ESCUELA POLITÉCNICA NACIONAL

DEPARTAMENTO DE MATEMÁTICA

SELECCIÓN DE VARIABLES A TRAVÉS DE MÁQUINAS DE SOPORTE VECTORIAL DISPERSAS: UN ENFOQUE DE OPTIMIZACIÓN BINIVEL

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN OPTIMIZACIÓN MATEMÁTICA

TESIS

CARLOS LUIS GONZÁLEZ VALLEJO

carlos.gonzalez@epn.edu.ec

DIRECTOR: PEDRO MERINO ROSERO, PhD.

pedro.merino@epn.edu.ec

CO-DIRECTOR: DAVID VILLACIS PROAÑO, PhD.

david.villacis01@epn.edu.ec

QUITO, ENERO 2024

DECLARACIÓN

Yo, CARLOS LUIS GONZÁLEZ VALLEJO, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Carlos Luis González Vallejo

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por CARLOS LUIS GONZÁLEZ VALLEJO, bajo mi supervisión.

Pedro Merino Rosero, PhD.

Director de Tesis

AGRADECIMIENTOS

Agradezco a Dios por permitirme alcanzar este objetivo, que hoy lo veo plasmado en este trabajo de investigación.

A mi familia, en especial a mi esposa, le dedico mi más sincero agradecimiento por su invaluable respaldo durante todo este logro académico. Su apoyo constante y comprensión han sido la fuente de mi fortaleza, inspirándome a superar cada desafío en este camino.

Agradezco a Pedro Merino y David Villacís por su dedicación, orientación y valiosos aportes que han enriquecido significativamente este trabajo.

Cada paso de este camino ha sido moldeado por aquellos que han dejado una huella en mi vida académica, y por ello, mi sincero agradecimiento a todos quienes han contribuido de alguna manera en este logro.

DEDICATORIA

Para María Reneé, Karla, Mateo/Amelia, Leonidas (+), Diana, Sofía y Daniel. Ustedes son todo mi universo.

Contenido

Lista de Figuras

Lista de Tablas

Lista de Algoritmos

Resumen

En esta tesis, abordamos el problema de selección de variables a través de Máquinas de Soporte Vectorial con regularización mixta (L_2 y L_1), denominado (L_2L_1-SVM). Presentamos un enfoque binivel, donde las funciones objetivo del nivel superior e inferior son el Error Cuadrático Medio (MSE) y (L_2L_1-SVM) respectivamente. Además, usamos aproximaciones locales suavizadas de las funciones hinge-loss y norma L_1, con el propósito de tener una función objetivo suave en el problema de nivel inferior y aplicar algoritmos que han sido ampliamente estudiados y mejorados. Este enfoque binivel permite seleccionar las variables más importantes para el problema de clasificación binaria; y a la vez optimizar el hiperparámetro de regularización dispersa L_1, manteniendo fijo el hiperparámetro de regularización L_2. Formulamos tres variantes del problema de optimización binivel, considerando el hiperparámetro de regularización dispersa L_1 como escalar, vectorial y vectorial agrupado. Para caracterizar la solución de todas las formulaciones del problema de optimización binivel, usamos las condiciones Karush-Kunh-Tucker. Con base a esta caracterización, proponemos un algoritmo eficiente llamado BiSVM L-BFGS para encontrar la solución numérica de las diferentes formulaciones del problema de optimización binivel. Los resultados experimentales demuestran la efectividad del enfoque propuesto, destacando mejoras en la interpretabilidad del modelo y la identificación de variables relevantes.

Palabras claves: Optimización Binivel, Selección de Variables, Aprendizaje Automático, Máquinas de Soporte Vectorial, Hiperparámetros.

Abstract

In this thesis, we address the problem of feature selection through Support Vector Machines with mixed regularization (L_2 and L_1), called (L_2L_1-SVM). We present a bilevel approach, where the upper and lower level objective functions are Mean Square Error (MSE) and (L_2L_1-SVM) respectively. Furthermore, we use smoothed local approximations of the hinge-loss function and L_1 norm, with the purpose of getting a smooth objective function in the lower-level problem and applying algorithms that have been widely studied and improved. This bilevel approach allows selecting the most important features for the binary classification problem and at the same time, optimizing the sparse regularization hyperparameter L_1, keeping the regularization hyperparameter L_2 fixed. We formulate three variants of the bilevel optimization problem, considering the sparse regularization hyperparameter L_1 as a scalar, vector, and grouped vector. To characterize the solution of all formulations of the bilevel optimization problem, we use the Karush-Kunh-Tucker conditions. Based on this characterization, we propose an efficient algorithm called BiSVM L-BFGS to find the numerical solution of the different formulations of the bilevel optimization problem. Experimental results demonstrate the effectiveness of the proposed approach, highlighting improvements in model interpretability and the identification of relevant features.

Key words: Bilevel Optimization, Feature Selection, Machine Learning, Support Vector Machines, Hyperparameters.

Capítulo 1

Introducción

La selección de variables es relevante en el campo del aprendizaje automático, cuyo propósito es identificar las variables más importantes e informativas de un conjunto de datos que ayudan a predecir una variable objetivo. Este proceso implica seleccionar un subconjunto de variables, las cuales pueden contribuir significativamente a la precisión del modelo de aprendizaje automático. Es decir, se busca determinar las variables del conjunto de datos que tienen mayor influencia para lograr una mejor predicción de la variable objetivo.

Este concepto gira entorno a la idea de que no todas las variables tienen la misma importancia para una tarea de aprendizaje en particular. En algunos casos, los conjuntos de datos tienen una gran cantidad de variables, de entre las cuales, algunas pueden ser irrelevantes, redundantes o ruidosas. Este tipo de variables mencionadas anteriormente, podrían provocar sobreajuste, mayor complejidad computacional o una menor interpretabilidad del modelo.

El sobreajuste ocurre cuando un modelo de aprendizaje automático se ajusta demasiado a las particularidades del conjunto de datos de entrenamiento, pero tiene dificultades para generalizar con un nuevo conjunto de datos que no ha sido utilizado en la etapa de entrenamiento. En esencia, el sobreajuste se traduce en una mala generalización del modelo frente a un nuevo conjunto de datos de validación. De acuerdo con [Shalev-Shwartz and Ben-David, 2014], el sobreajuste ocurre cuando un predictor ajusta “demasiado bien” los datos de entrenamiento. En consecuencia, la selección de variables contribuye a mejorar la capacidad de generalización del modelo.

En aplicaciones del mundo real, es común encontrarse con conjuntos de datos que poseen un elevado número de variables. Estas variables no sólo añaden complejidad computacional a la tarea de aprendizaje, sino que también pueden introducir ruido y sesgo en el modelo, lo que puede afectar negativamente en su capacidad predictiva. Al seleccionar

las variables más importantes de acuerdo a criterios de optimización, podemos reducir la complejidad del modelo, lo que a su vez mejora su precisión [Sarker, 2021].

Además, la selección de variables desempeña un papel fundamental en la interpretabilidad de un modelo. En algunos campos como salud, finanzas, comercio, entre otros, es fundamental comprender e interpretar las decisiones tomadas por los modelos de aprendizaje automático. En este sentido, al optar por las variables más relevantes, podemos simplificar el modelo, haciendo que sea más comprensible para la interpretación humana. El estudio de [Bazarra et al., 2021] respalda esta perspectiva al señalar que, si bien al incrementar la complejidad de un modelo puede lograr mejores resultados, también disminuye su interpretabilidad. A este fenómeno se lo conoce como el equilibrio de precisión/interpretabilidad.

Según las investigaciones llevadas a cabo por [Pudjihartono et al., 2022] y [Guyon and Elisseeff, 2003], existen varios métodos para la selección de variables. Estos métodos difieren en términos de las métricas de evaluación utilizadas, su complejidad computacional, los algoritmos empleados y el potencial para detectar interacciones o redundancias entre las variables. A continuación, se resumen estos métodos:

Métodos de Filtro: Estos métodos se pueden clasificar en dos categorías: univariantes y multivariantes. Los métodos univariantes analizan cada variable de forma individual con respecto a la variable de estudio, centrándose en determinar su relevancia. En cambio, los métodos multivariantes toman en cuenta un subconjunto de variables al mismo tiempo para determinar la interacción o redundancia entre ellas. En el contexto de los problemas de regresión, se utilizan varias métricas como: la correlación de *Pearson* [Sakr et al., 2019b], el coeficiente de determinación R^2 de una regresión [Thomas et al., 2023], t-test, ANOVA [Jafari and Azuaje, 2006], entre otras. Por otra parte, para los problemas de clasificación algunas métricas usadas son: FCBF (*Fast Correlation-Based Filter*) [Bolón-Canedo et al., 2014], test χ^2 [Liu and Setiono, 1995], área bajo la curva ROC (*Receiver Operating Characteristic*) [Guyon and Elisseeff, 2003] por nombrar los más relevantes. Una ventaja importante de estos métodos es que requieren un bajo costo computacional, lo cual permite aplicarse en conjuntos de datos con alta dimensionalidad.

Métodos Envolventes: Estos métodos utilizan funciones de costo tales como: RMSE, Accuracy, F1-Score, AUC, entre otras, como criterio para ayudar a identificar el mejor subconjunto de variables. A diferencia de los métodos de filtro, estos métodos implícitamente toman en consideración la interacción y redundancia de las variables

durante la selección del mejor subconjunto. A pesar que estos métodos pueden alcanzar mejores resultados que los métodos de filtro, la búsqueda exhaustiva de todas las posibles combinaciones de subconjuntos de variables resulta inaplicable desde el punto de vista computacional. Por tanto, no son métodos escalables a conjuntos de datos con alta dimensionalidad. A continuación, se mencionan algunos ejemplos de métodos dentro de esta categoría: búsqueda secuencial ([Xiong et al., 2001]; [Inza et al., 2004]), algoritmos de estimación de distribución [Inza et al., 2000], algoritmos genéticos [Sakr et al., 2019a], entre otros.

Métodos Integrados: A diferencia de los métodos envolventes y de filtro, estos métodos incorporan la selección de variables dentro de un algoritmo de aprendizaje automático, durante la etapa de entrenamiento del mismo. Es decir, el predictor ajusta sus parámetros y determina el peso o importancia de cada una de las variables con el fin de obtener una mejor precisión. Así, la búsqueda del mejor subconjunto de variables y la construcción del algoritmo de aprendizaje automático se lo realiza en un solo lazo. Desde el punto de vista computacional, estos métodos son más rápidos que los métodos envolventes, pero más lentos que los métodos de filtro. Algunos ejemplos de estos métodos son: bosques aleatorios [Diaz and Alvarez, 2006], Naïve Bayes ponderado [Duda et al., 2000], máquinas de soporte vectorial SVM ([Guyon et al., 2002]; [Maldonado and Weber, 2009]), regresión logística [Ma and Huang, 2005], entre otros.

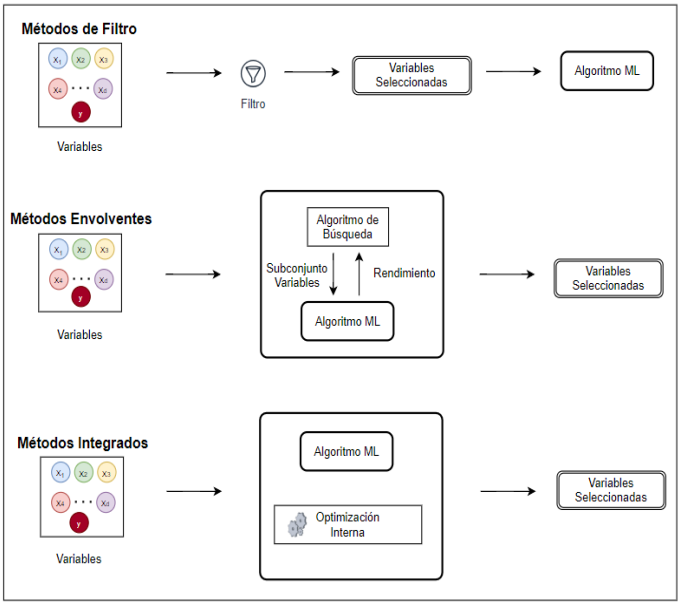


Figura 1.1: Ilustración de los métodos de selección de variables

Como podemos ver, la selección de variables es una técnica que requiere encontrar un equilibrio entre la simplicidad y la precisión del modelo. Por un lado, un modelo con muy pocas variables puede no ser capaz de capturar los patrones subyacentes en los datos. Esto podría dar como resultado un deficiente poder predictivo del modelo. Por otro lado, un número excesivo de variables puede hacer que el modelo sea demasiado complejo y propenso a sobreajuste. Esto conlleva a una menor capacidad de generalización.

Para abordar este desafío, se propone la selección de variables a partir de un enfoque de optimización binivel, una técnica que combina dos problemas de optimización de manera jerárquica. La estrategia es usar en el problema de nivel superior una función objetivo que evalúe el error de validación (este concepto se lo profundiza en el Capítulo 2 dentro de los elementos del aprendizaje estadístico); y por otro lado, en el problema de nivel inferior, se lleva a cabo el entrenamiento del modelo de aprendizaje automático.

En este contexto, tomando en cuenta los métodos integrados para la selección de variables, desde un punto de vista de optimización binivel, se establecen las siguientes consideraciones:

Sea $X \in \mathbb{R}^{n\times(d+1)}$ una matriz que contiene las observaciones del conjunto de entrenamiento, donde n es el número de observaciones y d es el número de variables más una columna de unos al final. De manera complementaria, se dispone del vector $y \in \{-1,1\}^n$ que corresponde a las etiquetas para cada observación en el conjunto de entrenamiento. Además, se cuenta con una matriz $V \in \mathbb{R}^{m\times(d+1)}$ que contiene las observaciones del conjunto de validación, donde m es el número de observaciones en ese conjunto. El vector de etiquetas correspondiente a cada observación del conjunto de validación se denota como $\eta \in \{-1,1\}^m$.

1.1. Planteamiento del problema general

El objetivo es abordar un problema de optimización binivel con la siguiente formulación:

$$\min_{\beta} \quad J(\beta, \hat{w}(\beta)) \tag{1.1a}$$

$$\text{s.a.} \quad \hat{w}(\beta) \in \arg\min_{w} E(w, \beta), \tag{1.1b}$$

donde (1.1a) es el problema de optimización de nivel superior, el cual que involucra una función objetivo evaluada en el conjunto de validación $V \in \mathbb{R}^{m\times(d+1)}$ para un hiperparámetros β, que representa una métrica del error de validación. Por otro lado, (1.1b) es el problema de optimización de nivel inferior sin restricciones. Específicamente, el problema L_2L_1-SVM

(*Support Vector Machine*). Cabe mencionar, que el problema de optimización inferior es también conocido por [Cui et al., 2021] y [Hajewski et al., 2018] como EN-SVM (*elastic-net* SVM) y L_1-SVM respectivamente.

El problema de nivel inferior L_2L_1-SVM representa una ampliación del enfoque tradicional de SVM. En este enfoque, se integran los términos de regularización L_1 y L_2 con el propósito de obtener un equilibrio entre la dispersión y robustez en el modelo.

En el contexto de la clasificación binaria, el problema L_2L_1-SVM se utiliza para encontrar el hiperplano óptimo $w \in \mathbb{R}^{d+1}$ (con el término de sesgo al final) que separa dos clases de observaciones. Este problema de optimización busca determinar los parámetros que definen el hiperplano de separación, de manera que se minimice la estimación del error de validación.

De manera general, el Error Cuadrático Medio (MSE por sus siglas en inglés), es utilizado como función de error de validación, el cual mide la calidad del ajuste del modelo.

$$J(\beta, w(\beta); V, \eta) := \frac{1}{2m}\|Vw(\beta)^\top - \eta\|_2^2. \tag{1.2}$$

Notemos que la función $J : \mathbb{R}^{(d+1)} \times \mathbb{R}^{(d+1)} \to \mathbb{R}$, que en el caso general considera hiperparámetro vectorial, pertenece a la clase $\mathcal{C}^2$ y w varía en función del parámetro de regularización β. Además, $V \in \mathbb{R}^{m\times(d+1)}$ representa el conjunto de validación. Es importante destacar que la función $J(\cdot)$ no es convexa en el parámetro β. Sólo lo es para w con un β fijo como lo indica [Klatzer, 2014].

Por otra parte, [Shalev-Shwartz and Ben-David, 2014] establece que los problemas de aprendizaje pueden reescribirse como un problema de minimización de la forma:

$$\arg\min_{w}\{L_S(w) + R_S(w)\}, \tag{1.3}$$

donde $L_S(w)$ es una función de riesgo empírico y $R_S(w)$ una función de regularización.

El problema de nivel inferior (1.1b), al cual llamaremos L_2L_1-SVM (escalar), puede ser formulado como:

$$\hat{w}(\beta) \in \arg\min_{w} E(w, \beta; X, y) := \frac{1}{n}\sum_{i=1}^{n}\ell(w, (x_i, y_i)) + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \beta\sum_{j=1}^{d+1}|w_j|, \tag{1.4}$$

donde $\alpha > 0$, $\beta > 0$ son parámetros de regularización que controlan la magnitud de los coeficientes y la dispersión del hiperplano w respectivamente, un vector de características x_i y la función *hinge-loss*:

$$\ell(w,(x_i,y_i)) = \text{máx}\{0, 1 - y_i\langle w, x_i\rangle\}, \tag{1.5}$$

para todo $i = 1, \ldots, n$.

Es importante destacar que la función $E(\cdot)$ del problema (1.4) es estrictamente convexa, pero no es diferenciable debido a la función *hinge-loss* (1.5) y a la norma L_1 definida por:

$$\|w\|_1 := \sum_{j=1}^{d+1} |w_j|. \tag{1.6}$$

Así, para esta formulación la función de riesgo empírico está dado por:

$$L_S(w) = \frac{1}{n}\sum_{i=1}^{n} \text{máx}\{0, 1 - y_i\langle w, x_i\rangle\},$$

y la función de regularización L_2L_1 por:

$$R_S(w) = \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \beta\sum_{j=1}^{d+1} |w_j|.$$

En el contexto de la selección de variables, el trabajo de [Guyon et al., 2002] combina el algoritmo de aprendizaje automático L_2-SVM con un proceso iterativo de eliminación de variables, conocido como RFE-SVM. El objetivo es seleccionar las variables de manera iterativa, eliminando las menos importantes y reentrenando el modelo hasta alcanzar un subconjunto de datos que maximiza la precisión del modelo L_2-SVM.

A diferencia del método recursivo $RFE-$SVM, nuestra propuesta adopta un enfoque binivel que incorpora la selección de variables durante el entrenamiento del L_2L_1-SVM. Es decir, la búsqueda del mejor subconjunto de variables y entrenamiento del algoritmo de aprendizaje automático se la realiza en un solo lazo, sin necesidad de hacerlo iterativamente. Esta mejora se obtiene al utilizar regularizadores que promueven la dispersión de los coeficientes del hiperplano w. Por lo tanto, w presenta un mayor número de componentes nulas. Además, al realizar la selección de variables a través del problema L_2L_1-SVM se superan algunas limitaciones que tiene el problema L_1-SVM a la hora de tratar conjuntos de datos con alta dimensionalidad y cuando las variables se encuentran altamente correlacionadas [Wang et al., 2006].

Por otra parte, nuestro enfoque binivel garantiza la optimalidad del hiperparámetro β y se aprovecha la continuidad de este. Cuando el hiperparámetro $\hat{\beta} \in \mathbb{R}^{d+1}$ es vectorial, puede emplearse para construir un score que mida la importancia de cada una de las variables del conjunto de datos. Este score de importancia proporcionado se convierte en una herramienta

que ayuda a evaluar la contribución relativa de cada variable en la tarea de aprendizaje automático.

1.1.1. Formulación modelo vectorial

Una extensión del problema (1.4) es cuando el hiperparámetro $\hat{\beta} \in \mathbb{R}^{d+1}$ posee una estructura vectorial. Es decir, tiene la forma $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_{d+1})$. En consecuencia, el problema L_2L_1-SVM (vectorial) se puede escribir de la siguiente manera:

$$\hat{w}(\hat{\beta}) \in \arg\min_{w} E(w, \hat{\beta}; X, y) := \frac{1}{n}\sum_{i=1}^{n} \text{máx}\{0, 1 - y_i\langle w, x_i\rangle\} + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \sum_{j=1}^{d+1} \hat{\beta}_j |w_j|. \quad (1.7)$$

Este tipo de hiperparámetro vectorial puede ser usado para determinar la importancia o relevancia de cada una de las $d+1$ variables (incluido el término de sesgo) de un conjunto de datos. Al asignar un hiperparámetro a cada variable, se permite que el modelo tenga más grados de libertad, lo que resulta en un mejor rendimiento en comparación con modelos que utilizan un único hiperparámetro escalar. Por ejemplo, [Klatzer, 2014] propone la optimización binivel para SVM con una regularización única de tipo L_2, tanto para SVM-Lineal como para SVM-Kernel. Ambos modelos utilizan el término de regularización de tipo vectorial y han sido experimentados con conjuntos de datos clásicos como: MNIST, Iris, Parkinsons, entre otros. Los resultados obtenidos muestran una mejora significativa en el rendimiento del modelo de clasificación en comparación con la formulación tradicional utilizando hiperparámetro escalar.

1.1.2. Formulación modelo vectorial - grupos

Otro caso se presenta cuando las $d+1$ variables se encuentran divididas en $g+1$ grupos no sobrepuestos de tamaño p_k (no necesariamente homogéneos) para $k = 1, \ldots, g+1$. Para este caso, el hiperparámetro de regularización admite una estructura de grupo de la forma $\tilde{\beta} = (\tilde{\beta}_1, \ldots, \tilde{\beta}_{g+1})$ para lo cual se usa la norma *Group Lasso* definida por:

$$\|w\|_{2,1}^{group} := \sum_{k=1}^{g+1} \|w_k\|_2. \quad (1.8)$$

En consecuencia, el problema L_2L_1-SVM (grupo) se puede escribir como:

$$\hat{w}(\tilde{\beta}) \in \arg\min_{w} E(w, \tilde{\beta}; X, y) := \frac{1}{n}\sum_{i=1}^{n} \text{máx}\{0, 1 - y_i\langle w, x_i\rangle\} + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \sum_{k=1}^{g+1} \tilde{\beta}_k \|w_k\|_2. \quad (1.9)$$

Notemos que la formulación vectorial (1.7) es un caso particular de la formulación en grupos (1.9) cuando se tiene un solo grupo de tamaño $p_1 = d$ más la columna del término de sesgo. Por su parte, [Tang et al., 2018] emplean la selección de variables agrupadas con SVM y concluye que esta formulación mejora tanto la interpretabilidad, como el rendimiento de los modelos; incluso al aplicar bases de datos del mundo real en áreas como la neurociencia, la genética, el reconocimiento de texto manual, entre otras.

1.2. Contribución

Este trabajo de investigación tiene por objetivo proponer una metodología para el problema de selección de variables desde un enfoque de optimización binivel, a través del problema de optimización L_2L_1-SVM con términos de regularización dispersa L_1 (1.6) y L_{21} (1.8). Esta metodología proporcionará el mejor subconjunto de variables y el hiperplano de separación óptimo, los cuales minimizan el error de validación.

A continuación, se detalla la metodología que proponemos en este trabajo de investigación:

1. Se buscará una aproximación local suavizada tanto para la función *hinge-loss* (1.5) como para las regularizaciones L_1 (1.6) y L_{21} (1.8).

2. Se estudiará las condiciones *Karush-Kuhn-Tucker* del problema L_2L_1-SVM (escalar). Este resultado nos ayudará a extender el análisis hacia otros escenarios, donde el término de regularización disperso posee una estructura vectorial y de grupo.

3. Se implementará un algoritmo para resolver el problema de nivel inferior L_2L_1-SVM con las diferentes formulaciones (escalar, vectorial y grupo).

4. Se buscará una caracterización numérica del hipergradiente del problema de optimización binivel, para el cual se aplicará un método quasi-Newton para su resolución. Este algoritmo realiza la selección de variables y el entrenamiento del problema L_2L_1-SVM en un solo lazo.

5. Se construirá un score para medir la importancia de las variables en el contexto del hiperparámetro vectorial del L_2L_1-SVM en diferentes experimentos numéricos.

Esta tesis está organizada de la siguiente manera: En el Capítulo 2 se presenta los preliminares acerca de la optimización no lineal, los elementos del aprendizaje estadístico y una breve introducción a la optimización binivel. El Capítulo 3 contiene una revisión del estado

del arte sobre algunas formas de caracterizar la solución de un problema de optimización binivel, los algoritmos numéricos que se usan para aproximar su solución en diferentes aplicaciones y la selección de variables a través de Máquinas de Soporte Vectorial SVM. En el Capítulo 4, se muestra la metodología que se usa para la caracterización del hipergradiente del problema de optimización binivel a través de las condiciones de optimalidad. Tanto para los casos cuando el hiperparámetro de dispersión es escalar, como el vectorial y vectorial agrupado. En el Capítulo 5 se muestra el algoritmo para resolver el problema de optimización binivel L_2L_1-SVM. El Capítulo 6 contiene experimentos y análisis de los resultados de este trabajo. Finalmente, en el Capítulo 7 presentamos las conclusiones de este trabajo de investigación.

Capítulo 2

Preliminares

En el presente capítulo, se presentan algunas nociones y conceptos fundamentales que serán utilizados en el desarrollo de este trabajo de investigación. Empezaremos con algunas definiciones y teoremas de la optimización no lineal. Luego, una introducción a los elementos del aprendizaje estadístico y una descripción general del problema de optimización binivel. Para profundizar estos temas se puede revisar las siguientes referencias: ([Nocedal and Wright, 2006], [Beck, 2017], [Boyd and Vandenberghe, 2004], [Shalev-Shwartz and Ben-David, 2014], [Hastie et al., 2016]).

2.1. Optimización no lineal

Definición 1 (Función continua)**.** *Sea $C \subset \mathbb{R}^n$. Una función $f : \mathbb{R}^n \to \mathbb{R}^k$ es continua en $x \in C$ si para todo $\epsilon < 0$, existe $\delta > 0$ tal que:*

$$\|f(x_1) - f(x_2)\| \leq \epsilon, \quad \forall x_1, x_2 \in C.$$

Definición 2 (Función Lipschitz continua)**.** *Sea $C \subset \mathbb{R}^n$. Una función $f : \mathbb{R}^n \to \mathbb{R}^k$ es Lipschitz continua, si existe $L > 0$ tal que:*

$$\|f(x_1) - f(x_2)\| \leq L\|x_1 - x_2\|, \quad \forall x_1, x_2 \in C.$$

Definición 3 (Función diferenciable)**.** *Sea $\Omega \subset \mathbb{R}^n$ abierto y $f : \Omega \to \mathbb{R}$. f se dice se dice diferenciable en $x_0 \in \Omega$ si existe una función lineal $L : \Omega \to \mathbb{R}$ tal que:*

$$\lim_{h \to 0} \frac{|f(x_0 + h) - f(x_0) - L(h)|}{\|h\|}$$

Definición 4 (Regla de la cadena)**.** *Supongamos que $x = g(t)$ y $y = h(t)$ dos funciones diferenciables de t y $z = f(x,y)$ una función diferenciable de x y y. Entonces $z = f(g(t),h(t))$ es una función diferenciable de t y, además:*

$$\frac{dz}{dt} = \frac{\partial z}{\partial x}\frac{dx}{dt} + \frac{\partial z}{\partial y}\frac{dy}{dt},$$

donde las derivadas ordinarias se evalúan en t y las derivadas parciales se evalúan en (x,y).

Teorema 1 (Teorema función implícita, [Dontchev and Rockafellar, 2014], pág. 20)**.** *Sea $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^n$ una función continuamente diferenciable en una vecindad de $(\bar{p},\bar{x})$ con valores $f(\bar{p},\bar{x})$, donde $\bar{p}$ es un parámetro y $\bar{x}$ es la variable a ser determinada, tal que $f(\bar{p},\bar{x}) = \mathbf{0}_n$. Además, $\nabla_x f(\bar{p},\bar{x})$ es no singular y se define el operador solución como:*

$$S : p \mapsto \{x \in \mathbb{R}^n : f(p,x) = \mathbf{0}_n, \quad \forall p \in \mathbb{R}^d\}.$$

Entonces, el mapeo solución S tiene una localización univaluada s alrededor de $\bar{p}$ para $\bar{x}$ que es continuamente diferenciable en una vecindad $\mathcal{Q}$ de $\bar{p}$ tal que:

$$\nabla s(p) = [\nabla_x f(p,s(p))]^{-1} \nabla_p f(p,s(p)), \quad \forall p \in \mathcal{Q}.$$

Definición 5 (Problema de optimización sin restricciones)**.** *Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función diferenciable. El siguiente problema es un problema de optimización sin restricciones:*

$$\min_{x \in \mathbb{R}^n} \quad f(x). \tag{2.1}$$

Para el problema (2.1), un vector $x^* \in \mathbb{R}^n$ se dice:

Solución local: si existe una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{N}$.

Solución local estricta: si existe una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) < f(x)$ para todo $x \in \mathcal{N}$ con $x \neq x^*$.

Solución local aislada: si existe una vecindad $\mathcal{N}$ de x^* tal que x^* es la única solución en la vecindad.

Solución global: si $f(x^*) \leq f(x)$ para todo $x \in \mathbb{R}^n$.

Solución global estricta: si $f(x^*) < f(x)$ para todo $x \in \mathbb{R}^n$.

A continuación, se presentan las condiciones de optimalidad para el problema de optimización sin restricciones (2.1).

Teorema 2 (Condiciones necesarias de primer orden, [Nocedal and Wright, 2006], pág. 14)**.** *Si x^* es una solución local y f una función continuamente diferenciable en una vecindad abierta $\mathcal{N}$ de x^*, entonces $\nabla f(x^*) = 0$.*

Definición 6 (Problema de optimización con restricciones)**.** *Sean $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ y $h : \mathbb{R}^n \to \mathbb{R}^p$ funciones continuamente diferenciables. Sea $\Omega = \{x \in \mathbb{R}^n : g(x) \leq \mathbf{0}_m, h(x) = \mathbf{0}_p\}$ un conjunto de soluciones factibles o admisibles.*
El siguiente problema es un problema de optimización con restricciones:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.a.} \quad & x \in \Omega. \end{aligned} \tag{2.2}$$

Para el problema (2.2), un vector $x^* \in \mathbb{R}^n$ se dice:

Solución local: si $x^* \in \Omega$ y existe una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) \leq f(x)$ para todo $x \in \mathcal{N} \cap \Omega$.

Solución local estricta: si $x^* \in \Omega$ y existe una vecindad $\mathcal{N}$ de x^* tal que $f(x^*) < f(x)$ para todo $x \in \mathcal{N} \cap \Omega$ con $x \neq x^*$.

Solución local aislada: si $x^* \in \Omega$ y existe una vecindad $\mathcal{N}$ de x^* tal que x^* es la única solución en $\mathcal{N} \cap \Omega$.

Solución global: si $f(x^*) \leq f(x)$ para todo $x \in \Omega$.

Solución global estricta: si $f(x^*) < f(x)$ para todo $x \in \Omega$.

Definición 7 (Conjunto restricciones activas)**.** *El conjunto de restricciones activas para cualquier punto factible $x \in \Omega$ se define por:*

$$\mathcal{A}(x) := \{\forall i \in [m] : g_i(x) = 0\} \bigcup \{\forall j \in [p] : h_j(x) = 0\}.$$

El conjunto de restricciones inactivas para cualquier punto factible $x \in \Omega$ se define por:

$$\mathcal{I}(x) := \{\forall i \in [m] : g_i(x) < 0\}.$$

Teorema 3 (Condiciones necesarias de primer orden, [Nocedal and Wright, 2006], pág. 329)**.** *Sea x^* una solución local del problema (2.2) que satisface una condición de calificación (LICQ) o (MFCQ). Entonces existen multiplicadores de Lagrange $\lambda^* \in \mathbb{R}^m$, $\mu^* \in \mathbb{R}^p$ tales que:*

$$\nabla_x f(x^*) + \lambda^{*\top} \nabla_x g(x^*) + \mu^{*\top} \nabla_x h(x^*) = \mathbf{0}_n, \tag{2.3}$$

$$h(x^*) = \mathbf{0}_p, \tag{2.4}$$

$$g(x^*) \leq \mathbf{0}_m \tag{2.5}$$

$$\lambda^* \geq \mathbf{0}_m \tag{2.6}$$

$$\lambda^{*\top} g(x^*) = 0. \quad \textit{(Condición de complementariedad)} \tag{2.7}$$

A estas condiciones de optimalidad de primer orden del problema (2.2) también se las conoce como las condiciones KKT (Karush-Kun-Tucker).

Cabe mencionar que de acuerdo al teorema anterior, para mostrar la existencia de multiplicadores de Lagrange, es necesario primeramente mostrar que se cumple un condición de calificación. A continuación, se presentan las condiciones de calificación más usuales.

Definición 8 (LICQ)**.** *Dado un punto $x \in \Omega$ y $\mathcal{A}(x)$, decimos que la condición de calificación de independencia lineal (LICQ) se cumple si el conjunto de gradientes de las restricciones activas $\{\nabla h_j(x), \nabla g_i(x)\}$ es linealmente independiente para todo $i, j \in \mathcal{A}(x)$.*

Definición 9 (MFCQ)**.** *Decimos que la condición de Mangasarian-Fromovitz si el conjunto de gradientes de las restricciones de igualdad $\{\nabla h_j(x)\}$ es linealmente independiente para todo $j \in [p]$ y además, existe un vector $w \in \mathbb{R}^n$ tal que:*

- $\langle \nabla g_i(x^*), w \rangle < 0, \quad \forall i \in \mathcal{A}(x^*)$,
- $\langle \nabla h_j(x^*), w \rangle = 0, \quad \forall j \in [p]$.

La condición de calificación (MFCQ) es más débil que la condición de calificación (LICQ), por lo que se puede mostrar que: LICQ $\Rightarrow$ MFCQ.

2.1.1. Problemas convexos

Definición 10 (Conjunto convexo)**.** *Un conjunto C se dice convexo si para cualquier par de puntos $x, y \in C$ y $\lambda \in [0, 1]$ se cumple:*

$$\lambda x + (1 - \lambda) y.$$

Definición 11 (Función convexa)**.** *Sea X un espacio Euclídeo, $f : X \to (-\infty, \infty]$. Entonces:*

- *f se dice convexa si satisface:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y), \quad \forall x, y \in X,\ \lambda \in [0,1]. \tag{2.8}$$

- *f se dice estrictamente convexa si satisface:*

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda)f(y), \quad \forall x, y \in X,\ \lambda \in)(0,1). \tag{2.9}$$

- *f se dice fuertemente convexa con módulo $\sigma > 0$ si satisface:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - \frac{\sigma}{2}\lambda(1-\lambda)\|x-y\|_2^2, \quad \forall x, y \in X,\ \lambda \in [0,1]. \tag{2.10}$$

Es fácil probar que si una función f es fuertemente convexa con módulo $\sigma > 0$, entonces f es estrictamente convexa. Además, para caracterizar la fuerte convexidad se utilizan los siguientes criterios:

- $f : X \to (-\infty, \infty]$ es fuertemente convexa con módulo $\sigma > 0$ si y solo la función $f(\cdot) - \frac{\sigma}{2}\|\cdot\|_2^2$ es convexa [Beck, 2017].
- Si f es dos veces diferenciable si el $dom(f)$ es convexo y su Hessiana cumple:

$$\nabla^2 f(x) \succeq \sigma \mathbb{I}$$

para todo $x \in dom(f)$ [Boyd and Vandenberghe, 2004].

Teorema 4 ([Beck, 2017], pág. 117)**.** *Sea X un espacio Euclídeo, $g : X \to -(\infty, \infty]$ una función convexa y $f : X \to (-\infty, \infty]$ una función fuertemente convexa de módulo $\sigma > 0$ y . Entonces, $f + g$ es fuertemente convexa de módulo $\sigma > 0$.*

Teorema 5 ([Beck, 2017], pág. 122)**.** *Sea X un espacio Euclídeo, $f : X \to (-\infty, \infty]$ una función propia, cerrada y fuertemente convexa de módulo $\sigma > 0$. Entonces, f tiene un único minimizador.*

Dado el siguiente problema:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.a.} \quad & x \in \Omega. \end{aligned} \tag{2.11}$$

donde $\Omega = \{x \in \mathbb{R}^n : g(x) \leq \mathbf{0}_m, h(x) = \mathbf{0}_p\}$ y $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^p$ funciones continuamente diferenciables. Se dice que (2.11) es un problema de optimización convexo si f es una función convexa y Ω es un conjunto convexo.

2.1.2. Método BFGS

El algoritmo BFGS (Broyden, Fletcher, Goldfarb y Shanno) utiliza una técnica de búsqueda local para la optimización. Pertenece a una categoría de métodos conocidos como métodos quasi-Newton. La principal idea detrás de este método es aproximar la matriz Hessiana de la función objetivo sin necesidad de calcularla directamente. En lugar de utilizar la matriz Hessiana real, este algoritmo utiliza una estimación que se actualiza iterativamente a medida que se exploran diferentes puntos en el espacio de búsqueda.

A continuación, se muestra el algoritmo general del método BFGS para resolver el siguiente problema de optimización:

$$\min_{x \in \mathbb{R}^n} \quad f(x). \tag{2.12}$$

Algoritmo 1 BFGS

Input: Punto de inicio x_0, aproximación inicial de la inversa de la matriz Hessiana H_0, tolerancia $\epsilon > 0$.
Output: $\hat{x}$ óptimo.

1: $k \leftarrow 0$;
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Calcular la dirección de búsqueda: $p_k = -H_k \nabla f(x_k)$
α_k se puede obtener con búsqueda lineal
4: Calcular: $x_{k+1} = x_k + \alpha_k p_k$
5: Definir: $s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad \rho_k = \frac{1}{y_k^\top s_k}$
6: Calcular: $H_{k+1} = (I - \rho_k s_k y_k^\top) H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top$
7: $k \leftarrow k + 1$
8: **end while**

Teorema 6 (Convergencia BFGS, [Nocedal and Wright, 2006], pág. 158)**.** *Supongamos que la función f es de clase C^2 y que las iteraciones generados por el algoritmo (1) convergen a un mínimo x^*. Además, que la Hessiana $\nabla^2 f(x^*)$ es Lipschitz continua en una vecindad de x^*; y además, $\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty$. Entonces la sucesión $\{x_k\}$ converge a x^* con una tasa superlineal, es decir,*

se cumple:

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Una desventaja del método BFGS es que se debe almacenar en cada iteración la aproximación de la matriz Hessiana H_k, que por lo general es una matriz densa. Por lo cual, esto no es viable para problemas de a gran escala. En este sentido, se propone un algoritmo alternativo llamado Limited-Memory BFGS o en su forma abreviada L-BFGS.

Algoritmo 2 L-BFGS

Input: Punto de inicio x_0, aproximación inicial de la inversa de la matriz Hessiana H_0, tolerancia $\epsilon > 0$, $m > 0$, $c_1 \in (0, 1)$, $c_2 \in (c_1, 1)$.
Output: $\hat{x}$ óptimo.

1: $k \leftarrow 0$;
2: **while** $\|\nabla f(x_k)\| > \epsilon$ **do**
3: Calcular la dirección de búsqueda: $p_k = -H_k \nabla f(x_k)$
α_k satisface las condiciones de Wolfe:
$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top p_k$
$\nabla f(x_k + \alpha_k p_k)^\top p_k \geq c_2 \nabla f(x_k)^\top p_k$
4: Calcular: $x_{k+1} = x_k + \alpha_k p_k$
5: **if** k>m **then**
6: Descartar el par de vectores $\{s_{k-m}, y_{k-m}\}$ del almacenamiento.
7: **end if**
8: Calcular: $s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.
9: $k \leftarrow k + 1$
10: **end while**

El paso 3 del algoritmo (2) se puede calcular con el siguiente algoritmo:

Algoritmo 3 L-BFGS two-loop recursion

1: $q \leftarrow \nabla f(x_k)$
2: **for** $i = k-1, k-2, \ldots, k-m$ **do**
3: $\alpha_i \leftarrow \rho_i s_i^\top q$
4: $q \leftarrow q - \alpha_i y_i$
5: **end for**
6: $r \leftarrow H_0$
7: **for** $i = k-m, k-m+1, \ldots, k-1$ **do**
8: $\beta \leftarrow \rho_i y_i^\top r$
9: $r \leftarrow r + s_i(\alpha_i - \beta)$
10: **end for**
11: Finalizar con $H_k \nabla f(x_k) = r$.

2.2. Elementos del aprendizaje estadístico

Dentro del marco del aprendizaje estadístico, para [Shwartz and David, 2014] un algoritmo de aprendizaje tiene los siguientes componentes:

- **Entrada**

  Dominio: Un conjunto arbitrario $\mathcal{X}$. Es el conjunto de objetos de interés los cuales se desea etiquetar. Al vector $x \in \mathcal{X}$ se le conoce como vector de características. A los elementos de este dominio se los conoce como instancias y a $\mathcal{X}$ como espacio de instancias.

  Etiquetas: Un conjunto $\mathcal{Y}$ que contiene las etiquetas de cada instancia del dominio $\mathcal{X}$. Generalmente, para un problema de clasificación binario se utilizan las etiquetas {0,1} o {-1,1}.

  Datos de Entrenamiento: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ una sucesión finita de pares en $\mathcal{X} \times \mathcal{Y}$, es decir, una sucesión de elementos del dominio con su correspondiente etiqueta. S también es conocido como conjunto de entrenamiento.

- **Salida**

  El algoritmo de aprendizaje genera una regla de predicción $h : \mathcal{X} \to \mathcal{Y}$ también conocida como predictor, hipótesis o clasificador. Este predictor h puede ser usado para predecir las etiquetas de los elementos de un nuevo dominio.

- **Modelo de generación de datos**

  Se asume que las instancias son generadas por una distribución de probabilidad desconocida $\mathcal{D}$ y que existe una función de etiquetado $f : \mathcal{X} \to \mathcal{Y}$, tal que $f(x_i) = y_i$ para todo $i = 1, \ldots, n$, que también es desconocida para el algoritmo de aprendizaje.

- **Medida de éxito**

  Se define el error de un predictor $h : \mathcal{X} \to \mathcal{Y}$ como:

$$L_{\mathcal{D},f}(h) := \mathbb{P}_{x \sim \mathcal{D}}[h(x) \neq f(x)] := \mathcal{D}(\{x : h(x) \neq f(x)\}). \tag{2.13}$$

  Es decir, es la probabilidad de escoger aleatoriamente una instancia $x \in \mathcal{X}$ tal que $h(x) \neq f(x)$. También se lo conoce como error de generalización, riesgo o error verdadero.

2.2.1. Minimización de riesgo empírico

El objetivo del algoritmo de aprendizaje es encontrar el predictor $h_S : \mathcal{X} \to \mathcal{Y}$ (el subíndice enfatiza que el predictor depende del conjunto de entrenamiento), el cual minimice

el error con respecto a $\mathcal{D}$ y f. De este modo, ya que $\mathcal{D}$ y f son desconocidas no se puede calcular directamente el error de generalización. Así, una forma alternativa es obtener el error de entrenamiento a través de:

$$L_S(h) := \frac{|\{h(x_i) \neq y_i, \forall i = 1, \ldots, n\}|}{n}. \tag{2.14}$$

Este error también se lo conoce como error empírico o riesgo empírico. Así, el encontrar un predictor h que minimice el error de entrenamiento $L_S(h)$ se conoce como minimización de riesgo empírico o ERM por sus siglas en inglés.

Minimización de riesgo empírico con sesgo inductivo

Se ha demostrado que la minimización de riesgo empírico (ERM) puede llevar al problema de sobreajuste. En este sentido, una solución común es aplicar la minimización de riesgo empírico en un espacio de búsqueda restringido.

Formalmente, el algoritmo de aprendizaje debe escoger un conjunto de predictores llamado clase de hipótesis, el cual se lo representar por $\mathcal{H}$. Entonces, dada una clase $\mathcal{H}$ y un conjunto de entrenamiento S, el $ERM_{\mathcal{H}}$ del algoritmo de aprendizaje usa el ERM para escoger un predictor $h \in \mathcal{H}$ con el error más bajo posible sobre S.

$$ERM_{\mathcal{H}}(S) \in \underset{h \in \mathcal{H}}{\arg\min}\, L_S(h). \tag{2.15}$$

Es decir, estamos sesgando la solución a un conjunto finito de predictores (sesgo inductivo).

Definición 12 (Supuesto de realizabilidad)**.** *Existe un $h^* \in \mathcal{H}$ tal que $L_{\mathcal{D},f}(h^*) = 0$.*

Esta definición implica que para cada hipótesis ERM se tiene $L_S(h_s) = 0$. Además, se asume que los elementos del conjunto de entrenamiento S son independientes e idénticamente distribuidos de acuerdo a una distribución $\mathcal{D}$ y se denota por $S \sim \mathcal{D}^n$, donde n es el tamaño de S. Por lo tanto, el error de generalización $L_{\mathcal{D},f}(h_S)$ se lo puede ver como una variable aleatoria.

Además, se puede mostrar que si $\mathcal{H}$ es una clase de hipótesis finita, entonces $ERM_{\mathcal{H}}$ no se sobreajustará, siempre que se base en una muestra de entrenamiento suficientemente grande.

Teorema 7 (No-Free-Lunch, [Shalev-Shwartz and Ben-David, 2014], pág. 37)**.** *Sea A un algoritmo de aprendizaje cualquiera para una tarea de clasificación binaria $\{0,1\}$ sobre un dominio $\mathcal{X}$. Además, sea n un número menor a $\frac{|\mathcal{X}|}{2}$ que representa el tamaño del conjunto de entrenamiento. Entonces, existe una distribución $\mathcal{D}$ sobre $\mathcal{X} \times \{0,1\}$ tal que:*

1. *Existe una función $f : \mathcal{X} \to \{0,1\}$ con $L_{\mathcal{D}}(f) = 0$.*
2. *Con probabilidad de al menos $\frac{1}{7}$ sobre una selección de $S \sim \mathcal{D}^n$ tenemos que $L_{\mathcal{D}}(A(S)) \geq \frac{1}{8}$.*

Este teorema establece que para cada algoritmo de aprendizaje, existe una tarea para la cual falla, aunque esa tarea puede ser aprendida con éxito por otro algoritmo de aprendizaje.

Por otra parte, para saber como escoger una buena clase de hipótesis, descomponemos el error de un predictor $ERM_{\mathcal{H}}$ en dos componentes como sigue. Sea h_S un predictor $ERM_{\mathcal{H}}$, entonces se puede escribir:

$$L_{\mathcal{D}}(h_S) = \epsilon_{app} + \epsilon_{est}. \tag{2.16}$$

donde, $\epsilon_{app} = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ es el error de aproximación y $\epsilon_{est} = L_{\mathcal{D}}(h_S) - \epsilon_{app}$ el error de estimación.

Error de aproximación: Es el menor valor de error de generalización que puede alcanzar un predictor en una clase de hipótesis específica. Este término mide cuanto sesgo inductivo se tiene. Además, no depende del tamaño del conjunto de entrenamiento. Si se agranda la clase de hipótesis, entonces disminuye el error de aproximación.

Error de estimación: Es la diferencia entre el error obtenido por un predictor ERM y el error de aproximación. La calidad de esta estimación depende del tamaño del conjunto de entrenamiento y del tamaño o complejidad de la clase de hipótesis representada por $|\mathcal{H}|$.

2.2.2. Función de costo generalizada

Dado un conjunto $\mathcal{H}$, que cumple el rol de nuestra clase de hipótesis o modelo, y un dominio Z. Sea una función $\ell : \mathcal{H} \times Z \to \mathbf{R}_+$ a la cual la llamamos función de costo. Para los problemas de aprendizaje supervisado, el dominio $Z = \mathcal{X} \times \mathcal{Y}$.

Función de riesgo: Se define como el costo esperado de un predictor $h \in \mathcal{H}$ con respecto a la distribución de probabilidad $\mathcal{D}$ sobre Z.

$$L_{\mathcal{D}}(h) := \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]. \tag{2.17}$$

Función de riesgo empírico: Se define como el costo esperado sobre un conjunto S.

$$L_S(h) := \frac{1}{n}\sum_{i=1}^{n} \ell(h, z_i). \tag{2.18}$$

2.2.2.1. Función hinge-loss

La función *hinge-loss* es utilizada en los problemas de aprendizaje automático supervisado de clasificación, especialmente en el contexto de las máquinas de soporte vectorial (SVM). Esta función se define como:

$$\ell(w, (x, y)) := \text{máx}\{0, 1 - y\langle w, x\rangle\}. \tag{2.19}$$

donde $\langle \cdot, \cdot \rangle$ representa el producto interno en $\mathbb{R}^d$ y $y \in \mathcal{Y}$. Esta función evalúa la precisión de las clasificaciones al penalizar las predicciones incorrectas, siendo especialmente efectiva en problemas donde la separación de clases no es trivial.

La función (2.19) también puede ser expresada como una función a trozos de la siguiente manera:

$$\ell(w, (x, y)) := \begin{cases} 0, & \text{si } y\langle w, x\rangle \geq 1, \\ 1 - y\langle w, x\rangle, & \text{si } y\langle w, x\rangle < 1. \end{cases} \tag{2.20}$$

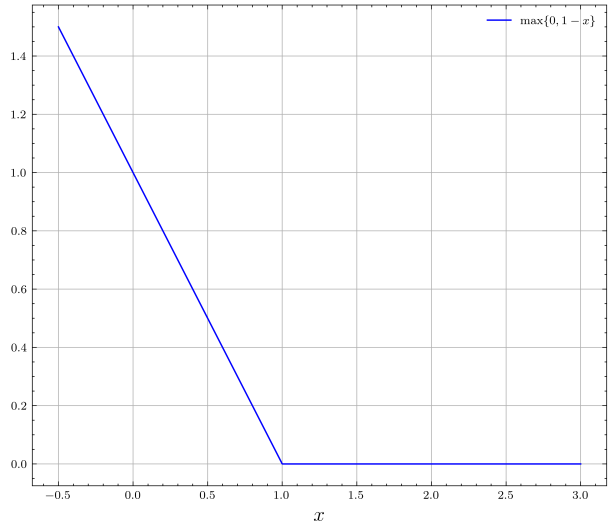


Figura 2.1: Función *hinge-loss* en un espacio de una dimensión.

Dentro de las principales propiedades se puede mencionar que la función *hinge-loss* es convexa, Lipschitz continua y no diferenciable.

2.2.3. Máquinas de soporte vectorial - SVMs

Las máquinas de soporte vectorial es una herramienta propuesta por [Vapnik, 1995] hace más de dos décadas, la cual es ampliamente utilizada en el aprendizaje automático supervisado. Esta herramienta tiene aplicaciones en problemas clásicos de regresión, como la predicción de series temporales [Müller et al., 1997] y problemas de clasificación como reconocimiento de objetos [Blanz et al., 1996] o reconocimiento facial [Osuna et al., 1997] por mencionar algunos.

Además, de acuerdo con [Shwartz and David, 2014], las máquinas de soporte vectorial plantean desafíos de complejidad computacional y complejidad de muestra cuando se trata de espacios de características con alta dimensionalidad. En términos generales, la complejidad computacional puede ser vista como el número de operaciones que requiere el algoritmo (en el peor de los casos) para resolver el problema, mientras que la complejidad de muestra se refiere a la cantidad de muestras de entrenamiento que se requieren para garantizar una solución aproximadamente correcta.

En este trabajo nos centramos en el problema de clasificación de dos clases o también llamado problema de clasificación binaria.

2.2.3.1. L_2-SVM

Definición 13 (Conjunto linealmente separable)**.** *Sea* $S = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\}$ *un conjunto de entrenamiento, donde cada vector de características* $\mathbf{x_i} \in \mathbb{R}^d$ *y las etiquetas* $y_i \in \{-1, +1\}^n$*. Se dice que el conjunto de entrenamiento es linealmente separable si existe un semiespacio* $(\mathbf{w}, b)$*, con* $\mathbf{w} \in \mathbb{R}^d$ *y* $b \in \mathbb{R}$*, tal que:*

$$y_i = sign(\langle \mathbf{w}, \mathbf{x_i} \rangle + b), \quad \forall i = 1, \ldots, n.$$

De forma alternativa, esta condición puede ser escrita como:

$$y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) > 0, \quad \forall i = 1, \ldots, n.$$

Dado un conjunto $S = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\}$ linealmente separable, entonces este puede ser separado por un hiperplano de la forma:

$$\langle \mathbf{w}, \mathbf{x_i} \rangle + b = 0. \tag{2.21}$$

Sin embargo, el conjunto S puede ser separado por infinitos hiperplanos de la forma (2.21), lo que nos lleva a buscar un hiperplano óptimo, el cual es definido por [Vapnik, 1995]

como aquel que separa sin error y la distancia entre el vector más cercano al hiperplano es máxima. De esta manera, se usa la siguiente forma canónica para describir el hiperplano de separación:

$$\langle \mathbf{w}, \mathbf{x_i} \rangle + b \geq 1, \quad \text{si } y = 1,$$
$$\langle \mathbf{w}, \mathbf{x_i} \rangle + b \leq -1, \quad \text{si } y = -1.$$

De forma compacta se puede escribir las desigualdades anteriores como sigue:

$$y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1, \quad \forall i = 1, \ldots, n. \tag{2.22}$$

Hard-Margin SVM

Una de las formulaciones más conocidas es Hard-SVM, la cual asume que el conjunto S es linealmente separable. Esta formulación usa la noción de margen, el cual se define como la distancia mínima entre un punto del conjunto de entrenamiento y el hiperplano. Esta formulación retorna un hiperplano que separa el conjunto de entrenamiento con el margen más grande posible como se muestra en la Figura 2.2.

Los vectores que están sobre los márgenes se los denomina vectores de soporte y cumplen lo siguiente para cada $i = 1, \ldots, n$:

$$\langle \mathbf{w}, \mathbf{x_i} \rangle + b = +1, \quad \text{si está sobre el margen positivo,}$$
$$\langle \mathbf{w}, \mathbf{x_i} \rangle + b = -1, \quad \text{si está sobre el margen negativo.}$$

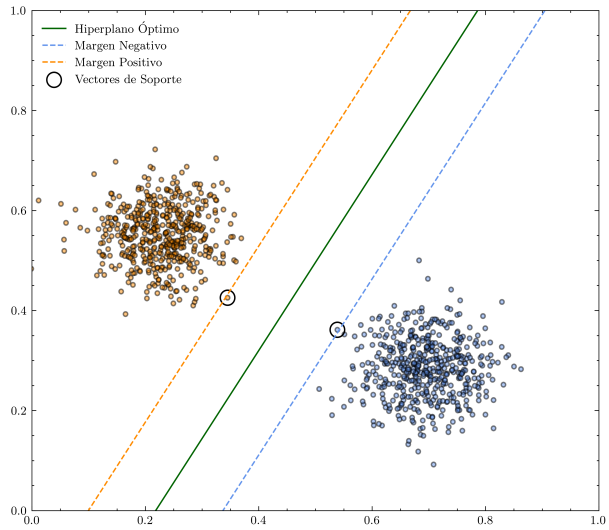


Figura 2.2: Hard-Margin SVM con datos linealmente separables.

Formalmente, Hard-Margin SVM se puede escribir como un problema de optimización de la siguiente forma:

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s.t.} \quad & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1, \quad \forall i = 1, \ldots, n. \end{aligned} \tag{2.23}$$

De este modo, resolviendo el problema 2.23 se obtiene el hiperplano óptimo de la forma $\langle \hat{\mathbf{w}}, \mathbf{x_i} \rangle + \hat{b} = 0$. Las nuevas etiquetas para un nuevo conjunto de observaciones x_j con $j = 1, \ldots, l$, se las obtiene a través de la siguiente ecuación:

$$y_j = sign(\langle \hat{\mathbf{w}}, \mathbf{x_j} \rangle + \hat{b}), \quad \forall j = 1, \ldots, l.$$

Soft-Margin SVM

Uno de los problemas con la formulación Hard-Margin SVM es que, por lo general, los conjuntos de datos no siempre son linealmente separables.

En este sentido, [Cortes and Vapnik, 1995] proponen una nueva formulación llamada Soft-Margin SVM, la cual introduce variables auxiliares no negativas $\xi_i \geq 0$, con $i = 1, \ldots, n$ que ayudan a relajar la restricción del problema 2.23 cuando sea necesario, es decir, cuando aquella restricción sea violada.

En la Figura 2.3 se puede observar un conjunto de entrenamiento que no es linealmente separable. Asimismo, se puede ver que el hiperplano óptimo no logra discriminar todos los datos correctamente en sus clases respectivas, es decir, esas malas clasificaciones violan la restricción del problema 2.23.

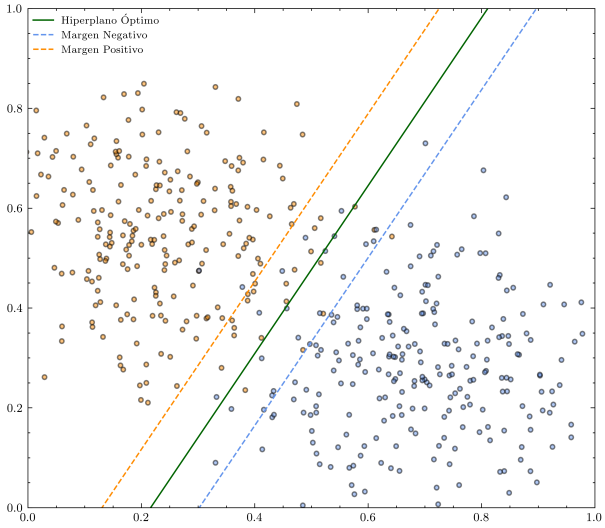


Figura 2.3: Soft-Margin SVM con datos que no son linealmente separables.

Por lo tanto, la formulación Soft-Margin SVM puede ser escrita como un problema de optimización de la siguiente forma:

$$\begin{aligned} \min_{w} \quad & \frac{\alpha}{2}\|w\|_2^2 + \sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad \forall i = 1, \ldots, n. \end{aligned} \tag{2.24}$$

donde $\alpha > 0$.

Así, la función objetivo es modificada por el término $\sum_{i=1}^{n} \xi_i$, que representa una cota superior para el número de errores de clasificación. Además, α es un parámetro de regularización que debe ser escogido por el usuario, también conocido como hiperparámetro, el cual sirve como un parámetro de compensación entre la maximización del margen y la minimización del error de clasificación. Un valor grande de α corresponde a asignar una penalidad mayor a la maximización del margen.

De este modo, la formulación Soft-Margin SVM o también conocida por L_2-SVM, puede ser presentada como el siguiente problema de minimización con la forma (1.3):

$$\min_{w} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(w, (x_i, y_i)) + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2. \tag{2.25}$$

donde $\ell(w, (x_i, y_i)) = \max\{0, 1 - y_i\langle w, x_i\rangle\}$ es la función *hinge-loss* y $R_S(w) = \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2$ se la conoce como la regularización Ridge o regularización de Tikhonov.

Cabe mencionar que la función objetivo del problema de minimización (2.25) es estrictamente convexa, ya que es la suma de una función estrictamente convexa como la norma L_2 y una función convexa como la función *hinge-loss*.

2.2.3.2. L_1-SVM

Esta es una variante a la formulación tradicional *Soft-Margin* SVM, la cual incorpora un término de regularización L_1 que promueve la dispersión de los coeficientes del hiperplano w estimado. Esta propiedad de dispersión de los coeficientes hace que esta formulación sea útil en la selección de variables, particularmente, en conjuntos de datos con alta dimensionalidad, reduciendo el impacto de variables irrelevantes o redundantes [Hastie et al., 2016].

De este modo, el problema de optimización L_1-SVM se formula de la siguiente manera:

$$\min_{w} \quad \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i\langle w, x_i\rangle\} + \beta\sum_{j=1}^{d+1} |w_j|. \tag{2.26}$$

donde $\beta > 0$ y $R_S(w) = \beta \sum_{j=1}^{d} |w_j|$ se la conoce como la regularización Lasso. Al ajustar del hiperparámetro β se puede controlar el equilibrio entre la dispersión de los coeficientes del hiperplano y el error de clasificación.

La principal diferencia entre las regularizaciones L_1 y L_2 en el contexto del L_1-SVM radica en su enfoque hacia la dispersión de los coeficientes del hiperplano [Hastie et al., 2016]. Mientras que la regularización L_1 fomenta la dispersión al reducir algunos coeficientes del hiperplano a cero, la regularización L_2 permite que todos los coeficientes del hiperplano sean pequeños pero diferentes de cero.

De acuerdo al trabajo de [Wang et al., 2006], la regularización L_1 tiene algunas ventajas sobre la regularización L_2 bajo ciertos escenarios como por ejemplo las variables redundantes. Sin embargo, se mencionan dos limitantes de esta regularización:

1. Cuando las variables se encuentran altamente correlacionadas en el conjunto de entrenamiento y todas son relevantes, la regularización L_1 tiende a tomar una de ellas (cualquiera) y el resto las hace cero.

2. En el caso cuando $d \gg n$ (el número de variables es mucho mayor que el número de observaciones) la regularización L_1 puede identificar, como máximo, n coeficientes ajustados distintos de cero, lo cual es poco probable para este escenario.

Además, es importante mencionar que tanto la regularización L_1 como la L_2 se pueden pueden beneficiar de la normalización o escalado de datos. Sin embargo, el impacto de la normalización en el rendimiento del modelo puede ser más crítico para la regularización L_1 en comparación con la regularización L_2.

2.2.3.3. L_2L_1-SVM

Esta formulación fue propuesta por [Zou and Hastie, 2005] con la finalidad de corregir las dos limitaciones que tiene el problema L_1-SVM. Este nuevo concepto es una mezcla de las regularizaciones L_1 y L_2, combinando las mejores características de ambas. Así, el problema de optimización L_2L_1-SVM o también conocido como SVM doblemente regularizado se presenta a continuación:

$$\min_{w} \quad \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i \langle w, x_i \rangle\} + \frac{\alpha}{2} \sum_{j=1}^{d+1} w_j^2 + \beta \sum_{j=1}^{d+1} |w_j|. \tag{2.27}$$

donde $\alpha; \beta$ son hiperparámetros positivos y $R_S(w) = \frac{\alpha}{2} \sum_{j=1}^{d+1} w_j^2 + \beta \sum_{j=1}^{d+1} |w_j|$ se la conoce como la regularización Elastic Net.

Por una parte, la regularización L_1 fomenta la dispersión al seleccionar tan solo un subconjunto de las variables originales más relevantes y; por otro lado, la regularización L_2 ayuda a prevenir el sobreajuste. Es decir, con esta combinación no solo selecciona las características más informativas sino que también evita que el modelo se vuelva demasiado complejo y sobreajuste los datos de entrenamiento.

Las nuevas ventajas de esta formulación son que ahora se pueden seleccionar juntos grupos de variables correlacionadas y que el número de variables seleccionadas ya no está limitado por n en el caso que el conjunto de entrenamiento tenga alta dimensionalidad.

Además, [Wang et al., 2006] menciona que el rol de la regularización L_1 es permitir la selección de variables, y el rol de la regularización L_2 es ayudar a que se seleccionen los grupos de variables correlacionadas. Como se mencionó anteriormente, la regularización L_2 tiende a hacer que las variables altamente correlacionadas tengan los coeficientes ajustados muy similares, lo cual es un efecto de agrupación.

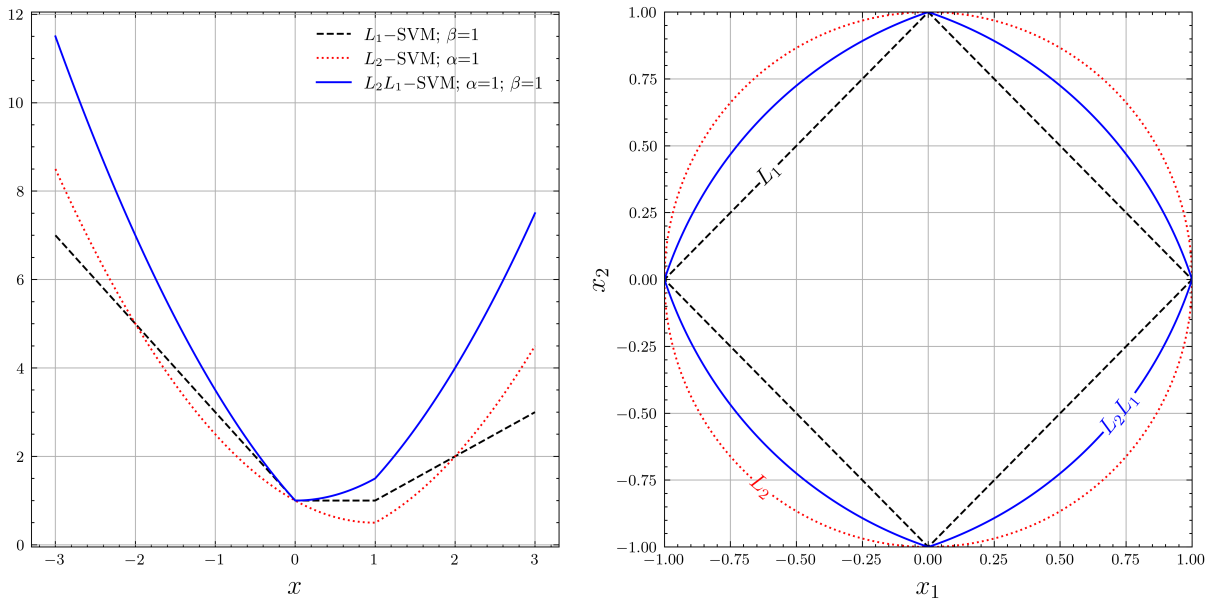


Figura 2.4: Izquierda: Funciones objetivo de las diferentes formulaciones SVM. Derecha: Ilustración geométrica (gráfico de contorno en $\mathbb{R}^2$) de los diferentes tipos de regularizadores; Reg. L_1: $\|x\|_1 = 1$, Reg. L_2: $\|x\|_2^2 = 1$, Reg. L_2L_1: $\|x\|_2^2 + \|x\|_1 = 1$

La Figura 2.4 compara las funciones objetivo de los tres problemas de optimización mostrados [2.25, 2.26, 2.27] y sus respectivas regularizaciones. En la parte izquierda, se puede ver que las funciones objetivo de los problemas L_2-SVM y L_2L_1-SVM son estrictamente convexas, mientras que la función objetivo del problema L_1-SVM es convexa.

2.3. Optimización binivel

Históricamente este problema fue introducido en 1934 por [Von-Stackelberg, 2011], el cual dio una descripción general en el aspecto de la teoría de juegos. Sin embargo, en 1973 los autores [Bracken and MacGill, 1973] presentan una formulación matemática general de este problema, aunque la noción de "programación binivel" puede ser vinculada probablemente a [Candler and Norton, 1977].

Desde el punto de vista de [Zhang et al., 2015], el problema de optimización binivel se lo puede ver como un caso especial de los problemas de optimización multi-nivel, en el cual existen solo dos niveles de optimización. De manera concisa, se puede decir que los problemas binivel son problemas que tienen por objetivo optimizar una función objetivo real (problema de optimización nivel superior) sujeta a restricciones relacionadas con un conjunto de soluciones óptimas de otro problema de optimización (problema de optimización nivel inferior). En otras palabras, el problema de optimización de nivel superior depende de la solución del problema de optimización de nivel inferior.

De forma general, según [Dempe, 2002] un problema de optimización binivel puede ser formulado como:

$$
\begin{aligned}
&\min_{x \in X} \quad F(x,y) && \text{(Nivel superior)},\\
&\text{s.a.} \quad G(x,y) \leq 0,\\
&\qquad \min_{y \in Y} \quad f(x,y), && \text{(Nivel inferior)},\\
&\qquad \text{s.a.} \quad g(x,y) \leq 0.
\end{aligned}
$$

donde $x \in X \subset \mathbb{R}^m$, $y \in Y \subset \mathbb{R}^n$ y las funciones $f, F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$, $G : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p$ y $g : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^q$ son continuas y dos veces diferenciables.

Entre las diversas aplicaciones de la optimización binivel y multi-nivel [Vicente and Calamai, 1994] y [Dempe and Zemkoho, 2020] destacan las siguientes:

- Problemas en teoría de juegos.
- Problemas de diseño óptimo en ingeniería.
- Problemas de segmentación y reconstrucción de imágenes.
- Problemas de transporte y estimación de viajes de demanda.
- Problemas de planificación: Servicios eléctricos, políticas agrícolas y producción.

- Problemas de aprendizaje automático, métodos de aprendizaje estadístico y aprendizaje de parámetros.
- Problemas de administración: Colocación de crédito, problema de fijación de precios (pricing) y seguros de salud.

En el contexto de la selección de variables, la estrategia en esta investigación es combinar dos problemas de optimización de manera jerárquica de la siguiente manera:

$$\min_{\beta} \quad J(\beta, \hat{w}(\beta)) \tag{2.28a}$$

$$\text{s.a.} \quad \hat{w}(\beta) \in \arg\min_{w} E(w, \beta). \tag{2.28b}$$

donde, en el nivel superior la función $J(\cdot)$ evalúa el error de validación; y en el nivel inferior la función de costo regularizada $E(\cdot)$ entrena el modelo de aprendizaje automático L_2L1-SVM. Más adelante, se estudiarán estas funciones haciendo énfasis en sus propiedades y aproximaciones.

Para resumir, esta sección se centra en los conceptos fundamentales que se requieren para abordar el problema de selección de variables a través de un enfoque binivel, haciendo uso de las máquinas de soporte vectorial con regularización L_2L_1. Posteriormente, en el Capítulo 4 se extenderá el estudio del problema de optimización binivel L_2L1-SVM general y su versión aproximada, con el fin de establecer las condiciones de optimalidad.

Capítulo 3

Revisión Estado del Arte

En esta sección se revisa algunas formas de caracterizar la solución del problema de optimización binivel (1.1), así como los algoritmos numéricos que se usan para aproximar la misma desde diferentes aplicaciones.

En el trabajo de [Crockett and Fessler, 2021] se presenta una revisión sobre métodos binivel para la reconstrucción de imágenes. Dentro de esta revisión, los autores:
[Samuel and Tappen, 2009], [Chen et al., 2014], [Gould et al., 2016] y [Holler et al., 2018] presentan una metodología binivel basada en el gradiente de la función objetivo del nivel superior (1.1a), también conocido como hipergradiente, para resolver el problema de optimización binivel. Los autores presentan dos métodos para caracterizarlo:

1. El primer método propuesto se basa en el teorema de la función implícita (IFT) y tiene por objetivo caracterizar el gradiente del operador solución con respecto al hiperparámetro β y de este modo obtener una caracterización del hipergradiente.

2. El segundo método adopta la perspectiva de las condiciones Karush-Kuhn-Tucker (KKT) del problema binivel, las cuales establecen que en un punto óptimo, el gradiente del Lagrangiano con respecto al hiperparámetro β es igual a cero.

Ambos métodos tienen las siguientes suposiciones:

H1. Los problemas de optimización superior e inferior no contienen restricciones de desigualdad.

H2. El problema de nivel inferior no contiene ninguna restricción.

H3. La función objetivo del nivel inferior (2.28b) es dos veces diferenciable con respecto al parámetro w y una vez diferenciable con respecto al hiperparámetro β.

H4. La matriz Hessiana $\nabla^2_{ww} E(w, \beta)$ es invertible para todo (w, β).

Dentro de los métodos resolución numérica se mencionan dos tipos. Primero, los algoritmos lazo doble (Hyperparameter Optimization with Approximate Gradient - HOAG [Pedregosa, 2016], Bilevel Approximation - BA y Bilevel Stochastic Approximation - BSA [Ghadimi and Wang, 2018]) que tienen la siguiente forma:

Algoritmo 4 Lazo Doble

Input: $J(\beta, w(\beta))$, $E(w, \beta)$, $w^{(0)}$, $\beta^{(0)}$, α_J.
Output: $\hat{\beta}$ óptimo.
1: **for** $u = 0, 1, \ldots$ **do**
2: $\boldsymbol{t = 0}$
3: **while** se cumpla un criterio de parada **do**
4: Paso de optimización del nivel inferior: $\boldsymbol{w^{(t+1)} \in \arg\min_w \{E(w^{(t)}, \beta^{(t)}; X, y)\}}$
5: $\boldsymbol{t = t + 1}$
6: **end while**
7: Calcular el hipergradiente: $\boldsymbol{g = \nabla J(\beta^{(u)}, w(\beta)^{(t)})}$
8: Actualizar: $\boldsymbol{\beta^{(u+1)} = \beta^{(u)} - \alpha_J g}$
9: **end for**

Estos algoritmos de lazo doble implican:

- Optimizar el nivel inferior para un cierto número de iteraciones o cierta tolerancia de convergencia (este paso puede implicar algunas iteraciones internas).
- Calcular el hipergradiente $\nabla J(\cdot)$.
- Tomar un paso de gradiente en el hiperparámetro β.
- Iterar.

Segundo, los algoritmos de lazo único (Two-Timescale Stochastic Approximation - TTSA [Hong et al., 2020] y Single-Timescale - STABLE [Chen et al., 2021]), son aquellos que involucran un lazo con cada iteración que contiene un paso de gradiente tanto para la variable de optimización del nivel inferior, w, como para la variable de optimización del nivel superior β y tienen la forma:

Algoritmo 5 Lazo Único

Input: $J(\beta, w(\beta))$, $E(w, \beta)$, $w^{(0)}$, $\beta^{(0)}$, α_J, α_E.
Output: $\hat{\beta}$ óptimo.
1: **for** $u = 0, 1, \ldots$ **do**
2: Actualizar: $\boldsymbol{w^{(u+1)} = w^{(u)} - \alpha_E \nabla E(w^{(u)}, \beta^{(u)})}$
3: Calcular el hipergradiente: $\boldsymbol{g = \nabla J(\beta^{(u)}, w(\beta)^{(u)})}$
4: Actualizar: $\boldsymbol{\beta^{(u+1)} = \beta^{(u)} - \alpha_J g}$
5: **end for**

En general, los algoritmos de lazo único destacan por su eficiencia computacional y resultan apropiados en escenarios donde el conjunto de datos de entrenamiento es grande. Por otro lado, los algoritmos de lazo doble sobresalen al proporcionar una mayor precisión superior al resolver el problema de optimización de nivel inferior.

Por su parte [Kunisch and Pock, 2013], en el contexto de estimación de parámetros en modelos variacionales proponen un enfoque de optimización binivel. El nivel superior se expresa por medio de una función de costo que penaliza los errores entre la solución del problema de nivel inferior y los datos verdaderos. El nivel inferior viene dado por el modelo variacional que está compuesto de un término de regularización y un término de fidelidad de datos. Los autores presentan diferentes formulaciones para el nivel inferior (L_1-model, L_2-model) con término de regularización escalar/vectorial y sus sistemas de optimalidad. Los algoritmos que se proponen para resolver el problema binivel son:

- Algoritmos Newton semi-suaves: Newton Learning - L_1, Newton Learning - L_2, Reduced Newton Learning for L_1, Iteratively Reweighted Learning L_2.
- Estos algoritmos tienen convergencia local superlineal.

En el mismo contexto de la eliminación de ruido de imágenes, [De los Reyes and Schönlieb, 2013] analizan una estrategia de aprendizaje binivel para encontrar los parámetros del modelo de ruido en espacios funcionales, a través de optimización no suave con restricciones que involucran ecuaciones en derivadas parciales. Para que el problema sea numéricamente manejable, emplean una versión regularizada del problema de nivel inferior utilizando una regularización de tipo Huber. Además, los valores óptimos de los parámetros se calculan numéricamente utilizando un método quasi-Newton, junto con algoritmos de tipo Newton semisuaves. Asimismo, [De los Reyes et al., 2017] proponen un enfoque de optimización binivel para el aprendizaje de parámetros en modelos de reconstrucción de imágenes de variación total de orden superior. Aparte del funcional de costo de mínimos cuadrados, proponen un funcional de costo basado en una regularización Huber de la seminorma TV. Para la solución numérica del problema de optimización binivel, se usa un algoritmo tipo quasi-Newton (BFGS) y Newton semi-suave combinado.

En el campo del aprendizaje automático, el trabajo de [Klatzer, 2014] aborda la resolución de un problema de optimización binivel aplicado a máquinas de soporte vectorial (SVM-C) en el contexto de clasificación. En este enfoque se incluye una regularización de tipo L_2. El objetivo es abordar este problema mediante la caracterización de la solución exacta a través de las condiciones de optimalidad. Por lo tanto, se exige que la función de costo sea dos

veces diferenciable. Cuando el conjunto de datos de entrenamiento es linealmente separable, en el nivel inferior se plantea el problema SVM para dos casos: uno donde el parámetro de regularización es escalar y otro donde es vectorial. Esto permite explorar diferentes enfoques y considerar la influencia del tipo de regularización en el rendimiento del modelo. Por el contrario, cuando el conjunto de datos de entrenamiento no es linealmente separable se establece en el nivel inferior el problema Kernel-SVM para los casos escalar y vectorial. En cuanto a los algoritmos de optimización, para encontrar la aproximación numérica del nivel inferior utiliza un método de primer orden, en específico, el algoritmo Fast Iterative Shrinkage-Thresholding - FISTA con búsqueda lineal [Beck and Teboulle, 2009]. Para resolver el problema de nivel superior se proponen dos algoritmos: Resilient Propagation - RPROP [Riedmiller and Braun, 1993] y LBFGS-B [Byrd et al., 1995].

Del mismo modo, [Li et al., 2022] abordan un problema de optimización binivel para encontrar el hiperparámetro óptimo en el contexto del problema de máquinas de soporte vectorial SVM para clasificación binaria. En el nivel superior, se busca minimizar el número promedio de clasificaciones incorrectas, mientras que en el nivel inferior se utiliza el SVM-C con parámetro de regularización L_2. Tanto el problema superior como inferior hacen uso de T conjuntos para la validación cruzada, lo que añade robustez a la solución. Sin embargo, los autores reformulan el problema Soft-Margin SVM convirtiéndolo en un problema de optimización convexo cuadrático con restricciones (relajación de la hipótesis $H2$ anteriormente descrita) al introducir variables de holgura. A continuación, establecen las condiciones Karush-Kuhn-Tucker (KKT) para convertirlo en un MPEC (Mathematical Program with Equilibrium Constraints). El método de solución propuesto es el algoritmo GRM (Global Relaxation Method). Otro estudio que también se basa en el enfoque de MPECs es el de [Kunapuli et al., 2008], donde abordan un problema de optimización binivel para la selección de parámetros utilizando L_0-SVM-C, L_1-SVM-C y L_2-SVM-C mediante la aplicación de validación cruzada; y también relaja la hipótesis $H1$ al incluir restricciones de caja en el problema de nivel superior. Para obtener la solución numérica, se emplean dos solvers: FILTER que utiliza un algoritmo de región de confianza y SNOPT que se basa en búsqueda lineal. Ambos hacen uso de la programación secuencial cuadrática (SQP) para la estimación de la solución.

Adicionalmente, el trabajo de [De los Reyes, 2023] investiga una familia de problemas de aprendizaje binivel en imágenes. En esta investigación, el problema de nivel inferior se define como un modelo variacional convexo con regularizadores dispersos no suaves de primer y segundo orden. En este contexto, el autor relaja la hipótesis de diferenciabilidad del problema de nivel inferior $H3$ y la hipótesis $H1$ al incluir restricciones de desigualdad en el problema de nivel superior. Además, aprovecha las propiedades geométricas de la reformulación primal-

dual del problema de nivel inferior e introduce variables auxiliares adecuadas para transformar el problema binivel original en un MPCC (Mathematical Program with Complementarity Constraints).

Por otro lado, [Gao et al., 2022] presentan el algoritmo VF-iDCA, basado en la descomposición de la función objetivo en partes convexas y cóncavas mediante una técnica de diferencia convexa con inexactitud. Este enfoque se apoya en las funciones de valor, que representan el valor esperado de la función objetivo en un rango de valores para los hiperparámetros. El VF-iDCA ha sido diseñado para abordar el desafío del ajuste de hiperparámetros en algoritmos de aprendizaje automático desde una perspectiva de optimización binivel. A diferencia de los algoritmos que dependen del cálculo del gradiente, el VF-iDCA no requiere que se cumplan las hipótesis $H3$ y $H4$. En otras palabras, este enfoque no exige que la función objetivo del problema de nivel inferior sea estrictamente convexa y diferenciable. En cuanto a su convergencia, el algoritmo VF-iDCA sigue un proceso secuencial, mejorando iterativamente la solución hasta alcanzar la convergencia.

Un trabajo destacado en el contexto de la selección de variables es el de [Guyon et al., 2002], donde se combinan el algoritmo de aprendizaje automático L_2-SVM con un proceso iterativo de eliminación de variables (*Recursive Feature Elimination*) al cual lo denomina RFE-SVM[1]. Este enfoque implica la eliminación iterativa de variables que presentan menor importancia para la tarea de clasificación, a través de los coeficientes del hiperplano w obtenido mediante la herramienta de aprendizaje automático L_2-SVM, u otras medidas de importancia como por ejemplo el discriminante lineal de *Fisher*. De este modo, se construye un nuevo hiperplano w con las variables restantes, obteniendo un conjunto de datos más compacto hasta que se cumpla un criterio de parada. La selección de un número predefinido de variables o el cambio en la función de costo cuando cae por debajo de un umbral determinado pueden ser usados como criterio de parada.

Otros autores como [Zhang et al., 2017] presentan un método de reducción de datos denominado Simultaneously Inactive Features and Samples - SIFS. Este método está diseñado para identificar de forma simultánea las variables y muestras inactivas del problema SVM de la forma (1.4). Además, permite una estimación precisa de la solución óptima tanto para el problema primal como para el dual del SVM, aprovechando la convexidad fuerte del mismo. La característica principal de este método es que realiza la identificación de las variables y muestras inactivas de manera estática. Es decir, se aplica el método SIFS una sola vez antes de la optimización. De hecho, el artículo demuestra que el método garantiza

[1]Se encuentra implementado en la librería Scikit-Learn (Machine Learning in Python). Los detalles se los puede encontrar en: Scikit-Learn: Feature Selection RFE

que todas las variables y muestras detectadas serán irrelevantes para los resultados, lo que implica que el modelo aprendido con los datos reducidos será idéntico al modelo aprendido con los datos completos.

Otra aproximación al problema de selección de variables se propone en [Wang et al., 2006], donde se utilizan las máquinas de soporte vectorial SVM con doble regularización (DrSVM). Esta regularización se la conoce como regularización Elastic Net; una mezcla de las regularizaciones L_1 y L_2. Los autores demuestran que el uso de esta regularización combinada, mejora dos limitaciones que se presentan al usar solamente la regularización L_1. Por una parte, fomenta la selección o eliminación de las variables altamente correlacionadas en conjunto; y por otra parte, mejora la selección de variables cuando se tienen conjuntos de datos de alta dimensionalidad. Se presentan algoritmos eficientes para la establecer una solución numérica del problema DrSVM. Para lograr la eficiencia de los algoritmos propuestos, se usa una versión regularizada de tipo Huber para la función *hinge-loss*.

Finalmente, el trabajo realizado por [Tang et al., 2018] aborda el problema de selección de variables agrupadas dentro del contexto de clasificación multiclase. Se utiliza el enfoque de máquinas de soporte vectorial SVM, donde el término de regularización contiene la norma de dispersión por grupos. Para resolver este desafío, emplean el algoritmo Alternating Direction Method of Multipliers - ADMM [Boyd et al., 2011], un método de ascenso dual que calcula de manera aproximada el gradiente de la función objetivo dual.

Los enfoques y algoritmos de optimización binivel presentados en esta sección brindan herramientas valiosas para abordar problemas de estimación de parámetros óptimos en diversos contextos como: el procesamiento de imágenes, clasificación de datos, selección de variables, entre otros. Estas metodologías proporcionan una caracterización detallada de la solución del problema y presentan métodos de aproximación numérica que resultan esenciales para el enfoque propuesto en este trabajo.

Nuestra propuesta utiliza una estrategia de optimización binivel para la selección de variables, donde el nivel superior evalúa el error de validación y el nivel inferior entrena el modelo de aprendizaje automático L_2L_1-SVM. Se utiliza una versión regularizada del problema de nivel inferior, lo cual permite aplicar algoritmos que han sido ampliamente estudiados y mejorados en la optimización de funciones diferenciables. A diferencia de otros trabajos de investigación, esta metodología realiza la búsqueda del mejor subconjunto de variables y el entrenamiento del modelo de aprendizaje automático en un único lazo mediante un algoritmo quasi-Newton. Además, se garantiza la optimalidad del hiperparámetro β, el cual es utilizado para construir un score que mide la importancia de cada una de las variables seleccionadas.

Capítulo 4

Metodología

La selección de variables tiene como propósito identificar las variables más relevantes en un conjunto de datos. Es decir, identificar aquellas variables que ejercen una mayor influencia en la capacidad predictiva en un modelo de aprendizaje automático. Para lograr este objetivo, en este trabajo utilizamos un método integrado de selección de variables, el cual realiza dicha selección dentro de un algoritmo de aprendizaje automático. En particular, usamos el problema de máquinas de soporte vectorial L_2L_1-SVM para clasificación binaria desde un enfoque de optimización binivel.

4.1. Problema binivel general

Para empezar, consideramos el siguiente problema binivel general como se muestra a continuación:

$$\min_{\beta} \quad J(\beta, \hat{w}(\beta); V, \eta) \tag{4.1a}$$

$$\text{s.a.} \quad \hat{w}(\beta) = \arg\min_{w} E(w, \beta; X, y), \tag{4.1b}$$

donde la función objetivo del problema de nivel superior evalúa el error de validación y el problema de nivel inferior corresponde al modelo de aprendizaje supervisado de clasificación binaria L_2L_1-SVM. Además, utilizaremos la siguiente notación:

S1. $X \in \mathbb{R}^{n\times(d+1)}$ una matriz que contiene n observaciones del conjunto de entrenamiento y d variables, más una columna de unos al final comúnmente incluido para representar el término de sesgo.

S2. $y \in \{-1, 1\}^n$ representa la etiqueta para cada observación en el conjunto de

entrenamiento.

S3. $V \in \mathbb{R}^{m\times(d+1)}$ es una matriz que contiene m observaciones del conjunto de validación y d variables, más una columna de unos al final comúnmente incluido para representar el término de sesgo.

S4. $\eta \in \{-1, 1\}^m$ es la etiqueta para cada observación en el conjunto de validación.

S5. $w \in \mathbb{R}^{(d+1)}$ representa el hiperplano de separación incluido el término de sesgo al final.

La forma como se expresa el problema de nivel inferior (4.1b), asume que tiene un minimizador único $\hat{w}(\beta) \in \mathbb{R}^{(d+1)}$. Este caso particular se cumple cuando E es una función estrictamente convexa en w.

El caso cuando la función E no es estrictamente convexa se lo deja para un estudio futuro.

Además, si se asume que la función E es diferenciable y dado que el problema de optimización (4.1b) no tiene restricciones, entonces se puede aplicar las condiciones necesarias de punto crítico. Es decir, si $\hat{w}$ es una solución local del problema (4.1b), entonces:

$$\min_{\beta} \quad J(\beta, \hat{w}(\beta); V, \eta) \tag{4.2a}$$

$$\text{s.a.} \quad \nabla_w E(\hat{w}, \beta; X, y) = \mathbf{0}_{d+1}. \tag{4.2b}$$

Notemos que, bajo las condiciones mencionadas anteriormente, el problema de optimización binivel (4.1) se convierte en un problema optimización de un solo nivel con $d+1$ restricciones de igualdad como se puede ver en (4.2).

Por otra parte, el gradiente con respecto al hiperparámetro β de la función objetivo J del problema (4.2) se lo conoce como hipergradiente. A través de la regla de la cadena y diferenciación implícita, el hipergradiente puede ser calculado por:

$$\nabla J(\beta, \hat{w}(\beta)) = \nabla_\beta J(\beta, \hat{w}(\beta)) + \left(\nabla_\beta \hat{w}(\beta)\right)^\top \nabla_w J(\beta, \hat{w}(\beta)). \tag{4.3}$$

Para encontrar la caracterización del elemento $\nabla_\beta \hat{w}(\beta)$, se lo puede realizar a través de dos enfoques: el teorema de la función implícita y las condiciones de optimalidad KKT [Crockett and Fessler, 2021].

4.1.1. Enfoque 1: Teorema de la función implícita

Desde esta perspectiva, notemos que el gradiente de la función E con respecto a w, considerando X, y fijos, está definida por:

$$\nabla_w E : \mathbb{R}^{(d+1)} \times \mathbb{R}^{(d+1)} \to \mathbb{R}^{(d+1)}. \tag{4.4}$$

El gradiente (4.4) es una función implícita continuamente diferenciable en una vecindad de $(\hat{w}, \hat{\beta})$, donde $\hat{\beta}$ es un hiperparámetro (vectorial para el caso general) y $\hat{w}$ es el hiperplano que se desea encontrar, tal que:

$$\nabla_w E(\hat{w}, \hat{\beta}) = \mathbf{0}_{d+1}. \tag{4.5}$$

Además, del supuesto anterior sabemos que la función E es estrictamente convexa. Entonces, la matriz Hessiana de E, denotada por $\nabla^2_{ww} E(\hat{w}, \hat{\beta})$, es definida positiva en $(\hat{w}, \hat{\beta})$. En consecuencia, la matriz Hessiana de E es invertible.

A continuación, se define el operador solución de la siguiente manera:

$$S : \beta \mapsto \{\hat{w} \in \mathbb{R}^{(d+1)} : \nabla_w E(\hat{w}, \beta) = \mathbf{0}_{d+1}, \quad \forall \beta \in \mathbb{R}^{(d+1)}\}. \tag{4.6}$$

Dado que se satisfacen las hipótesis de Teorema 1 de la función implícita para la función (4.4), entonces el operador solución definido en (4.6) tiene una localización univaluada $s \in S$ alrededor de $\hat{\beta}$ para $\hat{w}$, el cual es continuamente diferenciable en una vecindad $\mathcal{Q}$ de $\hat{\beta}$ tal que:

$$\nabla s(\beta) = -\left[\nabla^2_{ww} E(\hat{w}, \beta)\right]^{-1} \nabla_{w\beta} E(\hat{w}, \beta) = \nabla_\beta \hat{w}(\beta). \tag{4.7}$$

para cada $\beta \in \mathcal{Q}$.

Así, el hipergradiente se puede calcular en una vecindad $\mathcal{Q}$ de β para $\hat{w}$ a través de:

$$\nabla J(\beta, \hat{w}(\beta)) = \nabla_\beta J(\beta, \hat{w}(\beta)) - \left(\nabla_{w\beta} E(\hat{w}, \beta)\right)^\top \left[\nabla^2_{ww} E(\hat{w}, \beta)\right]^{-1} \nabla_w J(\beta, \hat{w}(\beta)). \tag{4.8}$$

4.1.2. Enfoque 2: Condiciones de optimalidad KKT

Desde el punto de vista de las condiciones de optimalidad KKT, primeramente es necesario mostrar la existencia de los multiplicadores de Lagrange. Para lograr este objetivo, se verifica que se cumpla una condición de calificación.

Lema 1. *Si A es una matriz definida positiva, entonces sus filas son linealmente independientes.*

Demostración. Como A es definida positiva, se tiene que:

$$x^\top Ax > 0, \qquad \forall x \neq 0. \qquad (1)$$

Por reducción al absurdo, supongamos que sus filas son linealmente dependientes. Es decir, existe un vector $x \neq 0$ tal que $Ax = 0$. Luego, $x^\top Ax = 0$ lo que contradice la Hipótesis 1. Así, el supuesto es falso. Por lo tanto, las filas de la matriz A son linealmente independientes. □

Como E es una función estrictamente convexa, entonces la matriz Hessiana de E es definida positiva. Así, gracias al Lema 1 sus filas son linealmente independientes. Es decir, el conjunto de gradientes de las restricciones de igualdad del problema (4.2) es linealmente independiente. Por lo tanto, se cumple la condición de calificación LICQ (Definición 8) lo que garantiza la existencia de los multiplicadores de Lagrange $\lambda \in \mathbb{R}^{(d+1)}$. Estos multiplicadores permiten encontrar soluciones que satisfacen tanto la función objetivo como las restricciones impuestas al problema (4.2).

Así, el Lagrangiano correspondiente al problema de optimización (4.2) es:

$$\mathcal{L}(w, \beta, \lambda) = J(\beta, w(\beta)) + \lambda^\top \nabla_w E(w, \beta), \quad \lambda \in \mathbb{R}^{(d+1)}. \tag{4.9}$$

Aplicando las condiciones KKT (Teorema 3), la primera condición establece que, en un punto óptimo del problema (4.2), el gradiente del Lagrangiano con respecto al parámetro w tiene que ser igual a cero.

$$\nabla_w \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \nabla_w J(\beta, \hat{w}(\beta)) + \nabla^2_{ww} E(\hat{w}, \beta)\hat{\lambda} = \mathbf{0}_{d+1}. \tag{4.10}$$

Podemos usar la ecuación (4.10) para caracterizar el multiplicador de Lagrange óptimo de la siguiente manera:

$$\hat{\lambda} = -\left[\nabla^2_{ww} E(\hat{w}, \beta)\right]^{-1} \nabla_w J(\beta, \hat{w}(\beta)). \tag{4.11}$$

Luego, el gradiente del Lagrangiano con respecto al parámetro β es:

$$\nabla_\beta \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \nabla_\beta J(\beta, \hat{w}(\beta)) + \left(\nabla_{w\beta} E(\hat{w}, \beta)\right)^\top \hat{\lambda}. \tag{4.12}$$

Reemplazando (4.11) en (5.13) se tiene:

$$\nabla_\beta \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \nabla_\beta J(\beta, \hat{w}(\beta)) - \left(\nabla_{w\beta} E(\hat{w}, \beta)\right)^\top \left[\nabla^2_{ww} E(\hat{w}, \beta)\right]^{-1} \nabla_w J(\beta, \hat{w}(\beta)). \tag{4.13}$$

De este modo, se puede observar que usando los dos enfoques las ecuaciones (4.8) y (4.13) son equivalentes cuando existe un minimizador único para la función objetivo del nivel inferior.

4.2. Problema binivel L_2L_1-SVM

En esta sección presentamos el problema de optimización binivel L_2L_1-SVM, el cual utiliza en el nivel superior la función de error cuadrático medio (MSE) para estimar el error de validación; y en el nivel inferior la función de costo del problema de aprendizaje automático supervisado para clasificación binaria L_2L_1-SVM.

4.2.1. Nivel inferior

Para ilustrar los conceptos de mejor manera, nos centramos en el caso cuando el hiperparámetro β es escalar. Por lo tanto, la función objetivo del problema de optimización de nivel inferior L_2L_1-SVM está definida por:

$$E(w, \beta; X, y) := \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i \langle w, x_i \rangle\} + \frac{\alpha}{2} \sum_{j=1}^{d+1} w_j^2 + \beta \sum_{j=1}^{d+1} |w_j|, \tag{4.14}$$

donde $\alpha, \beta \in \mathbb{R}_+$ son los hiperparámetros de regularización de las normas L_2 y L_1 respectivamente.

No obstante, es necesario que la función (4.14) sea diferenciable. Con este propósito, se propone emplear una aproximación local suave para las funciones no diferenciables, como la función *hinge-loss* y la norma L_1.

4.2.1.1. Regularización Pseudo-Huber de la norma L_1

Para la aproximación de la norma L_1, se propone una función de clase C^∞ conocida como Pseudo-Huber, la cual es usada por [Fountoulakis and Gondzio, 2015] en su trabajo sobre el método de segundo orden para problemas L_1-regularizados estrictamente convexos y se define de la siguiente manera:

$$H_\gamma(w_j) := \left(\gamma^2 + w_j^2\right)^{\frac{1}{2}} - \gamma, \tag{4.15}$$

donde $\gamma > 0$ para todo $j = 1, \ldots, d+1$.

Se puede observar en (4.15) y la Figura 4.1 que a medida que el parámetro γ se acerca a 0, se aproxima a la norma L_1. Es decir, la función Pseudo-Huber converge a la norma L_1 cuando $\gamma \to 0$.

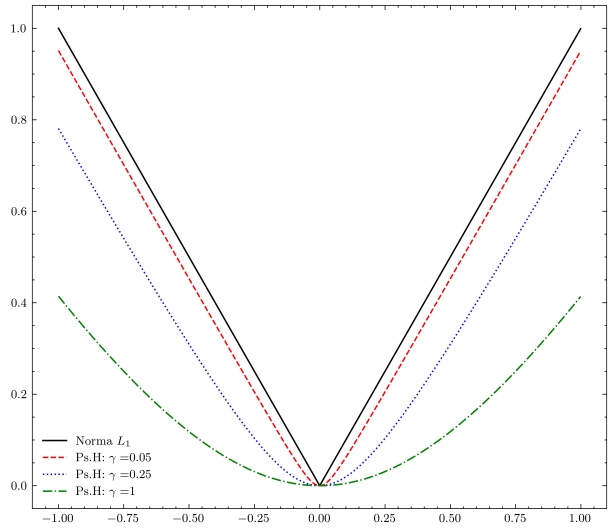


Figura 4.1: Comparación de la norma L_1 y Pseudo-Huber (Ps.H) con diferentes parámetros γ en un espacio de una dimensión.

Una de las ventajas de diferenciabilidad de la función Pseudo-Huber, es que se puede obtener información de primer y segundo orden. Esta información es útil a la hora de emplear métodos que calculan una dirección de descenso como los métodos Newton y quasi-Newton.

Así, el gradiente de la función Pseudo-Huber está dado por:

$$\nabla_w H_\gamma(w) = \left[w_1 \left(\gamma^2 + w_1^2\right)^{-\frac{1}{2}}, \ldots, w_{d+1} \left(\gamma^2 + w_{d+1}^2\right)^{-\frac{1}{2}}\right] \in \mathbb{R}^{(d+1)}, \tag{4.16}$$

y su matriz Hessiana está dada por:

$$\nabla^2_{ww} H_\gamma(w) = \gamma^2 \, diag\left(\left[\left(\gamma^2 + w_1^2\right)^{-\frac{3}{2}}, \ldots, \left(\gamma^2 + w_{d+1}^2\right)^{-\frac{3}{2}}\right]\right) \in \mathbb{R}^{(d+1)\times(d+1)}. \tag{4.17}$$

Proposición 1. *La función Pseudo-Huber (4.15) es convexa para todo $\gamma > 0$.*

Demostración. Para mostrar la convexidad de la función Pseudo-Huber, usamos el criterio de la segunda derivada. Sea la función Pseudo-Huber (4.15), la segunda derivada está definida

por:

$$H''_\gamma(w_j) = \gamma^2 \left(\gamma^2 + w_j^2\right)^{-\frac{3}{2}},$$

para todo $j = 1, \ldots, d+1$.

Notemos que $\gamma > 0$ y los coeficientes del hiperplano w están elevados al cuadrado. Por lo tanto, la segunda derivada es positiva para todo $\gamma > 0$, lo que implica que la función Pseudo-Huber $H_\gamma(\cdot)$ es convexa. □

Proposición 2 ([Fountoulakis and Gondzio, 2015], pág. 6). *Sean $H_\gamma(w_j)$ para todo $j = 1, \ldots, d+1$. definido por (4.15) y $\gamma > 0$. Entonces:*

a) El gradiente $\nabla_w H_\gamma(w)$ es Lipschitz continuo con constante $L = \frac{1}{\gamma}$.

b) La matriz Hessiana $\nabla^2_{ww} H_\gamma(w)$ es Lipschitz continua con constante $L = \frac{1}{\gamma^2}$.

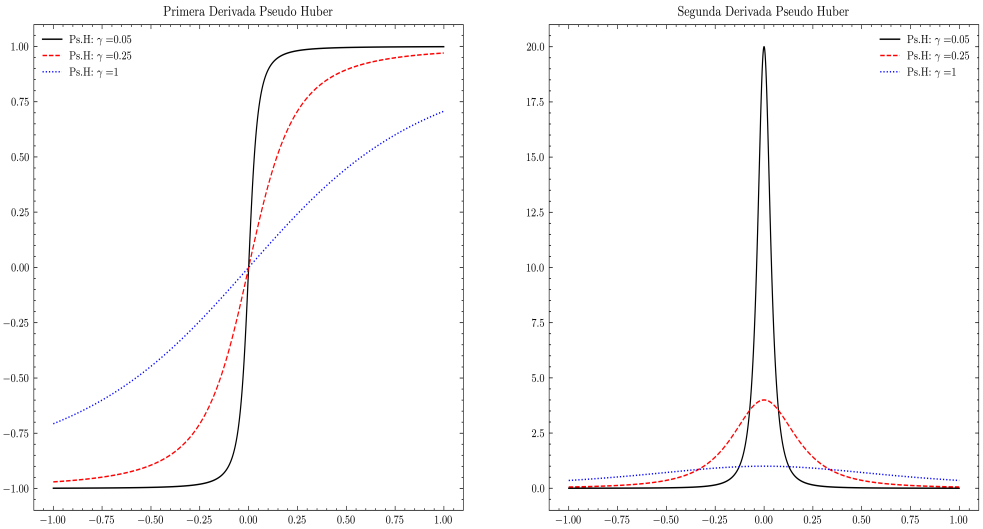


Figura 4.2: Derivadas de primer y segundo orden para la función Pseudo-Huber (Ps.H) con diferentes parámetros γ en un espacio de una dimensión.

4.2.1.2. Aproximación función Hinge-Loss

Del mismo modo, para aproximar la función *hinge-loss*, se utiliza la función de costo ℓ_μ de clase $\mathcal{C}^\infty$ propuesta en el trabajo de [Zhang et al., 2003], y está definida por:

$$\ell_\mu(1-t) := \mu \ln\left(1 + e^{\frac{1}{\mu}(1-t)}\right), \tag{4.18}$$

donde $\mu \in (0,1)$ y $t \in \mathbb{R}$.

La Figura 4.3 muestra que la función ℓ_μ converge a la función *hinge-loss* original cuando $\mu \to 0$.

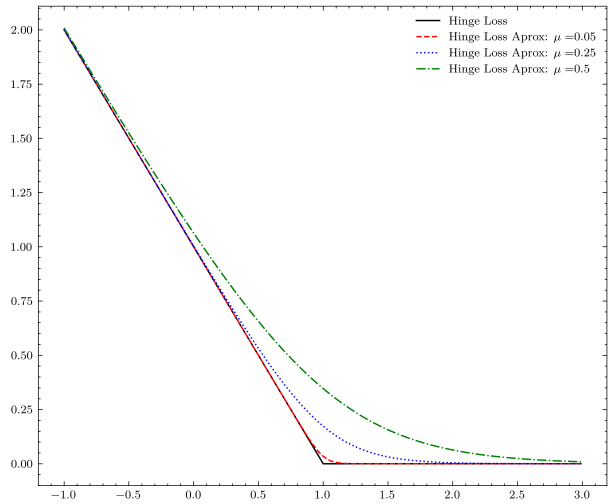


Figura 4.3: Comparación de la función *hinge-loss* y la función *hinge-loss* aproximada con diferentes parámetros μ en un espacio de una dimensión.

Además, ya que la función objetivo del nivel inferior necesita ser dos veces diferenciable, la deriva de primer orden es:

$$\ell'_\mu(1-t) = \frac{1}{1+e^{-\frac{1}{\mu}(1-t)}}, \tag{4.19}$$

y la derivada de segundo orden en $(1-t)$ es:

$$\ell''_\mu(1-t) = \frac{e^{-\frac{1}{\mu}(1-t)}}{\mu\left(1+e^{-\frac{1}{\mu}(1-t)}\right)^2}. \tag{4.20}$$

A continuación, la Figura 4.4 muestra la comparación de las derivadas de primer y segundo orden para diferentes parámetros $\mu \in (0,1)$.

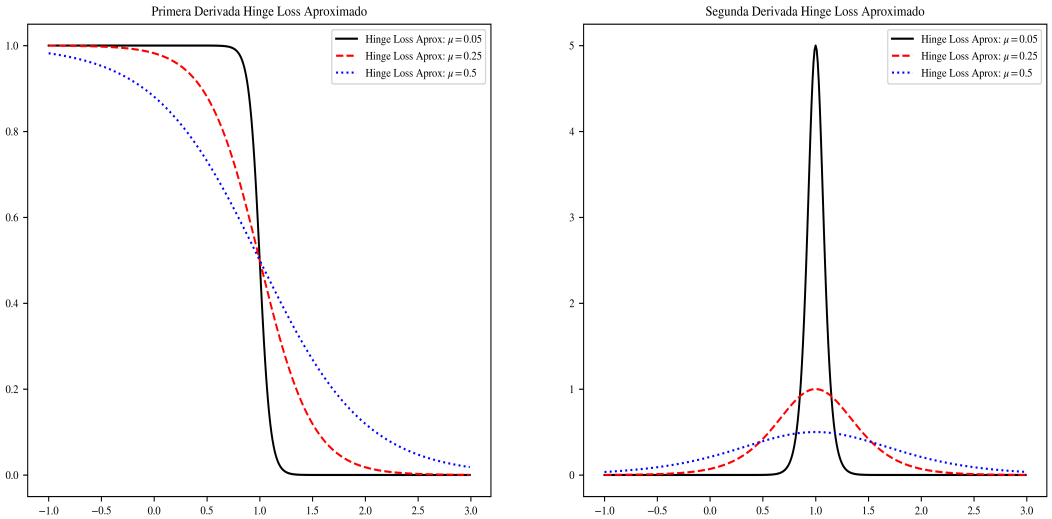


Figura 4.4: Derivadas de primer y segundo orden para la función *hinge-loss* aproximada con diferentes parámetros μ en un espacio de una dimensión.

Proposición 3. *La función hinge-loss aproximada (4.18) es convexa para todo $\mu > 0$.*

Demostración. Para mostrar la convexidad de la función *hinge-loss* aproximada, usamos el criterio de la segunda derivada.

Sea la función *hinge-loss* aproximada definida en (4.18), la segunda derivada está definida por:

$$\ell''_\mu(1-t) = \frac{e^{-\frac{1}{\mu}(1-t)}}{\mu\left(1+e^{-\frac{1}{\mu}(1-t)}\right)^2}.$$

Notemos que el denominador $\mu(1+e^{-\frac{1}{\mu}(1-t)})^2 > 0$, pues $\mu > 0$. Además, el numerador $e^{-\frac{1}{\mu}(1-t)} > 0$ es positivo. En consecuencia, la segunda derivada es positiva, lo que implica que la función *hinge-loss* aproximada $\ell_\mu(\cdot)$ es convexa. □

Por lo tanto, la función objetivo L_2L_1-SVM aproximada del problema de optimización de nivel inferior, que depende de los parámetros $\mu > 0$ y $\gamma > 0$, está dada por:

$$\tilde{E}(w,\beta;X,y) := \frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-y_it_i) + \frac{\alpha}{2}\sum_{j=1}^{d+1}w_j^2 + \beta\sum_{j=1}^{d+1}H_\gamma(w_j), \tag{4.21}$$

donde $t_i = \langle w, x_i\rangle$ para todo $i = 1,\ldots,n$.

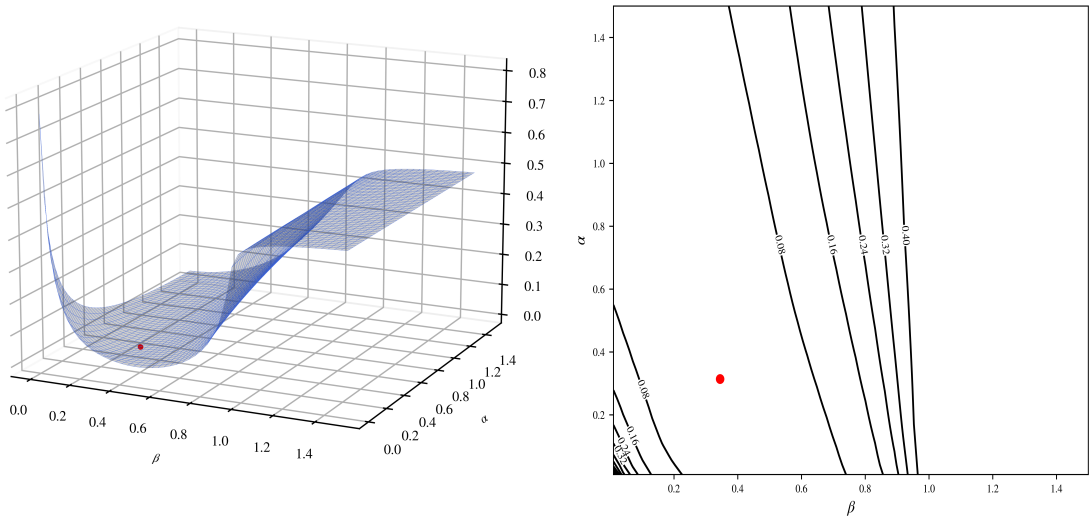
4.2.2. Nivel superior

En el contexto de la selección de variables, empleamos el error cuadrático medio (MSE) como función objetivo para el problema de nivel superior, con el propósito de evaluar el rendimiento del modelo en el conjunto de validación. Esta función cuantifica el promedio de los errores cuadráticos entre las predicciones generadas por el modelo de aprendizaje automático supervisado y los valores reales de las etiquetas en el conjunto de validación.

Para el caso general, cuando se busca los valores óptimos de los hiperparámetros α, β (escalares) de la función L_2L_1-SVM aproximada (4.21), el error medio cuadrático se expresa como:

$$J(\alpha,\beta,w(\alpha,\beta);V,\eta) := \frac{1}{2m}\|Vw(\alpha,\beta)^\top - \eta\|_2^2. \tag{4.22}$$

Notemos que el error medio cuadrático definido en (4.22), es dos veces diferenciable con respecto a w y se utiliza como aproximación del error de generalización del modelo.



(a) Gráfico de superficie.

(b) Gráfico de contorno.

Figura 4.5: Error cuadrático medio sobre $(\alpha, \beta) \in (0; 1{,}5] \times (0; 1{,}5]$ para el conjunto de datos Iris.

La Figura 4.5 muestra el error de validación a través del error cuadrático medio, a medida que varían los valores de los hiperparámetros α, β (escalares) en una búsqueda exhaustiva, también conocida como *Grid Search*, para el conjunto de datos Iris[1].

A continuación, se describe el proceso de búsqueda exhaustiva o *Grid Search*:

1. Resolver el problema de nivel inferior L_2L_1-SVM aproximado, donde la función objetivo está definida por (4.21) para los valores de $\alpha > 0$ y $\beta > 0$ (los detalles se presentan en el Capítulo 5).

2. Como resultado del paso 1, se obtiene un conjunto de hiperplanos de separación representados por $w(\alpha, \beta)$, donde cada par de valores α y β define un hiperplano de separación único.

3. Calcular el error de validación a través del MSE (4.22) para cada hiperplano $w(\alpha, \beta)$.

4. Identificar la combinación de α y β donde se minimiza el error de validación.

Al realizar el proceso de búsqueda exhaustiva descrito anteriormente, se observa en la Figura 4.5a la superficie de la función MSE, donde el menor error de validación alcanzado es $J_{min} = 0{,}026$. Además, la Figura 4.5b muestra el gráfico de contorno de la función MSE y los valores de $(\alpha, \beta) = (0{,}314\,;\, 0{,}345)$ donde se alcanza el mínimo valor. Este valor mínimo se obtiene cuando se resuelve el problema de nivel inferior L_2L_1-SVM aproximado para $\mu = 0{,}25$ y $\gamma = 0{,}01$.

[1]Los detalles se los presenta en el siguiente enlace:Iris Dataset.

No obstante, en este trabajo de investigación, dirigimos nuestra atención hacia la optimización del hiperparámetro β, el cual controla la dispersión de los coeficientes en el hiperplano de separación w estimado. De esta manera, la optimización de β desempeña un papel fundamental en el proceso de selección de variables. Por lo tanto, el hiperparámetro α se lo mantiene constante con un valor suficientemente pequeño.

Para este caso, el error de validación se puede definir como:

$$J(\beta, w(\beta); V, \eta) := \frac{1}{2m}\|Vw(\beta)^\top - \eta\|_2^2. \tag{4.23}$$

Del mismo modo, al aplicar el proceso de búsqueda exhaustiva para el caso cuando se fija el hiperparámetro $\alpha > 0$, se observa en la Figura 4.6, que el mínimo error de validación es $J_{min} = 0{,}0337$ para $\hat{\beta} = 0{,}48$. Este resultado se logra al resolver el problema de nivel inferior L_2L_1-SVM aproximado para $\mu = 0{,}25$ y $\gamma = 0{,}01$, con $\alpha = 0{,}01$ fijo.

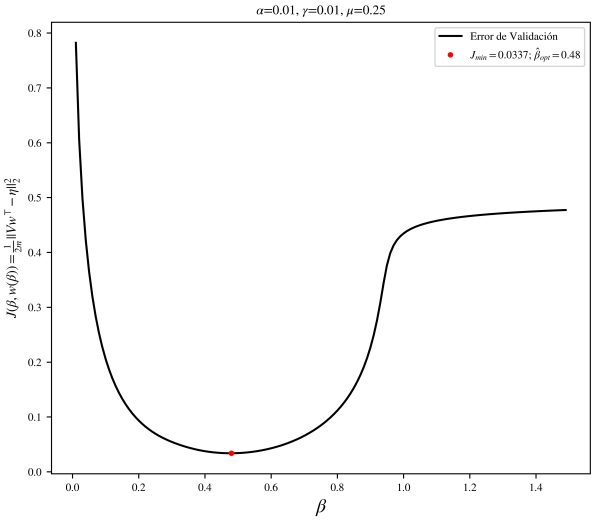


Figura 4.6: Error de validación sobre $\beta \in (0;\, 1,5]$ para el conjunto de datos Iris.

Es importante destacar que la función $J(\cdot)$ no es convexa para el hiperparámetro β, solo lo es en w para β fijo [Klatzer, 2014].

4.3. Formulación del modelo escalar

Considerando todos estos elementos, en esta sección se analiza el problema L_2L_1-SVM binivel aproximado. En particular, se aborda el caso cuando el hiperparámetro $\beta > 0$ es un escalar y el hiperparámetro $\alpha > 0$ es fijo, el cual se expresa de la siguiente

manera:

$$(P_E)\begin{cases}\min\limits_{\beta>0} & J(\beta, w(\beta); V, \eta) := \dfrac{1}{2m}\|V\hat{w}(\beta)^\top - \eta\|_2^2 \\ \text{s.a.} & \hat{w} = \arg\min\limits_{w} \tilde{E}(w, \beta; X, y),\end{cases} \tag{4.24}$$

donde la función objetivo del problema de nivel inferior, que depende de los parámetros $\mu > 0$ y $\gamma > 0$, está dada por:

$$\tilde{E}(w, \beta; X, y) := \frac{1}{n}\sum_{i=1}^{n} \ell_\mu(1 - y_i t_i) + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \beta \sum_{j=1}^{d+1} H_\gamma(w_j), \tag{4.25}$$

con $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $H_\gamma(\cdot)$ representa la función Pseudo-Huber (4.15) y $\ell_\mu(\cdot)$ es la función *hinge-loss* aproximada (4.18).

Lema 2. *Sean $X \in \mathbb{R}^{n\times(d+1)}$ un conjunto de entrenamiento, $V \in \mathbb{R}^{m\times(d+1)}$ un conjunto de validación, $w \in \mathbb{R}^{(d+1)}$ el hiperplano de separación, $y \in \{-1, 1\}^n$ un vector de etiquetas de los datos de entrenamiento y $\eta \in \{-1, 1\}^m$ un vector de etiquetas de los datos de validación para el problema de nivel inferior en (4.24). Entonces:*

a) Existe una solución única para el problema de nivel inferior en (4.24).

b) El gradiente de la función $\tilde{E}$ con respecto a w está dada por:

$$\nabla_w \tilde{E}(w, \beta; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yXw^\top)^\top X + \alpha w + \beta \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)}.$$

c) La Hessiana de $\tilde{E}$ con respecto a w está dada por:

$$\nabla^2_{ww} \tilde{E}(w, \beta; X, y) = \frac{1}{n} X^\top diag\left(\ell''_\mu(1 - yXw^\top)\right) X + \alpha \mathbb{I} + \beta \nabla^2_{ww} H_\gamma(\hat{w}) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

d) El gradiente con respecto a w y β de $\tilde{E}$ está dado por:

$$\nabla_{w\beta} \tilde{E}(w, \beta; X, y) = \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)}.$$

Demostración. Sea la función:

$$\tilde{E}(w, \beta; X, y) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_\mu(1 - y_i t_i) + \beta \sum_{j=1}^{d+1} H_\gamma(w_j)}_{f} + \underbrace{\frac{\alpha}{2}\|w\|_2^2}_{g},$$

donde $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $\alpha > 0$ y $\beta > 0$.

a) Para probar que existe una solución única para el problema de nivel inferior en (4.24), necesitamos probar que la función objetivo $\tilde{E} = f + g$ es fuertemente convexa.

Primeramente, gracias a los Teoremas 1 y 3 sabemos que las funciones $\ell_\mu(\cdot)$, $H_\gamma(\cdot)$ son convexas. Así, la función f, al ser una suma de funciones convexas, es convexa.

Luego, ya que g es dos veces diferenciable, se tiene que su Hessiana está dada por:

$$\nabla^2 g(w) = \alpha \mathbb{I},$$

donde $\alpha > 0$ e $\mathbb{I} \in \mathbb{R}^{(d+1)\times(d+1)}$ es la matriz identidad. De este modo, se tiene que:

$$\nabla^2 g(w) \succeq \alpha \mathbb{I},$$

lo que implica que la función g es fuertemente convexa con módulo $\alpha > 0$.

Por lo tanto, aplicando el Teorema 4 se tiene que $\tilde{E} = f + g$ es una función fuertemente convexa de módulo $\alpha > 0$. Finalmente, gracias al Teorema 5 se concluye que $\tilde{E} = f + g$ posee un único minimizador.

b) Para calcular el gradiente de la función $\tilde{E}$, analizamos sus componentes de manera individual.

$$\begin{aligned}
\left[\frac{\partial}{\partial w}\left(\frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-y_i t_i)\right)\right]_j &= \frac{\partial}{\partial w_j}\left(\frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-y_i t_i)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w_j}\left(\ell_\mu(1-y_i t_i)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial t_i}\left(\ell_\mu(1-y_i t_i)\right)\frac{\partial t_i}{\partial w_j} \\
&= -\frac{1}{n}\sum_{i=1}^{n} y_i\, \ell'_\mu(1-y_i t_i)\, x_i \frac{\partial w_i}{\partial w_j},
\end{aligned}$$

donde:

$$\frac{\partial w_i}{\partial w_j} = \begin{cases} 1, & \text{si } i = j, \quad \forall i, j = 1, \ldots, n., \\ 0, & \text{si } i \neq j, \quad \forall i, j = 1, \ldots, n. \end{cases}$$

Entonces, para $t = Xw^\top$ se tiene:

$$\nabla_w\left(\frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-y_i t_i)\right) = -\frac{1}{n} y \left(\ell'_\mu(1-yt)\right)^\top X \in \mathbb{R}^{(d+1)}. \tag{p1}$$

Por otra parte, fácilmente se puede calcular el gradiente de la norma L_2 como sigue:

$$\nabla_w \left(\frac{\alpha}{2} \|w\|_2^2 \right) = \alpha w \in \mathbb{R}^{(d+1)}. \tag{p2}$$

Finalmente, analizamos el último término de la función $\tilde{E}$.

$$\begin{aligned} \left[\frac{\partial}{\partial w} \left(\beta \sum_{j=1}^{d+1} H_\gamma(w_j) \right) \right]_k &= \frac{\partial}{\partial w_k} \left(\beta \sum_{j=1}^{d+1} H_\gamma(w_j) \right) \\ &= \beta \sum_{j=1}^{d+1} \frac{\partial}{\partial w_k} \left(H_\gamma(w_j) \right) \\ &= \beta \sum_{j=1}^{d+1} H'_\gamma(w_j). \end{aligned}$$

Luego, el gradiente con respecto a w es:

$$\nabla_w \left(\beta \sum_{j=1}^{d+1} H_\gamma(w_j) \right) = \beta \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)}, \tag{p3}$$

donde:

$$\nabla_w H_\gamma(w) = \left[w_1 \left(\gamma^2 + w_1^2 \right)^{-\frac{1}{2}}, \ldots, w_{d+1} \left(\gamma^2 + w_{d+1}^2 \right)^{-\frac{1}{2}} \right].$$

Por lo tanto, por $(p1)$, $(p2)$ y $(p3)$ el gradiente de $\tilde{E}$ con respecto a w es:

$$\nabla_w \tilde{E}(w, \beta; X, y) = -\frac{1}{n} y \, \ell'_\mu (1 - yt)^\top X + \alpha w + \beta \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)}.$$

c) Para calcular la Hessiana de $\tilde{E}$ analizamos cada término como en el literal anterior. Sea $t = Xw^\top$, entonces:

$$\begin{aligned} \frac{\partial}{\partial w} \left(-\frac{1}{n} y \left(\ell'_\mu (1 - yt) \right)^\top X \right) &= \frac{\partial}{\partial t} \left(-\frac{1}{n} y \left(\ell'_\mu (1 - yt) \right)^\top X \right) \frac{\partial t}{\partial w} \\ &= -\frac{1}{n} y (-y) \left(diag \left(\ell''_\mu (1 - yt) \right) X \right)^\top X \\ &= \frac{1}{n} X^\top diag \left(\ell''_\mu (1 - yt) \right) X. \end{aligned} \tag{p1.1}$$

Para la norma L_2 es fácil ver que:

$$\nabla_w (\alpha w) = \alpha \mathbb{I} \in \mathbb{R}^{(d+1) \times (d+1)}. \tag{p2.1}$$

Finalmente, para el término de la función Pseudo-Huber se tiene:

$$\nabla_w \left(\beta \nabla_w H_\gamma(w) \right) = \beta \nabla_{ww}^2 H_\gamma(w) \in \mathbb{R}^{(d+1) \times (d+1)}, \tag{p3.1}$$

donde:

$$\nabla^2_{ww} H_\gamma(w) = \gamma^2 \, diag\left(\left[\left(\gamma^2 + w_1^2\right)^{-\frac{3}{2}}, \ldots, \left(\gamma^2 + w_{d+1}^2\right)^{-\frac{3}{2}}\right]\right).$$

Por lo tanto, la Hessiana de $\tilde{E}$ con respecto a w es:

$$\nabla^2_{ww} \tilde{E}(w, \beta; X, y) = \frac{1}{n} X^\top \, diag\left(\ell''_\mu(1 - yt)\right) X + \alpha \mathbb{I} + \beta \nabla^2_{ww} H_\gamma(\hat{w}) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

d) Del literal b) se tiene que el gradiente de $\tilde{E}$ con respecto a w está definido por:

$$\nabla_w \tilde{E}(w, \beta; X, y) = -\frac{1}{n} \, y \, \ell'_\mu(1 - yt)^\top X + \alpha w + \beta \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)}.$$

De esta manera, si calculamos el gradiente con respecto a β se tiene:

$$\nabla_{w\beta} \tilde{E}(w, \beta; X, y) = \nabla_w H_\gamma(w) \in \mathbb{R}^{(d+1)},$$

donde:

$$\nabla_w H_\gamma(w) = \left[w_1 \left(\gamma^2 + w_1^2\right)^{-\frac{1}{2}}, \ldots, w_{d+1} \left(\gamma^2 + w_{d+1}^2\right)^{-\frac{1}{2}}\right].$$

□

Como el problema de nivel inferior en (4.24) es un problema de optimización sin restricciones y además su función objetivo es diferenciable, se puede aplicar las condiciones necesarias de primer orden. Entonces, el problema binivel (4.24) es equivalente al siguiente problema de optimización de un solo nivel:

$$(P'_E) \begin{cases} \min\limits_{\beta > 0} & J(\beta, w(\beta); V, \eta) := \dfrac{1}{2m} \|V \hat{w}(\beta)^\top - \eta\|_2^2 \\ \text{s.a.} & \nabla_w \tilde{E}(\hat{w}, \beta; X, y) = \mathbf{0}_{d+1}. \end{cases} \tag{4.26}$$

donde:

$$\nabla_w \tilde{E}(\hat{w}, \beta; X, y) = -\frac{1}{n} \, y \, \ell'_\mu(1 - yX\hat{w}^\top)^\top X + \alpha \hat{w} + \beta \nabla_w H_\gamma(\hat{w}).$$

Notar que las condiciones necesarias de primer orden también son suficientes gracias a la convexidad del problema de nivel inferior en (4.24).

Teorema 8 (Condiciones Karush-Kuhn-Tucker)**.** *Si $\hat{w}(\beta)$ es una solución local del problema (4.26) en el sentido de la Definición 5 tal que satisface la condición de calificación LICQ. Entonces, existe un un multiplicador de Lagrange $\hat{\lambda} \in \mathbb{R}^{(d+1)}$ tal que cumple las siguientes condiciones:*

a) $\nabla_w \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \mathbf{0}_{d+1}$,

b) $\nabla_\beta \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = 0$.

Donde:

$$\mathcal{L}(w, \beta, \lambda) = \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + \beta \nabla_w H_\gamma(w) \right).$$

Demostración. Para justificar la existencia del multiplicador de Lagrange, usamos el literal a) del Lema 2. Es decir, sabemos que la función $\tilde{E}$ es fuertemente convexa con módulo $\alpha > 0$, lo cual implica que la función $\tilde{E}$ es estrictamente convexa.

Además, sabemos que la convexidad estricta es una condición suficiente para que se cumpla la condición de calificación LICQ [Crockett and Fessler, 2021]. En consecuencia, se garantiza la existencia del multiplicador de Lagrange $\lambda \in \mathbb{R}^{d+1}$.

Luego, el Lagrangiano correspondiente al problema (4.26) está dado por:

$$\begin{aligned}\mathcal{L}(w, \beta, \lambda) &= J(w, \beta; V, \eta) + \lambda^\top \nabla_w \tilde{E}(w, \beta; X, y) \\ &= \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + \beta \nabla_w H_\gamma(w) \right).\end{aligned}$$

Sea $\hat{w} = \hat{w}(\beta)$ una solución del problema (4.26) y $\hat{\lambda} \in \mathbb{R}^{d+1}$. Entonces, el gradiente del Lagrangiano con respecto a w está dado por:

$$\begin{aligned}\mathbf{0}_{d+1} = \nabla_w \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) &= \nabla_w J(w, \beta; V, \eta) + \nabla^2_{ww} \tilde{E}(w, \beta; X, y)\hat{\lambda} \\ &= \frac{1}{m} V^\top (V\hat{w}^\top - \eta) + \left(\frac{1}{n} X^\top diag\left(\ell''_\mu(1 - yt) \right) X + \alpha \mathbb{I} + \beta \nabla^2_{ww} H_\gamma(\hat{w}) \right) \hat{\lambda}. \qquad (1)\end{aligned}$$

De (1) se puede caracterizar el multiplicador óptimo como sigue:

$$\hat{\lambda} = -\left[\frac{1}{n} X^\top diag(\ell''_\mu(1 - yt)) X + \alpha \mathbb{I} + \beta \nabla^2_{ww} H_\gamma(\hat{w}) \right]^{-1} \frac{1}{m} V^\top \left(V\hat{w}^\top - \eta \right),$$

donde $\mathbb{I} \in \mathbb{R}^{(d+1)\times(d+1)}$ es la matriz identidad.

Además, el gradiente con respecto al hiperparámetro β, también conocido como hipergradiente, se expresa de la siguiente manera:

$$\begin{aligned}0 = \nabla_\beta \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) &= \nabla_\beta J(w, \beta; V, \eta) + \nabla_{w\beta} \tilde{E}(w, \beta; X, y)^\top \hat{\lambda} \\ &= 0 + \left(\nabla_w H_\gamma(\hat{w}) \right)^\top \hat{\lambda} \\ &= -\left(\nabla_w H_\gamma(\hat{w}) \right)^\top G^{-1} \frac{1}{m} V^\top \left(V\hat{w}^\top - \eta \right),\end{aligned}$$

donde:

$$G := \left[\frac{1}{n} X^\top \, diag(\ell''_\mu(1 - yt))X + \alpha \mathbb{I} + \beta \nabla^2_{ww} H_\gamma(\hat{w})\right].$$

□

Este resultado nos ayuda a extender el análisis donde el hiperparámetro β posee una estructura vectorial como se muestra en las siguiente sección.

4.4. Formulación del modelo vectorial

En esta sección se analiza el problema L_2L_1-SVM binivel aproximado para el caso cuando el hiperparámetro $\hat{\beta} \in \mathbb{R}^{d+1}_+$ es un vector, es decir, tiene la forma $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_{d+1})$. Este tipo de hiperparámetro vectorial se usa para la selección de variables y para determinar la importancia o relevancia de cada una de las $d+1$ variables de un conjunto de datos (incluido el término de sesgo).

A continuación, la formulación para el problema L_2L_1-SVM binivel aproximado para el caso cuando el hiperparámetro $\hat{\beta}$ es vectorial y el hiperparámetro $\alpha > 0$ se mantiene fijo.

$$(P_V) \begin{cases} \min\limits_{\hat{\beta} \in \mathbb{R}^{d+1}_+} & J(\hat{\beta}, w(\hat{\beta}); V, \eta) := \dfrac{1}{2m} \| V\hat{w}(\hat{\beta})^\top - \eta \|_2^2 \\ \text{s.a.} & \hat{w} = \arg\min\limits_{w} \hat{E}(w, \hat{\beta}; X, y), \end{cases} \tag{4.27}$$

donde la función objetivo de nivel inferior está dada por:

$$\hat{E}(w, \hat{\beta}; X, y) := \frac{1}{n} \sum_{i=1}^{n} \ell_\mu(1 - yt_i) + \frac{\alpha}{2} \sum_{j=1}^{d+1} w_j^2 + \sum_{j=1}^{d+1} \hat{\beta}_j H_\gamma(w_j), \tag{4.28}$$

con $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $H_\gamma(\cdot)$ representa la función Pseudo-Huber (4.15) y $\ell_\mu(\cdot)$ es la función *hinge-loss* aproximada (4.18).

Lema 3. *Sean $X \in \mathbb{R}^{n \times (d+1)}$ un conjunto de entrenamiento, $V \in \mathbb{R}^{m \times (d+1)}$ un conjunto de validación, $w \in \mathbb{R}^{(d+1)}$ el hiperplano de separación, $y \in \{-1, 1\}^n$ un vector de etiquetas de los datos de entrenamiento y $\eta \in \{-1, 1\}^m$ un vector de etiquetas de los datos de validación para el problema de nivel inferior en (4.27). Entonces:*

a) Existe una solución única para el problema de nivel inferior en (4.27).

b) El gradiente de la función $\hat{E}$ con respecto a w está dada por:

$$\nabla_w \hat{E}(w, \hat{\beta}; X, y) = -\frac{1}{n} y \, \ell'_\mu(1 - yXw^\top)^\top X + \alpha\hat{w} + (\nabla_w H_\gamma(w))^\top \, diag(\hat{\beta}) \in \mathbb{R}^{(d+1)}.$$

c) La Hessiana de $\hat{E}$ con respecto a w está dada por:

$$\nabla^2_{ww}\hat{E}(w,\hat{\beta};X,y) = \frac{1}{n}X^\top diag\left(\ell''_\mu(1-yXw^\top)\right)X + \alpha\mathbb{I} + \hat{\beta}\odot\nabla^2_{ww}H_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)},$$

donde el símbolo $\odot$ representa la multiplicación punto a punto.

d) El gradiente con respecto a w y $\hat{\beta}$ de $\hat{E}$ está dado por:

$$\nabla_{w\hat{\beta}}\hat{E}(w,\hat{\beta};X,y) = diag(\nabla_w H_\gamma(w)) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

Demostración. Para la demostración de este teorema, nos centramos en el término de la función Pseudo-Huber, ya que los otros términos son los mismos que en la formulación escalar.

Sea la función:

$$\hat{E}(w,\hat{\beta};X,y) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-y_it_i) + \sum_{j=1}^{d+1}\hat{\beta}_jH_\gamma(w_j)}_{f} + \underbrace{\frac{\alpha}{2}\|w\|_2^2}_{g},$$

donde $t_i = \langle w, x_i\rangle$ para todo $i = 1,\ldots,n.$, $\alpha > 0$ y $\hat{\beta} \in \mathbb{R}_+^{(d+1)}$.

a) Para probar que existe una solución única para el problema de nivel inferior en (4.27), necesitamos probar que la función objetivo $\hat{E} = f + g$ es fuertemente convexa.

Gracias al Teorema 3 sabemos que $H_\gamma(\cdot)$ es convexa. Además, como $\hat{\beta}_j > 0$, entonces $\sum_{j=1}^{d+1}\hat{\beta}_jH_\gamma(w_j)$ es convexa. En consecuencia, $\hat{E} = f + g$ es fuertemente convexa con módulo $\alpha > 0$.

Por lo tanto, gracias al Teorema 5 se concluye que $\hat{E} = f + g$ posee un único minimizador.

b) Para calcular el gradiente de la función $\hat{E}$, analizamos el término de la función Pseudo-Huber.

$$\begin{aligned}\left[\frac{\partial}{\partial w}\left(\sum_{j=1}^{d+1}\hat{\beta}_jH_\gamma(w_j)\right)\right]_k &= \frac{\partial}{\partial w_k}\left(\sum_{j=1}^{d+1}\hat{\beta}_jH_\gamma(w_j)\right)\\ &= \sum_{j=1}^{d+1}\hat{\beta}_j\frac{\partial}{\partial w_k}\left(H_\gamma(w_j)\right)\\ &= \sum_{j=1}^{d+1}\hat{\beta}_jH'_\gamma(w_j)\end{aligned}$$

Luego, el gradiente con respecto a w es:

$$\nabla_w \left(\sum_{j=1}^{d+1} \hat{\beta}_j H_\gamma(w_j) \right) = \left(\nabla_w H_\gamma(w) \right)^\top diag(\hat{\beta}) \in \mathbb{R}^{(d+1)},$$

donde:

$$\nabla_w H_\gamma(w) = \left[w_1 \left(\gamma^2 + w_1^2 \right)^{-\frac{1}{2}}, \ldots, w_{d+1} \left(\gamma^2 + w_{d+1}^2 \right)^{-\frac{1}{2}} \right].$$

Por lo tanto, el gradiente de $\hat{E}$ con respecto a w es:

$$\nabla_w \hat{E}(w, \hat{\beta}; X, y) = -\frac{1}{n} y \, \ell'_\mu (1 - yXw^\top)^\top X + \alpha w + \left(\nabla_w H_\gamma(w) \right)^\top diag(\hat{\beta}) \in \mathbb{R}^{(d+1)}.$$

c) Para calcular la Hessiana de $\hat{E}$ analizamos el término de la función Pseudo-Huber.

$$\nabla_w \left(\left(\nabla_w H_\gamma(w) \right)^\top diag(\hat{\beta}) \right) = \hat{\beta} \odot \nabla^2_{ww} H_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)},$$

donde:

$$\nabla^2_{ww} H_\gamma(w) = \gamma^2 \, diag \left(\left[\left(\gamma^2 + w_1^2 \right)^{-\frac{3}{2}}, \ldots, \left(\gamma^2 + w_{d+1}^2 \right)^{-\frac{3}{2}} \right] \right).$$

Por lo tanto, la Hessiana de $\hat{E}$ con respecto a w es:

$$\nabla^2_{ww} \hat{E}(w, \hat{\beta}; X, y) = \frac{1}{n} X^\top diag \left(\ell''_\mu (1 - yXw^\top) \right) X + \alpha \mathbb{I} + \hat{\beta} \odot \nabla^2_{ww} H_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

d) Del literal b) se tiene que el gradiente de $\hat{E}$ con respecto a w está definido por:

$$\nabla_w \hat{E}(w, \hat{\beta}; X, y) = -\frac{1}{n} y \, \ell'_\mu (1 - yXw^\top)^\top X + \alpha w + \left(\nabla_w H_\gamma(w) \right)^\top diag(\hat{\beta}).$$

De esta manera, si calculamos el gradiente con respecto a $\hat{\beta}$ se tiene:

$$\nabla_{w\beta} \hat{E}(w, \hat{\beta}; X, y) = diag \left(\nabla_w H_\gamma(w) \right) \in \mathbb{R}^{(d+1)\times(d+1)},$$

donde:

$$\nabla_w H_\gamma(w) = \left[w_1 \left(\gamma^2 + w_1^2 \right)^{-\frac{1}{2}}, \ldots, w_{d+1} \left(\gamma^2 + w_{d+1}^2 \right)^{-\frac{1}{2}} \right].$$

□

Tomando en cuenta las mismas consideraciones del caso escalar para el problema de nivel inferior en (4.27), se puede aplicar las condiciones necesarias de primer orden.

Entonces, el problema (4.27) binivel es equivalente al siguiente problema de optimización de un solo nivel:

$$(P'_V)\begin{cases} \min\limits_{\hat{\beta}\in\mathbb{R}^{d+1}_+} & J(\hat{\beta}, w(\hat{\beta}); V, \eta) := \dfrac{1}{2m}\|V\hat{w}(\hat{\beta})^\top - \eta\|_2^2 \\ \text{s.a.} & \nabla_w \hat{E}(\hat{w}, \hat{\beta}; X, y) = \mathbf{0}_{d+1}, \end{cases} \tag{4.29}$$

donde:

$$\nabla_w \hat{E}(\hat{w}, \hat{\beta}; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yX\hat{w}^\top)X + \alpha\hat{w} + (\nabla_w H_\gamma(\hat{w}))^\top diag(\hat{\beta}) \in \mathbb{R}^{d+1}.$$

Teorema 9 (Condiciones Karush-Kuhn-Tucker)**.** *Si $\hat{w}(\hat{\beta})$ es una solución local del problema (4.26) en el sentido de la Definición 5 tal que satisface la condición de calificación LICQ. Entonces, existe un multiplicador de Lagrange $\hat{\lambda} \in \mathbb{R}^{(d+1)}$ tal que cumple las siguientes condiciones:*

a) $\nabla_w \mathcal{L}(\hat{w}, \hat{\beta}, \hat{\lambda}) = \mathbf{0}_{d+1}$,

b) $\nabla_{\hat{\beta}} \mathcal{L}(\hat{w}, \hat{\beta}, \hat{\lambda}) = 0$.

Donde:

$$\mathcal{L}(w, \hat{\beta}, \lambda) = \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + (\nabla_w H_\gamma(\hat{w}))^\top diag(\hat{\beta})\right).$$

Demostración. Para justificar la existencia del multiplicador de Lagrange, usamos el literal a) del Lema 3. Es decir, sabemos que la función $\hat{E}$ es fuertemente convexa con módulo $\alpha > 0$, lo cual implica que la función $\hat{E}$ es estrictamente convexa.

Sabemos que la convexidad estricta es una condición suficiente para que se cumpla la condición de calificación LICQ [Crockett and Fessler, 2021]. En consecuencia, se garantiza la existencia de un multiplicador de Lagrange $\lambda \in \mathbb{R}^{d+1}$.

Luego, el Lagrangiano correspondiente al problema (4.29) está dado por:

$$\begin{aligned} \mathcal{L}(w, \hat{\beta}, \lambda) &= J(w, \hat{\beta}; V, \eta) + \lambda^\top \nabla_w \hat{E}(w, \hat{\beta}; X, y) \\ &= \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + (\nabla_w H_\gamma(\hat{w}))^\top diag(\hat{\beta})\right). \end{aligned}$$

Sea $\hat{w} = \hat{w}(\hat{\beta})$ una solución del problema (4.29) y $\hat{\lambda} \in \mathbb{R}^{d+1}$. Entonces, el gradiente del Lagrangiano con respecto a w está dado por:

$$\begin{aligned} \mathbf{0}_{d+1} = \nabla_w \mathcal{L}(\hat{w}, \hat{\beta}, \hat{\lambda}) &= \nabla_w J(w, \hat{\beta}; V, \eta) + \nabla^2_{ww} \hat{E}(w, \hat{\beta}; X, y)\hat{\lambda} \\ &= \frac{1}{m} V^\top (V\hat{w}^\top - \eta) + \left(\frac{1}{n} X^\top\, diag(\ell''_\mu(1 - yt))X + \alpha\mathbb{I} + \hat{\beta} \odot \nabla^2_{ww} H_\gamma(\hat{w})\right)\hat{\lambda}. \end{aligned}$$

Luego, se puede caracterizar el multiplicador óptimo como sigue:

$$\hat{\lambda} = -\left[\frac{1}{n}X^\top diag(\ell''_\mu(1-yt))X + \alpha\mathbb{I} + \hat{\beta}\odot\nabla^2_{ww}H_\gamma(\hat{w})\right]^{-1}\frac{1}{m}V^\top\left(V\hat{w}^\top - \eta\right),$$

donde $\mathbb{I}\in\mathbb{R}^{(d+1)\times(d+1)}$ es la matriz identidad.

Además, el gradiente con respecto al hiperparámetro $\hat{\beta}$, también conocido como hipergradiente, se expresa de la siguiente manera:

$$\begin{aligned}0 = \nabla_{\hat{\beta}}\mathcal{L}(\hat{w},\hat{\beta},\hat{\lambda}) &= \nabla_{\hat{\beta}}J(w,\hat{\beta};V,\eta) + \nabla_{w\hat{\beta}}\hat{E}(w,\hat{\beta};X,y)\hat{\lambda}\\ &= 0 + diag(\nabla_w H_\gamma(\hat{w})\hat{\lambda}\\ &= -diag(\nabla_w H_\gamma(\hat{w})G^{-1}\frac{1}{m}V^\top\left(V\hat{w}^\top-\eta\right),\end{aligned}$$

donde:

$$G := \left[\frac{1}{n}X^\top diag(\ell''_\mu(1-yt))X + \alpha\mathbb{I} + \hat{\beta}\odot\nabla^2_{ww}H_\gamma(\hat{w})\right].$$

□

Notemos que a diferencia del modelo escalar, donde el hipergradiente es real, ahora el hipergradiente para este caso es vectorial de tamaño $d+1$.

4.4.1. Criterio para estimar la importancia de variables

En este apartado se propone una metodología para medir la importancia de las variables seleccionadas a través del problema L_2L_1-SVM binivel. Los coeficientes del hiperplano óptimo $\hat{w}(\hat{\beta})$ y el hiperparámetro $\hat{\beta}$ pueden proporcionar la base para establecer una puntuación aproximada de cada una de las variables. La metodología propuesta se describe en los siguientes pasos:

1. Se asume que las variables del dominio tienen la misma escala, es decir, que han sido escaladas antes de resolver el problema (4.27).

2. Usaremos la dispersión de los coeficientes de $\hat{w}(\hat{\beta})$ para crear un nuevo hiperplano como sigue:

$$\bar{w}_j = \begin{cases}1, & \text{si } |\hat{w}_j| > \epsilon, \quad \forall j = 1,\dots,d+1,\\ 0, & \text{si } |\hat{w}_j| \le \epsilon, \quad \forall j = 1,\dots,d+1,\end{cases} \tag{4.30}$$

donde $\epsilon > 0$ es un valor suficientemente pequeño.

3. Se calcula el nuevo vector de hiperparámetros para cada variable:

$$\tilde{\beta} = \bar{w} \odot \hat{\beta}, \tag{4.31}$$

donde $\tilde{\beta} \in \mathbb{R}^{d+1}$ tiene la misma dimensión que el hiperparámetro original $\hat{\beta}$.

4. Con estos elementos se propone la puntuación para cada una de las $d+1$ variables se define a continuación:

$$Score_j = \frac{\tilde{\beta}_j}{\sum_{j=1}^{d+1} \tilde{\beta}_j}, \quad \forall j = 1, \ldots, d+1, \tag{4.32}$$

donde $Score_j \in [0, 1]$.

Por lo tanto, el $Score_j$ representa la proporción de participación de cada una de las $d+1$ variables.

4.5. Formulación del modelo vectorial - grupos

En esta sección se analiza el problema L_2L_1-SVM binivel aproximado para el caso cuando las $d+1$ variables (incluida la columna del término de sesgo) se encuentran divididas en $g+1$ grupos no sobrepuestos (se toma como grupo individual la columna del término de sesgo) de tamaño p_k no necesariamente homogéneos para $k = 1, \ldots, g+1$. Notemos que el caso vectorial es un caso particular de la formulación en grupos cuando solo se tiene un solo grupo de tamaño $p_1 = d+1$. De este modo, el hiperparámetro $\bar{\beta}$ admite una estructura de grupo de la forma $\bar{\beta} = (\bar{\beta}_1, \ldots, \bar{\beta}_k, \ldots, \bar{\beta}_{g+1})$ tal que $\bar{\beta}_k \in \mathbb{R}^{p_k}$ y $\sum_{k=1}^{g+1} p_k = d+1$.

Además, se utiliza una aproximación de la norma *Group Lasso* (1.8) definida por:

$$\bar{H}_\gamma(w_k) := (\gamma^2 + \|w_k\|_2^2)^{\frac{1}{2}} - \gamma, \tag{4.33}$$

donde $\gamma > 0$ para todo $k = 1, \ldots, g+1$.

Además, el gradiente está dado por:

$$\nabla_w \bar{H}_\gamma(w) = \left[\underbrace{w_1(\gamma^2 + \|w_1\|_2^2)^{-\frac{1}{2}}}_{\text{tamaño } p_1}, \ldots, \underbrace{w_{g+1}(\gamma^2 + \|w_{g+1}\|_2^2)^{-\frac{1}{2}}}_{\text{tamaño } p_{g+1}} \right] \in \mathbb{R}^{d+1}, \tag{4.34}$$

y su matriz Hessiana está dada por:

$$\nabla^2_{ww}\bar{H}_\gamma(w) = \gamma^2\, diag\left((\gamma^2 + \|w_1\|_2^2)^{-\frac{3}{2}}, \ldots, (\gamma^2 + \|w_{g+1}\|_2^2)^{-\frac{3}{2}}\right) \in \mathbb{R}^{(d+1)\times(d+1)}. \tag{4.35}$$

A continuación, la formulación para el problema L_2L_1-SVM binivel aproximado para el caso cuando el hiperparámetro $\bar{\beta}$ es vectorial en grupos y $\alpha > 0$ es fijo, se expresa de la siguiente manera:

$$(P_G)\begin{cases} \min\limits_{\bar{\beta} \in \mathbb{R}^{g+1}_+} & J(\bar{\beta}, w(\bar{\beta}); V, \eta) := \dfrac{1}{2m}\|V\hat{w}(\bar{\beta})^\top - \eta\|_2^2 \\ \text{s.a.} & \hat{w} = \arg\min\limits_{w} \bar{E}(w, \bar{\beta}; X, y), \end{cases} \tag{4.36}$$

donde la función objetivo del nivel inferior está dada por:

$$\bar{E}(w, \bar{\beta}; X, y) := \frac{1}{n}\sum_{i=1}^{n} \ell_\mu(1 - yt_i) + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}_\gamma(w_k), \tag{4.37}$$

con $t_i = \langle w, x_i\rangle$ para todo $i = 1, \ldots, n.$, $\tilde{H}_\gamma(\cdot)$ es la aproximación de la norma Group Lasso y $\ell_\mu(\cdot)$ es la función *hinge-loss* aproximada (4.18).

Lema 4. *Sean $X \in \mathbb{R}^{n\times(d+1)}$ un conjunto de entrenamiento, $V \in \mathbb{R}^{m\times(d+1)}$ un conjunto de validación, $w \in \mathbb{R}^{(d+1)}$ el hiperplano de separación, $y \in \{-1, 1\}^n$ un vector de etiquetas de los datos de entrenamiento y $\eta \in \{-1, 1\}^m$ un vector de etiquetas de los datos de validación para el problema de nivel inferior en (4.36). Entonces:*

a) Existe una solución única para el problema de nivel inferior en (4.36).

b) El gradiente de la función $\bar{E}$ con respecto a w está dada por:

$$\nabla_w \bar{E}(w, \bar{\beta}; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yXw^\top)^\top X + \alpha\bar{w} + (\nabla_w \bar{H}_\gamma(\hat{w}))^\top diag(\bar{\beta}) \in \mathbb{R}^{(d+1)}.$$

c) La Hessiana de $\bar{E}$ con respecto a w está dada por:

$$\nabla^2_{ww}\bar{E}(w, \bar{\beta}; X, y) = \frac{1}{n}X^\top diag\left(\ell''_\mu(1 - yXw^\top)\right) X + \alpha\mathbb{I} + \bar{\beta} \odot \nabla^2_{ww}\bar{H}_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

donde el símbolo $\odot$ representa la multiplicación punto a punto.

d) El gradiente con respecto a w y $\bar{\beta}$ de $\bar{E}$ está dado por:

$$\nabla_{w\bar{\beta}}\bar{E}(w, \bar{\beta}; X, y) = diag(\nabla_w \bar{H}_\gamma(w)) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

Demostración. Para la demostración de este teorema, nos centramos en el término de la aproximación de la norma Group Lasso (4.33), ya que los otros términos son los mismos que en la formulación escalar.

Sea la función:

$$\bar{E}(w, \bar{\beta}; X, y) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell_\mu(1 - y_i t_i) + \sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}_\gamma(w_k)}_{f} + \underbrace{\frac{\alpha}{2}\|w\|_2^2}_{g},$$

donde $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $\alpha > 0$ y $\bar{\beta} \in \mathbb{R}_+^{(g+1)}$.

a) Para probar que existe una solución única para el problema de nivel inferior en (4.36), necesitamos probar que la función objetivo $\bar{E} = f + g$ es fuertemente convexa.

   Para esto, probemos que la aproximación de la norma Group Lasso (4.33) es convexa.

   La segunda derivada está definida por:

$$\bar{H}''_\gamma(w_k) = \gamma^2(\gamma^2 + \|w_k\|_2^2)^{-\frac{3}{2}},$$

   para todo $k = 1, \ldots, g + 1$, con $\gamma > 0$.

   Claramente, la segunda derivada es siempre positiva, por lo cual podemos inferir que $\bar{H}_\gamma(\cdot)$ es convexa. En consecuencia, $\bar{E} = f + g$ es fuertemente convexa con módulo $\alpha > 0$.

   Por lo tanto, gracias al Teorema 5 se concluye que $\bar{E} = f + g$ posee un único minimizador.

b) Para calcular el gradiente de la función $\bar{E}$, analizamos el término de la aproximación de la norma Group Lasso.

$$\begin{aligned}\left[\frac{\partial}{\partial w}\left(\sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}_\gamma(w_k)\right)\right]_i &= \frac{\partial}{\partial w_i}\left(\sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}_\gamma(w_k)\right) \\ &= \sum_{k=1}^{g+1} \bar{\beta}_k \frac{\partial}{\partial w_i}\left(\bar{H}_\gamma(w_k)\right) \\ &= \sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}'_\gamma(w_k),\end{aligned}$$

   Luego, el gradiente con respecto a w es:

$$\nabla_w\left(\sum_{k=1}^{g+1} \bar{\beta}_k \bar{H}_\gamma(w_k)\right) = \left(\nabla_w \bar{H}_\gamma(w)\right)^\top diag(\bar{\beta}) \in \mathbb{R}^{(d+1)},$$

donde:

$$\nabla_w \bar{H}_\gamma(w) = \left[w_1(\gamma^2 + \|w_1\|_2^2)^{-\frac{1}{2}}, \ldots, w_{g+1}(\gamma^2 + \|w_{g+1}\|_2^2)^{-\frac{1}{2}} \right].$$

Por lo tanto, el gradiente de $\bar{E}$ con respecto a w es:

$$\nabla_w \bar{E}(w, \bar{\beta}; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yXw^\top)^\top X + \alpha w + (\nabla_w \bar{H}_\gamma(w))^\top diag(\bar{\beta}) \in \mathbb{R}^{(d+1)}.$$

c) Para calcular la Hessiana de $\bar{E}$ analizamos la aproximación de la norma Group Lasso.

$$\nabla_w \left((\nabla_w \bar{H}_\gamma(w))^\top diag(\bar{\beta}) \right) = \bar{\beta} \odot \nabla^2_{ww} \bar{H}_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)},$$

donde:

$$\nabla^2_{ww} \bar{H}_\gamma(w) = \gamma^2\, diag\left((\gamma^2 + \|w_1\|_2^2)^{-\frac{3}{2}}, \ldots, (\gamma^2 + \|w_{g+1}\|_2^2)^{-\frac{3}{2}} \right).$$

Por lo tanto, la Hessiana de $\bar{E}$ con respecto a w es:

$$\nabla^2_{ww} \bar{E}(w, \beta; X, y) = \frac{1}{n} X^\top diag\left(\ell''_\mu(1 - yXw^\top) \right) X + \alpha \mathbb{I} + \bar{\beta} \odot \nabla^2_{ww} \bar{H}_\gamma(w) \in \mathbb{R}^{(d+1)\times(d+1)}.$$

d) Del literal b) se tiene que el gradiente de $\bar{E}$ con respecto a w está definido por:

$$\nabla_w \bar{E}(w, \bar{\beta}; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yXw^\top)^\top X + \alpha w + (\nabla_w \bar{H}_\gamma(w))^\top diag(\bar{\beta}).$$

De esta manera, si calculamos el gradiente con respecto a $\bar{\beta}$ se tiene:

$$\nabla_{w\bar{\beta}} \bar{E}(w, \beta; X, y) = diag\left(\nabla_w \bar{H}_\gamma(w) \right) \in \mathbb{R}^{(d+1)\times(d+1)},$$

donde:

$$\nabla_w \bar{H}_\gamma(w) = \left[w_1(\gamma^2 + \|w_1\|_2^2)^{-\frac{1}{2}}, \ldots, w_{g+1}(\gamma^2 + \|w_{g+1}\|_2^2)^{-\frac{1}{2}} \right].$$

□

De la misma manera como se ha tratado anteriormente, el problema (4.36) binivel es equivalente al siguiente problema de optimización de un solo nivel:

$$(P'_G) \begin{cases} \min\limits_{\bar{\beta} \in \mathbb{R}^{g+1}_+} & J(\bar{\beta}, w(\bar{\beta}); V, \eta) := \dfrac{1}{2m} \|V \hat{w}(\bar{\beta})^\top - \eta\|_2^2 \\ \text{s.a.} & \nabla_w \bar{E}(\hat{w}, \bar{\beta}; X, y) = \mathbf{0}_{d+1}, \end{cases} \tag{4.39}$$

donde:

$$\nabla_w \bar{E}(\hat{w}, \bar{\beta}; X, y) = -\frac{1}{n}\, y\, \ell'_\mu(1 - yX\hat{w}^\top)X + \alpha\hat{w} + (\nabla_w \tilde{H}_\gamma(\hat{w}))^\top diag(\bar{\beta}) \in \mathbb{R}^{d+1}.$$

Teorema 10 (Condiciones Karush-Kuhn-Tucker)**.** *Si $\hat{w}(\bar{\beta})$ es una solución local del problema (4.38) en el sentido de la Definición 5 tal que satisface la condición de calificación LICQ. Entonces, existe un multiplicador de Lagrange $\hat{\lambda} \in \mathbb{R}^{(d+1)}$ tal que cumple las siguientes condiciones:*

a) $\nabla_w \mathcal{L}(\hat{w}, \bar{\beta}, \hat{\lambda}) = \mathbf{0}_{d+1}$,

b) $\nabla_{\bar{\beta}} \mathcal{L}(\hat{w}, \bar{\beta}, \hat{\lambda}) = 0$.

Donde:

$$\mathcal{L}(w, \bar{\beta}, \lambda) = \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + (\nabla_w \bar{H}_\gamma(\hat{w}))^\top diag(\bar{\beta})\right).$$

Demostración. Para justificar la existencia del multiplicador de Lagrange, usamos el literal a) del Lema 4. Es decir, sabemos que la función $\bar{E}$ es fuertemente convexa con módulo $\alpha > 0$, lo cual implica que la función $\bar{E}$ es estrictamente convexa.

Asimismo como en las formulaciones anteriores, se sabe que la convexidad estricta es una condición suficiente para que se cumpla la condición de calificación LICQ [Crockett and Fessler, 2021]. En consecuencia, se garantiza la existencia de los multiplicadores de Lagrange $\lambda \in \mathbb{R}^{d+1}$.

Luego, el Lagrangiano correspondiente al problema (4.38) está dado por:

$$\begin{aligned}\mathcal{L}(w, \bar{\beta}, \lambda) &= J(w, \bar{\beta}; V, \eta) + \lambda^\top \nabla_w \bar{E}(w, \bar{\beta}; X, y) \\ &= \frac{1}{2m}\|Vw^\top - \eta\|_2^2 - \lambda^\top \left(\frac{1}{n}\, y\, \ell'_\mu(1 - yt)^\top X + \alpha w + (\nabla_w \bar{H}_\gamma(\hat{w}))^\top diag(\bar{\beta})\right).\end{aligned}$$

Sea $\hat{w} = \hat{w}(\bar{\beta})$ una solución del problema (4.38) y $\hat{\lambda} \in \mathbb{R}^{d+1}$. Entonces, el gradiente del Lagrangiano con respecto a w está dado por:

$$\begin{aligned}\mathbf{0}_{d+1} = \nabla_w \mathcal{L}(\hat{w}, \bar{\beta}, \hat{\lambda}) &= \nabla_w J(w, \bar{\beta}; V, \eta) + \nabla^2_{ww} \bar{E}(w, \bar{\beta}; X, y)\hat{\lambda} \\ &= \frac{1}{m} V^\top (V\hat{w}^\top - \eta) + \left(\frac{1}{n} X^\top\, diag(\ell''_\mu(1 - yt))X + \alpha\mathbb{I} + \bar{\beta} \odot \nabla^2_{ww} \bar{H}_\gamma(\hat{w})\right)\hat{\lambda}.\end{aligned}$$

Luego, se puede caracterizar el multiplicador óptimo como sigue:

$$\hat{\lambda} = -\left[\frac{1}{n} X^\top\, diag(\ell''_\mu(1 - yt))X + \alpha\mathbb{I} + \bar{\beta} \odot \nabla^2_{ww} \bar{H}_\gamma(\hat{w})\right]^{-1} \frac{1}{m} V^\top \left(V\hat{w}^\top - \eta\right),$$

donde $\mathbb{I} \in \mathbb{R}^{(d+1)\times(d+1)}$ es la matriz identidad.

Además, el gradiente con respecto al hiperparámetro $\bar{\beta}$, también conocido como hipergradiente, se expresa de la siguiente manera:

$$\begin{aligned} 0 = \nabla_{\bar{\beta}} \mathcal{L}(\hat{w}, \bar{\beta}, \hat{\lambda}) &= \nabla_{\bar{\beta}} J(w, \bar{\beta}; V, \eta) + \nabla_{w\bar{\beta}} \hat{E}(w, \beta; X, y)\hat{\lambda} \\ &= 0 + diag(\nabla_w \bar{H}_\gamma(\hat{w})\hat{\lambda} \\ &= -diag(\nabla_w \bar{H}_\gamma(\hat{w}) G^{-1} \frac{1}{m} V^\top \left(V\hat{w}^\top - \eta \right), \end{aligned}$$

donde:

$$G := \left[\frac{1}{n} X^\top diag(\ell''_\mu(1 - yt))X + \alpha\mathbb{I} + \hat{\beta} \odot \nabla^2_{ww} \bar{H}_\gamma(\hat{w}) \right].$$

□

4.5.1. Criterio para estimar la importancia de grupos

A diferencia de la formulación vectorial, ahora se propone una metodología para medir la importancia de los grupos seleccionados a través del problema L_2L_1-SVM binivel. Asimismo, a partir de los coeficientes del hiperplano óptimo $\tilde{w}(\tilde{\beta})$ y el hiperparámetro $\tilde{\beta}$ pueden proporcionar la base para establecer una puntuación aproximada de cada uno de los grupos de variables. La metodología propuesta se describe en los pasos siguientes:

1. Del mismo modo que el modelo vectorial, se asume que las variables del dominio tienen la misma escala, es decir, que han sido escaladas antes de resolver el problema (4.36).

2. Se construye un hiperplano auxiliar de la forma:

$$\overline{w}_k = \begin{cases} 1, & \text{si } \|\hat{w}_k\|_2^2 > \epsilon, \quad \forall k = 1, \ldots, g+1, \\ 0, & \text{si } \|\hat{w}_k\|_2^2 \leq \epsilon, \quad \forall k = 1, \ldots, g+1. \end{cases} \tag{4.40}$$

   donde $\epsilon > 0$ es un valor suficientemente pequeño.

3. Se calcula el nuevo vector de hiperparámetros agrupados:

$$\tilde{\beta} = \overline{w} \odot \tilde{\beta} \tag{4.41}$$

   donde $\tilde{\beta} \in \mathbb{R}^{g+1}$ tiene la misma dimensión que el hiperparámetro original $\tilde{\beta}$.

4. Con estos elementos se propone la puntuación para cada una de las $d+1$ variables se

define a continuación:

$$Score_k = \frac{\tilde{\beta}_k}{\sum_{k=1}^{g+1} \tilde{\beta}_k}, \quad \forall k = 1, \ldots, g+1. \tag{4.42}$$

donde $Score_k \in [0, 1]$.

Por lo tanto, el $Score_k$ representa la proporción de participación de cada uno de los grupos.

En resumen, la metodología propuesta para la selección de variables a través del problema L_2L_1-SVM desde un enfoque de optimización binivel, asume que los problemas de nivel superior e inferior no tienen restricciones de desigualdad. Además, la función que se utiliza para la estimación del error de validación es dos veces diferenciable con respecto al parámetro w. Asimismo, la función objetivo del nivel inferior es estrictamente convexa, lo cual garantiza la existencia de una solución única; y adicionalmente, es dos veces diferenciable con respecto a w. Finalmente, para cada una de las formulaciones del problema L_2L_1-SVM binivel, se usa una función objetivo aproximada del nivel inferior con el propósito de obtener las condiciones de optimalidad del problema y usar algoritmos ampliamente estudiados para su resolución.

Capítulo 5

Algoritmos de Optimización

En este capítulo presentamos una revisión de los algoritmos de optimización utilizados para resolver numéricamente el problema de optimización binivel L_2L_1-SVM aproximado con la siguiente formulación:

$$(P)\begin{cases}\min\limits_{\beta} & J(\beta, w(\beta); V, \eta) := \dfrac{1}{2m}\|V\hat{w}(\beta)^\top - \eta\|_2^2 \\ \text{s.a.} & \hat{w}(\beta) = \arg\min\limits_{w} E(w, \beta; X, y),\end{cases} \tag{5.1}$$

donde la función objetivo E del problema de nivel inferior representa el problema L_2L1-SVM, para cualquiera de los casos cuando el hiperparámetro β es escalar, vectorial o vectorial agrupado.

Para resolver el problema de nivel inferior (L_2L_1-SVM) y el problema de nivel superior aplicamos el algoritmo de memoria limitada L-BFGS [Byrd et al., 1995].

5.1. Solución nivel inferior

Existen diversos algoritmos para resolver un problema de optimización sin restricciones de la forma:

$$\min_{x \in \mathbb{R}^n} \quad f(x), \tag{5.2}$$

donde la función objetivo f es diferenciable.

Estos algoritmos para resolver problemas de optimización sin restricciones, inician a partir de un punto inicial $x^{(0)}$, generando una secuencia de iteraciones $\{x^{(k)}\}_{k\in\mathbb{N}}$ que convergen hacia la solución del problema con cierta precisión.

Las estrategias fundamentales para la transición de una iteración $x^{(k)}$ a una nueva iteración $x^{(k+1)}$ son: la búsqueda lineal y la región de confianza.

En el presente trabajo de investigación, nos centramos en la estrategia de búsqueda lineal para la actualización de las iteraciones.

Los métodos de búsqueda lineal presentan una iteración que calcula una dirección de búsqueda p_k, y un tamaño de paso $\tau_k > 0$ que determina la distancia a recorrer en esa dirección.

La iteración está dada por:

$$x^{(k+1)} = x^{(k)} + \tau_k p_k, \quad \forall k \in \mathbb{N}. \tag{5.3}$$

La mayoría de los métodos de búsqueda lineal exigen que la dirección de búsqueda p_k sea una dirección de descenso. Es decir, la dirección p_k debe cumplir con la siguiente condición:

$$p_k^\top \nabla_x f(x^{(k)}) < 0, \quad \forall k \in \mathbb{N}. \tag{5.4}$$

Además, la dirección de búsqueda adopta la forma general:

$$p_k = -B_k^{-1} \nabla_x f(x^{(k)}), \quad \forall k \in \mathbb{N}, \tag{5.5}$$

donde B_k es una matriz simétrica, definida positiva y no singular. Generalmente, se nota por H_k a la inversa de la matriz B_k, es decir, $H_k := B_k^{-1}$.

De acuerdo con [Nocedal and Wright, 2006], la elección de la matriz B_k clasifica los métodos de la siguiente manera:

- En el caso donde la matriz B_k es la matriz identidad $\mathbb{I}$ para todo $k \in \mathbb{N}$, se obtiene el método del descenso más profundo. Este enfoque, que utiliza únicamente información de primer orden, tiene la ventaja de simplicidad. Sin embargo, en problemas complejos, puede ser lento debido a su tasa de convergencia lineal.

- Cuando la matriz B_k es la matriz Hessiana exacta $\nabla^2_{xx} f(x^{(k)})$ para todo $k \in \mathbb{N}$, el método se conoce como el método de Newton. Aunque este método converge más rápido que el descenso más profundo, su alto costo computacional, derivado del cálculo de la matriz Hessiana, puede limitar su eficiencia.

- En el caso en que la matriz B_k sea una aproximación de la matriz Hessiana $\nabla^2_{xx} f(x^{(k)})$ para todo $k \in \mathbb{N}$, también se le denomina el método de quasi-Newton. Este método logra una convergencia más rápida que el descenso más profundo, aunque más lenta que el método de Newton, exhibiendo así una tasa de convergencia superlineal.

Por otra parte, los algoritmos de búsqueda lineal examinan una secuencia de valores candidatos para el tamaño de paso τ_k, deteniéndose para aceptar uno de estos valores cuando se cumplen ciertas condiciones conocidas como las condiciones de *Wolfe*.

Descenso suficiente: Esta condición establece que el tamaño de paso τ debe tener una disminución suficiente en la función objetivo f, la cual se puede medir a través de la siguiente desigualdad:

$$f(x^{(k)} + \tau p_k) \leq f(x^{(k)}) + c_1 \tau \nabla_x f(x^{(k)})^\top p_k \tag{5.6}$$

donde $c_1 \in (0,1)$ y $k \in \mathbb{N}$. Esta condición a veces se la conoce como la condición de Armijo.

Condición de curvatura: Sin embargo, la condición anterior por si sola no es suficiente, ya que el tamaño de paso τ podría ser muy pequeño. En este sentido, esta segunda condición requiere que τ cumpla con lo siguiente:

$$\nabla_x f(x^{(k)} + \tau p_k)^\top p_k \geq c_2 \nabla_x f(x^{(k)})^\top p_k \tag{5.7}$$

donde $c_2 \in (c_1, 1)$ y $k \in \mathbb{N}$.

A continuación, se describe el algoritmo *Backtracking Line Search*, utilizado para determinar el tamaño de paso τ.

Algoritmo 6 Backtracking Line Search

Input: $\bar{\tau} > 0$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $\rho \in (0,1)$.
1: $\tau \leftarrow \bar{\tau}$;
2: **while** (5.6) y (5.7) **do**
3: $\quad \tau \leftarrow \rho\tau$;
4: **end while**
5: Finalizar con $\tau_k = \tau$.

En este sentido, para resolver el problema de nivel inferior en (5.1), usamos el algoritmo L-BFGS que pertenece a la clase de los métodos quasi-Newton. Este método es útil para resolver problemas a gran escala, en especial, cuando la matriz Hessiana no se puede calcular con un costo computacional razonable o no tiene una estructura dispersa.

El algoritmo L-BFGS para determinar la solución del L_2L_1-SVM (nivel inferior) es:

Algoritmo 7 SVM L-BFGS

Input: $\beta^{(0)}$, $w^{(0)}$, $tol > 0$, $m > 0$, $c_1 \in (0,1)$, $c_2 \in (c_1,1)$, $\mu \in (0,1)$, $\gamma > 0$, $\alpha > 0$, X, y, V, η.
Output: $\hat{w}$ óptimo.

```
k ← 0;
H_0 = 𝕀;
while ‖∇_w E(w^(k), β^(k))‖ > tol do
    Calcular: p_k = −H_k ∇_w E(w^(k), β^(k)) (aplicar Algoritmo 3);
    Calcular: τ_k (aplicar Algoritmo 6);
    Calcular: w^(k+1) = w^(k) + τ_k p_k;
    if k >m then
        Descartar el par de vectores {s_{k−m}, y_{k−m}} del almacenamiento.
    end if
    Calcular: s_k = w^(k+1) − w^(k),   y_k = ∇_w E(w^(k+1), β^(k+1)) − ∇_w E(w^(k, β^(k));
    k ← k + 1.
end while
```

5.2. Solución binivel

Sabemos que gracias a la diferenciabilidad de la función objetivo E del nivel inferior, el problema binivel (5.1) es equivalente a:

$$(P') \begin{cases} \min\limits_{\beta} \quad J(\beta, w(\beta); V, \eta) := \dfrac{1}{2m} \|V\hat{w}(\beta)^\top - \eta\|_2^2 \\ \text{s.a.} \qquad \nabla_w E(\hat{w}, \beta; X, y) = \mathbf{0}_{d+1}. \end{cases} \tag{5.8}$$

Por lo tanto, para resolver el problema de un solo nivel con restricciones (5.8), usamos un método de descenso de tipo quasi-Newton (algoritmo L-BFGS), donde se busca una actualización del hiperparámetro β (para todos sus casos) de la forma:

$$\beta^{(k+1)} = \beta^{(k)} + \tau_k d_k, \quad \forall k \in \mathbb{N}. \tag{5.9}$$

donde d_k es la dirección de descenso y τ_k el tamaño de paso de búsqueda que puede ser calculado a través del Algoritmo 6.

Como se mostró anteriormente, la convexidad estricta de la función E es una condición suficiente para la existencia de un multiplicador de Lagrange $\lambda \in \mathbb{R}^{(d+1)}$ tal que:

$$\mathcal{L}(w, \beta, \lambda) = J(w, \beta) + \lambda^\top \nabla_w E(w, \beta) \tag{5.10}$$

Luego, aplicando las condiciones KKT (Teorema 3) al problema (5.8), la primera condición establece que:

$$\nabla_w \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \nabla_w J(\beta, \hat{w}(\beta)) + \nabla^2_{ww} E(\hat{w}, \beta)\hat{\lambda} = \mathbf{0}_{d+1}. \tag{5.11}$$

De la ecuación (5.11) se caracteriza el multiplicador de Lagrange óptimo como sigue:

$$\hat{\lambda} = -\left[\nabla^2_{ww} E(\hat{w}, \beta)\right]^{-1} \nabla_w J(\beta, \hat{w}(\beta)). \tag{5.12}$$

Luego, el gradiente con respecto al parámetro β (hipergradiente) es:

$$\nabla_\beta \mathcal{L}(\hat{w}, \beta, \hat{\lambda}) = \left(\nabla_{w\beta} E(\hat{w}, \beta)\right)^\top \hat{\lambda}. \tag{5.13}$$

el cual se usa como dirección de descenso para la actualización del hiperparámetro β.

Finalmente, el algoritmo L-BFGS para resolver el problema de optimización binivel L_2L_1-SVM aproximado se presenta a continuación:

Algoritmo 8 BiSVM L-BFGS

Input: $\beta^{(0)}$, $max\_iter > 0$, $m > 0$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $\mu \in (0,1)$, $\gamma > 0$, $\alpha > 0$, X, y, V, η.
Output: $\hat{\beta}$, $\hat{w}$ óptimos.

$k \leftarrow 0$;
$H_0 = \mathbb{I}$;
$w^{(0)} = \arg\min_w E(w, \beta^{(0)})$ (aplicar Algoritmo 7);
$\lambda^{(0)} = -[\nabla^2_{ww} E(w^{(0)}, \beta^{(0)})]^{-1} \nabla_w J(w^{(0)})$;
$d_0 = H_0 \nabla_\beta \mathcal{L}(w^{(0)}, \beta^{(0)}, \lambda^{(0)})$;
while $k \leq max\_iter$ **do**
    Calcular: τ_k (aplicar Algoritmo 6);
    Actualizar: $\beta^{(k+1)} = \beta^{(k)} + \tau_k d_k$;
    Actualizar: $w^{(k+1)} = \arg\min_w E(w, \beta^{(k+1)})$ (aplicar Algoritmo 7);
    Actualizar: $\lambda^{(k+1)} = -[\nabla^2_{ww} E(w^{(k+1)}, \beta^{(k+1)})]^{-1} \nabla_w J(w^{(k+1)})$;
    Actualizar: $d_{k+1} = -H_{k+1} \nabla_\beta \mathcal{L}(w^{(k+1)}, \beta^{(k+1)}, \lambda^{(k+1)})$, donde H_{k+1} es la aproximación BFGS de la inversa de la Hessiana construida en base al hipergradiente. (aplicar Algoritmo 3);
    if k >m **then**
        Descartar el par de vectores $\{s_{k-m}, y_{k-m}\}$ del almacenamiento.
    end if
    Calcular: $s_k = \beta^{(k+1)} - \beta^{(k)}, \quad y_k = \nabla_\beta \mathcal{L}(w^{(k+1)}, \beta^{(k+1)}, \lambda^{(k+1)}) - \nabla_\beta \mathcal{L}(w^{(k)}, \beta^{(k)}, \lambda^{(k)})$;
    $k \leftarrow k + 1$.
end while

Capítulo 6

Experimentos Numéricos

En este capítulo, evaluamos el desempeño del Algoritmo 8, detallado en la Sección 5.2, en cada variante del problema L_2L_1-SVM binivel, utilizando conjuntos de datos de referencia en el ámbito de la ciencia de datos, los cuales se describen a continuación:

Iris: Es uno de los primeros conjuntos utilizados en la literatura de los métodos de clasificación y aplicaciones de aprendizaje automático. Este conjunto contiene 3 especies (setosa, versicolor, virginica) de la planta iris con 50 observaciones cada una. Además, se usan 4 variables explicativas como el largo y ancho tanto del pétalo como del sépalo. Para el caso de clasificación binaria se toman las especies setosa y versicolor representadas con los valores de -1 y 1 respectivamente.

Breast Cancer: Es un conjunto de datos que contiene 569 observaciones acerca del diagnóstico de cancer de mama, el cual se representa con el valor de -1 si se diagnostica un tumor maligno y el valor de 1 si es benigno. Este conjunto de datos contiene 30 variables explicativas asociadas en 3 grupos (media, desviación estándar y peor valor reportado) de 10 variables.

Churn: Este conjunto de datos posee información de una empresa de telecomunicaciones que brinda servicio de banda ancha y desea predecir el abandono de sus clientes. El conjunto de datos contiene 18 variables explicativas y la variable objetivo binaria que toma el valor de 1 si el cliente abandona el servicio y -1 caso contrario. Se realiza un preprocesamiento de datos donde se obtienen 27405 observaciones únicas y 8 variables explicativas.

La descripción de las variables de cada uno de los conjuntos de datos utilizados y el repositorio de estos se encuentran en el Anexo A.

Para el escenario en el cual el hiperparámetro β es escalar, examinamos la sensibilidad de los parámetros $\mu \in (0,1)$ y $\gamma > 0$ de las funciones aproximadas *hinge-loss* y Pseudo-Huber respectivamente. Además, llevamos a cabo una comparación entre una técnica tradicional, como la búsqueda exhaustiva (*Grid Search*) y el Algoritmo BiSVM L-BFGS propuesto.

En el caso en que el hiperparámetro $\hat{\beta}$ es un vector, evaluamos la metodología propuesta para la selección de variables en comparación con otro método de selección de variables como *Tree-Based Feature Selection*.

Finalmente, evaluamos el algoritmo BiSVM L-BFGS propuesto para la selección de grupos de variables.

6.1. Hiperparámetro escalar

En esta sección se realizan los experimentos para la siguiente formulación:

$$(P_E)\begin{cases} \min\limits_{\beta > 0} & J(\beta, w(\beta); V, \eta) := \dfrac{1}{2m}\|V\hat{w}(\beta)^\top - \eta\|_2^2 \\ \text{s.a.} & \hat{w} = \arg\min\limits_{w} \tilde{E}(w, \beta; X, y), \end{cases} \tag{6.1}$$

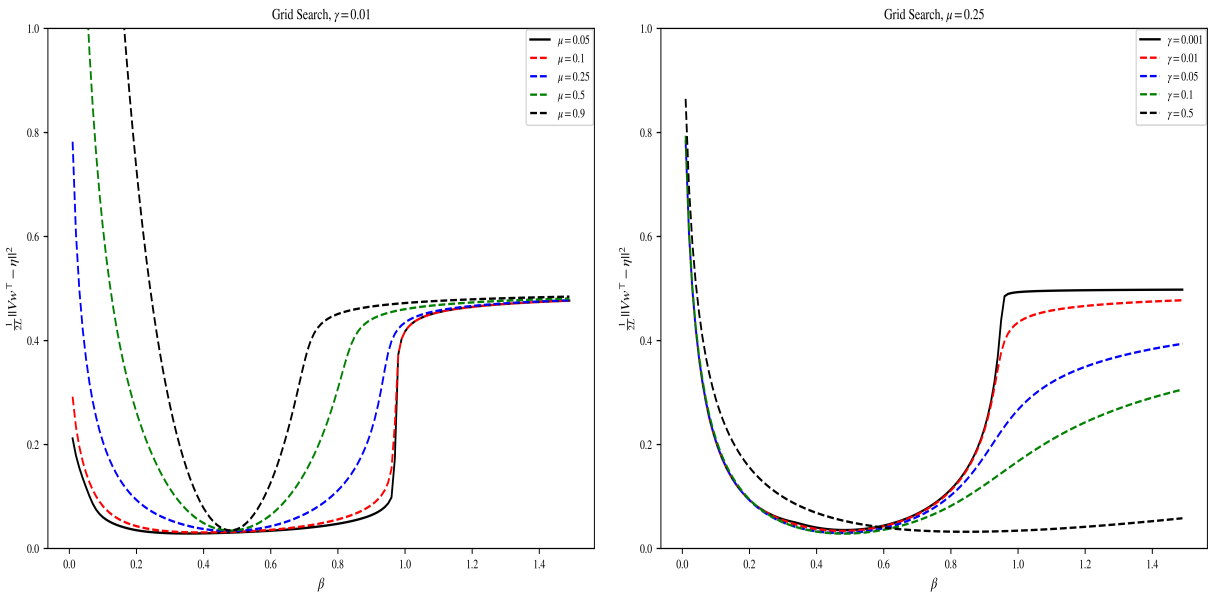
donde la función objetivo del problema de nivel inferior, que depende de los parámetros $\mu > 0$ y $\gamma > 0$, está dada por:

$$\tilde{E}(w, \beta; X, y) := \frac{1}{n}\sum_{i=1}^{n} \ell_\mu(1 - y_i t_i) + \frac{\alpha}{2}\sum_{j=1}^{d+1} w_j^2 + \beta \sum_{j=1}^{d+1} H_\gamma(w_j), \tag{6.2}$$

con $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $H_\gamma(\cdot)$ representa la regularización Pseudo-Huber de la norma L_1 (4.15) y $\ell_\mu(\cdot)$ es la aproximación de la función *hinge-loss* (4.18).

En una primera instancia, empleamos el conjunto de datos Iris para explorar la sensibilidad de los parámetros $\mu \in (0,1)$, asociado a la aproximación de la función *hinge-loss* y $\gamma > 0$ asociado a la regularización Pseudo-Huber, utilizando distintos valores. Además, para la partición de los datos, se asigna el $80\,\%$ al conjunto de entrenamiento X y el $20\,\%$ al conjunto de validación V.

Para lograr este objetivo, realizamos una búsqueda exhaustiva o *Grid Search* para el hiperparámetro β. Es decir, creamos un conjunto de 149 valores dentro del conjunto $(0;\, 1{,}5]$ para resolver el problema de nivel inferior con el Algoritmo 7. Luego, evaluamos cada hiperplano $w(\beta)$ para todo $\beta \in (0;\, 1{,}5]$ en la función de error de validación $J(\cdot)$. Además, fijamos el hiperparámetro de la norma L_2 en $\alpha = 0{,}01$.



(a) Grid Search para diferentes valores de μ.

(b) Grid Search para diferentes valores de γ.

Figura 6.1: Error de validación sobre $\beta \in (0;\ 1{,}5]$ para el conjunto de datos Iris.

La Figura 6.1a muestra la sensibilidad del parámetro $\mu \in (0,1)$ que corresponde a la aproximación *hinge-loss*, cuando se fija un valor $\gamma = 0{,}01$. Cuando $\mu \to 0$ (tiende a la función *hinge-loss* original) la función de error de validación $J(\cdot)$ ya no solo tiene un único minimizador β, sino que posee un conjunto de valores β donde se minimiza la función $J(\cdot)$. Esto generalmente sucede cuando se utiliza la función original *hinge-loss*.

Por otra parte, la Figura 6.1b muestra la sensibilidad del parámetro $\gamma > 0$ correspondiente a la regularización Psedo-Huber de la norma L_1, cuando se fija un valor $\mu = 0{,}25$. Cuando el valor de $\gamma > 0$ aumenta podemos ver que la función $J(\cdot)$ se suaviza de tal manera que no se estanca en ningún punto cuando se usa una dirección de descenso como el hipergradiente.

El análisis de sensibilidad de los parámetros $\mu \in (0,1)$ y γ que se usan en el L_2L_1-SVM aproximado nos da una idea de como se debe elegir cada uno de ellos.

Ahora, para el mismo conjunto Iris, podemos comparar la búsqueda exhaustiva con el Algoritmo 8 BiSVM L-BFGS para diferentes valores de μ y γ como se puede ver a continuación en la Figura 6.2:

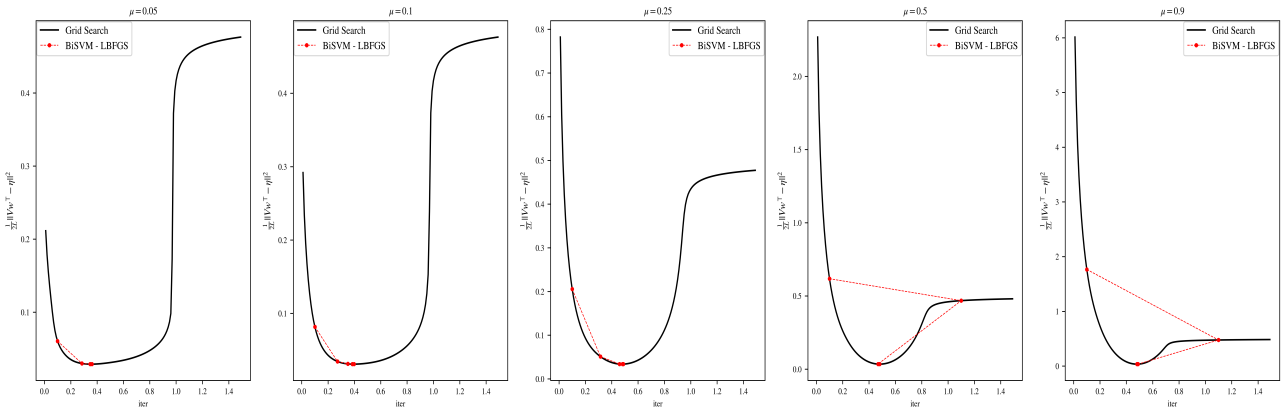


Figura 6.2: Comparación Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\gamma = 0{,}01$ fijo y diferentes valores de μ.

Parámetro	J_{min} GS	β GS	J_{min} BiSVM L-BFGS	β BiSVM L-BFGS	Iteraciones
$\mu = 0{,}05$	0,0287	0,36	0,0288	0,3610	3
$\mu = 0{,}10$	0,0308	0,39	0,0308	0,3979	4
$\mu = 0{,}25$	0,0337	0,48	0,0337	0,4658	3
$\mu = 0{,}50$	0,0344	0,48	0,0344	0,4809	3
$\mu = 0{,}90$	0,0346	0,48	0,0346	0,4806	3

Tabla 6.1: Resultados de Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\gamma = 0{,}01$ fijo y diferentes valores de μ

Podemos ver que el Algoritmo BiSVM L-BFGS realiza entre 3 y 4 iteraciones para cada caso y claramente es más eficiente que el método de búsqueda exhaustiva (*Grid Search*) donde se usó 149 operaciones para encontrar el mínimo de la función de error de validación $J(\cdot)$. Además, podemos que en algunos casos ($\mu = 0{,}5$ y $\mu = 0{,}9$) se alcanza un valor menor de la función de error de validación $J(\cdot)$ a lo alcanzado con la búsqueda exhaustiva. Asimismo, se puede aprovechar la continuidad del hiperparámetro óptimo encontrado a través del algoritmo propuesto. Es importante mencionar que para todos los experimentos el algoritmo inicia con $\beta_{inicial} = 0{,}01$.

Del mismo modo, podemos realizar el mismo experimento para el caso cuando fijamos un valor de $\mu = 0{,}25$ y variamos el parámetro γ como se muestra a continuación:

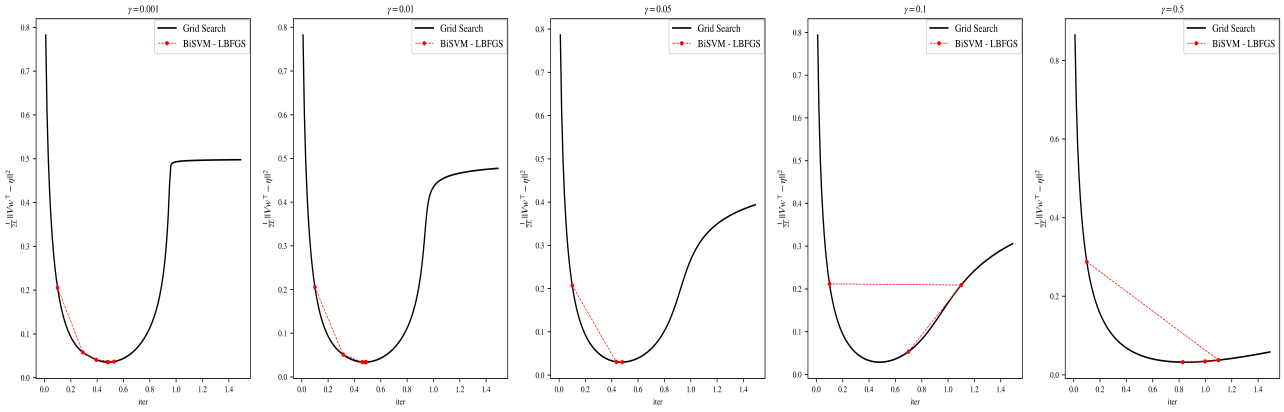


Figura 6.3: Comparación Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\mu = 0{,}25$ fijo y diferentes valores de γ.

Parámetro	J_{min} GS	β GS	J_{min} BiSVM L-BFGS	β BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}001$	0,0287	0,36	0,0354	0,4840	4
$\gamma = 0{,}01$	0,0308	0,39	0,0337	0,4817	3
$\gamma = 0{,}05$	0,0337	0,48	0,0297	0,4806	2
$\gamma = 0{,}10$	0,0344	0,48	0,0532	0,6995	2
$\gamma = 0{,}50$	0,0346	0,48	0,0321	0,8282	3

Tabla 6.2: Resultados de Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\mu = 0{,}25$ fijo y diferentes valores de γ.

El análisis de sensibilidad de parámetros nos ayuda a determinar los parámetros μ, γ adecuados para el problema de clasificación de especies, tomando en cuenta un nivel aceptable de error de validación.

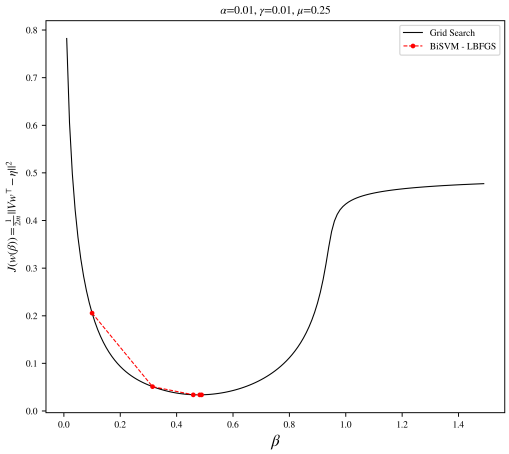


Figura 6.4: Comparación Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

Parámetros	J_{min} GS	β GS	J_{min} BiSVM L-BFGS	β BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}01$ $\mu = 0{,}25$	0,0337	0,48	0,0337	0,4817	4

Tabla 6.3: Resultados de Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\mu = 0{,}25$ y $\gamma = 0{,}01$.

En la Tabla 6.3 se observa que el Algoritmo BiSVM L-BFGS alcanza el mismo valor mínimo de la función de error de validación $J(\cdot)$ que se logró con *Grid Search*, pero en tan solo 4 iteraciones.

Por otro lado, se presenta el hiperplano de separación obtenido a través del Algoritmo BiSVM L-BFGS para el hiperparámetro óptimo $\beta = 0{,}4817$ el cual está dado por:

$$\hat{w} = [0{,}0093;\ -0{,}0188;\ 0{,}7241;\ 0{,}2291;\ -0{,}0003] \tag{6.3}$$

Podemos ver que el hiperplano de separación 6.3 tiene una estructura dispersa donde la primera y última componente se van acercando a cero. Esto se logra al utilizar la norma dispersa L_1 en el problema L_2L_1-SVM.

Otro aspecto importante para la visualización de los resultados es la trayectoria de regularización, la cual indica cómo cambian los coeficientes del hiperplano de separación w (eje y) a medida que varía el hiperparámetro β (eje x).

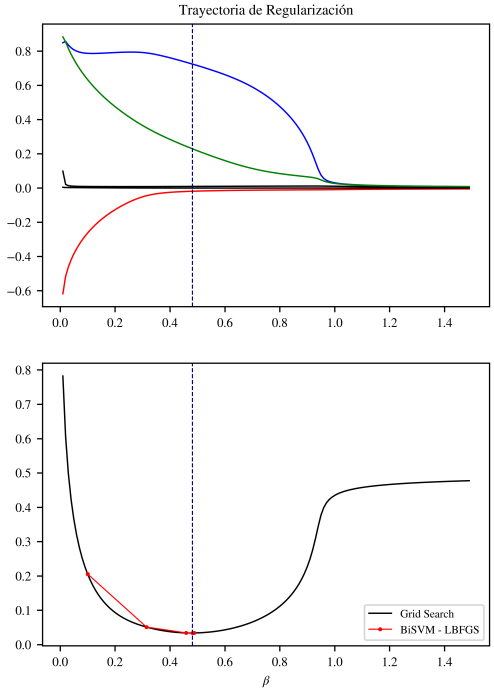


Figura 6.5: Trayectoria de regularización, Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Iris con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

La Figura 6.5 muestra que a medida que el algoritmo busca el hiperparámetro óptimo, las componentes del hiperplano de separación tienden a anularse.

En consecuencia, después de aplicar el Algoritmo BiSVM L-BFGS, el problema de clasificación de especies se puede realizar con tres variables de las cinco (incluido el término de sesgo) con un error de validación del $3{,}37\,\%$ (ver Tabla 6.3). Las variables seleccionadas son: *sepal width (cm)*, *petal length (cm)* y *petal width (cm)*.

Realizamos el mismo experimento con el conjunto Breast Cancer, el cual tiene 31 variables explicativas (incluido el término de sesgo al final) y 569 observaciones. La partición de los datos de entrenamiento y validación se mantiene en $80\,\%$ y $20\,\%$ respectivamente. Además, para el método *Grid Search* se usan los mismos 149 valores de búsqueda en el conjunto $(0; 1{,}5]$.

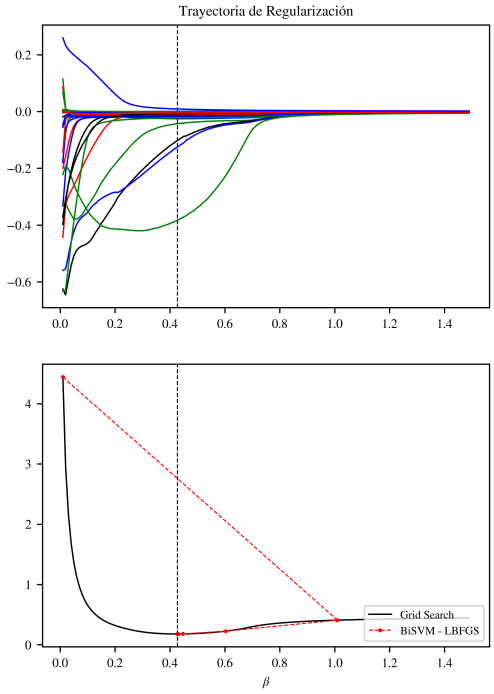


Figura 6.6: Trayectoria de regularización, Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0;\, 1,5]$ para el conjunto de datos Breast Cancer con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

La Figura 6.6 muestra un resumen de la trayectoria de regularización, el método *Grid Search* y el Algoritmo biSVM L-BFGS. Además, se observa que para resolver el problema de clasificación de tumores malignos y benignos, no es necesario utilizar las 31 variables del conjunto de datos Breast Cancer, sino que a medida que se encuentra el hiperparámetro óptimo laselección de variables se reduce a 12 variables explicativas.

Parámetros	J_{min} GS	β GS	J_{min} BiSVM L-BFGS	β BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}01$					
$\mu = 0{,}25$	0,1783	0,42	0,1783	0,4275	5

Tabla 6.4: Resultados de Algoritmo BiSVM LBFGS y Grid Search (GS) sobre $\beta \in (0; 1,5]$ para el conjunto de datos Breast Cancer con $\mu = 0{,}25$ y $\gamma = 0{,}01$.

La Tabla 6.4 muestra que aplicando el Algoritmo BiSVM L-BFGS, el hiperparámetro óptimo $\beta = 0{,}4275$ se encuentra realizando 5 iteraciones. Se logra alcanzar un error de validación de $0{,}1783$, igual al obtenido a través de *Grid Search*.

Por lo tanto, después de aplicar el Algoritmo BiSVM L-BFGS, el problema de clasificación de tumores malignos y benignos se puede realizar con 12 variables de las 31 (incluido el término de sesgo) con un error de validación del $17{,}83\,\%$ (ver Tabla 6.4). Las variables seleccionadas son: *mean radius*, *mean perimeter*, *mean area*, *mean smoothness*, *mean concavity*, *mean concave points*, *worst radius*, *worst perimeter*, *worst area*, *worst compactness*, *worst concavity*, *worst concave points*.

6.2. Hiperparámetro vectorial

Hasta el momento hemos podido seleccionar las variables a través de la aplicación del Algoritmo BiSVM L-BFGS, para el caso cuando el hiperparámetro β es escalar. Sin embargo, no sabemos la importancia de cada una de las variables seleccionadas.

Para lograr identificar la importancia de las variables seleccionadas, se resuelve la siguiente formulación:

$$(P_V) \begin{cases} \min\limits_{\hat{\beta} \in \mathbb{R}^{d+1}_+} & J(\hat{\beta}, w(\hat{\beta}); V, \eta) := \dfrac{1}{2m} \| V\hat{w}(\hat{\beta})^\top - \eta \|_2^2 \\ \text{s.a.} & \hat{w} = \arg\min\limits_{w} \hat{E}(w, \hat{\beta}; X, y), \end{cases} \tag{6.4}$$

donde la función objetivo de nivel inferior está dada por:

$$\hat{E}(w, \hat{\beta}; X, y) := \frac{1}{n} \sum_{i=1}^{n} \ell_\mu(1 - y t_i) + \frac{\alpha}{2} \sum_{j=1}^{d+1} w_j^2 + \sum_{j=1}^{d+1} \hat{\beta}_j H_\gamma(w_j), \tag{6.5}$$

con $t_i = \langle w, x_i \rangle$ para todo $i = 1, \ldots, n.$, $H_\gamma(\cdot)$ representa la función Pseudo-Huber (4.15) y $\ell_\mu(\cdot)$ es la función *hinge-loss* aproximada (4.18).

Una vez resuelto el problema (6.4) a través del Algoritmo 8, se procede a aplicar la metodología propuesta en la Sección (4.4.1).

El primer experimento que presentamos es con el conjunto de datos Iris bajo las mismas condiciones de particionamiento del conjunto de entrenamiento y validación. Al aplicar el Algoritmo BiSVM L-BFGS (vectorial) para el escenario cuando el hiperparámetro $\hat{\beta}$ es un vector se tiene los siguientes resultados:

Parámetros	J_{min} BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}01$		
$\mu = 0{,}25$	0,0227	11

Tabla 6.5: Resultados de Algoritmo BiSVM LBFGS (vectorial) para el conjunto de datos Iris con $\mu = 0{,}25$, $\gamma = 0{,}01$ y $\alpha = 0{,}01$.

Notemos que los resultados de la Tabla 6.5 muestra un error de validación igual a $0{,}0227$. Este resultado es menor que el error de validación obtenido en el caso escalar $0{,}0337$ (ver Tabla 6.3). Además, el algoritmo realiza 11 iteraciones para hallar el hiperparámetro vectorial óptimo.

Cabe mencionar que a medida que los hiperparámetros aumentan, el método *Grid Search* es impracticable ya que se debe evaluar el modelo para todas las combinaciones existentes.

El hiperparámetro vectorial óptimo y el hiperplano de separación asociado se presentan a continuación:

$$\hat{\beta} = [0{,}6467;\ 0{,}3346;\ 0{,}4649;\ 0{,}5367;\ 0{,}1945] \tag{6.6}$$

$$\hat{w} = [0{,}0065;\ -0{,}1717;\ 0{,}8545;\ 0{,}0199;\ 0{,}0000] \tag{6.7}$$

Con esta información del hiperparámetro óptimo y su hiperplano de separación asociado, podemos dar paso a calcular el *score* de importancia de cada una de las variables seleccionadas por el algoritmo. Notemos que el primero y el último coeficiente del hiperplano de separación $\hat{w}$ tienden a anularse al igual que el hiperplano de separación (6.3) del caso escalar. En este sentido, en el caso vectorial se seleccionan las mismas variables.

Para aplicar la metodología propuesta en la Sección (4.4.1), iniciamos fijando una tolerancia $\epsilon = 0{,}01$ para construir el hiperplano de separación auxiliar:

$$\bar{w} = [0;\ 1;\ 1;\ 1;\ 0] \tag{6.8}$$

Luego, calculamos el nuevo vector de hiperparámetros:

$$\bar{\beta} = [0{,}0000;\ 0{,}3346;\ 0{,}4649;\ 0{,}5367;\ 0{,}0000] \tag{6.9}$$

Finalmente, calculamos el *score* para cada una de las variables:

$$\text{Score} = [0{,}0000;\ 0{,}2504;\ 0{,}3480;\ 0{,}4016;\ 0{,}0000] \tag{6.10}$$

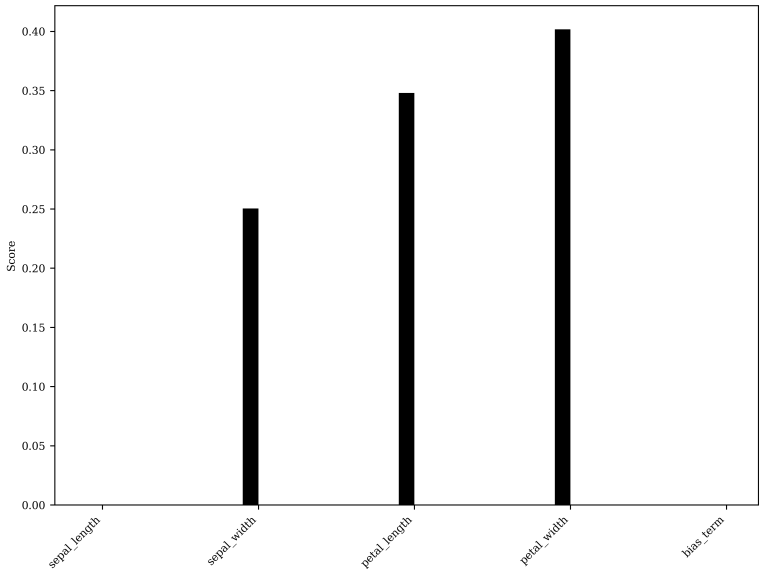


Figura 6.7: Importancia de variables seleccionadas a través del Algoritmo BiSVM LBFGS (vectorial) para el conjunto de datos Iris con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

Notemos que el $Score_j \in [0, 1]$ para cada $j = 1, \ldots, d + 1$. y la sumatoria de todos ellos es igual a uno.

La Figura 6.7 muestra que de las variables seleccionadas, la variable $petal\_width$ (ancho del pétalo) es la más importante con un score de $0{,}4016$ al momento de realizar una predicción en la clasificación de especies.

Además, podemos establecer un *ranking* de las variables de acuerdo a su medida de importancia. Para el caso del conjunto de datos Iris el *ranking* tiene el siguiente orden: $petal\_width$, $petal\_length$, $setal\_width$.

Por otra parte, podemos comparar nuestro algoritmo propuesto con otro método integrado de selección de variables como: *Tree-Based Feature Selection* (TB-FS)[1]. El método TB-FS se utiliza en el contexto de algoritmos de aprendizaje automático basados en árboles, como los árboles de decisión y los bosques aleatorios (Random Forest). Su objetivo es identificar las características más importantes o relevantes para un modelo predictivo.

[1]Los detalles de uso de la metodología se los puede encontrar en: Tree Based Feature Selection

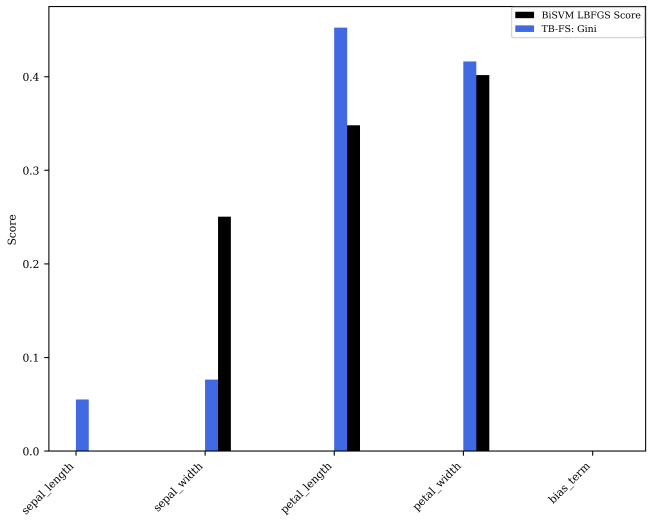


Figura 6.8: Comparación de la importancia de variables seleccionadas a través del Algoritmo BiSVM LBFGS (vectorial) y el método TB-FS para el conjunto de datos Iris con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

La variable más importante para el método TB-FS es *petal_length*. El score asignado por TB-FS a cada variable está dado por el siguiente vector:

$$Score_{TB-FS} = [0{,}112;\ 0{,}094;\ 0{,}395;\ 0{,}399] \tag{6.11}$$

La principal diferencia con nuestro algoritmo es que el método TB-FS asigna un score a todas las variables del conjunto de datos y no asigna ningún score al término de sesgo. Además, si se requiere hacer una selección de variables con el método TB-FS, el modelador debe establecer un umbral del índice de Gini para la selección, lo cual es subjetivo de cada individuo.

Un segundo experimento se realiza con el conjunto de datos Churn, el cual busca identificar el abandono de clientes de una empresa que ofrece sus servicios de banda ancha. Luego del procesamiento de datos, se obtiene 27405 registros únicos y 8 variables explicativas (incluida el término de sesgo al final). Para este conjunto de datos se usa una partición de $50\,\%$ para los datos de entrenamiento y $50\,\%$ para los datos de validación.

El conjunto de datos Churn contiene una variable categórica que identifica los nueve tipos de ancho de banda (*band width*) que ofrece la empresa. Para esta variable se aplica la técnica *One-Hot Encoding* para representar cada categoría como una variable binaria. En consecuencia, el nuevo conjunto de datos Churn contiene 16 variables numéricas (incluido el término de sesgo al final).

Al aplicar el Algoritmo BiSVM L-BFGS (vectorial) se tiene los siguientes resultados:

Parámetros	J_{min} BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}01$ $\mu = 0{,}25$	0,1559	7

Tabla 6.6: Resultados de Algoritmo BiSVM LBFGS (vectorial) para el conjunto de datos Churn con $\mu = 0{,}25$, $\gamma = 0{,}01$ y $\alpha = 0{,}01$.

La Tabla 6.6 muestra un error de validación igual a $0{,}1559$ luego de 7 iteraciones.

El score de las variables más importantes se lo presenta en la Figura 6.9. Además, se lo compara con el método TB-FS.

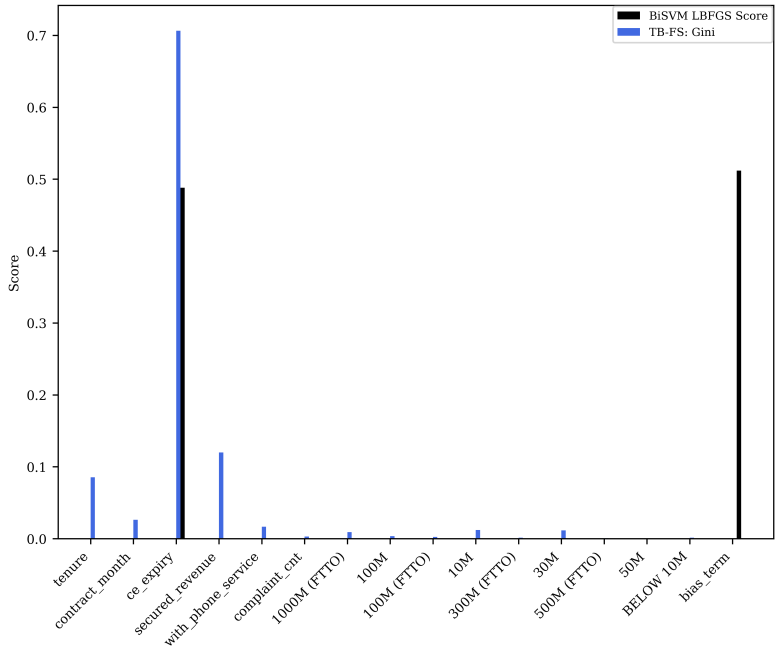


Figura 6.9: Comparación de la importancia de variables seleccionadas a través del Algoritmo BiSVM LBFGS (vectorial) y el método TB-FS para el conjunto de datos Churn con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

En la Figura 6.9 podemos ver claramente que la variable más importante identificada por el algoritmo es *ce_expirity*, la cual corresponde al vencimiento de los contratos y el término de sesgo. Además, el método TB-FS coincide con la variable más importante. Esto nos da una referencia al problema de abando de clientes de la empresa, pues de acuerdo a los datos, la variable más importante para pronosticar el abandono de un cliente tiene es la caducidad de los contratos.

Por otra parte, se realiza un último experimento con el conjunto de datos Breast Cancer.

Parámetros	J_{min} BiSVM L-BFGS	Iteraciones
$\gamma = 0{,}01$		
$\mu = 0{,}25$	0,1260	24

Tabla 6.7: Resultados de Algoritmo BiSVM LBFGS (vectorial) para el conjunto de datos Breast Cancer con $\mu = 0{,}25$, $\gamma = 0{,}01$ y $\alpha = 0{,}01$.

La Tabla 6.7 muestra que el Algoritmo BiSVM L-BFGS realiza 24 iteraciones para encontrar el hiperparámetro óptimo, donde se logra obtener un error de validación igual a $0{,}1260$; menor al que se obtuvo en el caso escalar $0{,}1783$ (ver Tabla 6.4).

Una comparación del método TB-FS y el algoritmo propuesto en base al score de importancia de variables se lo presenta en la siguiente figura:

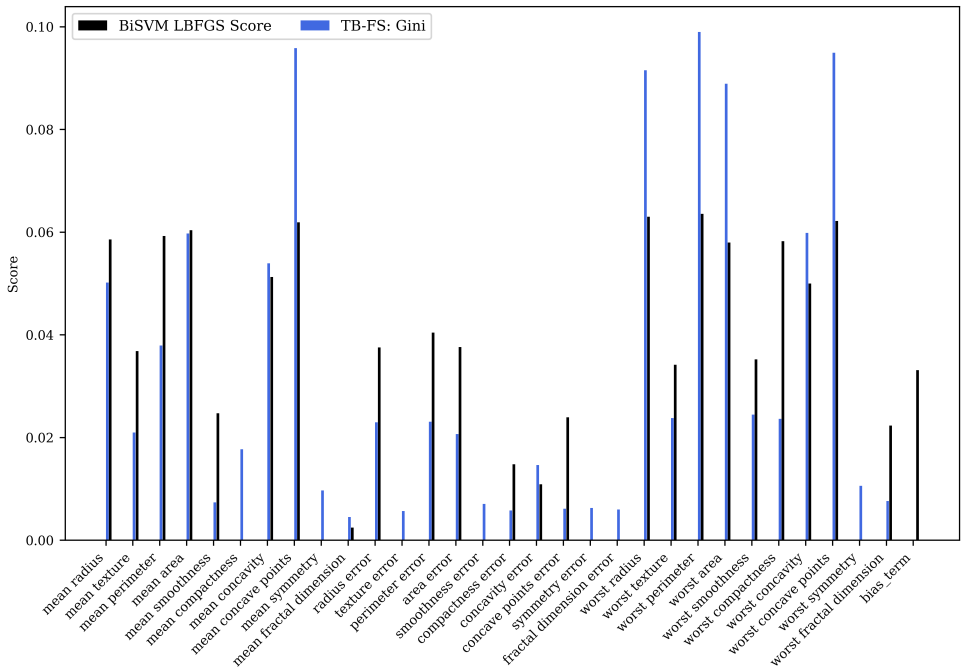
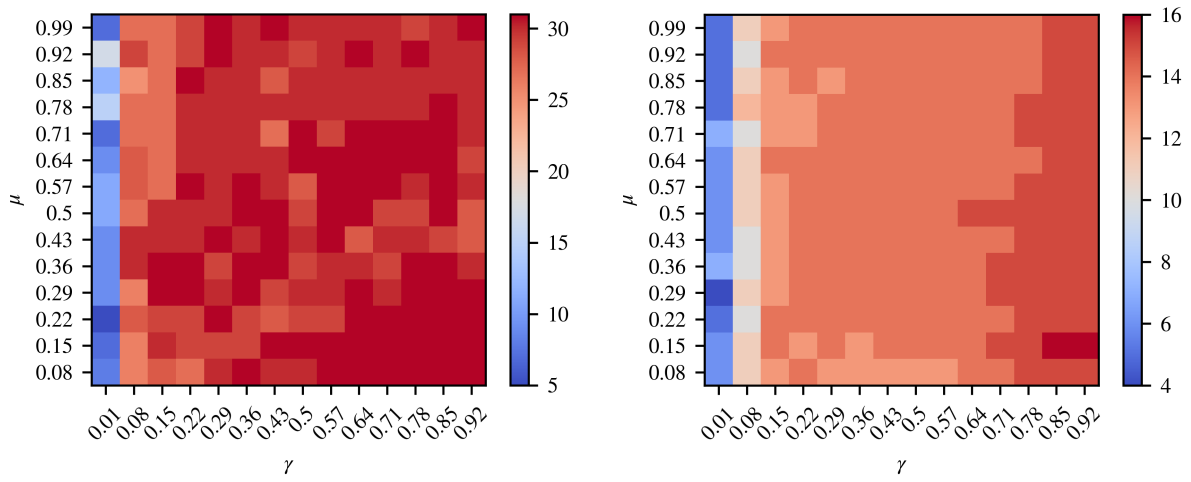


Figura 6.10: Comparación de la importancia de variables seleccionadas a través del Algoritmo BiSVM LBFGS (vectorial) y el método TB-FS para el conjunto de datos Breast Cancer con $\alpha = 0{,}01$ fijo, $\mu = 0{,}25$ y $\gamma = 0{,}01$.

Notemos que con esta versión del Algoritmo BiSVM L-BFGS (vectorial) se logró seleccionar más variables, en total 24 (incluido el término de sesgo) de las 31 variables originales, que en la versión escalar donde se seleccionaron 12 variables.

Finalmente, llevamos a cabo un experimento en el cual variamos los parámetros $\mu \in (0, 1)$ y $\gamma > 0$ de las funciones *Hinge Loss* aproximada (4.18) y Pseudo-Huber (4.15) respectivamente, con el fin de observar la convergencia numérica a medida que estos se aproximan cero.



(a) Conjunto Breast Cancer.

(b) Conjunto Churn.

Figura 6.11: Número de componentes no nulas en el hiperplano óptimo $\hat{w}$ luego de la evaluación del Algoritmo BiSVM LBFGS (vectorial) con $\alpha = 0{,}01$ fijo, $\mu \in (0,1)$ y $\gamma \in (0,1)$.

En la Figura 6.11, se presenta la evaluación del Algoritmo BiSVM LBFGS (vectorial) para diversas combinaciones de los parámetros $\mu \in (0,1)$ y $\gamma \in (0,1)$ de las funciones aproximadas. Se llevaron a cabo evaluaciones para un total de 225 combinaciones, considerando una componente como no nula si $|w_j| > 0{,}001$ para todo $j = 1, \ldots, d+1$. Esta ilustración asigna una escala de colores al número de componentes no nulas en el hiperplano solución $w(\hat{\beta})$. Cuantas más componentes no nulas posee el hiperplano solución, más intenso será el color rojo; caso contrario tomará un tono azul.

Se observa que a medida que el parámetro $\gamma > 0$ de la función Pseudo-Huber (aproximación de la norma L_1) se acerca a su límite, el número de componentes no nulas en el hiperplano solución disminuye. Esto sugiere que al reducir el parámetro $\gamma > 0$, se induce dispersión en el hiperplano solución, conduciendo así a una convergencia hacia la solución del problema original. Este comportamiento es coherente, dado que la norma L_1 promueve la dispersión en la solución original.

Además, se observa que manteniendo un valor bajo para $\gamma > 0$, el parámetro $\mu \in (0,1)$ de la función *Hinge Loss* aproximada puede variar dentro de su dominio, manteniendo así la dispersión de la solución.

6.3. Hiperparámetro vectorial - grupos

En esta sección se realiza un experimento para la siguiente formulación:

$$(P_G)\begin{cases}\min\limits_{\bar{\beta}\in\mathbb{R}^{g+1}_{+}} & J(\bar{\beta},w(\bar{\beta});V,\eta):=\dfrac{1}{2m}\|V\hat{w}(\bar{\beta})^{\top}-\eta\|_2^2\\ \text{s.a.} & \hat{w}=\arg\min\limits_{w}\bar{E}(w,\bar{\beta};X,y),\end{cases}\tag{6.12}$$

donde la función objetivo del nivel inferior está dada por:

$$\bar{E}(w,\bar{\beta};X,y):=\frac{1}{n}\sum_{i=1}^{n}\ell_\mu(1-yt_i)+\frac{\alpha}{2}\sum_{j=1}^{d+1}w_j^2+\sum_{k=1}^{g+1}\bar{\beta}_k\bar{H}_\gamma(w_k),\tag{6.13}$$

con $t_i=\langle w,x_i\rangle$ para todo $i=1,\ldots,n.$, $\tilde{H}_\gamma(\cdot)$ es la aproximación de la norma Group Lasso y $\ell_\mu(\cdot)$ es la función *hinge-loss* aproximada (4.18).

Para evaluar el Algoritmo BiSVM L-BFGS en esta formulación, usamos el conjunto de datos Breast Cancer. Este conjunto en realidad tiene 10 variables, pero medidas en diferentes formas: media, error estándar y el peor valor reportado. En total tiene 30 (tres grupos de 10 variables) más el término de sesgo al final que se lo toma como un grupo de una sola variable.

Parámetros	J_{min} BiSVM L-BFGS	Iteraciones
$\gamma=0{,}01$		
$\mu=0{,}25$	0,1734	4

Tabla 6.8: Resultados de Algoritmo BiSVM LBFGS (grupo) para el conjunto de datos Breas Cancer con $\mu=0{,}25$, $\gamma=0{,}01$ y $\alpha=0{,}01$.

La Tabla 6.8 muestra que el error de validación alcanzado es $0{,}1734$, que es ligeramente más bajo que el caso escalar. Además, el hiperparámetro grupal óptimo se lo encuentra en 4 iteraciones.

El hiperparámetro vectorial agrupado óptimo se presentan a continuación:

$$\tilde{\beta}=[0{,}9454;\ 0{,}3030;\ 0{,}0000;\ 0{,}1333]\tag{6.14}$$

Notemos que ahora se tiene un hiperparámetro para cada grupo de variables, incluido el término de sesgo como grupo unitario.

Para determinar la importancia de cada grupo de variables, aplicamos la metodología presentada en la Sección 4.5.1.

Tomamos una tolerancia $\epsilon=0{,}005$ para construir el hiperplano auxiliar. Las normas para

cada grupo de coeficientes del hiperplano obtenido son:

$$[0{,}0084;\ 0{,}0032;\ 0{,}0095;\ 0{,}0001] \tag{6.15}$$

Luego, el hiperplano auxiliar es:

$$\bar{w} = [1;\ 0;\ 1;\ 0] \tag{6.16}$$

Luego, calculamos el nuevo vector de hiperparámetros:

$$\bar{\beta} = [0{,}0084;\ 0{,}0000;\ 0{,}0095;\ 0{,}0000] \tag{6.17}$$

Finalmente, calculamos el *score* para cada una de las variables:

$$\text{Score} = [0{,}4693;\ 0{,}0000;\ 0{,}5307;\ 0{,}0000] \tag{6.18}$$

Por lo tanto, el conjunto más significativo de variables se refiere a aquellas que representan los valores más desfavorables (el peor valor registrado en la toma de datos) en cada caso. Además, observamos que los valores promedio juegan un papel predictivo crucial en la detección del cáncer en los tumores. En contraste, los valores correspondientes al error estándar de las variables y el término de sesgo no parecen tener relevancia para la detección de cáncer en este conjunto de datos.

Capítulo 7

Conclusiones

En este trabajo de investigación, presentamos la aplicación del enfoque binivel para la selección de variables mediante una herramienta de aprendizaje automático conocida como Máquinas de Soporte Vectorial con regularización mixta L_2 y L_1, la cual denominamos L_2L_1-SVM. Para alcanzar este objetivo, se requiere que tanto la función objetivo del problema de nivel inferior (L_2L_1-SVM) como la función objetivo del problema de nivel superior (error de validación) sean diferenciables. Esto nos lleva a emplear aproximaciones locales suavizadas de la norma L_1 y la función *hinge-loss* para la función objetivo del problema de nivel inferior. Asimismo, usamos el error cuadrático medio (MSE) como función objetivo del problema de nivel superior.

En el Capítulo 4 presentamos tres variantes del problema de optimización binivel L_2L_1-SVM para el hiperparámetro de la regularización dispersa L_1, abordando los casos en los que este hiperparámetro es: escalar, vectorial y vectorial agrupado.

La solución local del problema de optimización binivel L_2L_1-SVM, para sus tres variantes, se caracterizó a través de las condiciones de Karush-Kuhn-Tucker. Con base en esta caracterización, para encontrar su solución numérica utilizamos el método quasi-Newton L-BFGS, tanto para el problema de nivel inferior como para el superior. En este sentido, diseñamos un algoritmo eficiente al cual denominamos BiSVM L-BFGS.

Con el enfoque presentado en este trabajo, no solo se identifica el mejor subconjunto de datos para predecir una variable binaria, sino que también se encuentra el hiperparámetro de la regularización dispersa L_1 óptimo y el hiperplano w asociado en todas sus variantes.

Como se detalla en la sección experimental del Capítulo 6, el escenario donde el hiperparámetro de la regularización dispersa L_1 es escalar, presenta una clara ventaja sobre los métodos de búsqueda exhaustiva o Grid Search, ya que logra alcanzar el valor óptimo en un número reducido de iteraciones. Esta formulación, en lugar de calcular el error de

validación sobre un conjunto discretizado de posibles valores del hiperparámetro, realiza la búsqueda del hiperparámetro óptimo mediante una dirección de descenso empleando del Algoritmo BiSVM L-BFGS. Adicionalmente, el hiperparámetro óptimo encontrado es continuo, lo cual evita valores subóptimos asociados a la discretización en la búsqueda exhaustiva.

El beneficio se mejora cuando el hiperparámetro de la regularización dispersa L_1 adopta una estructura vectorial, incorporando un hiperparámetro para cada variable del conjunto de datos, incluyendo el término de sesgo. En los conjuntos de datos analizados, hemos logrado reducir el error de validación en comparación con la formulación escalar del problema de optimización binivel. Esta formulación nos permite calcular un indicador de importancia (*score*) que varía entre 0 y 1 para cada una de las variables seleccionadas por el Algoritmo BiSVM L-BFGS.

Al contrastar nuestra metodología de selección de variables, para el caso vectorial, con otro método integrado de selección como *Tree-Based Feature Selection* (TB-FS), notamos que en la mayoría de los experimentos realizados, coincidimos con la medida de importancia reportada por este método. Sin embargo, se destaca una diferencia significativa: nuestra metodología asigna una medida de importancia únicamente a las variables seleccionadas, a diferencia de TB-FS, que asigna una medida de importancia a todas las variables del conjunto de datos.

Además, se realizó un experimento para estudiar la convergencia numérica a medida que los parámetros de las funciones aproximadas del problema del nivel inferior se aproximan a sus límites. En este contexto, se observó que conforme el parámetro de la función Pseudo-Huber (aproximación de la norma L_1) se acerca a su límite, el hiperplano solución exhibe menos componentes no nulas, independientemente del valor que tome el parámetro de la función *Hinge Loss*. En otras palabras, las componentes del hiperplano óptimo muestran una mayor dispersión (sparsity), lo cual es consistente con las expectativas del problema original.

Por otra parte, al considerar el hiperparámetro de la regularización dispersa L_1 en forma vectorial agrupada, observamos un error de validación ligeramente inferior en comparación con la formulación escalar. Esta formulación resulta especialmente útil cuando las variables de un conjunto de datos están agrupadas según una categoría o naturaleza específica.

Tomando en cuenta los resultados experimentales, podemos ver que al aplicar nuestra metodología de selección de variables se puede obtener un buen resultado de clasificación binaria (bajo error de validación) y al mismo tiempo beneficiarnos de la complejidad reducida del modelo, obteniendo una mejor interpretabilidad de las variables relevantes para el modelo.

Finalmente, es importante señalar que si aplicamos nuestra metodología y observamos que todas las variables resultan ser importantes después de la optimización binivel, podría

sugerir que el conjunto de datos no requiere tantos grados de libertad. En este caso, podríamos considerar la aplicación de una formulación más simple, como la formulación binivel con un hiperparámetro escalar.

A continuación, se presentan las líneas de trabajo futuras:

- Las formulaciones presentadas en este trabajo pueden ampliarse para el caso donde la variable objetivo comprenda más de dos clases.
- La investigación puede expandirse el caso cuando los datos no son linealmente separables es el espacio de características original. Es decir, cuando los datos no pueden ser separados por un hiperplano. Para esto se puede usar funciones de kernel en el problema de nivel inferior, con el fin de mapear los datos a un espacio de características de mayor dimensión donde la separación lineal puede ser posible.
- Un desafío significativo consiste en abordar el problema de optimización binivel L_2L_1-SVM original, el cual presenta una función objetivo no diferenciable en el problema de nivel inferior. Este aspecto abre la puerta a una investigación más profunda para desarrollar métodos efectivos en la optimización de funciones no diferenciables en contextos de aprendizaje automático desde un enfoque binivel.

Referencias

[Bazarra et al., 2021] Bazarra, J., Droguett, E., and Martins, M. (2021). Towards interpretable deep learning: A feature selection framework for prognostics and health management using deep neural networks. *Sensors (Basel)*.

[Beck, 2017] Beck, A. (2017). *First-Order Methods in Optimization*. MOS-SIAM.

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009). Fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Science*.

[Blanz et al., 1996] Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C., Vapnik, V., and Vetter, T. (1996). Comparison of view-based object recognition algorithms using realistic 3d models. In *Artificial Neural Networks — ICANN 96*, pages 251–256. Springer Berlin Heidelberg.

[Bolón-Canedo et al., 2014] Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., Benítez, J., and Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. *Information Sciences*.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and trends in Machine Learning*.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Bracken and MacGill, 1973] Bracken, J. and MacGill, J. (1973). Mathematical programs with optimization problems in the constraints. In *Operations Research*, pages 37–44.

[Byrd et al., 1995] Byrd, R., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*.

[Candler and Norton, 1977] Candler, W. and Norton, R. (1977). Multilevel programming and development policy. In *Technical Report*, volume 258. World Bank Staff.

[Chen et al., 2021] Chen, T., Sun, Y., and Yin., W. (2021). A single-timescale stochastic bilevel optimization method. *Online*.

[Chen et al., 2014] Chen, Y., Ranftl, R., and Pock, T. (2014). Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. *IEEE Transactions on Image Processing*.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support vector networks. In *Machine Learning*, pages 273–297. Kluwer Academic Publishers, Boston.

[Crockett and Fessler, 2021] Crockett, C. and Fessler, J. (2021). Bilevel methods for image reconstruction.

[Cui et al., 2021] Cui, L., Shen, J., and Yao, S. (2021). The sparse learning of the support vector machine. *Journal of Physics: Conference Series*.

[De los Reyes, 2023] De los Reyes, J. (2023). Bilevel imaging learning problems as mathematical programs with complementarity constraints: Reformulation and theory. *SIAM Journal on Imaging Science*.

[De los Reyes and Schönlieb, 2013] De los Reyes, J. and Schönlieb, C.-B. (2013). Image denoising: Learning the noise model via nonsmooth pde-constrained optimization. *Inverse Problems and Imaging*.

[De los Reyes et al., 2017] De los Reyes, J., Schönlieb, C.-B., and Valkonen, T. (2017). Bilevel parameter learning for higher-order total variation regularisation models. *Journal of Mathematical Imaging and Vision*.

[Dempe, 2002] Dempe, S. (2002). Foundation of bilvel programming. In *Nonconvex Optimization and Its Applications*, volume 61. Kluwer Academic publishers.

[Dempe and Zemkoho, 2020] Dempe, S. and Zemkoho, A. (2020). Bilevel optimization: Advances and next challenges. In *Springer Optimization and Its Applications*, volume 161. Springer.

[Diaz and Alvarez, 2006] Diaz, R. and Alvarez, S. (2006). Gene selection and classification of microarray data using random forest. *Bioinformatics*.

[Dontchev and Rockafellar, 2014] Dontchev, A. and Rockafellar, R. (2014). *Implicit Functions and Solutions Mappings: A View from Variational Analysis*.

[Duda et al., 2000] Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*. Wiley.

[Fountoulakis and Gondzio, 2015] Fountoulakis, K. and Gondzio, J. (2015). A second-order method for strongly convex l_1-regularization problems. *Springer-MOS*.

[Gao et al., 2022] Gao, L., Ye, J., Yin, H., Zeng, S., and Zhang, J. (2022). Value function based difference-of-convex algorithm for bilevel hyperparameter selection problems. *Proceedings of Machine Learning Research*.

[Ghadimi and Wang, 2018] Ghadimi, S. and Wang, M. (2018). Approximation methods for bilevel programming.

[Gould et al., 2016] Gould, S., Fernando, B., Cherian, A., P. Anderson, R. S. C., and Guo, E. (2016). On differentiating parameterized argmin and argmax problems with application to bi-level optimization.

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3.

[Guyon et al., 2002] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning - Vol. 46*.

[Hajewski et al., 2018] Hajewski, J., Oliveira, S., and Stewart, D. (2018). Smoothed hinge loss and l1 support vector machines. *IEEE - International Conference on Data Mining Workshop*.

[Hastie et al., 2016] Hastie, T., Tibshirani, R., and Wainwright, M. (2016). *Statistical Learning with Sparsity*. CRC Press.

[Holler et al., 2018] Holler, G., Kunisch, K., and Barnard, R. C. (2018). A bilevel approach for parameter learning in inverse problems. *Inverse Problems, vol. 34*.

[Hong et al., 2020] Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. (2020). A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *Online*.

[Inza et al., 2004] Inza, I., Larrañaga, P., Blanco, R., and Cerrolaza, A. (2004). Filter versus wrapper gene selection approaches in dna microarray domains. *National Center for Biotechnology Information*.

[Inza et al., 2000] Inza, I., Larrañaga, P., Etxeberria, R., and Sierra, B. (2000). Feature subset selection by bayesian network-based optimization. *Artificial Intelligence*.

[Jafari and Azuaje, 2006] Jafari, P. and Azuaje, F. (2006). An assessment of recently published gene expression data analyses: Reporting experimental design and statistical factors. *National Center for Biotechnology Information*.

[Klatzer, 2014] Klatzer, T. (2014). Bi-level optimization for support vector machines. *Graz University of Technology*.

[Kunapuli et al., 2008] Kunapuli, G., Bennett, K. P., Hu, J., and Pang, J.-S. (2008). Bilevel model selection for support vector machines. *Centre de Recherches Mathématiques*.

[Kunisch and Pock, 2013] Kunisch, K. and Pock, T. (2013). A bilevel optimization approach for parameter learning in variational models. *SIAM J. IMAGING SCIENCES*.

[Li et al., 2022] Li, Q., Li, Z., and Zemkoho, A. (2022). Bilevel hyperparameter optimization for support vector classification: theoretical analysis and a solution method. *Mathematical Methods of Operations Research*.

[Liu and Setiono, 1995] Liu, H. and Setiono, R. (1995). Chi2: feature selection and discretization of numeric attributes. *International Conference on Tools for Artificial Intelligence (ICTAI)*.

[Ma and Huang, 2005] Ma, S. and Huang, J. (2005). Regularized roc method for disease classification and biomarker selection with microarray data. *Bioinformatics*.

[Maldonado and Weber, 2009] Maldonado, S. and Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences - Vol. 179*.

[Müller et al., 1997] Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V. (1997). Predicting time series with support vector machines. In *Artificial Neural Networks — ICANN 97*, pages 999–1004. Springer Berlin Heidelberg.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer.

[Osuna et al., 1997] Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: An application to face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 130–136. Springer Berlin Heidelberg.

[Pedregosa, 2016] Pedregosa, F. (2016). Hyperparameter optimization with approximate gradient. *Proceedings International Conference on Machine Learning, M. F. Balcan and K. Q. Weinberger*.

[Pudjihartono et al., 2022] Pudjihartono, N., Fadason, T., Kempa-Liehr, A., and O'Sullivan, J. (2022). A review of feature selection methods for machine learning-based disease risk prediction. *Frontiers in Bioinformatics*.

[Riedmiller and Braun, 1993] Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. *IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS*.

[Sakr et al., 2019a] Sakr, M., Tawfeeq, M., and El-Sisi, A. (2019a). An efficiency optimization for network intrusion detection system. *I. J. Computer Network and Information Security*.

[Sakr et al., 2019b] Sakr, M., Tawfeeq, M., and El-Sisi, A. (2019b). Filter versus wrapper feature selection for network intrusion detection system. *IEEE Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*.

[Samuel and Tappen, 2009] Samuel, G. and Tappen, M. (2009). Learning optimized map estimates in continuously-valued mrf models. *IEEE Conference on Computer Vision and Pattern Recognition*.

[Sarker, 2021] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learnig from Theory to Algorithms*. Cambridge University Press.

[Shwartz and David, 2014] Shwartz, S. S. and David, S. B. (2014). *Understanding Machine Learning from Theory to Algorithms*. Cambridge University Press, 1st edition.

[Tang et al., 2018] Tang, F., Adam, L., and Si, B. (2018). Group feature selection with multiclass support vector machine. *Neurocomputing*.

[Thomas et al., 2023] Thomas, J., Olson, J., Tapscott, S., and Zhao, L. (2023). An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. *Genome Research*.

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, 1st edition.

[Vicente and Calamai, 1994] Vicente, L. and Calamai, P. (1994). Bilevel and multilevel programming: A bibliography review. In *J Glob Optim*, volume 5.

[Von-Stackelberg, 2011] Von-Stackelberg, H. (2011). Market structure and equilibrium. Springer.

[Wang et al., 2006] Wang, L., Zhu, J., and Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica*.

[Xiong et al., 2001] Xiong, M., Fang, X., and Zhao, J. (2001). Biomarker identification by feature wrappers. *National Center for Biotechnology Information*.

[Zhang et al., 2015] Zhang, G., Lu, J., and Gao, Y. (2015). Multi-level decision making. models, methods and applications. In *Intelligent Systems Reference Library*, volume 82, pages 47–62. Springer.

[Zhang et al., 2003] Zhang, J., Jin, R., Yang, Y., and Huaptmann, A. (2003). Modifed logistic regression: An approximation to svm and its applications in large-scale text categorization. *International Conference on Machine Learning - ICML*.

[Zhang et al., 2017] Zhang, W., Hong, B., Liu, W., Ye, J., Cai, D., He, X., and Wang, J. (2017). Scaling up sparse support vector machines by simultaneous feature and sample reduction. *International Conference on Machine Learning*.

[Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Royal Statistical Society*.

Anexo A: Conjuntos de Datos

Conjunto de datos: IRIS

Variable	Descripción
`sepal length (cm)`	Longitud del sépalo en centímetros
`sepal width (cm)`	Ancho del sépalo en centímetros
`petal length (cm)`	Longitud del pétalo en centímetros
`petal width (cm)`	Ancho del pétalo en centímetros
`species`	Especie de la flor (setosa = -1, versicolor=1, virginica=2)

Tabla 7.1: Descripción de las variables del conjunto de datos Iris.

Dado que el estudio se centra para problemas de clasificación binaria, se toma en consideración sólo dos especies: setosa y versicolor. En total, el conjunto Iris utilizado tiene 100 observaciones y 4 variables explicativas.

Conjunto de datos: BREAST CANCER

Variable	Descripción
`mean radius`	Media de los radios de las células
`mean texture`	Media de la textura de las células
`mean perimeter`	Media del perímetro de las células
`mean area`	Media del área de las células
`mean smoothness`	Media de la suavidad de las células
`mean compactness`	Media de la compacidad de las células
`mean concavity`	Media de la concavidad de las células
`mean concave points`	Media de los puntos cóncavos de las células
`mean symmetry`	Media de la simetría de las células
`mean fractal dimension`	Media de la dimensión fractal de las células
`target`	Clase (-1 = maligno, 1 = benigno)

Tabla 7.2: Descripción de las variables del conjunto de datos Breast Cancer.

Este conjunto de datos contiene las variables descritas anteriormente en tres grupos: media, desviación estándar y el pero valor obtenido. En total, este conjunto utilizado tiene 569 observaciones y 30 variables explicativas.

Conjunto de datos: CHURN

Variable	Descripción
image	Mes y año de facturación
newacct_no	Identificación única del cliente
line_stat	Variable opcional (úsala en caso de encontrar una correlación)
bill_cycl	Variable opcional (úsala en caso de encontrar una correlación)
serv_type	Variable opcional (úsala en caso de encontrar una correlación)
serv_code	Variable opcional (úsala en caso de encontrar una correlación)
tenure	Meses que el cliente ha estado en el sistema
effc_strt_date	Fecha de inicio del contrato
effc_end_date	Fecha de finalización del contrato
contract_month	Tipo de contrato
ce_expiry	Fecha de vencimiento del contrato.
secured_revenue	Ingresos mensuales
bandwidth	Ancho de banda de Internet
term_reas_code	Código de motivo de terminación del contrato
term_reas_desc	Descripción del código de motivo de terminación del contrato
complaint_cnt	Número de llamadas de quejas realizadas por el cliente cada mes
with_phone_service	Si el cliente de banda ancha ha tomado un servicio telefónico por separado o no
churn	Si se ha producido la cancelación o no de un contrato (Sí = 1, No = -1)
current_mth_churn	El mes en que el cliente se canceló

Tabla 7.3: Descripción de las variables del conjunto de datos Churn.

Luego del preprocesamiento de datos, este conjunto utilizado tiene 27405 observaciones y 8 variables explicativas.