Compresión de Datos para Aprendizaje de Máquina

Bernardo Aurelio Gonzalez Torres

12 de mayo de 2016

Resumen

El aprendizaje de máquina o aprendizaje automático es una rama de las ciencias de la computación cuyo objeto de estudio son el conjunto de métodos y algoritmos que utilizan información o datos para mejorar el desempeño de alguna tarea o realizar predicciones precisas acerca de algún fenómeno. Estos métodos han sido ocupados con éxito en una gran variedad de aplicaciones, como reconocimiento de voz, motores de búsqueda, visión artificial, detección de rostros, diagnóstico médico, detección de fraudes, entre otros.

Uno de los métodos de aprendizaje más utilizados son las Máquinas de Soporte Vectorial (SVM, por sus siglas en inglés). Este método es utilizado principalmente en tareas de clasificación y su popularidad se debe en gran medida a que genera un modelo compacto, además de tener un fuerte sustento teórico y una elevada efectividad en la práctica. Sin embargo, este método tiene la desventaja de que su fase de entrenamiento es computacionalmente costosa, ya que su complejidad temporal es $O(m^3)$ y consume $O(m^2)$ espacio de memoria (siendo m el número de objetos en la muestra de entrenamiento), por lo que aplicarlo a conjuntos de datos grandes es problemático en la práctica.

En este trabajo se propone un método para disminuir el tamaño del conjunto de datos que recibe el método de SVM, con el fin de disminuir su tiempo de entrenamiento. Este método se basa en el algoritmo de aprendizaje de máquina llamado k-medoides (k-medoids en inglés), que es una variante del método de aprendizaje de máquina llamado k-medias (k-means en inglés). El método propuesto incorpora las estadísticas de los grupos formados por el algoritmo de k-medoides en el esquema de las máquinas de soporte vectorial de manera tal que el problema de optimización sigue siendo convexo.

El método propuesto será validado usando conjuntos de datos disponibles públicamente. Al aplicar este nuevo método se espera que el tiempo de entrenamiento de las SVM mejore, procurando que el error de generalización no aumente drásticamente.

Abstract

Machine learning is a branch of computer science that studies methods and algorithms that use information or data to improve performance or to make accurate predictions about some phenomenon. These methods have been successfully deployed in a variety of applications such as speech recognition, search engines, computer vision, face detection, medical diagnosis, fraud detection, among others.

One of the most commonly used learning algorithm is Support Vector Machines (SVM). This algorithm is used mainly for classification tasks and its popularity is due largely because it has a strong theoretical support and high effectiveness in practice. Moreover, the models produced by SVM learning algorithms are very compact, which make them very efficient once they are deployed. However, this algorithm has the disadvantage that its training phase is computationally expensive, consuming about $O(m^3)$ time and $O(m^2)$ memory space, so applying it to large data sets is problematic in practice.

In this work we propose a method to reduce the size of the data set received by the SVM algorithm, in order to reduce the training time. This method is based on the machine learning algorithm called k-medoids, which is a variant of the machine learning algorithm called k-means. The proposed method incorporates the statistics of the clusters formed by the algorithm k-medoids in the SVM scheme so that the optimization problem remains convex.

The proposed method will be validated using public data sets. By applying this new method it is expected that the training time of SVM improves, ensuring that the generalization error does not increase drastically.

Agradecimientos

Me considero una persona afortunada.

Gracias a mi familia, en especial a mis abuelos, a mis tíos y a mis primos. Siempre están allí cuando se les necesita.

Agradezco de manera muy especial a mi amiga Ariana Martínez, por haberme impulsado y motivado a realizar mis estudios de maestría.

Gracias a mi amigo Edgar Cortés por haberme recibido en Catalunya y ayudarme a cumplir un sueño que tenía desde hace aproximadamente siete años.

A mis amigos Gabriela Cortes, Brianda Córdova, Lucía Alonso, Monserrat Villanueva, Angel Alvarez, Israel Arellano e Israel Pérez. Gracias por haberme apoyado durante ese oscuro episodio. Fueron tiempos difíciles.

A la comunidad del CIC. Gracias absolutamente a todos, sobre todo a los que estuvieron en el día a día.

A mis compañeros de generación: Sarahi, Abraham, Alfonso, Asier, Gustavo, Humberto, Iván y Pablo. Gracias por todo.

A los miembros del jurado, sobre todo a los doctores Grigori Sidorov y Ricardo Barrón, gracias por su paciencia y confianza.

Toda mi gratitud para el Dr. Zvi. Entre bastidores lo manipulo todo. Y sin darme cuenta creo que fue él quien de manera sutil trazo la ruta, y cuando el curso peligraba o se desviaba de rumbo, salía a corregirlo. Gracias por su paciencia, confianza y amistad.

A Andrea y a Ángel. Sería impreciso tratar de adivinar cuál sería mi posición de no haber compartido mi tiempo junto a ustedes. Sin su ayuda, difícilmente hubiera logrado terminar esta tesis a tiempo. Gracias por su amistad, porque aún a 11000 km de distancia, no me dejaron solo.

A los hermanos Menchaca, en especial a Ricardo. Jefe, hay ocasiones en las que las palabras no son suficientes para expresar lo que uno quiere. Ésta es una de ellas. Mi gratitud hacia ti y hacia tu hermano es infinita, no sólo como alumno, sino como amigo y ser humano.

Finalmente, agradezco al CONACyT por el apoyo económico brindado durante mis estudios de maestría.

Aunque estos son unos agradecimientos y no una dedicatoria, quiero dedicarle este esfuerzo a mis padres, y muy en especial, a mis hermanos. Es un alivio tenerlos a mi lado.

Índice general

Índice de figuras

Notación

Símbolo	Significado
$\Re$	Conjunto de los números reales
$\Re^N$	Conjunto de los vectores de N dimensiones formados por números reales
$\Re_+^N$	Conjunto de los vectores de N dimensiones formados por números reales no negativos
$\mathbb{N}$	Conjunto de los números naturales
$\{a, b, c\}$	Conjunto formado por los elementos a, b y c
$\{1, 2, ..., m\}$	Conjunto formado por los números naturales $1, 2$, hasta el m
S	Conjunto arbitrario
$s \in S$	El elemento s pertenece al conjunto S
$\|S\|$	Número de elementos en el conjunto S
$\emptyset$	Conjunto vacío
$\bigcup_{i=1}^{k} S_i$	Unión de los conjuntos $S_1, S_2, ..., S_k$
$S_1 \cap S_2$	Intersección de los conjuntos S_1 y S_2
$S \times S$	Conjunto de parejas ordenadas (s_1, s_2) tal que $s_1, s_2 \in S$
$\mathcal{X}$	Espacio de entrada
$\mathcal{Y}$	Espacio objetivo
$b \in \Re$	El número b pertenece al conjunto de los números reales
$\|b\|$	Valor absoluto del número b
$\sum_{i=1}^{m} b_i$	$b_1 + b_2 + ... + b_m$
$\vee$	Disyunción lógica
$\mathbf{w}$	Vector arbitrario
w_i	i-ésimo componente del vector $\mathbf{w}$
$\mathbf{w} \cdot \mathbf{x}$	$\sum_{i=1}^{N} w_i x_i$. Producto interno entre los vectores $\mathbf{w}$ y $\mathbf{x}$
$\|\|\mathbf{w}\|\|$	$\sqrt{\mathbf{w} \cdot \mathbf{w}}$. Norma L_2 de $\mathbf{w}$
$F : S_1 \to S_2$	Función F con dominio S_1 y contradominio S_2
$F : \mathbf{w} \mapsto F(\mathbf{w})$	Función F que recibe un vector $\mathbf{w}$ y retorna $F(\mathbf{w})$
$d(\mathbf{w}, \mathbf{x})$	Distancia euclidiana entre los vectores $\mathbf{w}$ y $\mathbf{x}$
$d_M(\mathbf{x}, \mu)$	Distancia de Mahalanobis entre los vectores $\mathbf{x}$ y μ
$sgn(b)$	Función signo. Esta función devuelve $+1$ si $b \geq 0$ y -1 si $b < 0$
$\min_{\mathbf{x} \in S} f(\mathbf{x})$	Mínimo valor de la función $f(\mathbf{x})$ tal que el elemento $\mathbf{x}$ pertenezca al conjunto S
$\max_{\mathbf{x} \in S} f(\mathbf{x})$	Máximo valor de la función $f(\mathbf{x})$ tal que el elemento $\mathbf{x}$ pertenezca al conjunto S
$\operatorname{argmin}_{\mathbf{x} \in S} f(\mathbf{x})$	Elemento $\mathbf{x}$ que minimiza la función $f(\mathbf{x})$ tal que $\mathbf{x}$ pertenece al conjunto S
$\operatorname{argmax}_{\mathbf{x} \in S} f(\mathbf{x})$	Elemento $\mathbf{x}$ que maximiza la función $f(\mathbf{x})$ tal que $\mathbf{x}$ pertenece al conjunto S

Símbolo	Significado
s.t.	Sujeto a (del inglés subject to). Indicación que precede a las funciones de restricción de un problema de optimización
$\mathcal{L}$	Lagrangiano de un problema de optimización
$\mathbf{A}$	Matriz arbitraria
$\mathbf{A}^T$	Transpuesta de la matriz $\mathbf{A}$
$\mathbf{A} \succ 0$	La matriz $\mathbf{A}$ es definida positiva ($\mathbf{z}^T\mathbf{A}\mathbf{z} > 0$ para todo vector $\mathbf{z} \in \Re^N$, suponiendo que $\mathbf{A}$ es de tamaño $N \times N$)
$\mathbf{I}$	Matriz identidad
Σ	Matriz de covarianza
$\nabla_{\mathbf{w}} F(\mathbf{w})$	Gradiente de la función $F(\mathbf{w})$ con respecto a $\mathbf{w}$
$\nabla^2 F(\mathbf{w})$	Matriz hessiana de la función $F(\mathbf{w})$

Capítulo 1

Introducción

A pesar de que no existe consenso en la definición de aprendizaje de máquina, varias definiciones se han propuesto. En palabras de Murphy:

"We define machine learning as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty". [1]

"Definimos al aprendizaje de máquina como un conjunto de métodos que pueden detectar patrones en datos de manera automática, para luego utilizar los patrones descubiertos para predecir datos futuros o ejecutar algún tipo de toma de decisión bajo incertidumbre".

Mohri, Rostamizadeh y Talwalkar definen al aprendizaje de máquina de la siguiente manera:

"Machine learning can be broadly defined as computational methods using experience to improve performance or to make accurate predictions". [2]

"Aprendizaje de máquina puede definirse de manera general como métodos computacionales que utilizan la experiencia para mejorar el rendimiento o para realizar predicciones precisas".

Otra definición por Shalev-Shwartz y Ben-David es la siguiente:

"The term machine learning refers to the automated detection of meaningful patterns in data". [3]

"El término aprendizaje de máquina se refiere a la detección automática de patrones significativos en datos".

El término común que se puede observar en las definiciones presentadas es que el aprendizaje de máquina trata acerca del desarrollo de métodos y algoritmos que extraen conocimiento e información a partir de datos de forma automática, con el fin de descubrir estructuras o patrones subyacentes y utilizar éstos para construir un modelo general y preciso capaz de realizar predicciones en datos no observados con anterioridad.

1.1. Definiciones y terminología

A continuación se presentan algunas definiciones y categorías dentro del aprendizaje de máquina, prestando especial énfasis al aprendizaje supervisado y no supervisado, debido a la importancia de éstos en el presente trabajo.

- Objeto: También llamado ejemplo, instancia o patrón. Regularmente denotado como $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iN})$. Se refiere a un elemento de un conjunto de datos que puede ser utilizado para entrenamiento o evaluación. Se denotará al conjunto de todos los posibles objetos o instancias como $\mathcal{X}$. $\mathcal{X}$ también es llamado en ocasiones como el espacio de entrada.
- Conjunto de datos (Data set): Colección de datos usualmente representados en una tabla o una matriz, donde cada fila corresponde a un objeto o instancia y cada columna un atributo o característica.
- Atributos: También conocidos como características, son representadas usualmente como un vector asociado a un objeto. Comúnmente son mediciones tomadas del mundo real asociadas al objeto que las representa. Un atributo puede ser numérico, categórico (por ejemplo un color) o incluso un valor lógico (verdadero o falso). Dado un objeto $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iN})$, el valor de cada $x_{i1}, x_{i2}, ..., x_{iN}$ es un atributo del objeto $\mathbf{x}_i$.
- Dimensión: La dimensión de un objeto es el número de atributos asociados a éste. Dado un objeto $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iN})$, la dimensión del objeto $\mathbf{x}_i$ es N.
- Etiqueta: Valor o categoría asignada a un objeto. Regularmente denotado como y_i, asociado al objeto $\mathbf{x}_i$. Al conjunto de todas los posibles etiquetas se le denota como $\mathcal{Y}$ y también es llamado espacio objetivo.
- Muestra de entrenamiento: También llamado conjunto de entrenamiento, son los objetos utilizados para entrenar un método de aprendizaje.
- Muestra de validación: También llamado conjunto de validación, son los objetos utilizados para ajustar los parámetros de un método de aprendizaje. Los métodos de aprendizaje usualmente tienen uno o más parámetros, y la muestra de validación es utilizada para seleccionar valores apropiados para éstos parámetros.
- Muestra de prueba: También llamado conjunto de prueba, son los objetos utilizados para evaluar el desempeño de un método de aprendizaje. Ésta muestra regularmente no es accesible al método de aprendizaje durante la etapa de entrenamiento. Consiste en una colección de objetos para los que el método de aprendizaje ya entrenado debe predecir sus etiquetas basado en sus atributos. Estas predicciones luego son comparadas con las etiquetas de cada objeto de la muestra de prueba para medir el desempeño del método de aprendizaje.

Dependiendo del tipo de datos disponible, así como del orden y la manera en que estos datos ingresen al método de aprendizaje correspondiente, los métodos utilizados en aprendizaje de máquina se pueden clasificar dentro de las siguientes categorías: [2]

- Aprendizaje supervisado (Supervised learning)
- Aprendizaje no supervisado (Unsupervised learning)
- Aprendizaje semi-supervisado (Semi-supervised learning)

- Aprendizaje en línea (On-line learning)
- Aprendizaje activo (Active learning)

Estas categorías solamente son algunas de las más comunes y la lista no pretende abarcar todas las categorías posibles, sino introducir dos de las categorías más importantes: Aprendizaje supervisado y aprendizaje no supervisado. Para esta tesis en particular es importante definir cada una de ellas ya que el método de SVM pertenece a la categoría de aprendizaje supervisado y el método de k-medoides pertenece a la categoría de aprendizaje no supervisado.

Los métodos de aprendizaje supervisado se caracterizan por recibir una muestra de entrenamiento donde los objetos tienen una etiqueta asociada a ellos. El objetivo principal de este tipo de métodos es crear un modelo a partir de los datos de entrenamiento que pueda ser utilizado para predecir la etiqueta de nuevos objetos (diferentes a los del conjunto de entrenamiento), es decir, el método debe de ser capaz de generalizar. Algunos de los problemas mas comunes en este tipo de escenario son problemas de clasificación, donde se utiliza la muestra de entrenamiento para construir un modelo capaz de predecir la etiqueta de un nuevo objeto, que sólo puede ser categórica como por ejemplo A,B,C, etc. ó 1,2,3, etc. En problemas de clasificación, a la etiqueta también se le conoce como la clase del objeto. Una alta precisión al clasificar correctamente; comprensibilidad, que se refiere a la capacidad de un humano para entender el modelo de clasificación generado; y el generar un modelo compacto, son los objetivos principales para la tarea de clasificación [5].

Los métodos de aprendizaje no supervisado se caracterizan por recibir una muestra de entrenamiento donde los objetos no tienen una etiqueta asociada a ellos [25] [28]. Este tipo de métodos tiene como objetivo encontrar la estructura subyacente de los datos que reciben como entrada. Algunos de los problemas más comunes de este tipo son problemas de agrupamiento (clustering). Este tipo de problemas son parecidos a los problemas de clasificación, ya que asignan a cada objeto a un grupo (cluster) en particular, con la gran diferencia de que en problemas de clasificación se cuenta con la información de a que clase pertenece cada objeto de la muestra de entrenamiento, mientras que en problemas de agrupamiento se debe descubrir a que grupo pertenece cada objeto del conjunto de entrada, basándose normalmente en medidas de similitud o diferencia entre cada uno de ellos.

Las máquinas de soporte vectorial son un método de clasificación binaria que utilizan como modelo un hiperplano de separación óptima para clasificar. Los objetos del conjunto de datos de entrenamiento que determinan el hiperplano de separación óptima se denominan vectores de soporte (SV, por sus siglas en inglés) y se obtienen solucionando un problema de optimización de programación cuadrática (QPP, por sus siglas en inglés) [6].

1.2. Motivación

La información ha pasado de ser escasa a superabundante. Cuando el proyecto Sloan Digital Sky Survey comenzó a trabajar en el año 2000, su telescopio recolectó mas datos en sus primeras semanas de trabajo que todos los que habían sido recolectados en la historia de la astronomía. En 2010, su archivo contenía aproximadamente 140 terabytes de información. El Gran Telescopio para Sondeos Sinópticos en Chile recolectará la misma cantidad de datos cada cinco días. Wal-Mart maneja mas de un millón de transacciones por hora, alimentado a sus bases de datos, que se estiman en mas de 2.5 petabytes. La red social Facebook contiene mas de 40 mil millones de fotos [4].

El uso de sensores electrónicos en sistemas de seguridad, automóviles, sistemas de control industrial y la inclusión de sistemas como teléfonos inteligentes, agendas personales, sistemas de posicionamiento global, computadoras, etc. en nuestra vida diaria dieron como resultado una generación masiva de información. De acuerdo a un estudio presentado por la firma IDC en junio de 2011 [7] la cantidad de datos generados se estima superior a 1,8 zettabytes.

Todos los ejemplos mencionados indican que se cuenta con una cantidad enorme de datos digitales que crece cada día mas rápido. Estos datos nos brindan la oportunidad de encontrar tendencias en los negocios, prevenir enfermedades, combatir el crimen, entre algunas otras. Para que esto sea posible, es necesario encontrar y utilizar métodos que generen conocimiento o información útil a partir de éstos datos.

Las máquinas de soporte vectorial han sido utilizadas con éxito en muchas aplicaciones, tales como visión artificial [8], detección de rostros [9], calificación de crédito [10], bioinformática [11] y el filtrado de spam [12], entre otros. El modelo producido por las máquinas de soporte vectorial es compacto, geométricamente interpretable y su rendimiento por lo general supera la precisión de clasificación de otros métodos. A pesar de éstas características, las máquinas de soporte vectorial tienen un problema notorio: la fase de entrenamiento consume $O(m^3)$ tiempo y $O(m^2)$ espacio de memoria [13] [14] [39] (siendo m el número de objetos de la muestra de entrenamiento). Esto vuelve problemático el uso de las máquinas de soporte vectorial con conjuntos de datos de gran tamaño.

En resumen, desarrollar un método que permita entrenar a las máquinas de soporte vectorial de manera más rápida pero sin afectar de manera drástica su error de generalización ha sido lo que ha motivado este trabajo.

1.3. Hipótesis

Es posible entrenar a las máquinas de soporte vectorial sin utilizar una muestra de entrenamiento por completo, utilizando un método de aprendizaje no supervisado que procese de manera anticipada la muestra de entrenamiento, sin afectar de gran manera el error de generalización del modelo entregado.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un método de reducción de datos para mejorar el tiempo de entrenamiento de las máquinas de soporte vectorial. Este método permitirá utilizar las máquinas de soporte vectorial en conjuntos de datos grandes sin tener una pérdida apreciable en la precisión de la clasificación. El método por desarrollar tendrá como base un método de aprendizaje de máquina no supervisado.

1.4.2. Objetivos particulares

- Desarrollar un esquema que integre el número de grupos entregados por el método de agrupamiento, la fracción de objetos asignados a cada grupo y la varianza de cada grupo dentro del planteamiento de máquinas de soporte vectorial.
- Implementar el método desarrollado.
- Probar la eficiencia del método desarrollado utilizando conjuntos de datos públicos disponibles.

1.5. Contribuciones

Las principales contribuciones de esta tesis son las siguientes:

- Se propone un esquema de reducción de datos para acelerar el entrenamiento de las SVM.
- Se presenta una generalización del método de SVM utilizando la distancia de Mahalanobis. La distancia euclidiana corresponde a un caso donde la matriz de covarianza es igual a la matriz identidad.

1.6. Organización del documento

El resto de esta tesis esta organizada de la siguiente manera:

Capítulo 2. Marco teórico: máquinas de soporte vectorial. En este capítulo se presenta el método de máquinas de soporte vectorial para clasificación binaria. Se inicia presentado el caso separable, para después introducir el caso no separable y finalmente presentar el método de máquinas de soporte vectorial con ponderación. Se describe la formulación matemática del problema y su relación con problemas de optimización convexa.

Capítulo 3. Marco teórico: k-medoides. En este capítulo se presentan los método de agrupamiento k-medias y k-medoides. El capítulo comienza presentando el planteamiento general de un problema de agrupamiento. Después, se presenta el algoritmo de k-medias junto con algunas de sus limitaciones e inconvenientes. El capítulo termina presentado el algoritmo de k-medoides.

Capítulo 4. Trabajos relacionados. En este capítulo se describen algunas propuestas presentadas por otros autores para disminuir el tiempo de entrenamiento de las máquinas de soporte vectorial, cubriendo con mayor detalle los métodos de reducción de datos, ya que el método propuesto en esta tesis cae en esta categoría.

Capítulo 5. Propuesta. Este capítulo presenta el método propuesto, describiendo el algoritmo y explicando la manera en que se integrará la información del método de agrupamiento dentro del esquema de SVM.

Capítulo 6. Resultados experimentales. En este capítulo se describe la forma en que se realizó la experimentación y evaluación del método propuesto, se presentan los resultados obtenidos y la

comparación con otros métodos.

Capítulo 7. Conclusiones y trabajo futuro. Este capítulo presenta las conclusiones obtenidas con base en los estudios realizados y los resultados obtenidos. Finalmente, se presenta una posible línea de trabajo a seguir en el futuro, como continuación de esta tesis.

Capítulo 2

Marco teórico: máquinas de soporte vectorial

El método de máquinas de soporte vectorial fue presentado por Vapnik y sus colegas [6] [18] para resolver problemas de clasificación y regresión. En este capítulo se presentan los fundamentos teóricos de las máquinas de soporte vectorial para clasificación binaria, que es cuando el espacio objetivo $\mathcal{Y}$ se limita a sólo dos etiquetas $\mathcal{Y} = \{-1, +1\}$. Se comienza con el caso separable, que es donde se asegura de antemano que existe un hiperplano de separación que divide a todos los objetos de ambas clases. Después, se presenta el caso no separable, en donde no existe un hiperplano de separación que logré clasificar de manera correcta a todos los objetos en la muestra de entrenamiento. El desarrollo de éstos dos casos se basa en el desarrollo presentado en [2]. Finalmente, se presenta el método de máquinas de soporte vectorial con ponderación, que es una modificación del método original donde a cada objeto se le asigna un peso que denota la importancia que tiene ese objeto en la tarea de clasificación.

2.1. Máquinas de soporte vectorial. Caso separable

Se considera un espacio de entrada $\mathcal{X} \subseteq \Re^N$ donde $\Re$ denota al conjunto de los números reales y $N \geq 1$, un espacio objetivo $\mathcal{Y} = \{-1, +1\}$ y una muestra de entrenamiento S de tamaño m que es linealmente separable, es decir, existe un hiperplano que separa a todos los objetos con etiqueta $y_i = -1$ de todos los objetos con etiqueta $y = +1$, como se muestra en la imagen izquierda de la figura 2.1, tomada de [2]. La existencia del hiperplano de separación esta garantizada por el teorema del hiperplano de separación (un caso especial de la versión finito dimensional del teorema de separación de Hahn-Banach para espacios vectoriales topológicos):

Teorema 1. *Sean A y B dos subconjuntos convexos y no vacíos de $\Re^N$. Si $A \cap B = \emptyset$, entonces A y B pueden ser separados por un hiperplano.*

La idea clave del método de máquinas de soporte vectorial es la de retornar, de entre el número infinito de hiperplanos de separación, el hiperplano con el máximo margen o distancia a los objetos más cercanos de cada clase (ver figura 2.1). Como este es un problema de aprendizaje supervisado,

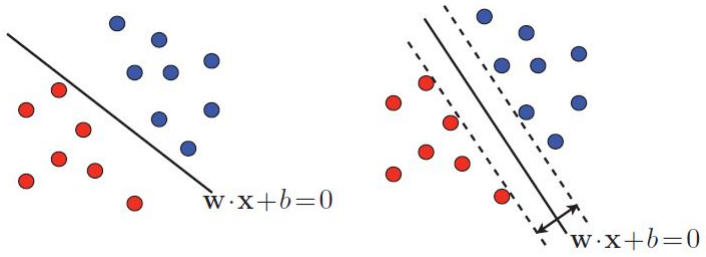

Figura 2.1: Dos posibles hiperplanos de separación. La figura del lado derecho muestra un hiperplano que maximiza el margen de separación

la muestra de entrenamiento S consta de parejas $(\mathbf{x}_i, y_i), i \in \{1, 2, ..., m\}$, donde $\mathbf{x}_i$ es un objeto y y_i es la etiqueta o clase correspondiente a ese objeto.

2.1.1. Problema de optimización primal

En esta parte se desarrolla el problema de clasificación binaria hasta llegar al problema de optimización relacionado a las máquinas de soporte vectorial. La ecuación general de un hiperplano en $\Re^N$ es

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{2.1}$$

donde $b \in \Re$ y $\mathbf{w} \in \Re^N$ es un vector normal al hiperplano. Es importante observar que esta definición es invariante a la multiplicación por un escalar $k \neq 0$:

$$k\ [\mathbf{w} \cdot \mathbf{x} + b = 0] \quad \Leftrightarrow \quad k\mathbf{w} \cdot \mathbf{x} + kb = 0 \quad \Leftrightarrow \quad \mathbf{w}' \cdot \mathbf{x} + b' = 0$$

donde $\mathbf{w}' \cdot \mathbf{x} + b' = 0$ describe al mismo hiperplano que $\mathbf{w} \cdot \mathbf{x} + b = 0$, sólo que representado con diferentes parámetros $\mathbf{w}' = k\mathbf{w}$ y $b' = kb$. Por lo tanto, para un hiperplano en el caso separable, se pueden escalar $\mathbf{w}$ y b de tal manera que $\min\limits_{(\mathbf{x},y)\in S} |\mathbf{w}\cdot\mathbf{x}+b| = 1$, es decir, se pueden escalar de tal manera que los hiperplanos con ecuación $\mathbf{w}\cdot\mathbf{x}+b = +1$ y $\mathbf{w}\cdot\mathbf{x}+b = -1$, llamados hiperplanos marginales, pasen por los objetos de cada clase más cercanos al hiperplano de separación $\mathbf{w} \cdot \mathbf{x} + b = 0$, como se muestra en la figura 2.2, tomada de [2].

Definiremos a la representación del hiperplano de separación que cumple con $\min\limits_{(\mathbf{x},y)\in S} |\mathbf{w} \cdot \mathbf{x} + b| = 1$ como el hiperplano canónico, definido por la pareja $(\mathbf{w}, b)$. La distancia de cualquier objeto $\mathbf{x}_0 \in \Re^N$ a un hiperplano definido por la ecuación (2.1) se puede calcular con la siguiente expresión:

$$\frac{|\mathbf{w} \cdot \mathbf{x}_0 + b|}{||\mathbf{w}||} \tag{2.2}$$

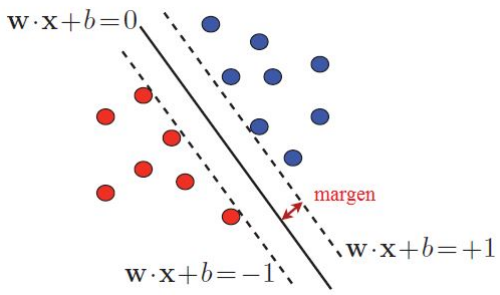

Figura 2.2: Hiperplano de separación e hiperplanos marginales (en líneas punteadas) con sus respectivas ecuaciones características.

Por lo tanto, para un hiperplano canónico, el margen, que es la distancia del hiperplano de separación a los objetos mas cercanos y se denota por la letra griega ρ, se puede calcular con la siguiente expresión

$$\rho = \min_{(\mathbf{x},y)\in S} \frac{|\mathbf{w}\cdot\mathbf{x}+b|}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||} \tag{2.3}$$

Un hiperplano definido por la pareja $(\mathbf{w}, b)$ clasifica de manera correcta un objeto de entrenamiento $\mathbf{x}_i$, $i \in \{1, 2, ..., m\}$ cuando el valor de $\mathbf{w}\cdot\mathbf{x}_i+b$ tiene el mismo signo que la correspondiente etiqueta y_i. Para un hiperplano canónico, por definición, se tiene que $|\mathbf{w}\cdot\mathbf{x}_i+b| \geq 1 \ \forall \ i \in \{1, 2, ..., m\}$, por lo tanto, el objeto $\mathbf{x}_i$ es correctamente clasificado cuando $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) \geq 1$. Como se dijo al principio de la sección, las máquinas de soporte vectorial pretenden maximizar el margen $\rho = \frac{1}{||\mathbf{w}||}$. Se puede observar que maximizar ρ es equivalente a minimizar $||\mathbf{w}||$ o $\frac{1}{2}||\mathbf{w}||^2$. Entonces, en el caso separable, la solución entregada por las máquinas de soporte vectorial, que es un hiperplano que maximiza el margen ρ y que clasifica de manera correcta todos los objetos de entrenamiento, puede ser expresada como la solución al siguiente problema de optimización convexa:

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\ &\text{s. t. } y_i(\mathbf{w}\cdot\mathbf{x}_i+b) \geq 1, \ \forall \ i \in \{1, 2, ..., m\} \end{aligned} \tag{2.4}$$

Al problema de optimización (2.4) se le conoce como problema de optimización primal de las máquinas de soporte vectorial. La función objetivo $F : \mathbf{w} \mapsto \frac{1}{2}||\mathbf{w}||^2$ es infinitamente diferenciable. Su gradiente es $\nabla_{\mathbf{w}} F(\mathbf{w}) = \mathbf{w}$ y su matriz hessiana es la matriz identidad $\nabla^2 F(\mathbf{w}) = \mathbf{I}$. Por lo tanto, $\nabla^2 F(\mathbf{w}) \succ 0$, lo que implica que F es una función estrictamente convexa. Además, las restricciones del problema de optimización son todas funciones afines $g_i : (\mathbf{w}, b) \mapsto 1 - y_i(\mathbf{w}\cdot\mathbf{x}_i+b)$. Por lo tanto, el problema de optimización (2.4) tiene una solución única $(\mathbf{w}^*, b^*)$, una propiedad importante y favorable que no se cumple para todos los métodos de aprendizaje. Además, dado que la función objetivo es cuadrática y las restricciones afines, el problema de optimización (2.4) es una instancia específica de programación cuadrática (QP, por sus siglas en inglés), una familia de problemas ampliamente estudiado en el campo de optimización [15].

2.1.2. Vectores de soporte

Cualquier problema de optimización con función objetivo convexa y diferenciable, y funciones de restricción afines y diferenciables, satisface la condición de que un punto es una solución al problema de optimización si y solo si cumple con las condiciones de Karush-Kuhn-Tucker (KKT) [16] [17]. Para el problema de máquinas de soporte vectorial, las condiciones de Karush-Kuhn-Tucker se obtienen igualando a cero el gradiente del Lagrangiano con respecto a las variables $\mathbf{w}$ y b, y la otra condición es conocida como la condición de holgura complementaria (complementary slackness) [15]:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{2.5}$$

$$\nabla_b \mathcal{L} = -\sum_{i=1}^{m} \alpha_i y_i = 0 \qquad \Longrightarrow \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{2.6}$$

$$\forall i, \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1] = 0 \qquad \Longrightarrow \qquad \alpha_i = 0 \vee y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1 \tag{2.7}$$

donde la expresión del Lagrangiano es la siguiente:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1] \tag{2.8}$$

El vector $\boldsymbol{\alpha}$ es un vector compuesto de m multiplicadores de Lagrange $\alpha_i \geq 0, i \in \{1, 2, ..., m\}$, por lo que $\boldsymbol{\alpha} \in \Re_+^m$.

Se puede observar de la ecuación (2.5) que el vector $\mathbf{w}$ que retornan las máquinas de soporte vectorial es una combinación lineal de los objetos $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ de la muestra de entrenamiento S. Un objeto $\mathbf{x}_i$ aparecerá en esta combinación lineal si y sólo si $\alpha_i \neq 0$. Estos objetos son llamados vectores de soporte. También se puede observar que debido a las condiciones de holgura complementaria (2.7), si $\alpha_i \neq 0$, entonces $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$. Por lo tanto, los vectores de soporte se encuentran en los hiperplanos marginales $\mathbf{w}\cdot\mathbf{x}_i + b = \pm 1$.

Los vectores de soporte definen completamente al hiperplano de separación de máximo margen que resuelve el problema de las máquinas de soporte vectorial, lo que justifica el nombre del algoritmo. Se puede observar que los objetos que no yacen en los hiperplanos marginales no afectan a la solución que retornan las máquinas de soporte vectorial, es decir, en su ausencia, la solución al problema de SVM se mantendría sin cambios. Es importante tener en cuenta que aunque que el vector $\mathbf{w}$ que retornan las SVM después de su entrenamiento es único, los vectores de soporte pueden no serlo, ya que en una dimensión de tamaño N, $N+1$ puntos son suficientes para definir un hiperplano. Por lo tanto, cuando hay mas de $N+1$ objetos en los hiperplanos marginales, se pueden formar diferentes configuraciones para la selección de los $N+1$ vectores de soporte necesarios para definir el hiperplano de separación.

2.1.3. Problema de optimización dual

Aplicando la propiedad distributiva del producto punto y de la multiplicación en la ecuación del Lagrangiano se obtiene lo siguiente:

$$\begin{aligned}\mathcal{L} &= \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1] \\ &= \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m}\alpha_i y_i(\mathbf{w}\cdot\mathbf{x}_i) - \sum_{i=1}^{m}\alpha_i y_i b + \sum_{i=1}^{m}\alpha_i \\ &= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i - b\sum_{i=1}^{m}\alpha_i y_i + \sum_{i=1}^{m}\alpha_i \end{aligned} \tag{2.9}$$

Al sustituir las ecuaciones (2.5) y (2.6) en la ecuación (2.9) se obtiene lo siguiente:

$$\begin{aligned} &= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\mathbf{w} + \sum_{i=1}^{m}\alpha_i \\ &= \frac{1}{2}||\mathbf{w}||^2 - ||\mathbf{w}||^2 + \sum_{i=1}^{m}\alpha_i \\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\mathbf{w}||^2 \\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i||^2 \\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j) \end{aligned} \tag{2.10}$$

La expresión (2.10) es conocida como la función dual del problema de máquinas de soporte vectorial $g(\boldsymbol{\alpha})$, $g : \Re^m \to \Re$. Una de las más importantes propiedades de la función dual de cualquier problema de optimización de minimización (convexo o no convexo) es que provee una cota inferior al valor óptimo del problema de optimización primal en cuestión, es decir, para cualquier $\boldsymbol{\alpha}$ tal que $\alpha_i \geq 0 \ \forall \ i \in \{1, 2, ..., m\}$ se cumple que

$$g(\boldsymbol{\alpha}) \leq p^* \tag{2.11}$$

donde p^* denota el valor óptimo del problema de optimización primal en cuestión, es decir, el mínimo valor que la función objetivo puede tomar y que además cumpla con las restricciones de optimización.

Al problema de maximizar la función dual $g(\boldsymbol{\alpha})$ con las restricciones $\alpha_i \geq 0 \ \forall \ i \in \{1, 2, ..., m\}$ y la restricción impuesta por la ecuación (2.6), se le conoce como el problema de optimización dual para las máquinas de soporte vectorial en su caso separable:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{s. t. } \sum_{i=1}^{m} \alpha_i y_i = 0,\ \alpha_i \geq 0\ \forall\ i \in \{1, 2, ..., m\} \end{aligned} \tag{2.12}$$

Las soluciones encontradas $\mathbf{w}^*$ y $\boldsymbol{\alpha}^*$ a los problemas de optimización primal y dual de las máquinas de soporte vectorial cumplen con la ecuación (2.5):

$$\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$$

Como los vectores de soporte yacen en los hiperplanos marginales, para cualquier vector de soporte $\mathbf{x}_i$ se cumple que $\mathbf{w}^* \cdot \mathbf{x}_i + b^* = y_i$, por lo que b^* puede calcularse con la siguiente ecuación:

$$b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i = y_i - (\sum_{j=1}^{m} \alpha_j^* y_j \mathbf{x}_j) \cdot \mathbf{x}_i = y_i - \sum_{j=1}^{m} \alpha_j^* y_j (\mathbf{x}_j \cdot \mathbf{x}_i) \tag{2.13}$$

Ya que se obtienen los valores de $\boldsymbol{\alpha}^*$, $\mathbf{w}^*$ y b^*, el modelo de clasificación que retornan las máquinas de soporte vectorial (también llamado hiperplano de separación óptima) es la función:

$$h(\mathbf{x}) = sgn(\mathbf{w}^* \cdot \mathbf{x} + b^*) = sgn(\sum_{i=1}^{m} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*) \tag{2.14}$$

donde $\mathbf{x}$ es un nuevo objeto por clasificar y la función $sgn : \Re \to \{+1, -1\}$ es la función signo, que devuelve $sgn(k) = +1\ \forall\ k \geq 0$ y $sgn(k) = -1\ \forall\ k < 0$.

2.2. Máquinas de soporte vectorial. Caso no separable

En la mayoría de los casos prácticos, la muestra de entrenamiento S recibida por las máquinas de soporte vectorial no es linealmente separable, es decir, para cualquier hiperplano con ecuación $\mathbf{w} \cdot \mathbf{x} + b = 0$, existe al menos un objeto $\mathbf{x}_i \in S$ tal que:

$$y_i[\mathbf{w} \cdot \mathbf{x}_i + b] \ngeq 1 \tag{2.15}$$

Esta condición es problemática ya que no existiría una solución que cumpla con las restricciones impuestas en el problema de optimización primal (2.4). Sin embargo, éstas restricciones pueden modificarse de tal manera que el problema de optimización primal aún pueda encontrar una solución. Para realizar esta modificación, se introducen variables de holgura (slack variables) $\xi_i \geq 0\ \forall\ i \in \{1, 2, ..., m\}$ tal que:

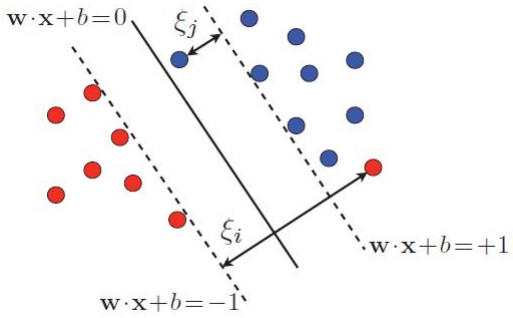

Figura 2.3: Hiperplano de separación que clasifica incorrectamente al objeto $\mathbf{x}_i$ y clasifica correctamente al objeto $\mathbf{x}_j$, aunque con un margen menor a 1.

$$y_i[\mathbf{w} \cdot \mathbf{x}_i + b] \geq 1 - \xi_i \tag{2.16}$$

La variable de holgura ξ_i mide la distancia por la que el objeto $\mathbf{x}_i$ sobrepasa la desigualdad deseada $y_i[\mathbf{w} \cdot \mathbf{x}_i + b] \geq 1$. La figura 2.3, tomada de [2], muestra esta situación. Dado un hiperplano de separación definido por la ecuación $\mathbf{w} \cdot \mathbf{x} + b = 0$, a un objeto $\mathbf{x}_i$ que tenga una variable de holgura $\xi_i > 0$ lo denominaremos anomalía (outlier en inglés). Cada objeto $\mathbf{x}_i$ debe de encontrarse en el lado correcto del espacio dividido por el correspondiente hiperplano marginal $\mathbf{w} \cdot \mathbf{x}_i + b = y_i$ para no ser considerado una anomalía. En las siguientes secciones, se presentan los problemas primal y dual en el caso no separable, y se caracterizarán los vectores de soporte para este caso.

2.2.1. Problema de optimización primal

La forma del problema de optimización primal en el caso no separable es simplemente una modificación del caso separable introduciendo en el problema las variables de holgura ξ_i. Se introduce la variable de penalización $C \geq 0$, que determina la manera en que interactuan la minimización del término $||\mathbf{w}||^2$ y la minimización de las variables de holgura $\sum_{i=1}^{m} \xi_i^p$ en el problema de optimización.

$$\begin{gathered} \min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} \xi_i^p \\ \text{s. t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall\ i \in \{1, 2, ..., m\} \end{gathered} \tag{2.17}$$

donde el vector de holgura $\boldsymbol{\xi}$ es un vector compuesto de m variables de holgura $\xi_i \geq 0, i \in \{1, 2, ..., m\}$, por lo que $\boldsymbol{\xi} \in \Re_+^m$.

La idea de incluir el término $\sum_{i=1}^{m} \xi_i^p$ en el problema de optimización (2.17) es tratar de minimizar la cantidad total de holgura provocada por los objetos que son anomalías. Como en el caso separable, el problema (2.17) es un problema de optimización convexa ya que las funciones de restricción son afines y la función objetivo es convexa para cualquier $p \geq 1$, por lo que, también presenta la

propiedad de tener una solución única $(\mathbf{w}^*, b^*)$. Los valores típicos utilizados para p son $p = 1$ o $p = 2$. El análisis presentado en las siguientes secciones se realiza con el valor de $p = 1$, que es el más comunmente utilizado.

2.2.2. Vectores de soporte

Al igual que en el caso separable, el problema de optimización en el caso no separable presenta una función objetivo convexa y diferenciable, y funciones de restricción afines y diferenciables, por lo que se pueden utilizar las condiciones de Karush-Kuhn-Tucker (KKT) para resolverlo. Se presentan las variables o multiplicadores de Lagrange $\alpha_i \geq 0, i \in \{1, 2, ..., m\}$ asociados a las primeras m restricciones, con su respectivo vector $\boldsymbol{\alpha} \in \Re_+^m$ y los multiplicadores $\beta_i \geq 0, i \in \{1, 2, ..., m\}$ asociados a las restricciones de no negatividad de las variables de holgura, con su respectivo vector $\boldsymbol{\beta} \in \Re_+^m$. El Lagrangiano queda entonces expresado de la siguiente manera:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i \tag{2.18}$$

Como en el caso separable, las condiciones de Karush-Kuhn-Tucker se obtienen igualando a cero el gradiente del Lagrangiano con respecto a las variables $\mathbf{w}$, b y $\xi_i \; \forall \; i \in \{1, 2, ..., m\}$ junto con las condiciones de holgura complementaria:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i \tag{2.19}$$

$$\nabla_b\mathcal{L} = -\sum_{i=1}^{m}\alpha_i y_i = 0 \qquad \Longrightarrow \qquad \sum_{i=1}^{m}\alpha_i y_i = 0 \tag{2.20}$$

$$\forall i, \nabla_{\xi_i}\mathcal{L} = C - \alpha_i - \beta_i = 0 \qquad \Longrightarrow \qquad C = \alpha_i + \beta_i \tag{2.21}$$

$$\forall i, \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] = 0 \qquad \Longrightarrow \qquad \alpha_i = 0 \vee y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1 - \xi_i \tag{2.22}$$

$$\forall i, \beta_i\xi_i = 0 \qquad \Longrightarrow \qquad \beta_i = 0 \vee \xi_i = 0 \tag{2.23}$$

De manera idéntica al caso separable, se observa en la ecuación (2.19) que el vector $\mathbf{w}$ que retornan las máquinas de soporte vectorial es una combinación lineal de los objetos $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ de la muestra de entrenamiento S. Un objeto $\mathbf{x}_i$ aparecerá en esta combinación lineal si y sólo si $\alpha_i \neq 0$. Estos objetos son llamados vectores de soporte. La diferencia con respecto al caso separable es que en este caso existen dos tipos de vectores de soporte. Debido a las condiciones de holgura complementaria (2.22), si $\alpha_i \neq 0$, entonces $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1 - \xi_i$. Si $\xi_i = 0$, entonces $y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1$ y el objeto $\mathbf{x}_i$ yace en algún hiperplano marginal, al igual que en el caso separable. De lo contrario, $\xi_i \neq 0$, lo que implica que el objeto $\mathbf{x}_i$ es una anomalía. En este caso, dado que $\xi_i \neq 0$, (2.23) implica que $\beta_i = 0$ y (2.21) implica que $\alpha_i = C$.

Por lo tanto, los vectores de soporte son anomalías (cuando $\alpha_i = C$) o se encuentran en los hiperplanos marginales $\mathbf{w} \cdot \mathbf{x}_i + b = \pm 1$ (cuando $0 < \alpha_i < C$).

Como en el caso separable, se puede observar que los objetos que no son anomalías ni yacen en los hiperplanos marginales no afectan a la solución que retornan las máquinas de soporte vectorial, por lo que, en su ausencia, la solución al problema de SVM se mantendría sin cambios. De manera similar, aunque el vector $\mathbf{w}$ retornado por las SVM es único, los vectores de soporte pueden no serlo.

2.2.3. Problema de optimización dual

Aplicando la propiedad distributiva del producto punto y de la multiplicación en la ecuación del Lagrangiano se obtiene lo siguiente:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i \\
&= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i y_i(\mathbf{w}\cdot\mathbf{x}_i) - \sum_{i=1}^{m}\alpha_i y_i b + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\beta_i\xi_i \\
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\sum_{i=1}^{m}\alpha_i y_i\mathbf{x}_i - b\sum_{i=1}^{m}\alpha_i y_i + \sum_{i=1}^{m}\alpha_i + \sum_{i=1}^{m}(C - \alpha_i - \beta_i)\xi_i
\end{aligned} \tag{2.24}$$

Al sustituir las ecuaciones (2.19),(2.20) y (2.21) en la ecuación (2.24) se obtiene lo siguiente:

$$\begin{aligned}
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\mathbf{w} + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}||\mathbf{w}||^2 - ||\mathbf{w}||^2 + \sum_{i=1}^{m}\alpha_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\mathbf{w}||^2 \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\sum_{i=1}^{m}\alpha_i y_i\mathbf{x}_i||^2 \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)
\end{aligned} \tag{2.25}$$

Se puede observar que la ecuación (2.25) es exactamente igual que la ecuación (2.10) del caso separable. Sin embargo, además de las restricciones del problema de optimización dual del caso separable, en este caso se debe añadir la restricción de los multiplicadores de Lagrange $\beta_i \geq 0$. Dada la ecuación (2.21), esta restricción es equivalente a la restricción $\alpha_i \leq C$. Por lo tanto, el problema de optimización dual en el caso no separable queda definido de la siguiente manera:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ &\text{s. t. } \sum_{i=1}^{m} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C\ \forall\ i \in \{1, 2, ..., m\} \end{aligned} \tag{2.26}$$

Al igual que en el caso separable, las soluciones encontradas $\mathbf{w}^*$ y $\boldsymbol{\alpha}^*$ a los problemas de optimización primal y dual cumplen con la ecuación (2.19):

$$\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$$

De manera similar, b^* puede calcularse ocupando cualquier vector de soporte que se encuentre en alguno de los hiperplanos marginales, es decir, cualquier objeto $\mathbf{x}_i$ con $0 < \alpha_i < C$. Para tales vectores de soporte, se cumple que $\mathbf{w}^* \cdot \mathbf{x}_i + b^* = y_i$, por lo tanto:

$$b^* = y_i - \mathbf{w}^* \cdot \mathbf{x}_i = y_i - \sum_{j=1}^{m} \alpha_j^* y_j (\mathbf{x}_j \cdot \mathbf{x}_i) \tag{2.27}$$

Como en el caso separable, ya que se obtienen los valores de $\boldsymbol{\alpha}^*$, $\mathbf{w}^*$ y b^*, el hiperplano de separación óptima que retornan las máquinas de soporte vectorial para un nuevo objeto $\mathbf{x}$ por clasificar es la función:

$$h(\mathbf{x}) = sgn(\mathbf{w}^* \cdot \mathbf{x} + b^*) = sgn(\sum_{i=1}^{m} \alpha_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*) \tag{2.28}$$

2.3. Máquinas de soporte vectorial con ponderación

Al utilizar las máquinas de soporte vectorial para tareas de clasificación, existen algunos casos en los que ciertos objetos del conjunto de entrenamiento son más importantes que otros. En casos como éstos, sería preferible que el hiperplano de separación clasificara de manera correcta a los objetos con mayor importancia, aún cuando se equivocará al clasificar a otros objetos en el conjunto de entrenamiento. El esquema tradicional de máquinas de soporte vectorial no tiene manera de manejar estos casos, ya que todos los objetos tienen la misma importancia en el planteamiento matemático del problema, así como en el problema de optimización asociado. El esquema de máquinas de soporte vectorial con ponderación [19] (WSVM, por sus siglas en inglés) es una modificación del método original que incorpora un peso $0 \leq \eta_i \leq 1\ \forall\ i \in \{1, 2, ..., m\}$ asociado a cada objeto $\mathbf{x}_i$ del conjunto de entrenamiento S. Este peso denota la importancia de cada objeto en la tarea de clasificación. Las máquinas de soporte vectorial con ponderación han sido utilizadas para análisis de sentimientos en redes sociales [20], diagnóstico de enfermedades [21], y predicción de la localización subcelular de proteínas [22]. En las siguientes secciones, se presentan los problemas primal y dual para las máquinas de soporte vectorial con ponderación, y se caracterizarán los vectores de soporte para este caso. Una modificación de este esquema es utilizada en el trabajo propuesto en esta tesis. Esta modificación integra en el esquema de máquinas de soporte vectorial con ponderación la información estadística encontrada por el algoritmo de agrupación utilizado.

2.3.1. Problema de optimización primal

La modificación que se realiza al esquema tradicional de las máquinas de soporte vectorial en el caso ponderado es introduciendo en el problema de optimización primal los pesos η_i.

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\eta_i\xi_i \\ &\text{s. t. } y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0\ \forall\ i \in \{1,2,...,m\} \end{aligned} \tag{2.29}$$

Denotaremos a $\boldsymbol{\eta}$ como el vector de pesos compuesto de m pesos $0 \leq \eta_i \leq 1, i \in \{1,2,...,m\}$, por lo que $\boldsymbol{\eta} \in \Re_+^m$.

Como en los demás casos, el problema (2.29) es un problema de optimización convexa ya que las funciones de restricción son afines y la función objetivo es convexa, por lo que, presenta la propiedad de tener una solución única $(\mathbf{w}^*, b^*)$.

2.3.2. Vectores de soporte

Como en los dos casos anteriores, el problema de optimización en el caso ponderado presenta una función objetivo convexa y diferenciable, y funciones de restricción afines y diferenciables, por lo que se pueden utilizar las condiciones de Karush-Kuhn-Tucker (KKT) para resolverlo. Siendo los multiplicadores de Lagrange $\alpha_i \geq 0, i \in \{1,2,...,m\}$ asociados a las primeras m restricciones, con su respectivo vector $\boldsymbol{\alpha} \in \Re_+^m$ y los multiplicadores $\beta_i \geq 0, i \in \{1,2,...,m\}$ asociados a las restricciones de no negatividad de las variables de holgura, con su respectivo vector $\boldsymbol{\beta} \in \Re_+^m$, el Lagrangiano queda expresado de la siguiente manera:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\eta_i\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i \tag{2.30}$$

Como en los casos anteriores, las condiciones de Karush-Kuhn-Tucker se obtienen igualando a cero el gradiente del Lagrangiano con respecto a las variables $\mathbf{w}$, b y $\xi_i\ \forall\ i \in \{1,2,...,m\}$ junto con las condiciones de holgura complementaria:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i \tag{2.31}$$

$$\nabla_b\mathcal{L} = -\sum_{i=1}^{m}\alpha_i y_i = 0 \qquad \Longrightarrow \qquad \sum_{i=1}^{m}\alpha_i y_i = 0 \tag{2.32}$$

$$\forall i, \nabla_{\xi_i}\mathcal{L} = C\eta_i - \alpha_i - \beta_i = 0 \qquad \Longrightarrow \qquad C\eta_i = \alpha_i + \beta_i \tag{2.33}$$

$$\forall i, \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] = 0 \qquad \Longrightarrow \qquad \alpha_i = 0 \vee y_i(\mathbf{w}\cdot\mathbf{x}_i + b) = 1 - \xi_i \tag{2.34}$$

$$\forall i, \beta_i \xi_i = 0 \qquad \Longrightarrow \qquad \beta_i = 0 \vee \xi_i = 0 \tag{2.35}$$

Al igual que en los casos anteriores, se observa de la ecuación (2.31) que el vector $\mathbf{w}$ que retornan las máquinas de soporte vectorial es una combinación lineal de los objetos $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m$ de la muestra de entrenamiento S. Un objeto $\mathbf{x}_i$ aparecerá en esta combinación lineal si y sólo si $\alpha_i \neq 0$. Estos objetos serán los vectores de soporte. Como en el caso no separable, existen dos tipos de vectores de soporte. Debido a las condiciones de holgura complementaria (2.34), si $\alpha_i \neq 0$, entonces $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 - \xi_i$. Si $\xi_i = 0$, entonces $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$ y el objeto $\mathbf{x}_i$ yace en algún hiperplano marginal. De lo contrario, $\xi_i \neq 0$, lo que implica que el objeto $\mathbf{x}_i$ es una anomalía. En este caso, dado que $\xi_i \neq 0$, (2.35) implica que $\beta_i = 0$ y (2.33) implica que $\alpha_i = C\eta_i$.

Por lo tanto, los vectores de soporte son anomalías (cuando $\alpha_i = C\eta_i$) o se encuentran en los hiperplanos marginales $\mathbf{w} \cdot \mathbf{x}_i + b = \pm 1$.

Como en los casos anteriores, se observa que los objetos que no son anomalías ni yacen en los hiperplanos marginales no afectan a la solución que retornan las máquinas de soporte vectorial, por lo que, en su ausencia, la solución al problema de SVM se mantendría sin cambios. De manera similar, aunque el vector $\mathbf{w}$ retornado por las SVM es único, los vectores de soporte pueden no serlo.

2.3.3. Problema de optimización dual

Aplicando la propiedad distributiva del producto punto y de la multiplicación en la ecuación del Lagrangiano se obtiene lo siguiente:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\eta_i\xi_i - \sum_{i=1}^{m}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i \\
&= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\eta_i\xi_i - \sum_{i=1}^{m}\alpha_i y_i(\mathbf{w}\cdot\mathbf{x}_i) - \sum_{i=1}^{m}\alpha_i y_i b + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\beta_i\xi_i \\
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\sum_{i=1}^{m}\alpha_i y_i\mathbf{x}_i - b\sum_{i=1}^{m}\alpha_i y_i + \sum_{i=1}^{m}\alpha_i + \sum_{i=1}^{m}(C\eta_i - \alpha_i - \beta_i)\xi_i
\end{aligned} \tag{2.36}$$

Al sustituir las ecuaciones (2.31),(2.32) y (2.33) en la ecuación (2.36) se obtiene lo siguiente:

$$
\begin{aligned}
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\mathbf{w} + \sum_{i=1}^{m}\alpha_i \\
&= \frac{1}{2}||\mathbf{w}||^2 - ||\mathbf{w}||^2 + \sum_{i=1}^{m}\alpha_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\mathbf{w}||^2 \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}||\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i||^2 \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j (\mathbf{x}_i\cdot\mathbf{x}_j)
\end{aligned} \tag{2.37}
$$

Se puede observar que la función objetivo (2.37) del problema de optimización dual en el caso ponderado es idéntica a los casos anteriores. Sin embargo, existe una ligera modificación en una de las restricciones con respecto al problema de optimización dual del caso no separable. Dada la ecuación (2.33), si $\beta_i \geq 0$, entonces $\alpha_i \leq C\eta_i \ \forall\ i \in \{1, 2, ..., m\}$. Por lo tanto, el problema de optimización dual en el caso ponderado queda definido de la siguiente manera:

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j (\mathbf{x}_i\cdot\mathbf{x}_j) \\
&\text{s. t. } \sum_{i=1}^{m}\alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C\eta_i \ \forall\ i \in \{1, 2, ..., m\}
\end{aligned} \tag{2.38}
$$

Al igual que en los casos anteriores, las soluciones encontradas $\mathbf{w}^*$ y $\boldsymbol{\alpha}^*$ a los problemas de optimización primal y dual cumplen con la ecuación (2.31):

$$\mathbf{w}^* = \sum_{i=1}^{m}\alpha_i^* y_i \mathbf{x}_i$$

De manera similar, b^* puede calcularse ocupando cualquier vector de soporte que se encuentre en alguno de los hiperplanos marginales. Para tales vectores de soporte, se cumple que $\mathbf{w}^*\cdot\mathbf{x}_i + b^* = y_i$, por lo tanto:

$$b^* = y_i - \mathbf{w}^*\cdot\mathbf{x}_i = y_i - \sum_{j=1}^{m}\alpha_j^* y_j (\mathbf{x}_j\cdot\mathbf{x}_i) \tag{2.39}$$

Obtenidos los valores de $\boldsymbol{\alpha}^*$, $\mathbf{w}^*$ y b^*, el modelo de clasificación para un nuevo objeto $\mathbf{x}$ por clasificar es la función:

$$h(\mathbf{x}) = sgn(\mathbf{w}^*\cdot\mathbf{x} + b^*) = sgn(\sum_{i=1}^{m}\alpha_i^* y_i (\mathbf{x}_i\cdot\mathbf{x}) + b^*) \tag{2.40}$$

Capítulo 3

Marco teórico: k-medoides

La práctica de agrupar objetos de acuerdo a similitudes percibidas entre dichos objetos es la base de gran parte de la ciencia. Organizar datos e información de manera coherente es uno de las expresiones mas fundamentales de la comprensión y el aprendizaje [25].

La tarea de agrupamiento (clustering en inglés) es una de las tareas de aprendizaje no supervisado que busca la organización de una colección de objetos en grupos, basándose en la similitud o diferencia entre ellos. Intuitivamente, los objetos dentro de un mismo grupo son más similares entre sí que a un objeto que pertenece a un grupo diferente. Un ejemplo de agrupación se muestra en la figura 3.1 [23]. Los objetos de entrada al método de agrupamiento se muestran en la figura 3.1 (a), y los grupos obtenidos se muestran en la figura 3.1 (b). A los objetos pertenecientes al mismo grupo se les asigna la misma etiqueta $1, 2, 3, 4, 5, 6$ ó 7, respectivamente. La variedad de técnicas para representar un dato, las posibles funciones que se pueden utilizar para medir la similitud o diferencia entre los objetos de entrada y los distintos criterios al agrupar objetos han producido una extensa variedad de métodos de agrupación.

En este capítulo se presentan los métodos de agrupamiento k-medias y k-medoides. Primero se presenta el algoritmo de k-medias, que realiza la partición de un conjunto de datos en k grupos, utilizando como criterio la distancia euclidiana entre los objetos en la muestra de entrenamiento. Después, se presenta el algoritmo de k-medoides, que es una modificación del algoritmo k-medias.

3.1. Planteamiento general de un problema de agrupamiento

En el campo del aprendizaje de máquina, Everitt indica que es difícil proporcionar una definición formal de grupo (cluster) [28]. Después, en 1980, Everitt documenta algunas de las siguientes definiciones:

"A cluster is a set of entities which are alike, and entities from different clusters are not alike". [27]

"Un grupo es un conjunto de entidades que son semejantes, y entidades de diferentes grupos no son semejantes".

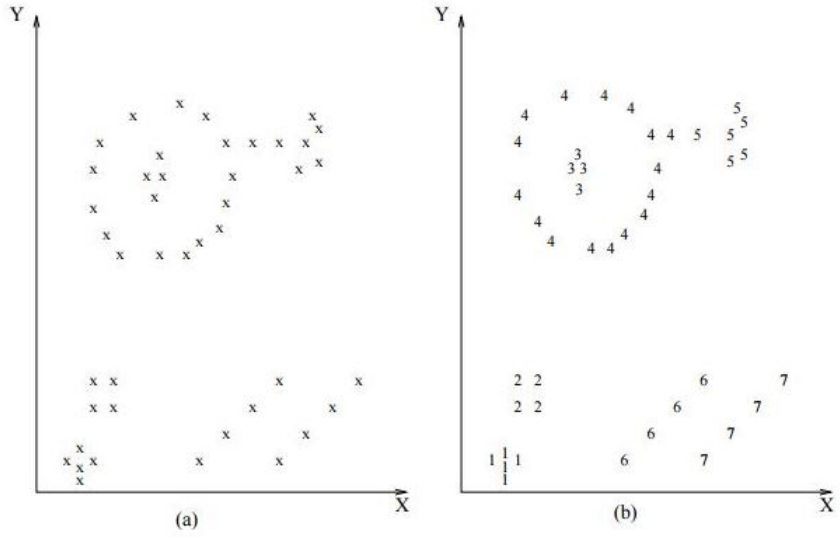

Figura 3.1: Ejemplo en un espacio de dos dimensiones de un posible agrupamiento de datos

"A cluster is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it". [27]

"Un grupo es una agregado de objetos en el espacio de prueba tal que la distancia entre dos objetos cualesquiera en el grupo es menor que la distancia entre cualquier objeto en el grupo y cualquier objeto fuera de él".

"Clusters may be described as connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points". [27]

"Los grupos pueden ser descritos como regiones conectadas de un espacio multidimensional conteniendo una relativa alta densidad de objetos, separado de otras regiones similares por una región que contiene una relativa baja densidad de objetos".

Las últimas dos definiciones asumen que los objetos que se pretende agrupar son representados como puntos en algún cierto espacio. Intuitivamente, los humanos reconocemos un grupo al observarlo en dos dimensiones, como se puede verificar en la figura 3.1, aunque no es del todo claro como realizamos esta acción de manera tan aparentemente sencilla [25].

Existen dos maneras principales de realizar agrupamiento: Agrupamiento particional y agrupamiento jerárquico. Los algoritmo de k-medias y k-medoides pertenecen al tipo de agrupamiento particional, que podemos definir de la siguiente manera [26] [31]:

Dado un conjunto de objetos $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ donde $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iN})$, un método de agrupamiento particional buscará dividir al conjunto $\mathbf{X}$ en k particiones ($m \geq k \geq 1, k \in \mathbb{N}$ donde $\mathbb{N}$ denota el conjunto de los números naturales) $\mathcal{C} = \{C_1, C_2, ..., C_k\}$ tal que:

- $C_i \neq \emptyset, \forall\ i \in \{1, 2, ..., k\}$
- $\bigcup_{i=1}^{k} C_i = \mathbf{X}$
- $C_i \cap C_j = \emptyset, \forall\ i, j \in \{1, 2, ..., k\}, i \neq j$

En el agrupamiento particional cada objeto $\mathbf{x}_i$ es asociado exclusivamente a un sólo grupo. Además, los objetos deben de estar agrupados de tal manera que los objetos en un mismo grupo sean más "similares" entre sí que lo que son a objetos de otros grupos. Para realizar lo anterior, se debe definir una función $f : \mathbf{X} \times \mathbf{X} \rightarrow \Re$ que "mida" la similitud (o diferencia) entre un par de objetos.

3.2. k-medias

El algoritmo de k-medias fue propuesto por MacQueen [24] en 1967. Fisher [29] y Cox [30] habían estudiado problemas similares dando versiones aproximadas del algoritmo. Un seguimiento mas a detalle de la historia del algoritmo de k-medias junto con la descripción de algunas variaciones del algoritmo es dada en [25]. La importancia del algoritmo de k-medias como técnica de agrupamiento y en el campo de minería de datos es resaltada en [32] y [33]. A continuación se presenta el algoritmo de k-medias, mencionando sus limitaciones e inconvenientes.

Se considera un conjunto de objetos $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ donde $\mathbf{x}_i \in \Re^N\ \forall\ i \in \{1, 2, ..., m\}$ y un número de grupos $k \geq 1$. El algoritmo comienza seleccionando k puntos en $\Re^N$ que servirán como los representantes o centroides iniciales de los k grupos, denotados como $\mathbf{c}_j\ \forall\ j \in \{1, 2, ..., k\}$, uno para cada grupo C_j. Existen diversas formas de elegir los centroides iniciales. Una de ellas es eligiendo k objetos al azar del conjunto $\mathbf{X}$ y otra es eligiendo k puntos al azar en el espacio $\Re^N$, por mencionar algunas. La versión del algoritmo de k-medias que se presenta en este documento elige k puntos al azar en el espacio $\Re^N$ como centroides iniciales. El siguiente paso es tomar cada objeto que pertenece al conjunto $\mathbf{X}$ y asignarlo al grupo con el centroide más cercano. Cuando todos los objetos han sido asignados a su respectivo grupo, se recalculan las posiciones de los k centroides como el centro o la media de los objetos asignados a cada grupo en el paso anterior. Esta es la razón por la que el algoritmo se llama k-medias. Una vez actualizados los k centroides, se vuelve a asignar cada objeto $\mathbf{x}_i$ del conjunto $\mathbf{X}$ al grupo con el centroide más cercano, y el proceso se repite. Existen diversos criterios de paro, pero el más común es detener el ciclo una vez que los k centroides no cambian su posición entre una iteración y otra. La ejecución del algoritmo se muestra de manera gráfica en un espacio $\Re^2$ de dos dimensiones en la figura 3.2, tomada de [31]. Cada iteración ocupa mk comparaciones, que determina la complejidad temporal de una iteración. El número de iteraciones requerido para la convergencia varía y puede depender de N, sin embargo, la complejidad de este algoritmo se puede considerar lineal en el tamaño del conjunto de objetos (m).

El algoritmo de k-medias busca una partición que minimice el error cuadrático expresado en la siguiente ecuación [32]:

$$\sum_{i=1}^{m} \left(\min_{j} ||\mathbf{x}_i - \mathbf{c}_j||^2 \right) \tag{3.1}$$

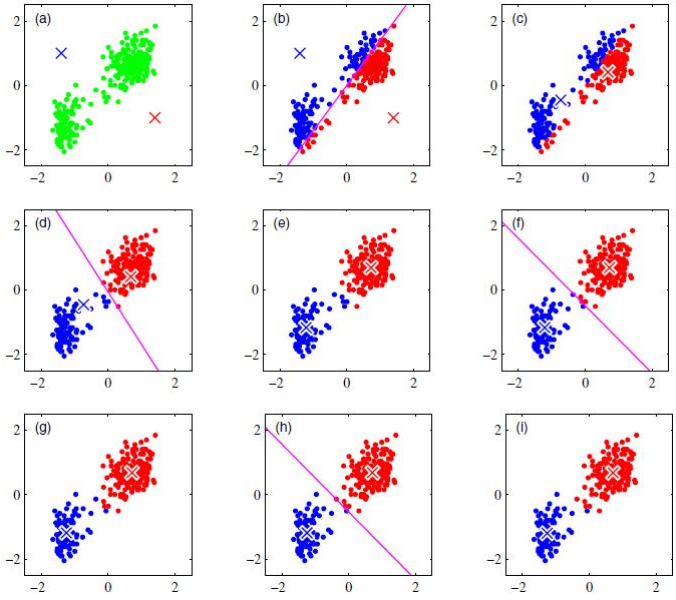

Figura 3.2: Ejecución del algoritmo k-medias. Los puntos verdes representan al conjunto de entrada $\mathbf{X}$ en un espacio de dos dimensiones. Los centroides se indican con las cruces azul y roja.

Diferentes elecciones de los k centroides iniciales pueden ocasionar diferentes agrupamientos finales debido a que un algoritmo basado en el criterio del error cuadrático puede converger a mínimos locales [25].

El algoritmo de k-medias es sencillo y fácil de implementar para resolver una gran cantidad de problemas prácticos. Su complejidad temporal es $O(mkN)$. Como k y N son usualmente mucho menores que m, en la práctica se considera que la complejidad del algoritmo de k-medias es lineal con respecto al tamaño m del conjunto de entrada.

Sin embargo, el algoritmo cuenta con algunas limitaciones e inconvenientes que a continuación se resumen. Como se menciono anteriormente, el procedimiento iterativo del algoritmo k-medias no garantiza converger al óptimo global. Además, como es necesario calcular las medias de los grupos, el algoritmo esta limitado a trabajar en $\Re^N$, es decir, no puede trabajar con objetos con atributos categóricos como colores (rojo, azul, etc.) o tamaños (grande, pequeño, mediano, etc.). Otro inconveniente es que, debido a que se utiliza como función de diferencia la distancia euclidiana, los grupos retornados por el algoritmo de k-medias tienden a tener geometría hiperesférica o convexa, por lo que, si los grupos por encontrar tuvieran otro tipo de geometría (una espiral, un anillo, etc.), el algoritmo de k-medias ya no sería efectivo. Finalmente, es importante mencionar que no se conoce un método eficaz y universal para identificar los centroides iniciales y el número de grupos k.

Algoritmo 1: Algoritmo de k-medias

Entrada: Conjunto de objetos $\mathbf{X}$, número k de grupos a formar
Salida : Una partición de $\mathbf{X}$

1 Seleccionar al azar k centroides $\mathbf{c}_j \in \Re^N \ \forall \ j \in \{1, 2, ..., k\}$;
2 Crear k grupos C_j tal que $C_j = \{\mathbf{c}_j\} \ \forall \ j \in \{1, 2, ..., k\}$;
3 **repeat**
4 Calcular y almacenar la distancia $d(\mathbf{x}_i, \mathbf{c}_j)$ entre cada objeto $\mathbf{x}_i \in \mathbf{X}$ y cada uno de los k centroides $\mathbf{c}_j$, $d(\mathbf{x}_i, \mathbf{c}_j) \longleftarrow ||\mathbf{x}_i - \mathbf{c}_j||$;
5 Asignar cada objeto $\mathbf{x}_i \in \mathbf{X}$ al grupo C_j que contenga el centroide $\mathbf{c}_j$ más cercano, $C_j \longleftarrow C_j \bigcup \mathbf{x}_i : \underset{j}{\operatorname{argmin}}\, d(\mathbf{x}_i, \mathbf{c}_j)$;
6 Calcular la nueva posición del centroide de cada grupo, $\mathbf{c}_j \longleftarrow \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} \mathbf{x}$;
7 **until** *el conjunto de centroides* $\mathbf{c}_j$ *sea idéntico que el de la iteración anterior*;

3.3. k-medoides

Una de las limitantes del algoritmo de k-medias es que sólo puede ocuparse con objetos en un espacio $\Re^N$, ya que es necesario calcular las medias de cada grupo. Una modificación al algoritmo de k-medias fue propuesta en 1987 por Kaufman y Rousseeuw [34] [35]. El algoritmo modificado es llamado k-medoides, y en lugar de actualizar los centroides $\mathbf{c}_j$ de cada grupo C_j utilizando la media de los objetos que pertenecen al grupo C_j, el algoritmo de k-medoides actualiza los centroides $\mathbf{c}_j$ utilizando el medoide del correspondiente grupo C_j, que es el objeto $\mathbf{x}^*$ más representativo del grupo C_j.

Debido a que ya no se calculan las medias de los grupos, el algoritmo ya no está limitado a trabajar en $\Re^N$, por lo que se pueden ocupar otras funciones de semejanza o diferencia $f(\mathbf{x}_i, \mathbf{x}_j)$ entre objetos. El medoide de un grupo C_j se puede definir como el objeto $\mathbf{x} \in C_j$ cuyo promedio de semejanza a todos los objetos en el grupo es máxima, es decir, es el objeto más "céntrico"del grupo. El algoritmo de k-medoides busca maximizar la suma de semejanzas entre cada objeto y su centroide correspondiente.

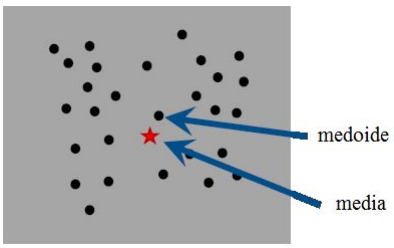

Figura 3.3: Ejemplo de la diferencia entre la media y el medoide de un grupo de objetos

La principal diferencia del algoritmo k-medoides con respecto al algoritmo de k-medias es que los centroides $\mathbf{c}_j$ de cada grupo C_j tienen que pertenecer al conjunto de datos entrada $\mathbf{X}$ que se pretende agrupar, como se muestra en la figura 3.3, tomada de [36].

Algoritmo 2: Algoritmo de k-medoides

Entrada: Conjunto de objetos $\mathbf{X}$, número k de grupos a formar
Salida : Una partición de $\mathbf{X}$

1 Seleccionar al azar k objetos $\mathbf{x}_p \in \mathbf{X}$ del conjunto de entrada como centroides iniciales $\mathbf{c}_j$;
2 Crear k grupos C_j tal que $C_j = \{\mathbf{c}_j\}\ \forall\ j \in \{1, 2, ..., k\}$;
3 **repeat**
4 Calcular y almacenar la semejanza $f(\mathbf{x}_i, \mathbf{c}_j)$ entre cada objeto $\mathbf{x}_i \in \mathbf{X}$ y cada uno de los k centroides $\mathbf{c}_j$;
5 Asignar cada objeto $\mathbf{x}_i \in \mathbf{X}$ al grupo C_j que contenga el centroide $\mathbf{c}_j$ más semejante a $\mathbf{x}_i$, $C_j \longleftarrow C_j \bigcup \mathbf{x}_i : \underset{j}{\operatorname{argmax}}\, f(\mathbf{x}_i, \mathbf{c}_j)$;
6 Calcular la nueva posición del centroide de cada grupo, $\mathbf{c}_j \longleftarrow \mathbf{x} : \underset{\mathbf{x} \in C_j}{\operatorname{argmax}} \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} f(\mathbf{x}, \mathbf{x}_i)$;
7 **until** *el conjunto de centroides* $\mathbf{c}_j$ *sea idéntico que el de la iteración anterior*;

Capítulo 4

Trabajos relacionados

Debido a la complejidad temporal $O(m^3)$ y espacial $O(m^2)$ del entrenamiento de las máquinas de soporte vectorial, una porción significativa de la investigación relacionada a las máquinas de soporte vectorial se enfoca en el desarrollo de métodos eficientes para su entrenamiento [38]. Estos métodos han sido clasificados en las siguientes categorías: reducción de datos, descomposición, algunas variantes de las máquinas de soporte vectorial y otros métodos que utilizan heurísticas o cómputo paralelo [13] [37]. En este capítulo se expone una breve descripción de estos métodos, enfatizando y cubriendo con mayor detalle los métodos de reducción de datos, debido a que el método propuesto en esta tesis cae en esta categoría.

4.1. Métodos de descomposición

El hiperplano de separación óptima descrito por la ecuación (2.14) que se obtiene al utilizar las máquinas de soporte vectorial es computado resolviendo el problema de optimización de programación cuadrática descrito en (2.4), (2.17) ó (2.29), respectivamente.

Los métodos de descomposición abordan el problema del entrenamiento de las máquinas de soporte vectorial optimizando de manera iterativa sólo las variables que pertenecen a un cierto subconjunto llamado conjunto de trabajo (working set en inglés). Las variables que no pertenecen al conjunto de trabajo se consideran fijas. El éxito del método depende de gran manera de la selección del conjunto de trabajo [51] [52]. Osuna, Freund y Girosi [53] encuentran el conjunto de trabajo escogiendo las variables que no cumplen con las condiciones de Karush-Kuhn-Tucker.

Probablemente el algoritmo de descomposición más famoso es el algoritmo de optimización secuencial mínima (SMO por sus siglas en inglés) [54]. SMO considera el tamaño más pequeño posible para el conjunto de trabajo: sólo dos variables. Otros métodos de descomposición aparecen en [52], [53] y [55].

4.2. Variantes de máquinas de soporte vectorial

Algunos investigadores han modificado el problema de optimización de programación cuadrática original asociado a las máquinas de soporte vectorial para acelerar su tiempo de entrenamiento, a costa de perder precisión en la clasificación. Una de éstas modificaciones es llamada versión de mínimos cuadrados de las máquinas de soporte vectorial (LS-SVM, por sus siglas en inglés) y fue presentada por Suykens y Vandewalle en 1999 [49]. En esta modificación no se necesita resolver un problema de optimización de programación cuadrática, sino sólamente un sistema de ecuaciones lineales.

Otra variante presentada por Fung y Mangasarian [50], es llamada máquinas de soporte vectorial proximales (PSVM, por sus siglas en inglés). Los hiperplanos proximales se definen de tal manera que los objetos de una clase se agrupen alrededor del hiperplano proximal correspondiente y que los hiperplanos proximales se encuentren tan alejados uno del otro tanto como sea posible. Esta variante clasifica a los objetos asignándolos a uno de los dos hiperplanos proximales, específicamente, al hiperplano proximal al que se encuentre más cercano.

4.3. Otros métodos

En esta sección se mencionan algunos métodos que no se consideran en las demás categorías.

4.3.1. Máquinas de soporte vectorial en paralelo

Como su nombre lo sugiere, este tipo de métodos utilizan la potencia del cómputo en paralelo para intentar resolver el problema del entrenamiento de las máquinas de soporte vectorial. Sin embargo, la ejecución en paralelo es difícil debido a la dependencia entre pasos computacionales [39].

Uno de los enfoques más comunes al paralelizar es dividir el conjunto de entrenamiento en subconjuntos y procesar cada uno de éstos subconjuntos en paralelo [39] [40]. Otro tipo de enfoques al paralelizar son presentados en [41] y [42].

4.3.2. Semillas alfa (Alpha seeding)

El enfoque de semillas alfa (alpha seeding en inglés) consiste en proporcionar estimaciones iniciales de los valores α_i de los multiplicadores de Lagrange del problema de optimización dual de las máquinas de soporte vectorial (descrito en la ecuación (2.12), (2.26) ó (2.38), respectivamente) para resolverlo de manera mas eficiente y mejorar el tiempo de entrenamiento. Métodos basados en este enfoque se presentan en [43] y [44]. Es importante recordar que los valores α_i de los multiplicadores de Lagrange son esenciales para el problema de máquinas de soporte vectorial ya que son los que definen cuales objetos son vectores de soporte.

4.3.3. Entrenamiento en línea (On-line training)

Existen ocasiones en los que la adquisición de los datos de entrenamiento es caro y toma largos periodos de tiempo. En consecuencia, es común que éstos datos se encuentren disponibles por pequeños paquetes cada cierto periodo de tiempo, es decir, que no sea posible para un método como las máquinas de soporte vectorial tener acceso a la muestra de entrenamiento por completo, sino por "pedazos" entregados cada cierto periodo de tiempo. En escenarios como éste, es necesario actualizar el modelo de clasificación de manera que tome en cuenta los nuevos objetos que le son proporcionados sin comprometer la precisión de clasificación de los objetos con los que ya había sido entrenado [45].

Los métodos incrementales son métodos que se desarrollaron para abordar este problema. Algunos de los métodos basados en este enfoque se presentan en [45], [46], [47] y [48].

4.4. Métodos de reducción de datos

Como se menciono en el capítulo 2, los objetos que no son vectores de soporte no afectan a la solución que retornan las máquinas de soporte vectorial, es decir, en su ausencia, la solución al problema de SVM se mantendría sin cambios.

En el caso separable de las máquinas de soporte vectorial, sólo los objetos que yacen sobre los hiperplanos marginales son candidatos a ser vectores de soporte. En el caso no separable, además de los objetos que yacen sobre los hiperplanos marginales, sólo los objetos considerados anomalías (outliers) son candidatos a ser vectores de soporte. Considerando lo anterior, es probable que la gran mayoría de objetos en el conjunto de entrenamiento no sean candidatos a ser vectores de soporte, ya que en la mayoría de los casos prácticos pocos objetos del conjunto de entrenamiento son anomalías o yacen en los hiperplanos marginales.

Tomando en cuenta lo anterior, el enfoque de los métodos de reducción de datos para entrenar máquinas de soporte vectorial es crear o seleccionar un subconjunto $\mathcal{R}$ de tamaño $n < m$ a partir del conjunto de entrenamiento S y entrenar el método de máquinas de soporte vectorial con este subconjunto. Debido a que $n < m$, el tiempo de entrenamiento utilizando sólo n objetos es menor.

Se han propuesto diversas maneras de escoger el subconjunto $\mathcal{R}$. Algunas de ellas eliminan objetos del conjunto de entrenamiento original, mientras que otras construyen el subconjunto $\mathcal{R}$ de manera incremental agregando objetos que, mediante algún criterio, se estiman que pueden ser fuertes candidatos para ser vectores de soporte. Algunas de éstas técnicas utilizan heurísticas, mientras que otras utilizan otros algoritmos de aprendizaje supervisado para seleccionar a los candidatos a ser vectores de soporte.

Debido a la naturaleza de éste trabajo, se dividirán estas propuestas en dos tipos: las que utilizan métodos de agrupamiento de aprendizaje de máquina para seleccionar el subconjunto $\mathcal{R}$ y las que utilizan cualquier otro tipo de estrategia para seleccionar el subconjunto $\mathcal{R}$. Ya que el método propuesto en esta tesis utiliza un método de agrupamiento para seleccionar el subconjunto $\mathcal{R}$, se cubrirá con mayor detalle este tipo de métodos.

4.4.1. Métodos de reducción de datos que no ocupan métodos de agrupamiento

En [58], Keerthi y sus colegas presentan un método que de manera voraz (greedily en inglés) va incrementando un conjunto de vectores de soporte, tratando de maximizar la precisión de clasificación del modelo obtenido. Bottou y sus colegas [59] proponen un algoritmo que remueve objetos del conjunto de entrenamiento usando estimaciones probabilísticas inspirado en el algoritmo presentado en [60]. Balcázar y sus colegas [56] [57] utilizan técnicas de muestreo aleatorio para seleccionar los objetos del conjunto de entrenamiento que serán candidatos a ser vectores de soporte. Schohn y Cohn [61] presentan una heurística de aprendizaje activo para reducir el número de objetos en el conjunto de entrenamiento.

Lee y sus colegas [62] [63] proponen un método llamado máquinas de soporte vectorial reducidas (RSVM, por sus siglas en inglés), en donde se asume que objetos del conjunto de entrenamiento escogidos al azar pueden representar a todos los objetos en el conjunto de entrenamiento. En [64], Peres y Pedreira proponen un esquema para seleccionar objetos clave en una muestra de entrenamiento. Este esquema no sólo es aplicable a SVM, sino que también puede ser utilizado en otros algoritmos de aprendizaje. La principal herramienta utilizada por este esquema es la divergencia de Cauchy-Schwartz, que es utilizada como una medida de la diferencia entre dos densidades de probabilidad.

Los métodos presentados en [65], [66], [67] y [68] asumen que los objetos de la muestra de entrenamiento cuyos vecinos más cercanos pertenecen a otra clase posiblemente se encuentren cerca del hiperplano de separación. Algunos de éstos métodos incorporan en su esquema el algoritmo de aprendizaje supervisado llamado k-vecinos más cercanos (k-NN por sus siglas en inglés).

López Chau presenta en [13] un par de métodos. Uno de ellos se basa en calcular la cáscara convexa de los objetos de cada clase. El otro método combina el uso de árboles de decisión y del discriminante lineal de Fisher.

4.4.2. Reducción de datos utilizando métodos de agrupamiento

Los algoritmos de k-medias y k-medoides fueron presentados en el capítulo 3 de esta tesis como algoritmos de agrupamiento. Además de este par de algoritmos, existe una gran variedad de algoritmos de agrupamiento que son utilizados en tareas de aprendizaje no supervisado. Algunos de ellos han sido utilizados para pre-procesar el conjunto de entrenamiento de las máquinas de soporte vectorial de distintas maneras. A continuación se presentarán algunos de estos enfoques.

En 2003, Yu y sus colegas [69] proponen un método llamado máquinas de soporte vectorial basadas en agrupamiento (CB-SVM, por sus siglas en inglés) que utiliza un algoritmo de agrupamiento jerárquico para proveer a las SVM de un conjunto de entrenamiento reducido. El algoritmo de agrupamiento jerárquico que se utiliza en este trabajo es llamado Agrupamiento y reducción iterativo balanceado usando jerarquías (BIRCH, por sus siglas en inglés) propuesto en [70] por Zhang y sus colegas.

Xiong y sus colegas [71] utilizan un método de agrupamiento sustractivo propuesto por Chiu [72] para reducir el tamaño del conjunto de entrenamiento tratando de mantener la mayor cantidad de información posible. En [73], Linda y Manic utilizan el algoritmo de gas neuronal creciente (GNG, por sus siglas en inglés) propuesto por Fritzke en [74] para pre-procesar el conjunto de

entrenamiento de las SVM. El algoritmo GNG es un algoritmo de agrupamiento capaz de aprender las relaciones topológicas de un conjunto de datos.

Un método llamado mapas opuestos (Opposite maps en inglés) es propuesto por Neto y Barreto [75] en 2013. Este método puede utilizar cualquier algoritmo de agrupamiento para realizar el pre-procesamiento del conjunto de entrenamiento de las SVM, aunque específicamente en el artículo utilizan los algoritmos de k-medias, k-medias con kernel, mapa auto-organizado y GNG. La idea fundamental de este método es realizar el agrupamiento de los objetos en cada clase para después utilizar una heurística que busca los agrupamientos mas cercanos a los objetos de la clase contraria, con la idea de que los vectores de soporte son objetos que usualmente se encuentran cercanos a los objetos de la clase contraria.

De Almeida y sus colegas [80] presentan un método que ejecuta el algoritmo de k-medias sobre el conjunto de entrenamiento original S sin tomar en cuenta la clase o etiqueta de los objetos. Los grupos que contienen objetos de una misma clase son descartados y se almacenan solamente los centroides de este tipo de grupos. De manera contraria, los grupos que contienen objetos de clases distintas son preservados y se almacenan todos los objetos pertenecientes a este tipo de grupos. Con el conjunto formado por los objetos y centroides almacenados después de este proceso, el método de SVM es entrenado.

Koggalage y Halgamuge [79] proponen un método que puede utilizar cualquier algoritmo de agrupamiento para realizar el pre-procesamiento del conjunto de entrenamiento de las SVM para identificar los grupos iniciales. Al igual que en el método propuesto por De Almeida y sus colegas, el agrupamiento se realiza sobre el conjunto de entrenamiento S sin tomar en cuenta la clase o etiqueta de los objetos. Una vez detectados los grupos iniciales, se identifican los grupos que contengan solamente objetos de la misma clase. A estos grupos se les llama grupos crujientes (crisp clusters en inglés). El resto del método consiste en implementar una técnica para refinar los grupos crujientes de tal manera que se extraigan de ellos los posibles vectores de soporte. Aunque se mencionó que el método puede utilizar cualquier algoritmo de agrupamiento, Koggalage y Halgamuge utilizan el algoritmo de k-medias en su artículo.

Un método parecido al propuesto por De Almeida y sus colegas es propuesto en [81] por Tran y sus colegas. Este método también ejecuta el algoritmo de k-medias sobre el conjunto de entrenamiento original S. Sin embargo, en esta propuesta el agrupamiento de los objetos se realiza en cada clase, tomando en cuenta la etiqueta de los objetos. Otra diferencia es que en este caso el método de SVM es entrenado con un conjunto formado solamente por los centroides encontrados por el algoritmo de k-medias.

Un método para acelerar el entrenamiento de las SVM basado en el algoritmo de k-medias es propuesto en [88]. Al igual que el método propuesto en esta tesis, este método utiliza los centroides devueltos por el algoritmo de k-medias para entrenar a las SVM. Sin embargo, este método se enfoca en presentar una heurística para determinar el número k de grupos por formar y no presenta una manera de incorporar la información estadística de los grupos formados en el esquema de SVM.

En [89], Yao y sus colegas presentan un algoritmo parecido al presentado en [80]. La idea básica es la misma: ejecutar el algoritmo de k-medias sobre el conjunto de entrenamiento original S sin tomar en cuenta la clase o etiqueta de los objetos y, una vez que se formaron los grupos, se buscan los objetos que han sido agrupados junto con otros objetos de etiqueta distinta. Una vez encontrados, Yao y sus colegas proponen utilizar el algoritmo de k vecinos más cercanos (k-NN por sus siglas en inglés) para realizar una búsqueda de los posibles candidatos a vectores de soporte. Al igual que sucede con el método propuesto en [88], la principal diferencia de éste método con el método

propuesto en esta tesis es que éste método no incorpora la información de los grupos formados por k-medias en el esquema de SVM.

Gu y Han presentan en [90] un método llamado máquinas de soporte vectorial agrupadas (CSVM por sus siglas en inglés). Este algoritmo divide el conjunto de entrenamiento en varios grupos utilizando el algoritmo de k-medias, y luego realiza el entrenamiento de SVM en cada grupo, realizando una separación local para cada grupo. El método utiliza un término de regularización global, que afecta a las soluciones encontradas de manera local de tal manera que hace que éstas se "acerquen" entorno a una referencia global.

En [76], Li y sus colegas combinan el algoritmo de k-medias junto con una idea basada en el grafo de Gabriel. El grafo de Gabriel es un grafo definido sobre un conjunto de vértices $\mathcal{V}$ en un espacio euclidiano que expresa la noción de proximidad entre estos vértices. El grafo de Gabriel fue propuesto en [77] por Gabriel y Sokal en 1969. En [76], el grafo de Gabriel es utilizado para identificar de manera aproximada cuales objetos son los posibles vectores de soporte. Otro algoritmo que utiliza una combinación de k-medias junto con otro algoritmo para pre-procesar el conjunto de entrenamiento de las SVM es presentado por Lu y sus colegas en [78].

Yang y sus colegas [82] integran al esquema de máquinas de soporte vectorial con ponderación un método de pre-procesamiento para el conjunto de entrenamiento que además genera los pesos η_i basado en el algoritmo de agrupación llamado c-medias posibilístico (Possibilistic c-means en inglés) [84].

De manera similar, Nguyen y su colegas [83] integran al esquema de máquinas de soporte vectorial con ponderación un método de pre-procesamiento para el conjunto de entrenamiento. En este caso, el método de pre-procesamiento puede ser cualquier algoritmo de agrupamiento, aunque en el artículo utilizan el algoritmo de k-medias para este fin. Al igual que en el método propuesto por Tran y sus colegas, en este caso el método de SVM es entrenado con un conjunto formado solamente por los centroides encontrados por el algoritmo de k-medias. Este esquema propone dos maneras para generar los pesos η_i. La primera es generando los pesos de manera que sean proporcionales al tamaño de la clase correspondiente. Para clasificación binaria, el correspondiente peso η_i del centroide c_i se calcula con la siguiente expresión:

$$\eta_i = \frac{z_i}{2\sum_{j=1}^{2} m_j \gamma_{i,j}} \tag{4.1}$$

donde z_i es la cantidad de objetos asignados al grupo C_i con centroide c_i, m_1 y m_2 son la cantidad de objetos pertenecientes a cada clase, de tal manera que $m_1 + m_2 = m$ donde m es el número de objetos en el conjunto de entrenamiento S y $\gamma_{i,j}$ es una variable tal que $\gamma_{i,j} = 1$ si el centroide c_i pertenece a la clase j y $\gamma_{i,j} = 0$ de cualquier otra manera.

La segunda manera es generando los pesos de manera que sean proporcionales al tamaño del grupo correspondiente. Entonces, el correspondiente peso η_i del centroide c_i se calcula con la siguiente expresión:

$$\eta_i = \frac{z_i}{m} \tag{4.2}$$

Capítulo 5

Propuesta

Un esquema para el entrenamiento de las máquinas de soporte vectorial llamado máquinas de soporte vectorial con ponderación probabilística (PWSVM por sus siglas en inglés) es presentado en este capítulo.

La idea clave del enfoque propuesto es reducir el tamaño del conjunto de entrenamiento original S utilizando el algoritmo de agrupación k-medoides. Con el fin de evitar la pérdida de información durante la agrupación, se propone utilizar al conjunto de centroides de los grupos entregados por el algoritmo de agrupación como el nuevo conjunto de entrenamiento $\mathcal{R}$. También se propone utilizar información acerca de la dispersión de los objetos de cada grupo.

Además, el esquema original de máquinas de soporte vectorial con pesos propuesto en [19] es modificado para aprovechar la información estadística encontrada por el algoritmo de agrupación.

5.1. Método para la reducción de datos

El esquema propuesto considera un espacio de entrada $\mathcal{X} \subseteq \Re^N$ con $N \geq 1$, un espacio objetivo $\mathcal{Y} = \{-1, +1\}$ y una muestra de entrenamiento S de tamaño m con m_1 objetos $\mathbf{x}_p$ con etiqueta $y_p = +1$ y m_2 objetos $\mathbf{x}_q$ con etiqueta $y_q = -1$ tal que $m_1 + m_2 = m$.

La primera parte del esquema reduce el conjunto de entrenamiento S utilizando el algoritmo de agrupamiento k-medoides, que entrega un conjunto de entrenamiento reducido $\mathcal{R}$ de tamaño k, formado por los centroides de los grupos obtenidos por el algoritmo de k-medoides. La agrupación de los objetos se realiza de acuerdo a su clasificación, es decir, por un lado, se utiliza el algoritmo de k-medoides para agrupar a los objetos $\mathbf{x}_p$ con etiqueta $y_p = +1$, formando un conjunto de centroides $\mathcal{R}^+$ de tamaño k_+ y después se ejecuta nuevamente el algoritmo de k-medoides para agrupar a los objetos $\mathbf{x}_q$ con etiqueta $y_q = -1$, formando un conjunto de centroides $\mathcal{R}^-$ de tamaño k_-, de tal forma que $\mathcal{R}^+ \cup \mathcal{R}^- = \mathcal{R}$. Los valores de k_+ y k_- son definidos de manera que la proporción $\frac{m_1}{m_2} \approx \frac{k_+}{k_-}$ se mantenga lo mejor posible.

Además de entregar el conjunto de centroides $\mathcal{R}$, durante la ejecución del algoritmo de k-medoides se calcula el número de objetos asignados a cada grupo y la varianza de la distancia de los objetos

pertenecientes a cada grupo con respecto a su centroide calculado. El número de objetos de cada grupo se denomina con la letra z y se almacena en un conjunto $\mathcal{Z}$ de tamaño k de tal manera que $\mathcal{Z}^+ \cup \mathcal{Z}^- = \mathcal{Z}$, de manera similar a la manera en que se formó el conjunto de centroides $\mathcal{R}$. Los valores de varianza se denominan σ^2 y se almacenan en un conjunto $\mathcal{D}$ de tamaño k de tal manera que $\mathcal{D}^+ \cup \mathcal{D}^- = \mathcal{D}$, de manera similar a la manera en que se formó el conjunto de centroides $\mathcal{R}$. Los valores almacenados en los conjuntos $\mathcal{Z}$ y $\mathcal{D}$ serán utilizados como medida de dispersión de los grupos encontrados por el algoritmo de k-medoides, y se integrarán en el esquema propuesto.

Una de las ventajas de utilizar el algoritmo de k-medoides (o incluso k-medias) para pre-procesar el conjunto de entrenamiento para las SVM, es que el tamaño del nuevo conjunto de entrenamiento $\mathcal{R}$ es controlado simplemente escogiendo el número de grupos k. Además, a diferencia de otros algoritmos de agrupamiento como k-medias ó c-medias difuso (fuzzy c-means en inglés), los centroides entregados por k-medoides son objetos pertenecientes al conjunto de entrenamiento original S.

Algoritmo 3: Algoritmo de k-medoides para PWSVM

Entrada: Muestra de entrenamiento S, número k de grupos a formar

Salida : Un par de conjuntos de centroides: $\mathcal{R}^+ = \{\mathbf{c}_1^+, \mathbf{c}_2^+, ..., \mathbf{c}_{k_+}^+\}$ y $\mathcal{R}^- = \{\mathbf{c}_1^-, \mathbf{c}_2^-, ..., \mathbf{c}_{k_-}^-\}$, un par de conjuntos de parámetros de dispersión: $\mathcal{D}^+ = \{\sigma_{+1}^2, \sigma_{+2}^2, ..., \sigma_{+k_+}^2\}$ y $\mathcal{D}^- = \{\sigma_{-1}^2, \sigma_{-2}^2, ..., \sigma_{-k_-}^2\}$ y un par de conjuntos de tamaños de grupos: $\mathcal{Z}^+ = \{z_1^+, z_2^+, ..., z_{k_+}^+\}$ y $\mathcal{Z}^- = \{z_1^-, z_2^-, ..., z_{k_-}^-\}$

1 Calcular la cantidad de grupos k_+ para los objetos de clase $y_i = +1$, $k_+ \longleftarrow \lceil \frac{km_1}{m} \rceil$;
2 Calcular la cantidad de grupos k_- para los objetos de clase $y_i = -1$, $k_- \longleftarrow k - k_+$;
3 Seleccionar al azar k_+ objetos $\mathbf{x}_p^+$ con etiqueta $y_i = +1$ de la muestra de entrenamiento S como centroides iniciales $\mathbf{c}_j^+$;
4 Crear k_+ grupos C_j^+ tal que $C_j^+ = \{\mathbf{c}_j^+\}\ \forall\ j \in \{1, 2, ..., k_+\}$;
5 Se inicializa un contador de iteraciones $t = 0$;
6 **repeat**
7 $\quad t = t + 1$;
8 $\quad$ Calcular y almacenar la distancia $d(\mathbf{x}_i^+, \mathbf{c}_j^+)$ entre cada objeto $\mathbf{x}_i^+ \in S$ con etiqueta $y_i = +1$ y cada uno de los k_+ centroides $\mathbf{c}_j^+$, $d(\mathbf{x}_i^+, \mathbf{c}_j^+) \longleftarrow ||\mathbf{x}_i^+ - \mathbf{c}_j^+||$;
9 $\quad$ Asignar cada objeto $\mathbf{x}_i^+ \in S$ con etiqueta $y_i = +1$ al grupo C_j^+ que contenga el centroide $\mathbf{c}_j^+$ más cercano, $C_j^+ \longleftarrow C_j^+ \bigcup \mathbf{x}_i^+ : \underset{j}{\operatorname{argmin}}\, d(\mathbf{x}_i^+, \mathbf{c}_j^+)$;
10 $\quad$ Calcular la nueva posición del centroide de cada grupo $\mathbf{c}_j^+ \longleftarrow \mathbf{x}^+ : \underset{\mathbf{x}^+ \in C_j^+}{\operatorname{argmin}} \frac{1}{|C_j^+|} \sum_{\mathbf{x}_i^+ \in C_j^+} d(\mathbf{x}^+, \mathbf{x}_i^+)$;
11 **until** *el conjunto de centroides* $\mathbf{c}_j^+$ *sea idéntico que el de la iteración anterior ó* $t = 100$;
12 Almacenar los centroides obtenidos $\mathbf{c}_j^+$ en el conjunto $\mathcal{R}^+$, $\mathcal{R}^+ = \{\mathbf{c}_1^+, \mathbf{c}_2^+, ..., \mathbf{c}_{k_+}^+\}$;
13 Calcular el número de objetos z_j^+ asignados a cada grupo $C_j^+\ \forall\ j \in \{1, 2, ..., k_+\}$, $z_j^+ \longleftarrow |C_j^+|$;
14 Almacenar los valores z_j^+ obtenidos en el paso anterior en el conjunto $\mathcal{Z}^+$, $\mathcal{Z}^+ = \{z_1^+, z_2^+, ..., z_{k_+}^+\}$;
15 Calcular el parámetro de dispersión σ_{+j}^2 de cada grupo $C_j^+\ \forall\ j \in \{1, 2, ..., k_+\}$, $\sigma_{+j}^2 \longleftarrow \frac{1}{z_j^+} \sum_{\mathbf{x}^+ \in C_j^+} d^2(\mathbf{x}^+, \mathbf{c}_j^+)$;
16 Almacenar los parámetro de dispersión σ_{+j}^2 obtenidos en el paso anterior en el conjunto $\mathcal{D}^+$, $\mathcal{D}^+ = \{\sigma_{+1}^2, \sigma_{+2}^2, ..., \sigma_{+k_+}^2\}$;
17 Seleccionar al azar k_- objetos $\mathbf{x}_q^-$ con etiqueta $y_i = -1$ de la muestra de entrenamiento S como centroides iniciales $\mathbf{c}_h^-$;
18 Crear k_- grupos C_h^- tal que $C_h^- = \{\mathbf{c}_h^-\}\ \forall\ h \in \{1, 2, ..., k_-\}$;
19 Se inicializa un contador de iteraciones $t = 0$;
20 **repeat**
21 $\quad t = t + 1$;
22 $\quad$ Calcular y almacenar la distancia $d(\mathbf{x}_e^-, \mathbf{c}_h^-)$ entre cada objeto $\mathbf{x}_e^-$ con etiqueta $y_e = -1$ y cada uno de los k_- centroides $\mathbf{c}_h^-$, $d(\mathbf{x}_e^-, \mathbf{c}_h^-) \longleftarrow ||\mathbf{x}_e^- - \mathbf{c}_h^-||$;
23 $\quad$ Asignar cada objeto $\mathbf{x}_e^-$ con etiqueta $y_e = -1$ al grupo C_h^- que contenga el centroide $\mathbf{c}_h^-$ más cercano $C_h^- \longleftarrow C_h^- \bigcup \mathbf{x}_e^- : \underset{h}{\operatorname{argmin}}\, d(\mathbf{x}_e^-, \mathbf{c}_h^-)$;
24 $\quad$ Calcular la nueva posición del centroide de cada grupo $\mathbf{c}_h^- \longleftarrow \mathbf{x}^- : \underset{\mathbf{x}^- \in C_h^-}{\operatorname{argmin}} \frac{1}{|C_h^-|} \sum_{\mathbf{x}_e^- \in C_h^-} d(\mathbf{x}^-, \mathbf{x}_e^-)$;
25 **until** *el conjunto de centroides* $\mathbf{c}_h^-$ *sea idéntico que el de la iteración anterior ó* $t = 100$;
26 Almacenar los centroides obtenidos $\mathbf{c}_{k_-}^-$ en el conjunto $\mathcal{R}^- = \{\mathbf{c}_1^-, \mathbf{c}_2^-, ..., \mathbf{c}_{k_-}^-\}$;
27 Calcular el número de objetos z_h^- asignados a cada grupo $C_h^-\ \forall\ h \in \{1, 2, ..., k_-\}$, $z_h^- \longleftarrow |C_h^-|$;
28 Almacenar los valores z_h^- obtenidos en el paso anterior en el conjunto $\mathcal{Z}^-$, $\mathcal{Z}^- = \{z_1^-, z_2^-, ..., z_{k_-}^-\}$;
29 Calcular el parámetro de dispersión σ_{-h}^2 de cada grupo $C_h^-\ \forall\ h \in \{1, 2, ..., k_-\}$, $\sigma_{-h}^2 \longleftarrow \frac{1}{|C_h^-|} \sum_{\mathbf{x}^- \in C_h^-} d^2(\mathbf{x}^-, \mathbf{c}_h^-)$;
30 Almacenar los parámetro de dispersión σ_{-h}^2 obtenidos en el paso anterior en el conjunto $\mathcal{D}^-$, $\mathcal{D}^- = \{\sigma_{-1}^2, \sigma_{-2}^2, ..., \sigma_{-k_-}^2\}$;

5.2. Máquinas de soporte vectorial con ponderación probabilística (PWSVM)

El trabajo propuesto pretende utilizar la información estadística encontrada por el algoritmo de agrupación dentro del esquema de máquinas de soporte vectorial con ponderación. Para realizar esto, se propone lo siguiente. Primero, se define que el nuevo conjunto de entrenamiento sea el conjunto $\mathcal{R}$ de tamaño k encontrado por el algoritmo de k-medoides. Luego, se propone definir el peso η_j^+ del centroide correspondiente $\mathbf{c}_j^+ \ \forall \ j \in \{1, 2, ..., k_+\}$ de manera que sea proporcional al número de objetos asignados al grupo C_j^+:

$$\eta_j^+ = \frac{z_j^+}{m} \tag{5.1}$$

Los pesos η_h^- de los centroides correspondientes $\mathbf{c}_h^- \ \forall \ h \in \{1, 2, ..., k_-\}$ se definen de manera idéntica:

$$\eta_h^- = \frac{z_h^-}{m} \tag{5.2}$$

Esta manera de definir los pesos es la misma que la propuesta en [83]. Cabe recordar que m es el número de objetos en el conjunto de entrenamiento original S.

Finalmente, la manera de integrar las varianzas σ^2 calculadas en el esquema de máquinas de soporte vectorial con ponderación es tratando de alterar la distancia euclidiana utilizada hasta ahora de tal manera que incluya la información de la varianza. La forma en que se realizó esto es utilizando la distancia de Mahalanobis [85]. Para un punto $\mathbf{x} \in \Re^N$, la distancia de Mahalanobis se define como:

$$d_M(\mathbf{x}, \mu) = \left[(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right]^{\frac{1}{2}} \tag{5.3}$$

donde μ es la media de una cierta muestra de referencia y Σ es la matriz de covarianza de la misma muestra de referencia. Para nuestro caso, los centroides $\mathbf{c}_j^+$ y $\mathbf{c}_h^-$ harían el papel de μ para cada grupo, y las correspondientes matrices de covarianza tendrían que calcularse. De aquí en adelante, se realizará el desarrollo de la propuesta utilizando un centroide genérico $\mathbf{c}_j$ que sin pérdida de generalidad puede ser del tipo $\mathbf{c}_j^+$ o del tipo $\mathbf{c}_h^-$.

Tomando en cuenta lo anterior, la distancia de Mahalanobis de cualquier centroide $\mathbf{c}_j$ al hiperplano de separación sería:

$$d_M(\mathbf{h}_j, \mathbf{c}_j) = \left[(\mathbf{h}_j - \mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{h}_j - \mathbf{c}_j)\right]^{\frac{1}{2}} \tag{5.4}$$

donde $\mathbf{h}_j$ es el punto más cercano a $\mathbf{c}_j$ que yace sobre el hiperplano de separación. Ahora, definimos un vector de distancia $\mathbf{d}_j$ tal que:

$$\mathbf{c}_j + \mathbf{d}_j = \mathbf{h}_j \tag{5.5}$$

es decir, $\mathbf{d}_j$ es un vector paralelo al vector normal $\mathbf{w}$ del hiperplano de separación, con una cierta magnitud δ_j. Ya que $\mathbf{d}_j$ es paralelo al vector $\mathbf{w}$, podemos expresarlo de la siguiente manera:

$$\mathbf{d}_j = \delta_j \frac{\mathbf{w}}{||\mathbf{w}||} \tag{5.6}$$

Despejando $\mathbf{d}_j$ de la ecuación (5.5) y sustituyéndola en (5.6):

$$\mathbf{d}_j = \mathbf{h}_j - \mathbf{c}_j = \delta_j \frac{\mathbf{w}}{||\mathbf{w}||} \tag{5.7}$$

Sustituyendo la ecuación (5.7) en la ecuación (5.4):

$$d_M(\mathbf{h}_j, \mathbf{c}_j) = \left[\left(\delta_j \frac{\mathbf{w}}{||\mathbf{w}||} \right)^T \Sigma_j^{-1} \left(\delta_j \frac{\mathbf{w}}{||\mathbf{w}||} \right) \right]^{\frac{1}{2}} = \left[\frac{\delta_j^2}{||\mathbf{w}||^2} \mathbf{w}^T \Sigma_j^{-1} \mathbf{w} \right]^{\frac{1}{2}} = \frac{\delta_j}{||\mathbf{w}||} \left[\mathbf{w}^T \Sigma_j^{-1} \mathbf{w} \right]^{\frac{1}{2}} \tag{5.8}$$

Además, como se mencionó en el capítulo 2, el valor de δ_j se puede calcular con la expresión (2.2):

$$\delta_j = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||} \tag{5.9}$$

Sustituyendo la ecuación (5.9) en la ecuación (5.8):

$$d_M(\mathbf{h}_j, \mathbf{c}_j) = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\mathbf{w}^T \Sigma_j^{-1} \mathbf{w} \right]^{\frac{1}{2}} \tag{5.10}$$

Analizaremos los posibles esquemas resultantes dependiendo de la forma que tengan las matrices de covarianza Σ_j. Se denota como $\mathbf{I}$ a la matriz identidad.

5.2.1. Caso 1. $\Sigma_j = \mathbf{I}$

En el caso cuando $\Sigma_j = \mathbf{I} \; \forall \; j \in \{1, 2, ..., k\}$, sucede lo siguiente en la expresión (5.10):

$$\begin{aligned} d_M(\mathbf{h}_j, \mathbf{c}_j) &= \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\mathbf{w}^T \Sigma_j^{-1} \mathbf{w} \right]^{\frac{1}{2}} = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\mathbf{w}^T \mathbf{I}^{-1} \mathbf{w} \right]^{\frac{1}{2}} = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\mathbf{w}^T \mathbf{w} \right]^{\frac{1}{2}} \\ &= \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[||\mathbf{w}||^2 \right]^{\frac{1}{2}} = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} ||\mathbf{w}|| = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||} \end{aligned} \tag{5.11}$$

Es fácil observar que esta ecuación es la misma ecuación que (2.2), debido a que, cuando las matrices de covarianza Σ_j son iguales a la matriz identidad $\mathbf{I}$, la distancia de Mahalanobis $d_M(\mathbf{x},\mu) = \left[(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right]^{\frac{1}{2}}$ simplemente se vuelve la distancia euclidiana $d(\mathbf{x},\mu) = ||\mathbf{x}-\mu||$. Si continuáramos con el desarrollo en este caso, llegaríamos al esquema típico de SVM, con el problema de optimización primal.

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\ &\text{s. t. } y_i(\mathbf{w}\cdot\mathbf{c}_j + b) \geq 1,\ \forall\ j \in \{1,2,...,k\} \end{aligned} \tag{5.12}$$

El problema es que este esquema no permite incorporar las varianzas σ_j^2, por lo que exploraremos el siguiente caso, cuando $\Sigma_j = \sigma_j^2\mathbf{I}$.

5.2.2. Caso 2. $\Sigma_j = \sigma_j^2\mathbf{I}$

En el caso cuando $\Sigma_j = \sigma_j^2\mathbf{I}$, sucede lo siguiente en la expresión (5.10):

$$\begin{aligned} d_M(\mathbf{h}_j,\mathbf{c}_j) &= \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{||\mathbf{w}||^2}\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}} = \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{||\mathbf{w}||^2}\left[\mathbf{w}^T\frac{1}{\sigma_j^2}\mathbf{I}\mathbf{w}\right]^{\frac{1}{2}} = \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{||\mathbf{w}||^2}\left[\frac{1}{\sigma_j^2}\mathbf{w}^T\mathbf{w}\right]^{\frac{1}{2}} \\ &= \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{\sigma_j||\mathbf{w}||^2}\left[||\mathbf{w}||^2\right]^{\frac{1}{2}} = \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{\sigma_j||\mathbf{w}||^2}||\mathbf{w}|| = \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{\sigma_j||\mathbf{w}||} \end{aligned} \tag{5.13}$$

En este caso, aunque la ecuación (5.13) es muy parecida a la ecuación (2.2), la ecuación (5.13) tiene en el denominador el término σ_j, que es una medida de la dispersión del grupo C_j. Al igual que en el capítulo 2, es importante observar que $\mathbf{w}$ y b se pueden escalar de tal manera que $\min_{\mathbf{c}_j\in\mathcal{R}} \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{\sigma_j} = 1$. Por lo tanto, en este caso, el margen ρ_M que es la distancia de Mahalanobis del hiperplano de separación a los objetos mas cercanos se puede calcular con la siguiente expresión

$$\rho_M = \min_{\mathbf{c}_j\in\mathcal{R}} \frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{\sigma_j||\mathbf{w}||} = \frac{1}{||\mathbf{w}||} \tag{5.14}$$

Al igual que en el planteamiento original de SVM, lo que se busca es maximizar el margen $\rho_M = \frac{1}{||\mathbf{w}||}$, que es equivalente a minimizar $||\mathbf{w}||$ o $\frac{1}{2}||\mathbf{w}||^2$. Por lo tanto, se propone que el problema de optimización primal a resolver sea el siguiente:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{k}\eta_j\xi_j \\ &\text{s. t. } \tfrac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j+b) \geq 1-\xi_j,\ \xi_j \geq 0\ \forall\ j \in \{1,2,...,k\} \end{aligned} \tag{5.15}$$

El vector $\boldsymbol{\eta} \in \Re_+^k$ es el vector de pesos compuesto de k pesos $0 \leq \eta_j \leq 1, j \in \{1, 2, ..., k\}$. La manera en que se calculan estos pesos es explicada al principio de esta sección.

Como en los demás esquemas de SVM, el problema (5.15) es un problema de optimización convexa ya que las funciones de restricción son afines y la función objetivo es convexa, por lo que, presenta la propiedad de tener una solución única $(\mathbf{w}^*, b^*)$. Además, se pueden utilizar las condiciones de Karush-Kuhn-Tucker (KKT) para resolverlo. Siendo los multiplicadores de Lagrange $\alpha_j \geq 0, j \in \{1, 2, ..., k\}$ asociados a las primeras k restricciones, con su respectivo vector $\boldsymbol{\alpha} \in \Re_+^k$ y los multiplicadores $\beta_j \geq 0, j \in \{1, 2, ..., k\}$ asociados a las restricciones de no negatividad de las variables de holgura, con su respectivo vector $\boldsymbol{\beta} \in \Re_+^k$, el Lagrangiano queda expresado de la siguiente manera:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{k}\eta_j\xi_j - \sum_{j=1}^{k}\alpha_j\left[\frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) - 1 + \xi_i\right] - \sum_{j=1}^{k}\beta_j\xi_j \tag{5.16}$$

Las condiciones de Karush-Kuhn-Tucker se obtienen igualando a cero el gradiente del Lagrangiano con respecto a las variables $\mathbf{w}$, b y $\xi_j \ \forall \ j \in \{1, 2, ..., k\}$ junto con las condiciones de holgura complementaria:

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j\mathbf{c}_j = 0 \qquad \Longrightarrow \qquad \mathbf{w} = \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j\mathbf{c}_j \tag{5.17}$$

$$\nabla_b\mathcal{L} = -\sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j = 0 \qquad \Longrightarrow \qquad \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j = 0 \tag{5.18}$$

$$\forall j, \nabla_{\xi_j}\mathcal{L} = C\eta_j - \alpha_j - \beta_j = 0 \qquad \Longrightarrow \qquad C\eta_j = \alpha_j + \beta_j \tag{5.19}$$

$$\forall j, \alpha_j\left[\frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) - 1 + \xi_j\right] = 0 \qquad \Longrightarrow \qquad \alpha_j = 0 \vee \frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) = 1 - \xi_j \tag{5.20}$$

$$\forall j, \beta_j\xi_j = 0 \qquad \Longrightarrow \qquad \beta_j = 0 \vee \xi_j = 0 \tag{5.21}$$

Al igual que en los casos anteriores, se observa de la ecuación (5.17) que el vector $\mathbf{w}$ es una combinación lineal de los objetos en la muestra de entrenamiento $\mathcal{R}$. Un objeto $\mathbf{c}_j$ aparecerá en esta combinación lineal si y sólo si $\alpha_j \neq 0$. Estos objetos serán los vectores de soporte. Al igual que en el esquema de SVM con ponderación, existen dos tipos de vectores de soporte. Debido a las condiciones de holgura complementaria (5.20), si $\alpha_j \neq 0$, entonces $\frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) = 1 - \xi_j$. Si $\xi_j = 0$, entonces $\frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) = 1$ y el objeto $\mathbf{c}_j$ yace en algún hiperplano marginal. De lo contrario, $\xi_j \neq 0$, lo que implica que el objeto $\mathbf{c}_j$ es una anomalía. En este caso, dado que $\xi_j \neq 0$, (5.21) implica que $\beta_j = 0$ y (5.19) implica que $\alpha_j = C\eta_j$.

Como en los demás esquemas de SVM, se observa que los objetos que no son anomalías ni yacen en los hiperplanos marginales no afectan a la solución retornada, por lo que, en su ausencia, la solución al problema se mantendría sin cambios. De manera similar, aunque el vector $\mathbf{w}$ retornado es único, los vectores de soporte pueden no serlo.

Para encontrar el problema de optimización dual, se aplica la propiedad distributiva del producto punto y de la multiplicación a la ecuación del Lagrangiano:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{k}\eta_j\xi_j - \sum_{j=1}^{k}\alpha_j\left[\frac{y_j}{\sigma_j}(\mathbf{w}\cdot\mathbf{c}_j + b) - 1 + \xi_i\right] - \sum_{j=1}^{k}\beta_j\xi_j \\
&= \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{k}\eta_j\xi_j - \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j(\mathbf{w}\cdot\mathbf{c}_j) - \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j b + \sum_{j=1}^{k}\alpha_j - \sum_{j=1}^{k}\alpha_j\xi_j - \sum_{j=1}^{k}\beta_j\xi_j \\
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j\mathbf{c}_j - b\sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j + \sum_{j=1}^{k}\alpha_j + \sum_{j=1}^{k}(C\eta_j - \alpha_j - \beta_j)\xi_j
\end{aligned} \tag{5.22}$$

Al sustituir las ecuaciones (5.17),(5.18) y (5.19) en la ecuación (5.22) se obtiene lo siguiente:

$$\begin{aligned}
&= \frac{1}{2}||\mathbf{w}||^2 - \mathbf{w}\cdot\mathbf{w} + \sum_{j=1}^{k}\alpha_j \\
&= \frac{1}{2}||\mathbf{w}||^2 - ||\mathbf{w}||^2 + \sum_{j=1}^{k}\alpha_j \\
&= \sum_{j=1}^{k}\alpha_j - \frac{1}{2}||\mathbf{w}||^2 \\
&= \sum_{j=1}^{k}\alpha_j - \frac{1}{2}||\sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j\mathbf{c}_j||^2 \\
&= \sum_{j=1}^{k}\alpha_j - \frac{1}{2}\sum_{i,j=1}^{k}\alpha_i\alpha_j\frac{y_i}{\sigma_i}\frac{y_j}{\sigma_j}(\mathbf{c}_i\cdot\mathbf{c}_j)
\end{aligned} \tag{5.23}$$

Se puede observar que la función objetivo (5.23) del problema de optimización dual es muy parecida a los demás esquemas. Sin embargo, en este esquema el problema de optimización dual incluye tanto en la función objetivo como en las restricciones a la raiz cuadrada de las varianzas σ_j^2, integrando la información acerca de las dispersión de los grupos C_j en el problema de optimización. Dada la ecuación (5.19), si $\beta_j \geq 0$, entonces $\alpha_j \leq C\eta_j \ \forall\ j \in \{1, 2, ..., k\}$. Por lo tanto, el problema de optimización dual en el esquema propuesto de PWSVM queda definido de la siguiente manera:

$$\begin{aligned}
&\max_{\boldsymbol{\alpha}} \sum_{j=1}^{k}\alpha_j - \frac{1}{2}\sum_{i,j=1}^{k}\alpha_i\alpha_j\frac{y_i}{\sigma_i}\frac{y_j}{\sigma_j}(\mathbf{c}_i\cdot\mathbf{c}_j) \\
&\text{s. t. } \sum_{j=1}^{k}\frac{y_j}{\sigma_j}\alpha_j = 0,\ 0 \leq \alpha_j \leq C\eta_j \ \forall\ j \in \{1, 2, ..., k\}
\end{aligned} \tag{5.24}$$

Finalmente, una vez que se resuelve el problema de optimización y se encuentran los valores de $\mathbf{w}^*$ y b^*, el modelo de clasificación para un nuevo objeto $\mathbf{x}$ por clasificar es la función:

$$h(\mathbf{x}) = sgn(\mathbf{w}^* \cdot \mathbf{x} + b^*) \tag{5.25}$$

Este es el esquema propuesto de PWSVM, y se probará de manera experimental utilizando conjuntos de datos fabricados de manera artificial (llamados ejemplos de juguete o toy examples en inglés) y conjuntos de datos públicos. Antes de presentar el capítulo de resultados experimentales, a continuación se extiende la idea de PWSVM en dos casos mas. Estos casos hacen menos suposiciones acerca de las matrices de covarianza Σ_j.

5.2.3. Caso 3. Σ_j es una matriz diagonal

Se supone el siguiente caso:

$$\Sigma_j = \begin{bmatrix} \sigma_{j1}^2 & 0 & ... & 0 \\ 0 & \sigma_{j2}^2 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \sigma_{jN}^2 \end{bmatrix}$$

En este caso la expresión (5.10) resulta en lo siguiente:

$$d_M(\mathbf{h}_j, \mathbf{c}_j) = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\mathbf{w}^T \Sigma_j^{-1} \mathbf{w}\right]^{\frac{1}{2}} = \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\sum_{i=1}^{N} \frac{w_i^2}{\sigma_{ji}^2}\right]^{\frac{1}{2}} \tag{5.26}$$

donde $\mathbf{w} = (w_1, w_2, ..., w_N)$. Al igual que en los demás casos, $\mathbf{w}$ y b se pueden escalar de tal manera que $\min\limits_{\mathbf{c}_j \in \mathcal{R}} |\mathbf{w} \cdot \mathbf{c}_j + b| \left[\sum\limits_{i=1}^{N} \frac{w_i^2}{\sigma_{ji}^2}\right]^{\frac{1}{2}} = 1$. Por lo tanto, el margen ρ_M que es la distancia de Mahalanobis del hiperplano de separación a los objetos mas cercanos se puede calcular con la siguiente expresión

$$\rho_M = \min_{\mathbf{c}_j \in \mathcal{R}} \frac{|\mathbf{w} \cdot \mathbf{c}_j + b|}{||\mathbf{w}||^2} \left[\sum_{i=1}^{N} \frac{w_i^2}{\sigma_{ji}^2}\right]^{\frac{1}{2}} = \frac{1}{||\mathbf{w}||^2} \tag{5.27}$$

Al igual que en el planteamiento original de SVM, lo que se busca es maximizar el margen $\rho_M = \frac{1}{||\mathbf{w}||^2}$, que es equivalente a minimizar $||\mathbf{w}||^2$ o $\frac{1}{2}||\mathbf{w}||^2$. Por lo tanto, se propone que el problema de optimización primal a resolver sea el siguiente:

$$\begin{aligned} &\min_{\mathbf{w}, b, \boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{k} \eta_j \xi_j \\ &\text{s. t. } y_j(\mathbf{w} \cdot \mathbf{c}_j + b) \left[\sum_{i=1}^{N} \frac{w_i^2}{\sigma_{ji}^2}\right]^{\frac{1}{2}} \geq 1 - \xi_j, \, \xi_j \geq 0 \; \forall \; j \in \{1, 2, ..., k\} \end{aligned} \tag{5.28}$$

La cuestión que surge en este caso es que, a diferencia de los demás casos presentados, es difícil determinar si el problema de optimización (5.28) es convexo o no, debido a la forma de la restricción $y_j(\mathbf{w}\cdot\mathbf{c}_j+b)\left[\sum_{i=1}^{N}\frac{w_i^2}{\sigma_{ji}^2}\right]^{\frac{1}{2}} \geq 1-\xi_j$. Aunque se ha realizado la aportación de plantear el problema de optimización (5.28), queda como problema abierto determinar la convexidad del mismo.

5.2.4. Caso 4. Σ_j es una matriz de covarianza

En este caso, no se hace ninguna suposición, y solamente se sabe que las matrices Σ_j son matrices de covarianza.

En este caso la expresión (5.10) queda sin alterarse, quedando la distancia simple y sencillamente como $d_M(\mathbf{h}_j,\mathbf{c}_j)=\frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{||\mathbf{w}||^2}\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}}$

Sin embargo, se puede observar que, al igual que en los demás casos, $\mathbf{w}$ y b se pueden escalar de tal manera que $\min\limits_{\mathbf{c}_j\in\mathcal{R}}|\mathbf{w}\cdot\mathbf{c}_j+b|\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}}=1$. Por lo tanto, el margen ρ_M que es la distancia de Mahalanobis del hiperplano de separación a los objetos mas cercanos se puede calcular con la siguiente expresión

$$\rho_M=\min_{\mathbf{c}_j\in\mathcal{R}}\frac{|\mathbf{w}\cdot\mathbf{c}_j+b|}{||\mathbf{w}||^2}\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}}=\frac{1}{||\mathbf{w}||^2} \tag{5.29}$$

Como en los demás planteamientos, se busca maximizar el margen $\rho_M=\frac{1}{||\mathbf{w}||^2}$, que equivale a minimizar $||\mathbf{w}||^2$ o $\frac{1}{2}||\mathbf{w}||^2$. Por lo tanto, se propone que el problema de optimización primal a resolver sea el siguiente:

$$\begin{aligned}&\min_{\mathbf{w},b,\boldsymbol{\xi}}\frac{1}{2}||\mathbf{w}||^2+C\sum_{j=1}^{k}\eta_j\xi_j \\ &\text{s. t. } y_j(\mathbf{w}\cdot\mathbf{c}_j+b)\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}}\geq 1-\xi_j,\ \xi_j\geq 0\ \forall\ j\in\{1,2,...,k\}\end{aligned} \tag{5.30}$$

Al igual que en el caso anterior, surge la cuestión de que es difícil determinar si el problema de optimización (5.30) es convexo o no, debido a la forma de la restricción $y_j(\mathbf{w}\cdot\mathbf{c}_j+b)\left[\mathbf{w}^T\Sigma_j^{-1}\mathbf{w}\right]^{\frac{1}{2}}\geq 1-\xi_j$. De manera similar al caso anterior, queda como problema abierto determinar la convexidad del problema (5.30).

Capítulo 6

Resultados experimentales

Este capítulo presenta los resultados experimentales obtenidos al evaluar el desempeño del método propuesto. Para realizar esta evaluación, se utilizaron cuatro conjuntos de datos. Uno de estos conjuntos fue creado para esta tesis en específico utilizando una distribución normal en dos dimensiones. Los otros conjuntos fueron tomados del repositorio de la Universidad de California en Irvine (UCI) [86]. Estos conjuntos son comúnmente utilizados para evaluar el desempeño de algoritmos de clasificación.

El método propuesto fue comparado con el método tradicional de SVM en el caso no separable, como el presentado en la sección 2 del capítulo 2 de este documento, y con una reducción de datos realizada al azar a la que llamaremos RSVM por sus siglas en inglés (Random SVM). RSVM funciona escogiendo k objetos al azar del conjunto original para realizar el entrenamiento de SVM con este subconjunto. Todos los experimentos fueron realizados en una computadora con las siguientes características:

- Procesador: AMD A6-4400M APU 2.70 GHz
- Memoria RAM: 4.00 GB

Todos los algoritmos y métodos fueron implementados en MATLAB. Para resolver los problemas de optimización que derivan del método de SVM, se utilizó CVX, una paquetería para especificar y resolver problemas de optimización convexa [87].

6.1. Diseño experimental e interpretación de los resultados

Cada experimento consiste en realizar 50 ejecuciones independientes. Cada ejecución consta de escoger un conjunto de datos, un método (SVM, RSVM ó PWSVM) y, específicamente para los métodos RSVM y PWSVM, un valor de k que determina el tamaño del conjunto reducido. Con las especificaciones anteriores, se entrena el método seleccionado con el conjunto de datos seleccionado y luego se evalua con el correspondiente conjuto de datos de prueba. Para el método de PWSVM, el desempeño se evaluó en los siguientes casos:

- PWSVM1: Utilizando sólamente los centroides devueltos por el algoritmo de k-medoides como entrada al método de SVM.
- PWSVM2: Utilizando los centroides devueltos por el algoritmo de k-medoides, la cantidad de objetos asignados a cada centroide y la varianza calculada para cada grupo en el esquema de PWSVM.

Una vez realizadas las 50 repeticiones para un determinado conjunto de datos con un determinado método y sus parámetros, se obtienen 50 soluciones, a partir de las cuales se calcularon la media y la desviación estandar. Las mediciones realizadas fueron:

- Porcentaje de clasificación correcta [%]
- Número de vectores de soporte
- Tiempo que tardó en realizarse el agrupamiento [s]
- Tiempo que tardó el entrenamiento del respectivo esquema de SVM [s]
- Tiempo total que tardó el método por completo (agrupamiento + entrenamiento de SVM) [s]

6.2. Conjunto de datos artificial

El primer conjunto de datos con el que se probó el método propuesto fue construido a partir de nueve distribuciones normales distintas en dos dimensiones. Cada una de estas distribuciones esta determinada por su media μ y su matriz de covarianza Σ. Cada una de ellas genero un grupo de q objetos centrados alrededor de μ y dispersos acorde a los valores en la matriz Σ. De los nueve grupos, cuatro de ellos se conforman de objetos con etiqueta $y_i = +1$ y los otro cinco grupos se conforman de objetos con etiqueta $y_i = -1$. La figura 6.1 muestra el conjunto de entrenamiento y la figura 6.2 muestra el conjunto de prueba. Ambos conjuntos fueron creados con las mismas nueve distribuciones normales. Los objetos con etiqueta $y_i = +1$ son representados con cuadros verdes y los objetos con etiqueta $y_i = -1$ son representados con círculos amarillos. Se observa que el conjunto de entrenamiento no es linealmente separable. Todos los grupos tienen una matriz de covarianza de la forma:

$$\Sigma_i = \begin{bmatrix} \sigma_i^2 & 0 \\ 0 & \sigma_i^2 \end{bmatrix}$$

con σ_i distinto para cada grupo. La información acerca de los grupos generados que conforman los conjuntos de entrenamiento y de prueba se pueden observar en los cuadros 6.1 y 6.2.

El conjunto de entrenamiento consta de 5000 objetos de dimensión 2, 2220 con etiqueta $y_i = +1$ y 2780 con etiqueta $y_i = -1$. El conjunto de prueba tiene 1000 objetos de dimensión 2, 450 con etiqueta $y_i = +1$ y 550 con etiqueta $y_i = -1$.

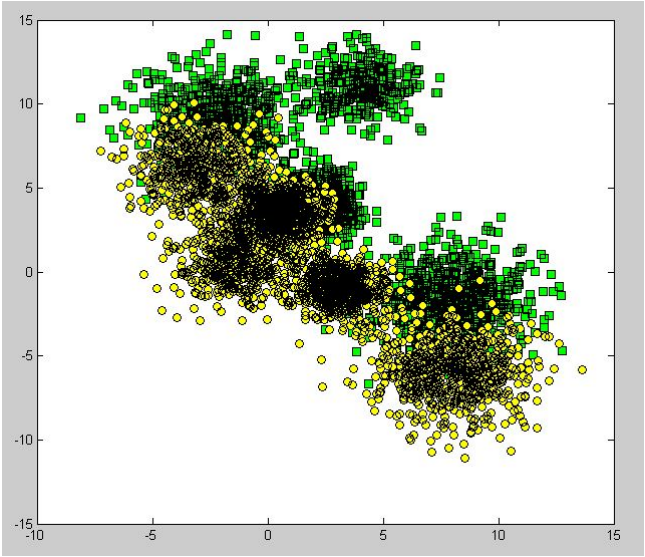

Figura 6.1: Conjunto de datos de entrenamiento para clasificación creado de manera artificial

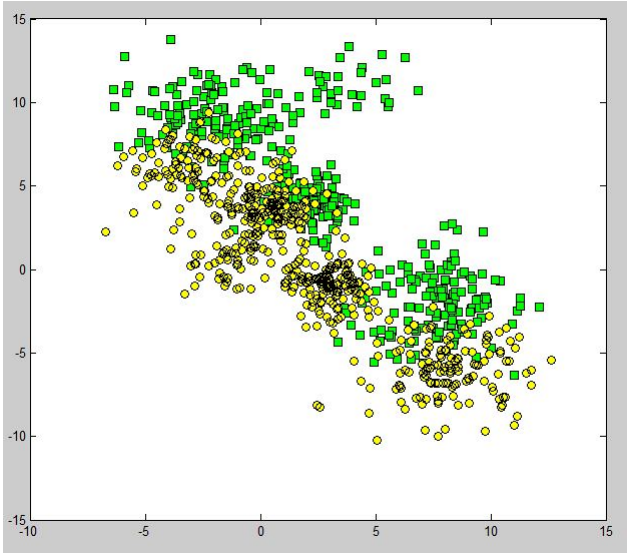

Figura 6.2: Conjunto de datos de prueba creado de manera artificial

Cuadro 6.1: Información acerca de los grupos que conforman al conjunto de entrenamiento artificial

Grupo	Etiqueta	Número de objetos en el grupo	Media	σ_i^2
1	+1	270	[4,11]	1.5
2	+1	550	[-2,9]	3
3	+1	700	[2,4]	1
4	+1	700	[8,-2]	3
5	-1	480	[-3,6]	2
6	-1	700	[.5,3.5]	1
7	-1	400	[-1.5,.5]	1.5
8	-1	600	[3,-1]	1
9	-1	600	[8,-6]	3

Cuadro 6.2: Información acerca de los grupos que conforman al conjunto de prueba artificial

Grupo	Etiqueta	Número de objetos en el grupo	Media	σ_i^2
1	+1	30	[4,11]	1.5
2	+1	140	[-2,9]	3
3	+1	140	[2,4]	1
4	+1	140	[8,-2]	3
5	-1	95	[-3,6]	2
6	-1	135	[.5,3.5]	1
7	-1	50	[-1.5,.5]	1.5
8	-1	135	[3,-1]	1
9	-1	135	[8,-6]	3

6.2.1. Resultados utilizando el conjunto de datos artificial

En el cuadro 6.3 se presentan los resultados obtenidos con los distintos métodos para este conjunto de datos. La primera columna indica el método utilizado. La segunda columna indica el valor del parámetro C. La tercera columna indica el promedio del porcentaje de acierto en la clasificación en el conjunto de prueba. La cuarta columna indica el máximo porcentaje de acierto alcanzado en la clasificación en el conjunto de prueba. La quinta columna indica el promedio de la cantidad de vectores de soporte del modelo. Las columnas seis, siete y ocho indican el promedio del tiempo en segundos que tardo cada método en realizar el agrupamiento, el entrenamiento y el tiempo total, respectivamente. Los valores del parámetro C para los distintos métodos se calcularon mediante validación cruzada (n-cross validation en inglés) con $n = 5$.

Se puede observar que el método propuesto logra reducir en promedio el tiempo de entrenamiento de las SVM hasta aproximadamente la mitad perdiendo en promedio menos de un punto porcentual en el acierto de clasificación para casi todos los valores de k, excepto cuando $k = 500$. Además, para todos los valores de k, existieron ejecuciones del método propuesto en las que se mejoró el porcentaje de acierto de clasificación, como se observa en la cuarta columna del cuadro 6.3, alcanzado el máximo porcentaje (81.7 % con $k = 100$) por sobre los demás métodos probados. Sin embargo, el tiempo promedio que tarda en realizarse la agrupación es bastante mayor que el tiempo de entrenamiento original, como puede observarse en el cuadro 6.3.

Debido a la restricción $0 \leq \alpha_i \leq C\eta_i$ del problema de optimización dual (5.24), en ocasiones es difícil reconocer que objetos son vectores de soporte, ya que los valores α_i devueltos por CVX

Cuadro 6.3: Resultados de la evaluación con el conjunto de datos artificial

Método	C	Prom. Clasif. [%]	Max. Clasif. [%]	# VS	$T_{AGRUPAR}$[s]	$T_{ENTRENAMIENTO}$[s]	T_{TOTAL}[s]
SVM	5	81.1±0	81.1	2020±0	0	1.26±0.01	1.26±0.01
RSVM(k=9)	5	74.99±6.08	81.6	3.72±1.31	.0012±.000319	0.25±0.02	0.25±0.02
RSVM(k=25)	20	78.49±2.79	81.2	10.04±3.94	.0012±.000286	0.29±0.03	0.29±0.03
RSVM(k=100)	50	80.48±0.79	81.6	38.78±7	.0012±.000096	0.48±0.05	0.48±0.05
RSVM(k=250)	50	80.81±0.54	81.6	99.78±12.14	.0012±.000134	0.7±0.05	0.7±0.05
RSVM(k=500)	1	80.93±0.44	81.6	198.28±14.83	.0012±.000114	2.26±0.17	2.26±0.17
PWSVM1(k=9)	1	79.7±1.22	81.6	2.4±0.52	32.41±9.76	0.25±0.02	32.67±9.77
PWSVM1(k=25)	1	80.75±0.44	81.3	8.2±1.62	19.85±3.78	0.3±0.03	20.16±3.78
PWSVM1(k=100)	1	80.76±0.44	81.6	39.8±4.32	10.35±1.66	0.45±0.03	10.8±1.68
PWSVM1(k=250)	50	80.85±0.56	81.6	103.3±7.23	15.89±1.25	0.83±0.06	16.72±1.23
PWSVM1(k=500)	5	81.05±0.17	81.2	206.4±9.51	25.27±4.42	2.09±0.14	27.36±4.42
PWSVM2(k=9)	50	80.32±0.8	81.4	sin medición	43.91±16.72	0.3±0.01	44.2±16.72
PWSVM2(k=25)	50	80.87±0.68	81.6	6.4±0.84	18.45±3.54	0.33±0.06	18.79±3.52
PWSVM2(k=100)	20	80.72±0.59	81.7	38.5±1.9	11.44±1.19	0.52±0.04	11.96±1.2
PWSVM2(k=250)	20	80.87±0.38	81.3	87.9±3.25	16.18±1.73	0.85±0.06	17.03±1.76
PWSVM2(k=500)	20	80.9±0.48	81.5	177±9.32	24.99±2.02	2.56±0.12	27.55±2.1

para los objetos que no son vectores de soporte no siempre son exactamente cero, y escoger un umbral inferior para reconocer que valores considerar como distintos a cero se complica debido a la condición $\alpha_i \leq C\eta_i$. Debido a lo anterior hay casillas del cuadro 6.3 que no cuentan con mediciones.

Las figuras 6.3 y 6.4 muestran el comportamiento de los diferentes métodos aplicados a este conjunto de datos.

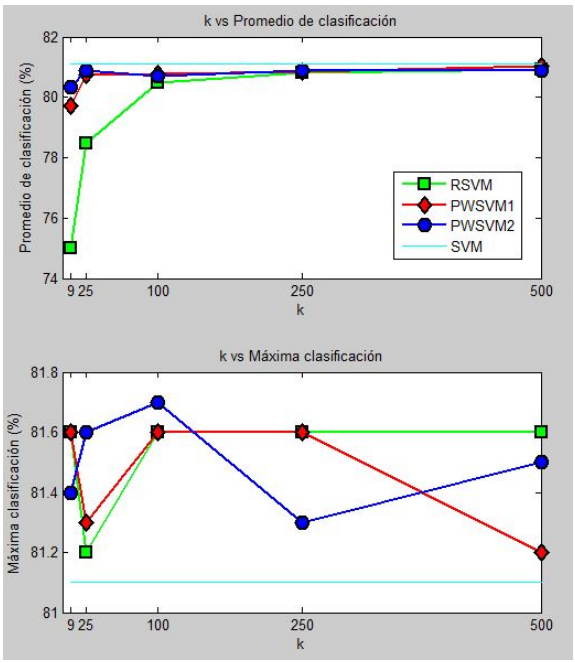

Figura 6.3: Gráficas de los porcentajes de clasificación con respecto al valor de k (conjunto de datos artificial)

En todos los gráficos se presenta con el color cian el comportamiento del método tradicional de las SVM utilizando el conjunto de datos completo. El comportamiento del método aleatorio se presenta con el color verde. Los métodos presentados al inicio de este capítulo como PWSVM1

y PWSVM2 son mostrados con los colores rojo y azul, respectivamente. La figura 6.3 muestra el comportamiento del promedio de clasificación y de la máxima clasificación reportada por éstos métodos al variar el valor de k.

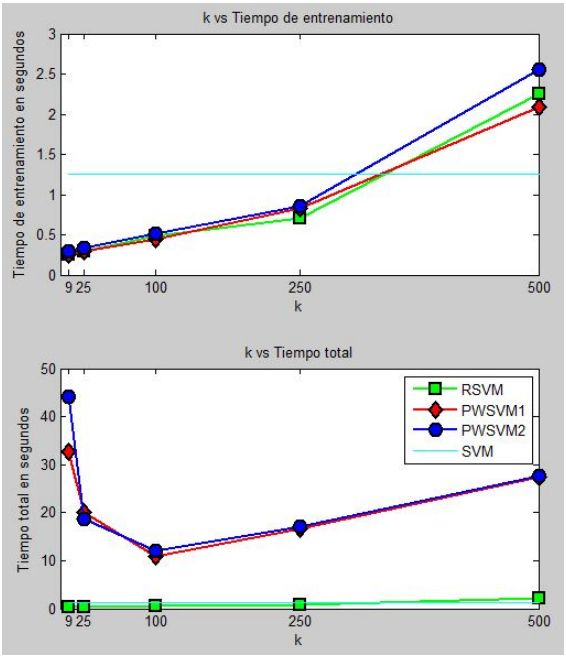

Figura 6.4: Gráficas de los tiempos de ejecución con respecto al valor de k (conjunto de datos artificial)

De manera similar, la figura 6.4 muestra el comportamiento del tiempo de entrenamiento y del tiempo total utilizado por cada método al variar el valor de k.

6.3. Conjuntos de datos públicos

6.3.1. Conjunto de datos de cáncer de pecho de Wisconsin

El método propuesto se probó con uno de los conjuntos más utilizados para poner a prueba métodos de clasificación: el conjunto de datos de cáncer de pecho de Wisconsin (Breast cancer Wisconsin data set). Este conjunto fue tomado del repositorio de la Universidad de California en Irvine (UCI) [86].

El conjunto consta de 569 objetos de dimensión 30, 357 de ellos clasificados como benignos, a los cuales se les asignó la etiqueta $y_i = +1$ y 212 clasificados como malignos, a los cuales se les asignó la etiqueta $y_i = -1$. El conjunto se dividió utilizando una proporción 80/20. El 80 % de los datos (455) se utilizaron como conjunto de entrenamiento, mientras el 20 % restante (114) se utilizaron como conjunto de prueba.

6.3.2. Resultados utilizando el conjunto de datos de cáncer de pecho de Wisconsin

En el cuadro 6.4 se presentan los resultados obtenidos con los distintos métodos para este conjunto de datos. La disposición del cuadro es la misma que la de la cuadro 6.3. Al igual que con el conjunto de datos artificiales, los valores del parámetro C para los distintos métodos se calcularon mediante validación cruzada (n-cross validation en inglés) con $n = 5$.

Cuadro 6.4: Resultados de la evaluación con el conjunto de datos de cáncer de pecho de Wisconsin

Método	C	Ac. Clasif. [%]	Max. Clasif. [%]	# VS	$T_{AGRUPAR}$[s]	$T_{ENTRENAMIENTO}$[s]	T_{TOTAL}[s]
SVM	5	97.36±0	97.36	32±0	0	3.53±0.04	3.53±0.04
RSVM(k=10)	5	87.05±3.58	92.98	2.54±0.81	.000428±.00009	0.27±0.01	0.27±0.01
RSVM(k=25)	20	88.82±3.7	95.1	3.42±1.21	.000457±.000086	0.32±0.02	0.32±0.02
RSVM(k=50)	1	90.33±3.1	94.74	5.02±1.7	.000481±.000122	0.34±0.02	0.34±0.02
RSVM(k=100)	1	92.4±2.17	95.49	8.58±3.34	.000524±.000105	0.51±0.03	0.51±0.03
PWSVM1(k=10)	1	89.47±1.6	92.1	2.6±0.7	0.23±0.06	0.28±0.01	0.51±0.07
PWSVM1(k=25)	5	91.23±1.94	92.98	3±0.94	0.18±0.04	0.33±0.02	0.51±0.04
PWSVM1(k=50)	50	92.02±2.53	94.74	5±1.33	.21±.03	0.35±0.02	0.57±0.04
PWSVM1(k=100)	100	92.8±2.44	95.59	6.5±0.85	0.3±0.03	0.58±0.02	0.88±0.04
PWSVM2(k=10)	20	88.25±2.16	92.1	sin medición	0.27±0.08	0.26±0.01	0.52±0.08
PWSVM2(k=25)	50	90.35±2.95	92.1	sin medición	0.2±0.06	0.28±0.03	0.48±0.07
PWSVM2(k=50)	50	92.19±0.77	93.85	7.1±2.08	.21±.03	0.3±0.02	0.52±0.03
PWSVM2(k=100)	100	93.77±0.97	95.61	11.5±1.78	0.3±0.03	0.49±0.06	0.79±0.08

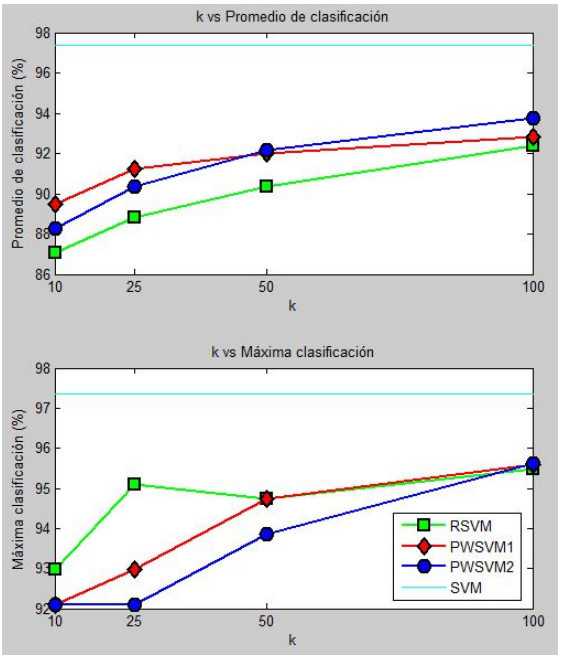

Figura 6.5: Gráficas de los porcentajes de clasificación con respecto al valor de k (conjunto de datos de cáncer de pecho)

Se puede observar que el método propuesto logra reducir de manera drástica el tiempo promedio de entrenamiento de las SVM para todos los valores de k, con la desventaja de perder hasta nueve puntos porcentuales en el acierto de clasificación cuando $k = 10$. Sin embargo, para el valor de $k = 100$, el promedio del porcentaje de acierto de clasificación es de 93.77 %, bastante cercano al 97.36 % alcanzado al utilizar el conjunto de entrenamiento por completo. Además, con este valor de

k, se llegó a alcanzar el segundo mejor porcentaje máximo de acierto de clasificación, con 95.61 % de acierto, sólo por debajo del porcentaje alcanzado utilizando todo el conjunto de entrenamiento. El tiempo total de ejecución de PWSVM para este valor de k promedia en 0.79 s, mucho menor al tiempo de entrenamiento promedio al entrenar SVM con el conjunto por completo, que es de 3,53 s.

Las figuras 6.5 y 6.6 muestran el comportamiento de los diferentes métodos aplicados a este conjunto de datos.

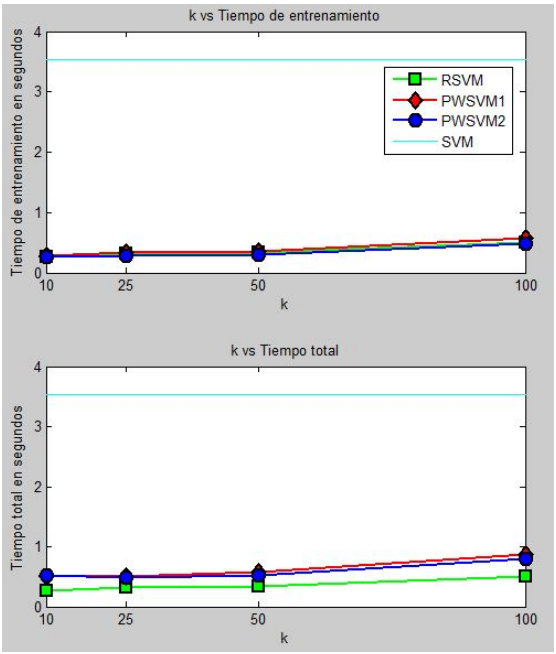

Figura 6.6: Gráficas de los tiempos de ejecución con respecto al valor de k (conjunto de datos de cáncer de pecho)

6.3.3. Conjunto de datos de abulones

El método propuesto se probó con el conjunto de datos de abulones (Abalone data set). Un abulón es un caracol marino de California. Este conjunto fue tomado del repositorio de la Universidad de California en Irvine (UCI) [86].

El conjunto consta de 4177 objetos de dimensión 8. Originalmente, este conjunto de datos es utilizado para tareas de multiclasificación, ya que que cuenta con 29 clases, donde cada clase es la edad en años del abulón correspondiente. La manera en que se adaptó este conjunto de datos para poder utilizarlo en clasificación binaria fue muy sencilla: si el abulón reporta una edad mayor a nueve años (si su etiqueta correspondiente es $y_i > 9.$), pertenece a la clase madura y se le asigna una etiqueta $y_i = +1$. Por otro lado, si el abulón reporta una edad menor o igual a nueve años (si su etiqueta correspondiente es $y_i \leq 9.$), pertenece a la clase joven y se le asigna una etiqueta $y_i = -1$.

Con la modificación anterior, el conjunto de datos de abulones presenta 2081 objetos con etiqueta

$y_i = +1$ y 2096 objetos con etiqueta $y_i = -1$. Para obtener un conjunto de entrenamiento y un conjunto de prueba, el conjunto se dividió utilizando una proporción 80/20. El 80 % de los datos (3342) se utilizaron como conjunto de entrenamiento, mientras el 20 % restante (835) se utilizaron como conjunto de prueba.

6.3.4. Resultados utilizando el conjunto de datos de abulones

En el cuadro 6.5 se presentan los resultados obtenidos con los distintos métodos para este conjunto de datos. Al igual que con los demás conjuntos de datos, los valores del parámetro C para los distintos métodos se calcularon mediante validación cruzada con $n = 5$.

Cuadro 6.5: Resultados de la evaluación con el conjunto de datos de abulones

Método	C	Prom. Clasif. [%]	Max. Clasif. [%]	# VS	$T_{AGRUPAR}$[s]	$T_{ENTRENAMIENTO}$[s]	T_{TOTAL}[s]
SVM	10	78.1±0	78.1	1654±0	0	1.25±0.04	1.25±0.04
RSVM(k=10)	20	66.78±7.58	77.37	4.76±2.2	.0013±.0008	0.29±0.03	0.29±0.03
RSVM(k=25)	20	71.35±5.57	80.96	11.58±3.23	.0014±.0012	0.34±0.06	0.34±0.06
RSVM(k=100)	100	76.33±2.68	81.2	46.48±5.65	.0013±.0003	0.52±0.07	0.52±0.07
RSVM(k=250)	50	77.47±1.47	80	119.92±13.57	.0015±.0002	0.83±0.09	0.83±0.09
RSVM(k=500)	5	78.07±1.42	80.6	248.62±12.44	.0014±.0007	2.87±0.24	2.87±0.24
PWSVM1(k=10)	50	77.2±1.98	80.84	4.18±1.14	15.49±4.62	0.34±0.04	15.84±4.62
PWSVM1(k=25)	20	78.9±1.8	81.44	7.66±1.48	7.9±1.27	0.39±0.04	8.29±1.28
PWSVM1(k=100)	5	78.94±1.7	81.8	48.76±4.78	4.8±0.7	0.52±0.05	5.33±0.71
PWSVM1(k=250)	5	78.62±0.87	80.36	122.3±9.94	7.08±0.9	0.85±0.08	7.93±0.91
PWSVM1(k=500)	5	77.95±1.12	79.88	250.9±13.7	11.71±1.05	2.91±0.24	14.62±1.07
PWSVM2(k=10)	100	76.29±2.7	80.48	sin medición	16.48±5.2	0.31±0.03	16.79±5.21
PWSVM2(k=25)	100	78.91±1.4	81.44	6.66±1.73	8.46±1.83	0.36±0.07	8.82±1.84
PWSVM2(k=100)	50	80.5±0.68	82.4	41.3±4.58	4.76±0.79	0.51±0.06	5.27±0.8
PWSVM2(k=250)	100	80.42±0.68	81.56	101.84±6.5	6.93±0.85	0.82±0.07	7.75±0.86
PWSVM2(k=500)	100	80.21±0.59	81.32	198.22±8.84	11.78±1.73	2.96±0.41	14.74±1.97

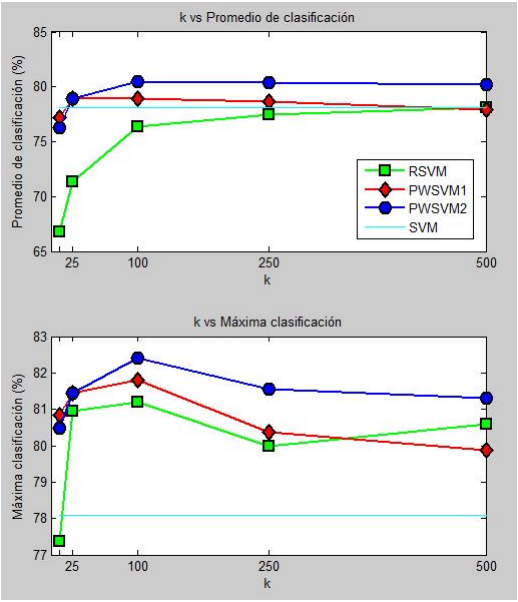

Figura 6.7: Gráficas de los porcentajes de clasificación con respecto al valor de k (conjunto de datos de abulones)

Se observa que el método propuesto logra reducir el tiempo promedio de entrenamiento de las SVM en la mayoría de los casos, excepto en el caso cuando $k = 500$. Además, salvo en el caso cuando $k = 10$, el método propuesto tiene un promedio de clasificación mejor que la mayoría de los demás métodos probados, incluso llegando a alcanzar un acierto de clasificación de 82.4 % cuando $k = 100$ (como se observa en la cuarta columna del cuadro 6.5), alcanzado el máximo porcentaje de acierto y superando por más de 4 puntos porcentuales el acierto de clasificación del método tradicional de SVM utilizando el conjunto de datos completo. Sin embargo, el tiempo promedio que tarda en realizarse la agrupación es bastante mayor que el tiempo de entrenamiento original.

Las figuras 6.7 y 6.8 muestran el comportamiento de los diferentes métodos aplicados a este conjunto de datos.

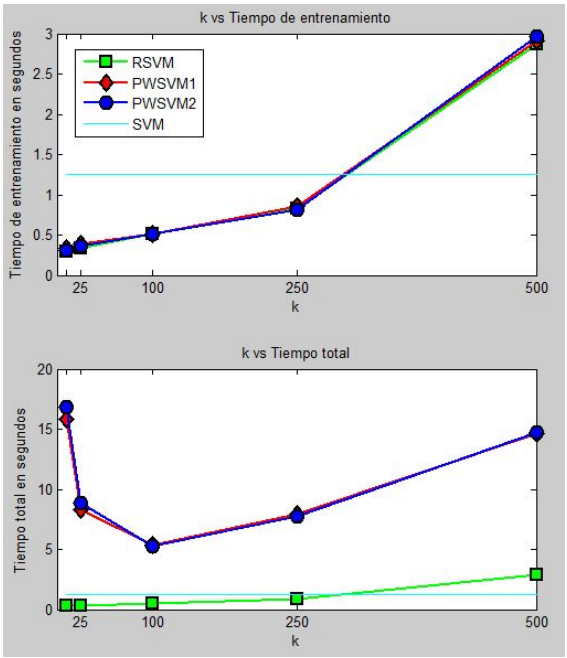

Figura 6.8: Gráficas de los tiempos de ejecución con respecto al valor de k (conjunto de datos de abulones)

6.3.5. Conjunto de datos de crédito alemán

El método propuesto se probó con el conjunto de datos de crédito alemán (German credit data set). La tarea a realizar con este conjunto es clasificar el riesgo crediticio de un cliente con base en una serie de características. El riesgo puede ser clasificado como bueno o malo, dependiendo de las características que presente el cliente. Este conjunto fue tomado del repositorio de la Universidad de California en Irvine (UCI) [86].

El conjunto consta de 1000 objetos de dimensión 20, 700 de ellos clasificados como buenos, a los cuales se les asignó la etiqueta $y_i = -1$ y 300 clasificados como malos, a los cuales se les asignó la etiqueta $y_i = +1$. El conjunto se dividió utilizando una proporción 80/20. El 80 % de los datos (800) se utilizaron como conjunto de entrenamiento, mientras el 20 % restante (200) se utilizaron como conjunto de prueba.

6.3.6. Resultados utilizando el conjunto de datos de crédito alemán

En el cuadro 6.6 se presentan los resultados obtenidos con los distintos métodos para este conjunto de datos. La disposición del cuadro es la misma que la de los cuadros anteriores. Al igual que con los demás conjuntos de datos, los valores del parámetro C para los distintos métodos se calcularon mediante validación cruzada (n-cross validation en inglés) con $n = 5$.

Cuadro 6.6: Resultados de la evaluación con el conjunto de datos de crédito alemán

Método	C	Prom. Clasif. [%]	Max. Clasif. [%]	# VS	$T_{AGRUPAR}$[s]	$T_{ENTRENAMIENTO}$[s]	T_{TOTAL}[s]
SVM	5	73±0	73	410±0	0	7.9±0.67	7.9±0.67
RSVM(k=10)	1	64.31±5.83	73	4.9±1.45	.0005±.00004	0.26±0.02	0.26±0.02
RSVM(k=25)	1	65.17±4.95	73	11.44±1.95	.0005±.0003	0.3±0.05	0.3±0.05
RSVM(k=50)	1	66.02±4.02	71	18.4±3.98	.0008±.0006	0.44±0.07	0.44±0.07
RSVM(k=150)	1	70.8±2.19	74	71.92±8.81	.0017±.0027	0.8±0.14	0.81±0.14
PWSVM1(k=10)	100	67.08±4.46	73.5	6.28±1.09	0.64±0.12	0.3±0.03	0.94±0.13
PWSVM1(k=25)	1	66.85±3.39	75	10.46±1.95	0.47±0.1	0.3±0.06	0.77±0.13
PWSVM1(k=50)	1	67.45±3.87	76	18.56±4.74	0.44±0.05	0.32±0.04	0.76±0.08
PWSVM1(k=150)	100	71.59±1.83	75.5	60.46±7.98	0.86±0.09	0.68±0.09	1.54±0.14
PWSVM2(k=10)	1	70±0	70	5.62±0.73	0.63±0.17	0.25±0.03	0.88±0.18
PWSVM2(k=25)	20	70.31±0.95	74	14.6±1.36	0.48±0.1	0.3±0.05	0.75±0.11
PWSVM2(k=50)	50	71.81±2.05	75.5	25.7±3.05	0.45±0.06	0.32±0.04	0.77±0.08
PWSVM2(k=150)	50	73.03±1.66	76	79.76±4.52	0.85±0.07	0.55±0.04	1.4±0.09

Se puede observar que el método propuesto logra reducir de manera drástica el tiempo promedio de entrenamiento de las SVM para todos los valores de k, con la desventaja de perder hasta tres puntos porcentuales en el acierto de clasificación cuando $k = 10$. Sin embargo, para el valor de $k = 150$, el promedio del porcentaje de acierto de clasificación es de 73.03 %, ligeramente mayor al 73 % alcanzado al utilizar el conjunto de entrenamiento por completo. Además, con este valor de k, se llegó a alcanzar el mejor porcentaje máximo de acierto de clasificación, con 76 % de acierto, superando por tres puntos porcentuales el porcentaje alcanzado utilizando todo el conjunto de entrenamiento. Además, el tiempo total de ejecución de PWSVM para cualquier valor de k es mucho menor al tiempo de entrenamiento promedio al entrenar SVM con el conjunto por completo.

Las figuras 6.9 y 6.10 muestran el comportamiento de los diferentes métodos aplicados a este conjunto de datos.

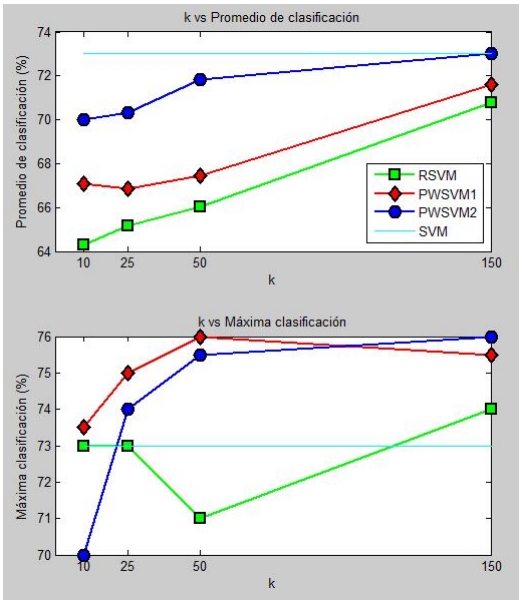

Figura 6.9: Gráficas de los porcentajes de clasificación con respecto al valor de k (conjunto de datos de crédito alemán)

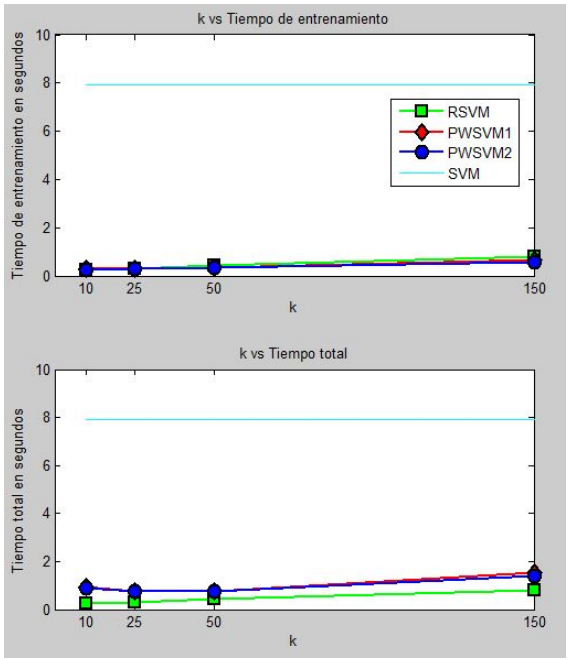

Figura 6.10: Gráficas de los tiempos de ejecución con respecto al valor de k (conjunto de datos de crédito alemán)

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Esta tesis presenta un nuevo método llamado máquinas de soporte vectorial con ponderación probabilística (PWSVM) que integra un método de aprendizaje no supervisado al esquema de máquinas de soporte vectorial con ponderación (WSVM). PWSVM aplica el algoritmo de k-medoides al conjunto de entrenamiento original tomando en cuenta las etiquetas de los objetos, encontrando k grupos, cada uno de ellos con un objeto representativo (medoide) del grupo. Los medoides de cada grupo, la cantidad de objetos pertenecientes a cada grupo y una medida de dispersión de cada grupo son obtenidos e integrados en un nuevo esquema de SVM que considera la información de cada grupo para realizar el entrenamiento de las máquinas de soporte.

Se presenta una comparación del desempeño del método con respecto al esquema tradicional de SVM y a una reducción del conjunto de entrenamiento realizada al azar. Los resultados experimentales indican que para conjuntos de datos de baja dimensión como el conjunto de datos artificial y el conjunto de datos de abulones, el método propuesto logra reducir en la mayoría de los casos el tiempo de entrenamiento de las máquinas de soporte incluyendo la información obtenida acerca de los grupos, logrando un porcentaje de acierto en la clasificación cercana e incluso en ocasiones superior a la alcanzada utilizando todo el conjunto de entrenamiento. Sin embargo, el tiempo de ejecución del algoritmo de agrupamiento es mayor al tiempo que tarda el entrenamiento utilizando el conjunto de datos original. Esto puede deberse a la táctica que utiliza el paquete CVX para resolver el problema de optimización. Puede que CVX ocupe algún algoritmo que aproveche la baja dimensionalidad del problema en cuestión y explote esta característica para resolver el problema de optimización de manera eficiente.

Además, los resultados experimentales muestran que al utilizar el método propuesto con los conjuntos de datos de cáncer de pecho de Wisconsin y de crédito alemán, con dimensión $N = 30$ y $N = 20$ respectivamente, se logra reducir el tiempo de entrenamiento de las SVM afectando muy poco la eficiencia en el conjunto de prueba. Además, el tiempo que tarda el algoritmo de agrupamiento en encontrar los medoides y recolectar la información requerida de los grupos es bastante bajo, tanto que el tiempo total que tarda el agrupamiento mas el tiempo que tarda el entrenamiento utilizando los conjuntos reducidos es bastante menor que el tiempo que tarda el entrenamiento utilizando los conjuntos originales. Los resultados observados utilizando estos conjuntos de datos

son prometedores. El método propuesto se podría aplicar a conjuntos de datos de alta dimensión ($N \geq 20$) en situaciones en las que se pueda sacrificar un poco la eficiencia de clasificación en virtud de mejorar el tiempo de entrenamiento.

7.2. Trabajo a futuro

- Demostrar la convexidad (o no convexidad) de los problemas de optimización (5.28) y (5.30).
- En caso de que alguno de los problemas de optimización (5.28) y (5.30) sea convexo, explotar esta característica y utilizarla de manera similar al esquema de PWSVM.
- Demostrar la complejidad temporal y espacial de PWSVM.
- Extender PWSVM a problemas de clasificación con mas de dos clases.
- Experimentar con otros algoritmos de agrupamiento para realizar la reducción del conjunto de datos de entrenamiento original.

Bibliografía

[1] Murphy, Kevin P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[2] Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[3] Shalev-Shwartz, Shai, and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[4] Cukier, Kenneth. *Data, data everywhere: A special report on managing information*. Economist Newspaper, 2010.

[5] Abbass, Hussein A., et al. "Online adaptation in learning classifier systems: stream data mining." *Urbana* 51 (2004): 61801.

[6] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks."*Machine learning* 20.3 (1995): 273-297.

[7] Gantz, John, and David Reinsel. "Extracting value from chaos."*IDC iview* 1142 (2011): 1-12.

[8] Vedaldi, Andrea, and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms."*Proceedings of the 18th ACM international conference on Multimedia*. ACM, 2010.

[9] Osuna, Edgar, Robert Freund, and Federico Girosi. "Support vector machines: Training and applications."(1997).

[10] Chen, Wun-Hwa, and Jen-Ying Shih. "A study of Taiwan's issuer credit rating systems using support vector machines."*Expert Systems with Applications* 30.3 (2006): 427-435.

[11] Byvatov, Evgeny, and G. Schneider. "Support vector machine applications in bioinformatics."*Applied bioinformatics* 2.2 (2002): 67-77.

[12] Sculley, David, and Gabriel M. Wachman. "Relaxed online SVMs for spam filtering."*Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007.

[13] López Chau, Asdrúbal. *Métodos de reducción de datos para clasificación con Máquinas de Soporte Vectorial*. Tesis de doctorado, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2013.

[14] Tsang, Ivor W., James T. Kwok, and Pak-Ming Cheung. "Core vector machines: Fast SVM training on very large data sets."*Journal of Machine Learning Research*. 2005.

[15] Boyd, Stephen, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[16] Karush, William. *Minima of functions of several variables with inequalities as side constraints*. Diss. Master's thesis, Dept. of Mathematics, Univ. of Chicago, 1939.

[17] Kuhn, H. W. and A. W. Tucker. "Nonlinear Programming."*Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, (1951): 481-492.

[18] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers". *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992.

[19] Yang, Xulei, Qing Song, and Yue Wang. "A weighted support vector machine for data classification."*International Journal of Pattern Recognition and Artificial Intelligence* 21.05 (2007): 961-976.

[20] Asiaee T, Amir, et al. "If you are happy and you know it... tweet."*Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012.

[21] Tomar, Divya, and Sonali Agarwal. "Hybrid feature selection based weighted least squares twin support vector machine approach for diagnosing breast cancer, hepatitis, and diabetes."*Advances in Artificial Neural Systems* 2015 (2015): 1.

[22] Tian, Jiang, et al. "Robust prediction of protein subcellular localization combining PCA and WSVMs."*Computers in biology and medicine* 41.8 (2011): 648-652.

[23] Jain, Anil K., M. Narasimha Murty, and Patrick J. Flynn. "Data clustering: a review."*ACM computing surveys (CSUR)* 31.3 (1999): 264-323.

[24] MacQueen, James. "Some methods for classification and analysis of multivariate observations."*Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. No. 14. 1967.

[25] Jain, Anil K., and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[26] Xu, Rui, and Don Wunsch. *Clustering*. Vol. 10. John Wiley & Sons, 2008.

[27] Everitt, B. S. *Cluster analysis*. London : Heinemann Educational on Behalf of the Social Science Research Council, 1980.

[28] Everitt, Brian, Sabine Landau, and Morven Leese. *Cluster Analysis*. London: Arnold, 2001.

[29] Fisher, Walter D. "On grouping for maximum homogeneity."*Journal of the American statistical Association* 53.284 (1958): 789-798.

[30] Cox, Douglas R. "Note on grouping."*Journal of the American Statistical Association* 52.280 (1957): 543-547.

[31] Bishop, Christopher M. "Pattern Recognition."*Machine Learning* (2006).

[32] Wu, Xindong, et al. "Top 10 algorithms in data mining."*Knowledge and information systems* 14.1 (2008): 1-37.

[33] Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms."*Neural Networks, IEEE Transactions* on 16.3 (2005): 645-678.

[34] Kaufman, L. and Rousseeuw, P.J. "Clustering by means of Medoids."*Statistical Data Analysis Based on the L_1–Norm and Related Methods*, edited by Y. Dodge. North-Holland, (1987): 405–416.

[35] Kaufman, L. and Rousseeuw, P.J. *Finding groups in data: an introduction to cluster analysis.* John While & Sons (1990).

[36] Godoy, Salvador. "Análisis del algoritmo k-means". Curso de reconocimiento de patrones. CIC-IPN, Ciudad de México. Otoño, 2014. Presentación en clase.

[37] Wang, Guosheng. "A survey on training algorithms for support vector machine classifiers."*Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference* on. Vol. 1. IEEE, 2008.

[38] Doumpos, Michael. "An experimental comparison of some efficient approaches for training support vector machines."*Operational Research* 4.1 (2004): 45-56.

[39] Graf, Hans P., et al. "Parallel support vector machines: The cascade svm."*Advances in neural information processing systems.* 2004.

[40] Collobert, Ronan, Samy Bengio, and Yoshua Bengio. "A parallel mixture of SVMs for very large scale problems."*Neural computation* 14.5 (2002): 1105-1114.

[41] Zanghirati, Gaetano, and Luca Zanni. "A parallel solver for large quadratic programs in training support vector machines."*Parallel computing* 29.4 (2003): 535-551.

[42] Poulet, Francois. "Multi-way distributed SVM algorithms."*Proc. of ECML/PKDD.* 2003.

[43] DeCoste, Dennis, and Kiri Wagstaff. "Alpha seeding for support vector machines."*Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2000.

[44] Feng, Du, et al. "A new alpha seeding method for support vector machine training."*Advances in Natural Computation.* Springer Berlin Heidelberg, 2005. 679-682.

[45] Liu, Yangguang, et al. "An incremental updating method for support vector machines."*Advanced Web Technologies and Applications.* Springer Berlin Heidelberg, 2004. 426-435.

[46] Hao, Zhifeng, et al. "Online LS-SVM learning for classification problems based on incremental chunk."*Advances in Neural Networks–ISNN 2004.* Springer Berlin Heidelberg, 2004. 558-564.

[47] Orabona, Francesco, et al. "On-line independent support vector machines."*Pattern Recognition* 43.4 (2010): 1402-1412.

[48] Bordes, Antoine, and Léon Bottou. "The Huller: a simple and efficient online SVM."*Machine Learning: ECML 2005.* Springer Berlin Heidelberg, 2005. 505-512.

[49] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers."*Neural processing letters* 9.3 (1999): 293-300.

[50] Fung, Glenn, and Olvi L. Mangasarian. "Incremental Support Vector Machine Classification."*SDM.* 2002.

[51] List, Niko, and Hans Ulrich Simon. "A general convergence theorem for the decomposition method."*Learning Theory.* Springer Berlin Heidelberg, 2004. 363-377.

[52] Hsu, Chih-Wei, and Chih-Jen Lin. "A simple decomposition method for support vector machines."*Machine Learning* 46.1-3 (2002): 291-314.

[53] Osuna, Edgar, Robert Freund, and Federico Girosi. "An improved training algorithm for support vector machines."*Neural Networks for Signal Processing [1997] VII. Proceedings of the 1997 IEEE Workshop*. IEEE, 1997.

[54] Platt, John. *Sequential minimal optimization: A fast algorithm for training support vector machines*. Microsoft Research. Technical Report MSR-TR-98-14, (1998).

[55] Joachims, Thorsten. *Making large scale SVM learning practical*. Universität Dortmund, 1999.

[56] Balcázar, Jose, Yang Dai, and Osamu Watanabe. "A random sampling technique for training support vector machines."*Algorithmic Learning Theory*. Springer Berlin Heidelberg, 2001.

[57] Balcazar, Jose L., Yang Dai, and Osamu Watanabe. "Provably fast training algorithms for support vector machines."*Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001.

[58] Keerthi, S. Sathiya, Olivier Chapelle, and Dennis DeCoste. "Building support vector machines with reduced classifier complexity."*The Journal of Machine Learning Research* 7 (2006): 1493-1515.

[59] Bottou, Léon, Jason Weston, and Gökhan H. Bakir. "Breaking SVM complexity with cross-training."*Advances in neural information processing systems*. 2004.

[60] Devijver, Pierre A., and Josef Kittler. *Pattern recognition: A statistical approach*. Vol. 761. London: Prentice-Hall, 1982.

[61] Schohn, Greg, and David Cohn. "Less is more: Active learning with support vector machines."*ICML*. 2000.

[62] Lee, Yuh-Jye, and Olvi L. Mangasarian. "RSVM: Reduced Support Vector Machines."*SDM*. Vol. 1. 2001.

[63] Lee, Yuh-Jye, and Su-Yun Huang. "Reduced support vector machines: A statistical theory."*Neural Networks, IEEE Transactions on* 18.1 (2007): 1-13.

[64] Peres, Rodrigo T., and Carlos E. Pedreira. "Generalized risk zone: Selecting observations for classification."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.7 (2009): 1331-1337.

[65] Angiulli, Fabrizio. "Fast nearest neighbor condensation for large data sets classification."*Knowledge and Data Engineering, IEEE Transactions on* 19.11 (2007): 1450-1464.

[66] Wang, Jigang, Predrag Neskovic, and Leon N. Cooper. "Training data selection for support vector machines."*Advances in natural computation*. Springer Berlin Heidelberg, 2005. 554-564.

[67] Lyhyaoui, Abdelouahid, et al. "Sample selection via clustering to construct support vector-like classifiers."*Neural Networks, IEEE Transactions on* 10.6 (1999): 1474-1481.

[68] Shin, Hyunjung, and Sungzoon Cho. "Neighborhood property–based pattern selection for support vector machines."*Neural Computation* 19.3 (2007): 816-855.

[69] Yu, Hwanjo, Jiong Yang, and Jiawei Han. "Classifying large data sets using SVMs with hierarchical clusters."*Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003.

[70] Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. "BIRCH: an efficient data clustering method for very large databases."*ACM Sigmod Record*. Vol. 25. No. 2. ACM, 1996.

[71] Xiong, Sheng-Wu, Xiao-Xiao Niu, and Hong-Bing Liu. "Support vector machines based on subtractive clustering."*Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on.* Vol. 7. IEEE, 2005.

[72] Chiu, Stephen L. "Fuzzy model identification based on cluster estimation."*Journal of Intelligent & Fuzzy Systems* 2.3 (1994): 267-278.

[73] Linda, Ondrej, and Milos Manic. "GNG-SVM framework-classifying large datasets with Support Vector Machines using Growing Neural Gas."*Neural Networks, 2009. IJCNN 2009. International Joint Conference on.* IEEE, 2009.

[74] Fritzke, Bernd. "A growing neural gas network learns topologies."*Advances in neural information processing systems* 7 (1995): 625-632.

[75] Neto, Ajalmar RR, and Guilherme A. Barreto. "Opposite maps: Vector quantization algorithms for building reduced-set svm and lssvm classifiers."*Neural processing letters* 37.1 (2013): 3-19.

[76] Li, Xia, Na Wang, and Shu-Yuan Li. "A fast training algorithm for SVM via clustering technique and Gabriel graph."*Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques.* Springer Berlin Heidelberg, 2007. 403-412.

[77] Gabriel, K. Ruben, and Robert R. Sokal. "A new statistical approach to geographic variation analysis."*Systematic Biology* 18.3 (1969): 259-278.

[78] Lu, Shu-xia, Jie Meng, and Gui-en Cao. "Support vector machine based on a new reduced samples method."*Machine Learning and Cybernetics (ICMLC), 2010 International Conference on.* Vol. 3. IEEE, 2010.

[79] Koggalage, Ravindra, and Saman Halgamuge. "Reducing the number of training samples for fast support vector machine classification."*Neural Information Processing-Letters and Reviews* 2.3 (2004): 57-65.

[80] De Almeida, Marcelo Barros, Antônio de Pádua Braga, and João Pedro Braga. "SVM-KM: speeding SVMs learning with a priori cluster selection and k-means."*Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on.* IEEE, 2000.

[81] Tran, Quang-Anh, Qian-Li Zhang, and Xing Li. "Reduce the number of support vectors by using clustering techniques."*Machine Learning and Cybernetics, 2003 International Conference on.* Vol. 2. IEEE, 2003.

[82] Yang, Xulei, Qing Song, and Yue Wang. "A weighted support vector machine for data classification."*International Journal of Pattern Recognition and Artificial Intelligence* 21.05 (2007): 961-976.

[83] Nguyen, Giang Hoang, Son Lam Phung, and Abdesselam Bouzerdoum. "Efficient SVM training with reduced weighted samples."*Neural Networks (IJCNN), The 2010 International Joint Conference on.* IEEE, 2010.

[84] Krishnapuram, Raghu, and James M. Keller. "A possibilistic approach to clustering."*Fuzzy Systems, IEEE Transactions on* 1.2 (1993): 98-110.

[85] Mahalanobis, Prasanta Chandra. "On the generalised distance in statistics."*Proceedings of the National Institute of Sciences of India* 2 (1) (1936): 49–55.

[86] Lichman, M. *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2013.

[87] Michael Grant and Stephen Boyd. *CVX: Matlab software for disciplined convex programming, version 2.0 beta* [http://cvxr.com/cvx]. 2013.

[88] Wang, Jiaqi, Xindong Wu, and Chengqi Zhang. "Support vector machines based on K-means clustering for real-time business intelligence systems."*International Journal of Business Intelligence and Data Mining* 1.1 (2005): 54-64.

[89] Yao, Yukai, et al. "K-SVM: An effective SVM algorithm based on K-means clustering."*Journal of computers* 8.10 (2013): 2632-2639.

[90] Gu, Quanquan, and Jiawei Han. "Clustered support vector machines."*Proceedings of the sixteenth international conference on artificial intelligence and statistics*. 2013.