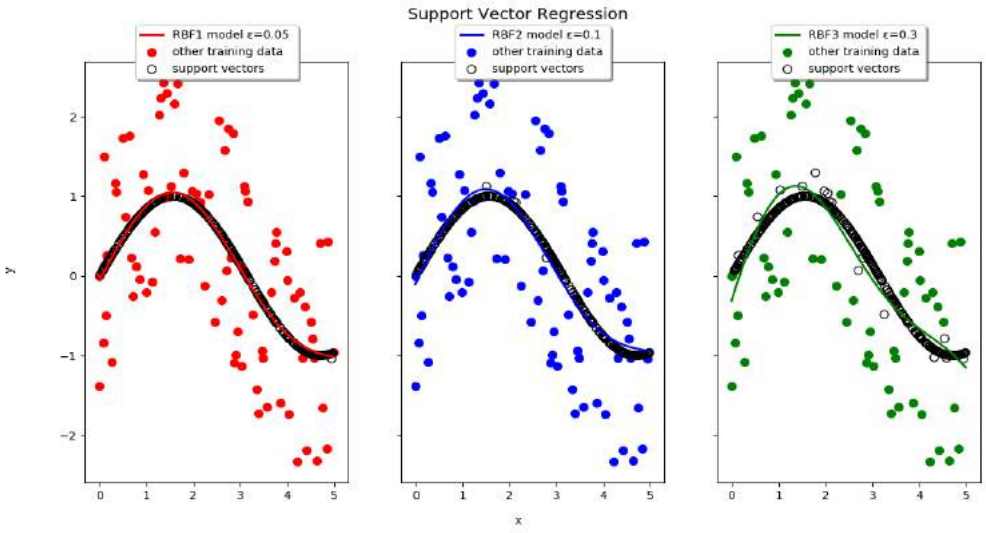


Figura 4.6: Modelos de SVR con kernel gaussiano donde ε toma los valores 0.05, 0.1, y 0.3 (de izquierda a derecha).

```
    svr_rbf1 = SVR(kernel="rbf", C=100, gamma=0.01, epsilon=0.05)
    svr_rbf2 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.05)
    svr_rbf3 = SVR(kernel="rbf", C=100, gamma=0.9, epsilon=0.05)

    svr_rbf21 = SVR(kernel="rbf", C=100, gamma=0.01, epsilon=0.1)
    svr_rbf22 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.1)
    svr_rbf23 = SVR(kernel="rbf", C=100, gamma=0.9, epsilon=0.1)

    svr_rbf31 = SVR(kernel="rbf", C=100, gamma=0.01, epsilon=0.3)
    svr_rbf32 = SVR(kernel="rbf", C=100, gamma=0.1, epsilon=0.3)
    svr_rbf33 = SVR(kernel="rbf", C=100, gamma=0.9, epsilon=0.3)
```

Código Fuente 4.15: Construcción de los modelos de SVR con kernel gaussiano en función de los valores de ε y γ.

Utilizando el error cuadrático medio de cada modelo de SVR con kernel gaussiano y el comando *sns.heatmap* de la librería *seaborn*, construímos un mapa de calor. Dicho mapa de calor puede verse en la Figura 4.7. donde los valores de ε se encuentran en el eje de abcisas y los valores de γ en el eje de ordenadas. Según podemos observar en esta figura, el SVR con kernel gaussiano, $\varepsilon = 0.1$ y $\gamma = 0.9$ es el mejor modelo ya que es el que tiene mejor error cuadrático medio. Además, puede observar que a mayor valor de γ mejor es el ajuste del modelo.

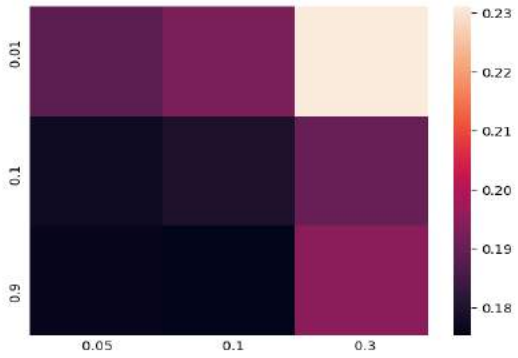


Figura 4.7: Mapa de calor basado en el error cuadrático medio de nueve modelos de SVR con kernel gaussiano en función de los valores de ε (eje de abcisas) y γ (eje de ordenadas).

4.3. SVDD

En esta última sección, vamos a realizar un ejemplo práctico sobre el algoritmo visto para el SVDD en la Sección 3.2. Este ejemplo está basado en el cógido fuente de [Qiu, 2022]. En primer lugar, vamos a generar 100 datos objetivo y 5 *outliers* de la base de datos BananaDataSet (ver Figura 4.8).

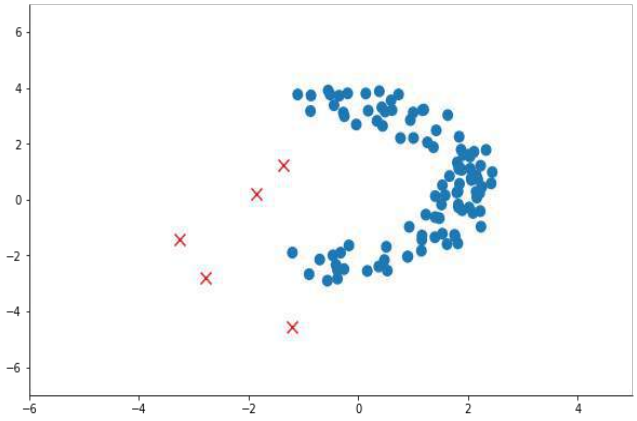


Figura 4.8: Representación del conjunto total de datos generados.

De forma análoga al experimento de la Sección 4.1, vamos a separar los datos en una matriz X formada por las variables explicativas, y un vector $\mathbf{y}$ con la variable respuesta de cada dato. Además de realizar esta separación de los datos, estos serán divididos en dos conjuntos, uno de entrenamiento para la construcción del modelo SVDD y otro test. Esta división se realizará con el comando *.split*(), tomando el 70 % de los datos para el conjunto de entrenamiento y el 30 % para el conjunto test.

```
import sys
sys.path.append("..")
from src.BaseSVDD import BaseSVDD, BananaDataset

X, y = BananaDataset.generate(number_p=100, number_n=5, display='on')
X_train, X_test, y_train, y_test = BananaDataset.split(X, y,
                                        ratio=0.3)
```

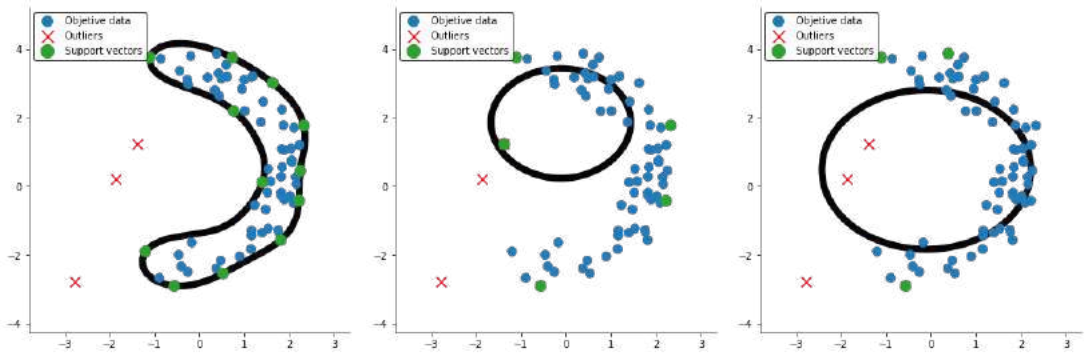
Código Fuente 4.16: Obtención y división de los datos.

A continuación, construimos tres modelos del SVDD utilizando las funciones kernels gaussiano, polinomial y lineal con $C = 0.9$.

```
kL = {"1": BaseSVDD(C=0.9, kernel='rbf', gamma=0.3, display='on'),
      "2": BaseSVDD(C=0.9, kernel='poly', degree=2, display='on'),
      "3": BaseSVDD(C=0.9, kernel='linear', display='on')}

for i in range(len(kL)):
    svdd = kL.get(str(i+1))
    svdd.fit(X_train,  y_train)
```

Código Fuente 4.17: Modelos del SVDD.



(a) (b) (c)

Figura 4.9: Representación de los modelos de SVDD: (a) SVDD con kernel gaussiano, (b) SVDD con kernel polinomial y (c) SVDD con kernel lineal.

La solución de los tres modelos de SVDD la podemos ver gráficamente en la Figura 4.9. En ella, claramente se visualiza que el SVDD con kernel polinomial y SVDD con kernel lineal, no son buenos modelos. Efectivamente, utilizando el comando *accuracy* obtenemos que el SVDD con kernel gaussiano tiene un rendimiento del 93.15 %, el SVDD con kernel polinomial del 20.55 % y el SVDD con kernel lineal del 36.99 %. Por lo tanto, el mejor modelo para estos datos es el SVDD con kernel gaussiano.

Capítulo 5

Conclusiones

El objetivo principal de este trabajo ha sido realizar un análisis de los fundamentos matemáticos de las Máquinas de Vectores Soporte, cuya base teórica está basada en la teoría del aprendizaje estadístico, y extenderlo a los modelos de SVR y SVDD. Además, también se han realizado varios experimentos computacionales que muestran la eficiencia y gran utilidad de estos métodos.

Desde el punto de vista teórico, hemos desarrollado el problema de optimización, tanto en su versión primal como en la dual. Además, gracias a la teoría derivada de las funciones kernel, hemos podido adaptar los métodos SVM, SVR y SVDD al caso no lineal.

Por otro lado, desde el punto de vista práctico, hemos llevado a cabo distintos experimentos computacionales con los métodos SVM, SVR y SVDD. En dichos experimentos, se ha comparado la eficiencia del método usando varias funciones kernel. También se ha estudiado el comportamiento de los modelos según los valores de los distintos hiperpárametros, como por ejemplo ε y γ.

Apéndice A

Definiciones

En este apéndice definiremos algunos conceptos auxiliares que se han mencionado en el trabajo.

Definición A.1. Sea $X \in \mathbb{R}^d$. Decimos que X es compacto si para toda familia $\{O_i\}_{i\in I}$ de abiertos de $\mathbb{R}^d$ tal que $X \subset \bigcup_{i\in I} O_i$, se verifica que existen $i_1, ..., i_p \in I$ tales que $X \subset \bigcup_{j}^{p} O_{ij}$.

Definición A.2. Sea X un conjunto compacto de $\mathbb{R}^d$. $L_2(X)$ denota el espacio vectorial de las funciones cuadradas integrables sobre dicho conjunto, y sobre el que se definen la suma y la multiplicación. Es decir,

$$L_2(X) = \left\{ f : \int_X f(x)^2 dx < \infty \right\}.$$

Definición A.3. Sea $X \subset \mathbb{R}^d$ y $f_n : X \longrightarrow \mathbb{R}^m$ una sucesión de funciones. Se dice que f_n converge uniformemente a la función f definida en X si para todo $\epsilon > 0$ existe un número natural n_0 tal que si $n \geq n_0$ se verifica $\|f_n(x) - f(x)\| < \epsilon$ para todo $x \in X$.

Definición A.4. Una sucesión, $\{h_n\}_{n\geq 1}$, decimos que es de Cauchy si satisface la siguiente propiedad:

$$\sup_{m>n} \|h_n - h_m\| \longrightarrow 0, \text{ cuando } n \longrightarrow \infty.$$

Definición A.5. Un espacio $\mathcal{F}$ es completo si toda sucesión de Cauchy $\{h_n\}_{n\geq 1}$ de elementos de $\mathcal{F}$ converge a un elemento $h \in \mathcal{F}$.

Definición A.6. Un espacio $\mathcal{F}$ es separable si para cualquier $\epsilon > 0$ existe un conjunto finito de elementos $h_1, ..., h_n$ de $\mathcal{F}$ tal que para todo $h \in \mathcal{F}$

$$\min_i \|h_i - h\| < \epsilon$$

Definición A.7. Un espacio de Hilbert $\mathcal{F}$ es un espacio de producto escalares completo y separable.

Definición A.8. El producto tensorial de dos matrices, $K_1 \in \mathcal{M}_{i\times\ell}$ y $K_2 \in \mathcal{M}_{p\times q}$, es

$$K_1 \otimes K_2 = \begin{pmatrix} (K_1)_{11}K_2 & (K_1)_{12}K_2 & \cdots & (K_1)_{1\ell}K_2 \\ (K_1)_{21}K_2 & (K_1)_{22}K_2 & \cdots & (K_1)_{2\ell}K_2 \\ \vdots & \vdots & \ddots & \vdots \\ (K_1)_{i1}K_2 & (K_1)_{i2}K_2 & \cdots & (K_1)_{i\ell}K_2 \end{pmatrix}$$

Más explícitamente, tenemos

$$K_1 \otimes K_2 = \begin{pmatrix} (K_1)_{11}(K_2)_{11} & (K_1)_{11}(K_2)_{12} & \cdots & (K_1)_{11}(K_2)_{1q} & \cdots & \cdots & (K_1)_{1\ell}(K_2)_{1q} \\ (K_1)_{11}(K_2)_{21} & (K_1)_{11}(K_2)_{22} & \cdots & (K_1)_{11}(K_2)_{2q} & \cdots & \cdots & (K_1)_{1\ell}(K_2)_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots \\ (K_1)_{11}(K_2)_{p1} & (K_1)_{11}(K_2)_{p2} & \cdots & (K_1)_{11}(K_2)_{pq} & \cdots & \cdots & (K_1)_{1\ell}(K_2)_{pq} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots \\ (K_1)_{i1}(K_2)_{p1} & (K_1)_{i1}(K_2)_{p2} & \cdots & (K_1)_{i1}(K_2)_{pq} & \cdots & \cdots & (K_1)_{i\ell}(K_2)_{pq} \end{pmatrix}$$

Bibliografía

H. A. Abu Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. Eyal Salman, and V. S. Prasath. Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data*, 7(4):221–248, 2019.

F. Aiolli, A. Sperduti, and Y. Singer. Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6(5), 2005.

B. Baesens, R. Setiono, C. Mues, and J. Vanthienen. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management science*, 49(3):312–329, 2003.

S. Benítez-Peña, E. Carrizosa, V. Guerrero, M. D. Jiménez-Gamero, B. Martín-Barragán, C. Molero-Río, P. Ramírez-Cobo, D. R. Morales, and M. R. Sillero-Denamiel. On sparse ensemble methods: An application to short-term predictions of the evolution of covid-19. *European Journal of Operational Research*, 295(2):648–663, 2021.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

G. Biau and E. Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

N. Boodhun and M. Jayabalan. Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems*, 4(2):145–154, 2018.

S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

E. Carrizosa and D. Romero Morales. Supervised classification and mathematical optimization. *Computers & Operations Research*, 40(1):150–165, 2013. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2012.05.015.

E. Carrizosa, C. Molero-Río, and D. Romero Morales. Mathematical optimization in classification and regression trees. *Top*, 29(1):5–33, 2021.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

P. Cunningham and S. J. Delany. K-nearest neighbour classifiers - a tutorial. *ACM Comput. Surv.*

H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. *Advances in neural information processing systems*, 9, 1996.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

T. Fawcett and F. Provost. Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3):291–316, 1997.

Á. Fernández, J. Bella, and J. R. Dorronsoro. Supervised outlier detection for classification and regression. *Neurocomputing*, 486:77–92, 2022.

R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

Y. M. Goh, C. U. Ubeynarayana, K. L. X. Wong, and B. H. Guo. Factors influencing unsafe behaviors: A supervised learning approach. *Accident Analysis & Prevention*, 118:77–85, 2018.

J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang. A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, 115:356–372, 2019.

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

G.-B. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2011.

A. Jiménez-Cordero, J. M. Morales, and S. Pineda. A novel embedded min-max approach for feature selection in nonlinear support vector machine classification. *European Journal of Operational Research*, 293(1):24–35, 2021.

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

V. M. Patro and M. R. Patra. Augmenting weighted average with confusion matrix to enhance classification accuracy. *Transactions on Machine Learning and Artificial Intelligence*, 2(4):77–91, 2014.

K. Qiu. SVDD-Python. `https://github.com/iqiukp/SVDD-Python`, 2022.

ScikitLearn. SVM en ScikitLearn. `https://scikit-learn.org/stable/modules/svm.html#classification`, 2022a.

ScikitLearn. SVR en ScikitLearn. `https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html#sphx-glr-auto-examples-svm-plot-svm-regression-py`, 2022b.

J. Shawe-Taylor, N. Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

E. J. C. Suárez. Tutorial sobre máquinas de vectores soporte (svm). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*, pages 1–12, 2014.

A. Tewari and P. L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(5), 2007.

G. Valenzuela-González. Supervised_learning_TFG. `https://github.com/gemavg/Supervised_Learning_TFG`, 2022.

V. Vapnik. Statistical learning theory. *John Wiley and Sons*, 1998.

V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.

Q. Wang, Y. Ma, K. Zhao, and Y. Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212, 2022.

Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.