Universidad de Valladolid

FACULTAD DE CIENCIAS

Trabajo Fin de Grado

Grado en Estadística

Técnicas para la corrección del desplazamiento de covarianzas en clasificación supervisada

Autor:

Javier Ramos Jimeno

Tutor:

José Ignacio Segovia Martín

Año:

2024–2025

Resumen

En los métodos de aprendizaje supervisado se suele asumir que la distribución de los datos de entrenamiento y test son iguales lo que permite simplificar el desarrollo de los modelos. Sin embargo, en muchas ocasiones esto no es cierto, lo que puede llevar a malos resultados. Por este motivo, se han desarrollado metodologías que permiten adaptarse a estos cambios en las distribuciones.

En concreto, este trabajo se centra en el *covariate shift* donde las distribuciones marginales de las instancias son distintas pero las condicionales respecto de las etiquetas permanecen constantes. Para adaptarse al *covariate shift* se utiliza un peso, al que se denomina importancia, que se aplica a las muestras de entrenamiento para intentar corregir la diferencia entre las distribuciones. De forma que, el valor del peso dependerá, únicamente, de cómo de probable es que una muestra de entrenamiento pueda aparecer dentro del conjunto de test. El problema es que para calcularlo de manera exacta es necesario conocer las distribuciones de los datos, lo que en la práctica, no es posible.

Por este motivo, existen diferentes métodos que permiten estimar la importancia. En este trabajo se profundiza en algunos de los métodos del estado del arte para hacer esta estimación como son KDE, obtención de los pesos usando regresión logística, KMM, KLIEP, LSIF o uLSIF. Además, también se han implementado todos en Python y con ellos se han realizado multitud de experimentos usando datos sintéticos.

Abstract

In supervised learning methods, it is usually assumed that the distribution of the training and test data is the same, which simplifies the development of models. However, in many cases this is not true, which can lead to bad results. For this reason, methodologies have been developed to adapt to these changes in the distributions.

Specifically, this work focuses on *covariate shift*, where the marginal distributions of the instances are different but the conditional distributions regarding the labels remain constant. To adapt to *covariate shift*, a weight, known as importance, is applied to the training samples to try to correct the difference between the distributions. The value of the weight will depend only on how likely it is that a training sample could appear in the test set. The problem is that, to calculate it exactly, it is necessary to know the data distributions, which in practice is not possible.

For this reason, there are different methods to estimate importance. In this work, some state-of-the-art methods to make this estimation, such as KDE, obtaining weights using logistic regression, KMM, KLIEP, LSIF, and uLSIF are discussed in detail. In addition, all of them have been implemented in Python, and several experiments have been carried out using synthetic data.

Agradecimientos

Quiero agradecer a todas las personas que han hecho posible la realización de este trabajo. En primer lugar, a mi tutor, José Ignacio, por su ayuda que ha sido fundamental para poder llevar a cabo este trabajo.

También quiero agradecer a mis compañeros de clase que me han acompañado durante la carrera y que me han ayudado a lo largo de la misma.

Por último, a mi familia, en especial a mi tío Antonio, que me han ayudado durante todos estos años y han sido una fuente de motivación y apoyo.

Índice general

1. Introducción **1**
1.1. Aprendizaje supervisado . . . 2
1.1.1. Modelos para el aprendizaje de funciones . . . 3
1.1.2. Error de generalización . . . 4
1.1.3. Funciones de pérdida . . . 5
1.1.3.1. Error cuadrático medio . . . 5
1.1.3.2. Error absoluto medio . . . 5
1.1.3.3. Función de pérdida 0/1 . . . 5
1.1.3.4. Entropía cruzada . . . 6

2. El problema del covariate shift **7**
2.1. Adaptación al covariate shift . . . 7
2.1.1. Minimización del riesgo empírico . . . 8
2.1.2. Minimización del riesgo empírico ponderado por importancia . . . 9

3. Métodos de estimación de importancia **13**
3.1. Kernel Density Estimation (KDE) . . . 14
3.2. Obtención de pesos por regresión logística . . . 15
3.3. Kernel Mean Matching (KMM) . . . 16
3.4. Procedimiento de Kullback-Leibler (KLIEP) . . . 18
3.5. Ajuste de importancia por mínimos cuadrados (LSIF) . . . 20
3.6. Comparación de los métodos de estimación . . . 22

4. Ejemplos de los métodos de estimación de importancia **23**
4.1. Generación de los datos . . . 23
4.2. Necesidad de la importancia . . . 24
4.3. Métodos de estimación de la importancia . . . 26
4.4. Comparación de los métodos de estimación de la importancia . . . 28
4.5. Efecto de aplanar los pesos . . . 30
4.6. Modificación de la distribución de una de las covariables . . . 33

5. Conclusiones **37**
5.1. Líneas de trabajo futuras . . . 38

Bibliografía **40**

A. Códigos **41**
A.1. Generación de datos . . . 41
A.2. Clasificador sin importancia . . . 42
A.3. Importancia real . . . 42

A.4. Métodos de estimación de la importancia 43
A.4.1. Kernel Density Estimation (KDE) 43
A.4.2. Kernel Mean Matching (KMM) 44
A.4.3. Obtención de pesos por regresión logística 45
A.4.4. Procedimiento de Kullback-Leibler (KLIEP) 45
A.4.5. Ajuste de importancia por mínimos cuadrados (LSIF) 46
A.4.6. Ajuste de importancia por mínimos cuadrados sin restricciones (uLSIF) . . . 47

Lista de Figuras

1.1. Ejemplo de conjuntos con distribuciones distintas (idea extraída de [4]). 2

4.1. Datos generados para evaluar los métodos de estimación de la importancia. 23

4.2. Resultados del clasificador sin importancia y del que usa la importancia real. 25

4.3. Comparación de los pesos utilizados con ambos clasificadores. 25

4.4. Importancia real aplicada en las muestras de entrenamiento. 26

4.5. Rectas de clasificación obtenidas con los métodos de estimación de la importancia. 27

4.6. Representación conjunta de las rectas de clasificación obtenidas por los diferentes métodos de estimación de la importancia. 28

4.7. Errores de clasificación de los métodos en los conjuntos de entrenamiento y test. . 29

4.8. Errores de clasificación sobre el conjunto de test de los diferentes métodos de estimación de la importancia. 30

4.9. Efecto de aplanar el peso de la importancia en los métodos de estimación. 31

4.10. Representación conjunta de las rectas de clasificación obtenidas por los diferentes métodos de estimación de la importancia. 32

4.11. Distintas distribuciones utilizadas para comprobar el funcionamiento de los métodos. 34

4.12. Errores de clasificación al variar la distribución de $X_{tr}^{(1)}$. 35

4.13. Errores de clasificación al variar la distribución de $X_{tr}^{(1)}$. 35

Lista de Tablas

3.1. Ventajas e inconvenientes de los métodos de estimación de la importancia 22

4.1. Errores de clasificación sobre el conjunto de test para distintos valores de γ. 32

1. Introducción

El aprendizaje automático, también conocido como *machine learning*, es una rama de la inteligencia artificial que se centra en desarrollar técnicas que permitan a los ordenadores aprender a partir de unos datos [1].

Dentro de este ámbito, existen diversos métodos de aprendizaje con el objetivo de extraer información acerca de los datos pero con diferentes enfoques. Los paradigmas de aprendizaje se pueden clasificar en [2][1]:

- **Aprendizaje supervisado**: Tiene el objetivo de encontrar una función que permita asociar los valores de entrada con sus salidas correspondientes. Para ello, es necesario disponer de un conjunto de datos de entrenamiento etiquetado, es decir, en donde se tienen las entradas con sus respectivas salidas. En función del tipo de la salida se tienen problemas de clasificación (cuando la salida es discreta) o de regresión (si la salida es continua).
- **Aprendizaje no supervisado**: Tiene el objetivo de extraer información acerca de los datos para encontrar posibles patrones ocultos. Los métodos de aprendizaje no supervisado utilizan un conjunto de datos no etiquetados.
- **Aprendizaje semisupervisado**: Tiene el objetivo de intentar mejorar los resultados combinando técnicas de aprendizaje supervisado y no supervisado. Como se emplean técnicas de ambos paradigmas, es necesario disponer de un conjunto de datos etiquetados y de un conjunto de datos no etiquetados.
- **Aprendizaje por refuerzo**: Tiene el objetivo de aprender las acciones que debe realizar un modelo, suponiendo que puede interactuar con el medio en el que se encuentra, ante distintas situaciones. Por tanto, se pretende encontrar la función que permita tomar estas decisiones para maximizar una recompensa. De esta forma, en el aprendizaje por refuerzo no es necesario disponer de un conjunto de datos etiquetados ya que el modelo aprende a partir de la experiencia obtenida al interactuar con su entorno [3].

Este trabajo se centra solamente en el primero de ellos, el aprendizaje supervisado, y más concretamente en un problema de clasificación supervisada. Por tanto, se utiliza un conjunto de datos etiquetados donde las salidas son discretas y se corresponden con la clase a la que pertenece cada una de las instancias del conjunto.

1.1. Aprendizaje supervisado

En el ámbito del *machine learning* y por tanto, también en el aprendizaje supervisado, cuando se desarrollan modelos se asume que los datos que se utilizan como conjunto de entrenamiento siguen la misma distribución que los de prueba. El problema es que, en las aplicaciones reales, esto no suele ser cierto por lo que se crea la necesidad de desarrollar técnicas de aprendizaje que tengan en cuenta estos cambios en las distribuciones y permitan adaptarse a ellos.

A modo de ejemplo se presenta un caso en el que esta suposición no se cumple. Supongamos que se quiere desarrollar un modelo para predecir si una persona es mayor de edad o no, es decir se tienen las clases: 1.Edad <18 y 2.Edad $\geq$ 18. Si para hacer el entrenamiento se utilizan datos de una población general pero después se comprueba su funcionamiento en una población mayoritariamente infantil, como las personas que están en un colegio, la distribución de las edades es claramente diferente ya que en un colegio hay una mayor proporción de niños y una menor proporción de adultos que en la población general. Esta situación se ilustra en la Figura 1.1 en la que el círculo azul representa la población general y el rombo amarillo representa la población infantil.

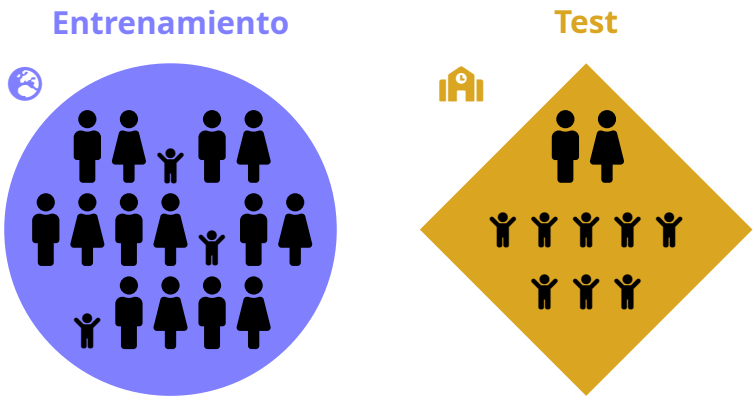


Figura 1.1: Ejemplo de conjuntos con distribuciones distintas (idea extraída de [4]).

Es razonable pensar que si no existe conexión alguna entre los datos de entrenamiento y los de test, no será posible aprender nada acerca de las muestras de test utilizando las de entrenamiento. Por este motivo, es necesario asumir alguna relación entre la distribución subyacente de los datos de entrenamiento y los de test.

Entre las distintas suposiciones que se pueden analizar, este trabajo se centra en una conocida como desplazamiento de covarianzas o *covariate shift*. El *covariate shift* se centra en la situación en la que las distribuciones marginales sobre las instancias son distintas en los conjuntos de entrenamiento y test, es decir, $P_{\text{tr}}(\boldsymbol{instancias}) \neq P_{\text{te}}(\boldsymbol{instancias})$, pero las distribuciones condicionales sobre las etiquetas o clases son constantes, es decir, $P_{\text{tr}}(\boldsymbol{clases}|\boldsymbol{instancias}) = P_{\text{te}}(\boldsymbol{clases}|\boldsymbol{instancias})$. La idea clave es escoger las muestras de entrenamiento más significativas considerando la importancia de cada una de ellas en la predicción de las muestras de test.

Antes de profundizar, es necesario definir la notación utilizada y también algunos conceptos básicos propios del aprendizaje automático, que se emplean a lo largo del trabajo y que son fundamentales para entender correctamente el contexto en el que se encuentra este problema.

En cuanto a la notación, se utilizan letras en negrita para denotar vectores $\boldsymbol{v}$, donde $v^{(i)}$ denota al i-ésimo elemento, y matrices $\boldsymbol{M}$, donde $M_{i,j}$ representa el elemento de la fila i columna j. Además, el superíndice ^{T} se utiliza para indicar la trasposición de estos elementos. Por otra parte, las variables aleatorias se indican con letrás maýusculas (X, Y, ...) y sus realizaciones con la letra minúscula correspondiente (x, y, ...). Las esperanzas matemáticas se denotan como $\mathbb{E}(X)$ o $\mathbb{E}X$. Las normas se representan como $\|\cdot\|_{norma}$, donde el subíndice indica la norma correspondiente. El soporte de una distribución P se expresa como $supp(P)$. Por último, para referirse a estimaciones o aproximaciones de un parámetro θ se emplea $\hat{\theta}$.

En el aprendizaje supervisado se construyen modelos que obtienen clasificadores, a los que se denota como f, a partir de los datos para hacer predicciones. Para conseguirlo, se necesita utilizar un conjunto de datos de entrenamiento para que el modelo pueda obtener el clasificador y otro de prueba o test para evaluar los resultados:

- **Conjunto de muestras de entrenamiento**: $(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}})_{i=1}^{n_{tr}}$
  - Las instancias $(\boldsymbol{x}_{\text{tr, i}})$ son una muestra independiente e igualmente distribuida con distribución de probabilidad $P_{\text{tr}}(\boldsymbol{x})$.
  - Las clases $(y_{\text{tr, i}})$ siguen una distribución condicionada $y_{\text{tr, i}} \sim f\,(y|\boldsymbol{x} = \boldsymbol{x}_{\text{tr, i}})$.
- **Conjunto de muestras de prueba o test**: $(\boldsymbol{x}_{\text{te, i}}, y_{\text{te, i}})_{i=1}^{n_{te}}$
  - Las instancias $(\boldsymbol{x}_{\text{te, i}})$ son una muestra independiente e igualmente distribuida con distribución de probabilidad $P_{\text{te}}(\boldsymbol{x})$.
  - Las clases $(y_{\text{te, i}})$ siguen una distribución condicionada $y_{\text{te, i}} \sim f\,(y|\boldsymbol{x} = \boldsymbol{x}_{\text{te, i}})$.

Con estos conjuntos, el objetivo de un problema de clasificación supervisada es crear un modelo que, con los datos de entrenamiento, obtenga una función que permita predecir correctamente las clases del conjunto de prueba a partir de sus instancias. Para evaluar el resultado de los modelos se utiliza el error de generalización (explicado más adelante en la Sección 1.1.2) y dependiendo de la función utilizada para hacer las predicciones existen distintos modelos.

1.1.1. Modelos para el aprendizaje de funciones

En el aprendizaje supervisado, se busca una función (f) que permita estimar el valor real de las clases (y) dadas sus instancias $(\boldsymbol{x})$ [2]. Es decir, se quiere obtener una función que:

$$y \approx f(\boldsymbol{x}; \boldsymbol{\theta}) \tag{1.1}$$

De esta forma, se utilizan los datos de entrenamiento para aprender el valor de los parámetros del modelo $\left(\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_b)^T \in \boldsymbol{\Theta} \subset \mathbb{R}^b\right)$ que permitan relacionar las instancias con sus clases.

Se pueden plantear distintos modelos según la forma de la función utilizada. A continuación, se explica el modelo de mayor relevancia para este trabajo.

Este es el modelo lineal en los parámetros, denominado de esta forma al utilizar una función $(f(\boldsymbol{x}; \boldsymbol{\theta}))$ que es lineal en el parámetro $\boldsymbol{\theta}$. Para unas funciones fijas y linealmente independientes de las instancias $\left(\{\rho_i(\boldsymbol{x})\}_{i=1}^b\right)$ se expresa como:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{i=1}^{b} \theta_i \rho_i(\boldsymbol{x}) \tag{1.2}$$

donde $\rho(\boldsymbol{x})$ son las funciones base que se utilizan para construir el modelo. Habitualmente, estas funciones son lineales $(\rho(\boldsymbol{x}) = \boldsymbol{x})$, polinómicas $(\rho(\boldsymbol{x}) = \boldsymbol{x}^p, p \in \mathbb{Z})$ o funciones de tipo núcleo $(\rho(\boldsymbol{x}) = K(\boldsymbol{x}, \boldsymbol{x'}))$.

Cuando esta función base es lineal, el modelo se conoce como modelo lineal en las entradas porque, además de ser lineal para el parámetro $(\boldsymbol{\theta})$, es lineal para las entradas $(\boldsymbol{x})$. Este modelo, suponiendo que las entradas son $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$, se expresaría como:

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \theta_0 + \sum_{i=1}^{n} \theta_i x_i \tag{1.3}$$

Es importante mencionar que estos modelos en casos multidimensionales, en donde las instancias $(\boldsymbol{x})$ tienen varias dimensiones, se suelen construir combinando modelos unidimensionales de forma aditiva o multiplicativa.

1.1.2. Error de generalización

Para evaluar el funcionamiento de un modelo se minimiza el error esperado que se produce sobre las muestras de prueba o test, conocido como error de generalización [2]. La expresión matemática de dicho error es la siguiente:

$$Gen = \mathbb{E}_{(\boldsymbol{x}_{\text{te}}, y_{\text{te}})} \left[\ell\left(\boldsymbol{x}_{\text{te}}, y_{\text{te}}, f\left(\boldsymbol{x}_{\text{te}}; \boldsymbol{\theta}\right)\right)\right] \tag{1.4}$$

El objetivo del proceso de aprendizaje es encontrar las reglas de clasificación que minimizan la esperanza de la función de pérdida, es decir, el error de generalización. A continuación se describe lo que es una función de pérdida junto con algunas de las más utilizadas.

1.1.3. Funciones de pérdida

Las funciones de pérdida son las que, dada una instancia $(\boldsymbol{x})$ cuantifican la diferencia que existe entre las etiquetas reales (y) y las predichas por el modelo $(f(\boldsymbol{x};\boldsymbol{\theta}))$. Es decir, con ellas se mide el error que comete un modelo.

Existen numerosas funciones de pérdida en función del problema que se esté tratando. A continuación se definen las más comunes.

1.1.3.1. Error cuadrático medio

La función de pérdida del error cuadrático medio, también conocida como *Mean Squared Error (MSE)*, es una de las más utilizadas en problemas de regresión [5][6]. Esta función eleva al cuadrado la diferencia que existe entre las salidas reales y las predichas. Su expresión es:

$$\ell_{MSE}(\boldsymbol{x}, y, f) = (f(\boldsymbol{x};\boldsymbol{\theta}) - y)^2 \tag{1.5}$$

1.1.3.2. Error absoluto medio

La función de pérdida del error absoluto medio, también conocida como *Mean Absolute Error (MAE)*, es también una de las más utilizadas en problemas de regresión [6]. Esta función calcula el valor absoluto de la diferencia entre las salidas reales y las predichas de forma que se expresa como:

$$\ell_{MAE}(\boldsymbol{x}, y, f) = |f(\boldsymbol{x};\boldsymbol{\theta}) - y| \tag{1.6}$$

1.1.3.3. Función de pérdida 0/1

La función de pérdida 0/1, también conocida como *Zero-One Loss*, es una de las más utilizadas en problemas de clasificación [2]. Esta función asigna un valor 1 si hay un error al clasificar una instancia y 0 si no lo hay. Se expresa como:

$$\ell_{0/1}(\boldsymbol{x}, y, f) = \begin{cases} 0 & \text{si } f(\boldsymbol{x}, \boldsymbol{\theta}) = y, \\ 1 & \text{en otro caso} \end{cases} \tag{1.7}$$

1.1.3.4. Entropía cruzada

La función de pérdida de la entropía cruzada, también conocida como *Cross Entropy Loss (CE)*, es también una de las más utilizadas en problemas de clasificación [6]. Esta función, de manera general, para un problema con N clases se expresa como:

$$\ell_{CE}\left(\boldsymbol{x}, y, f\right) = -\sum_{i=1}^{N}\left(y_i \log\left(f_i\left(\boldsymbol{x}, \boldsymbol{\theta}\right)\right)\right) \tag{1.8}$$

De forma que, si el problema es de clasificación binaria, es decir, la etiqueta solo puede tomar dos valores ($y \in \{0, 1\}$), la expresión es:

$$\ell_{CE}\left(\boldsymbol{x}, y, f\right) = -\left[y \log\left(f\left(\boldsymbol{x}; \boldsymbol{\theta}\right)\right) + (1 - y) \log\left(1 - f\left(\boldsymbol{x}; \boldsymbol{\theta}\right)\right)\right] \tag{1.9}$$

Es importante mencionar que esta función de pérdida también se conoce como *log loss* o pérdida logarítmica, al utilizar el logaritmo de las probabilidades predichas por el modelo [7].

2. El problema del covariate shift

Dentro del paradigma del aprendizaje supervisado, una suposición muy común es que los datos de entrenamiento y los de test pertenecen a la misma distribución. Normalmente, se utiliza esta suposición porque permite simplificar el desarrollo de los modelos al evaluar el rendimiento en unas condiciones bastante más favorables al permitir estimar las esperanzas en el conjunto de test con muestras del conjunto de entrenamiento.

El problema es que esta suposición en la práctica no suele ser cierta. Esto hace que los modelos no sean capaces de generalizar bien en otros escenarios. Por este motivo, dentro de este paradigma existen otras suposiciones como son las siguientes:

- **Covariate Shift**: Se centra en la situación en la que las distribuciones marginales en los conjuntos de entrenamiento y test son distintas $(P_{\text{tr}}(\boldsymbol{x}) \neq P_{\text{te}}(\boldsymbol{x}))$ pero en donde las distribuciones condicionales permanecen constantes $(P_{\text{tr}}(y|x) = P_{\text{te}}(y|x))$. Es decir, que la distribución de las instancias cambia pero la relación entre las instancias y sus clases es constante [2] [8].

- **Label Shift**: Se refiere al escenario donde las distribuciones marginales de las clases cambia entre los conjuntos de entrenamiento y test $(P_{\text{tr}}(\boldsymbol{y}) \neq P_{\text{te}}(\boldsymbol{y}))$ pero las distribuciones condicionales de las instancias dada la clase permanecen constantes $(P_{\text{tr}}(x|y) = P_{\text{te}}(x|y))$ [8] [9].

- **Concept Drift**: Hace referencia a la situación en donde la distribución de las instancias permanece constante pero la relación entre las instancias y las clases cambia a lo largo del tiempo. Es decir, si llamamos t_1 y t_2 a dos instantes de tiempo diferentes, entonces $P_{t_1}(Clases|Instancias) \neq P_{t_2}(Clases|Instancias)$ [10].

A lo largo de este trabajo se profundiza en la primera de ellas, en la suposición del *covariate shift*.

2.1. Adaptación al covariate shift

El objetivo de cualquier modelo es encontrar el valor del parámetro $(\boldsymbol{\theta})$ que minimiza el error de generalización (Ecuación (1.4)). Es decir, se busca resolver el siguiente problema:

$$\boldsymbol{\theta} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathbb{E}_{(\boldsymbol{x}_{\text{te}}, y_{\text{te}})} \left[\ell \left(\boldsymbol{x}_{\text{te}}, y_{\text{te}}, f \left(\boldsymbol{x}_{\text{te}}; \boldsymbol{\theta} \right) \right) \right] \tag{2.1}$$

Como el error de generalización no es conocido, habitualmente para resolver este problema se utiliza un método de estimación conocido como minimización del riesgo empírico. El riesgo empírico es una aproximación del error de generalización donde se aproxima la esperanza del conjunto de test utilizando el conjunto de datos de entrenamiento [11]:

$$R_{emp}\left(f\right) = \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \ell\left(\boldsymbol{x}, y, f\right) \tag{2.2}$$

De forma que si en el problema (Ecuación (2.1)) se sustituye el error de generalización por el riesgo empírico se podría obtener un valor estimado como:

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, R_{emp}\left(f\right) \tag{2.3}$$

El problema es que, bajo la situación del *covariate shift* donde $P_{tr} \neq P_{te}$, el estimador que produce no es consistente, es decir, no converge al verdadero valor del parámetro $\left(\widehat{\theta} \underset{n\to\infty}{\not\mapsto} \theta\right)$ ya que se está minimizando el error de generalización en la distribución de entrenamiento en lugar de la de test cuando no coinciden. Por este motivo, es necesario modificar el método para afrontar este problema [2].

2.1.1. Minimización del riesgo empírico

El método habitual para resolver el problema de optimización (Ecuación (2.1)) es la minimización del riesgo empírico, también conocida como *Empirical Risk Minimization (ERM)*.

En la situación que se suele utilizar en donde las distribuciones de entrenamiento y test son iguales $(P_{\text{tr}}(x, y) = P_{\text{te}}(x, y))$, es decir, cuando no nos encontramos en *covariate shift*, el ERM se define como [2]:

$$\begin{aligned} \text{ERM} &= \min_{f} Gen \\ &= \min_{f} \mathbb{E}_{P_{te}(\boldsymbol{x}_{\text{te}}, y_{\text{te}})}\left[\ell\left(\boldsymbol{x}_{\text{te}}, y_{\text{te}}, f\right)\right] \\ &\text{Asumiendo que: } P_{tr} = P_{te} \\ &= \min_{f} \mathbb{E}_{P_{tr}(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}})}\left[\ell\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}, f\right)\right] \end{aligned} \tag{2.4}$$

Se puede obtener una aproximación para este valor si se utilizan las muestras etiquetadas que están disponibles durante el entrenamiento. De esta forma, se puede sustituir en la Ecuación (2.4) las esperanzas por medias muestrales llegando a lo siguiente:

$$\widehat{\text{ERM}} = \min_{f}\left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \ell\left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f\right)\right] \tag{2.5}$$

Por otra parte, como la función f es de la forma $f(\boldsymbol{x};\boldsymbol{\theta})$, es decir, reglas de clasificación paramétricas, también se puede obtener el valor estimado del parámetro $\left(\widehat{\boldsymbol{\theta}}\right)$ que minimiza el riesgo empírico. La expresión que permite calcularlo es la siguiente:

$$\widehat{\boldsymbol{\theta}}_{ERM} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \ell\left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f\left(\boldsymbol{x}_{\text{tr, i}}; \boldsymbol{\theta}\right)\right) \right] \tag{2.6}$$

El problema es que estos estimadores $\left(\widehat{\text{ERM}}\text{ y }\widehat{\boldsymbol{\theta}}_{ERM}\right)$ solo son consistentes si las distribuciones de entrenamiento y test coinciden ya que, de no ser así, el último paso realizado en la Ecuación (2.4) no se verificaría que $P_{tr} = P_{te}$.

Por este motivo, existen técnicas que permiten ajustar el ERM para que se tengan en cuenta las diferencias entre las distribuciones de los conjuntos.

2.1.2. Minimización del riesgo empírico ponderado por importancia

La minimización del riesgo empírico ponderado por importancia, también conocida como *Importance Weighted Empirical Risk Minimization (IWERM)*, es una variante de ERM (Sección 2.1.1) que tiene en cuenta las diferencias entre las distribuciones de los datos de entrenamiento y los de test.

Para hacerlo este método minimiza el error de generalización, al igual que ERM, pero se supone que las distribuciones marginales de ambos conjuntos son diferentes $(P_{\text{tr}}(x) \neq P_{\text{te}}(x))$ pero las condicionales son iguales $(P_{\text{tr}}(y|x) = P_{\text{te}}(y|x))$. De esta forma, el riesgo empírico ponderado por importancia, bajo la suposición del *covariate shift*, se define como [12]:

$$\begin{aligned}
\text{IWERM} &= \min_f Gen \\
&= \min_f \mathbb{E}_{P_{te}(\boldsymbol{x}_{\text{te}}, y_{\text{te}})} \left[\ell\left(\boldsymbol{x}_{\text{te}}, y_{\text{te}}, f\right)\right] \\
&= \min_f \mathbb{E}_{P_{te}(\boldsymbol{x}_{\text{te}}, y_{\text{te}})} \left[\frac{P_{tr}\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}\right)}{P_{tr}\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}\right)} \ell\left(\boldsymbol{x}_{\text{te}}, y_{\text{te}}, f\right)\right] \\
&\quad \text{Asumiendo que: } supp(P_{te}(\boldsymbol{x}_{\text{te}})) \subseteq supp(P_{tr}(\boldsymbol{x}_{\text{tr}})) \\
&= \min_f \mathbb{E}_{P_{tr}(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}})} \left[\frac{P_{te}\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}\right)}{P_{tr}\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}\right)} \ell\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}, f\right)\right] \\
&= \min_f \mathbb{E}_{P_{tr}(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}})} \left[\frac{P_{te}\left(\boldsymbol{x}_{\text{tr}}\right) P_{te}\left(y_{\text{tr}}|\boldsymbol{x}_{\text{tr}}\right)}{P_{tr}\left(\boldsymbol{x}_{\text{tr}}\right) P_{tr}\left(y_{\text{tr}}|\boldsymbol{x}_{\text{tr}}\right)} \ell\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}, f\right)\right] \\
&\quad \text{Bajo covariate shift: } P_{te}(y|x) = P_{tr}(y|x) \\
&= \min_f \mathbb{E}_{P_{tr}(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}})} \left[\frac{P_{te}(\boldsymbol{x}_{\text{tr}})}{P_{tr}(\boldsymbol{x}_{\text{tr}})} \ell\left(\boldsymbol{x}_{\text{tr}}, y_{\text{tr}}, f\right)\right]
\end{aligned} \tag{2.7}$$

Es posible obtener una estimación del IWERM utilizando las muestras. Para ello, se sustituyen las esperanzas por las medias muestrales de forma que se puede obtener una aproximación del IWERM $\left(\widehat{\text{IWERM}}\right)$ con la siguiente expresión:

$$\widehat{\text{IWERM}} = \min_{f} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{P_{te}(\boldsymbol{x}_{\text{tr, i}})}{P_{tr}(\boldsymbol{x}_{\text{tr, i}})} \ell \left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f \right) \right] \tag{2.8}$$

En el problema de optimización (2.8) aparece el término conocido como importancia que se expresa como $w(x) = \frac{P_{te}(x)}{P_{tr}(x)}$ y que será de especial interés en este trabajo.

Por otra parte, al igual que ocurría en el ERM, en lugar de obtener este valor se puede calcular un valor estimado para el parámetro $\left(\widehat{\boldsymbol{\theta}}\right)$ que minimiza el riesgo empírico ponderado. Para hacerlo, se considera la siguiente expresión:

$$\widehat{\boldsymbol{\theta}}_{IWERM} = \underset{\boldsymbol{\theta}}{\text{argmin}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{P_{\text{te}}(\boldsymbol{x}_{\text{tr, i}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{tr, i}})} \ell \left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f \left(\boldsymbol{x}_{\text{tr, i}}; \boldsymbol{\theta} \right) \right) \right] \tag{2.9}$$

Es importante mencionar que, aunque IWERM permite generar una estimación bajo la suposición del *covariate shift*, los estimadores que se producen (Ecuaciones (2.8) y (2.9)) pueden ser inestables. Esto se debe a que el peso de la importancia puede ser muy grande si el valor de la probabilidad de la muestra de test es mucho mayor que el de la muestra de entrenamiento, o muy pequeño si ocurre justo lo contrario. El problema se produce cuando la importancia es muy grande, ya que puede hacer que el resultado solo dependa de unas pocas muestras.

Por este motivo, existen algunas variantes del IWERM, que se analizarán a continuación, para mejorar la estabilidad de los estimadores, aplanando el peso de la importancia al elevar el término a un exponente o añadiendo un término de regularización.

El **IWERM adaptativo** (minimización del riesgo empírico ponderado por importancia adaptativo) es la variante donde se aplana el peso de la importancia. Para hacerlo, se utiliza un parámetro (γ), conocido como parámetro de aplanamiento, que toma valores entre 0 y 1. Si se añade este parámetro, resaltando en color azul las diferencias, los estimadores de IWERM (Ecuaciones (2.8) y (2.9)) se definen como:

$$\begin{aligned} \widehat{IWERM}_{\gamma} &= \min_{f} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(\frac{P_{te}(\boldsymbol{x}_{\text{tr, i}})}{P_{tr}(\boldsymbol{x}_{\text{tr, i}})} \right)^{\gamma} \ell \left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f \right) \right] \\ \widehat{\boldsymbol{\theta}}_{\gamma} &= \underset{\boldsymbol{\theta}}{\text{argmin}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left(\frac{P_{\text{te}}(\boldsymbol{x}_{\text{tr, i}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{tr, i}})} \right)^{\gamma} \ell \left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f \left(\boldsymbol{x}_{\text{tr, i}}; \boldsymbol{\theta} \right) \right) \right] \end{aligned} \tag{2.10}$$

Se puede observar que los estimadores del IWERM adaptativo (Ecuación (2.10)) son equivalentes a los del estimador ERM (Ecuaciones (2.5) y (2.6)) cuando $\gamma = 0$ y a los del estimador IWERM (Ecuaciones (2.8) y (2.9)) cuando $\gamma = 1$.

El **IWERM con regularización** (minimización del riesgo empírico ponderado por importancia con regularización) es la otra variante en donde se añade un término de regularización al estimador. Este término $(R(\boldsymbol{\theta}))$ intenta evitar que el estimador sea inestable y tiene un parámetro de regularización $(\lambda \geq 0)$. Si se añade este término, resaltando en color azul las diferencias, los estimadores de IWERM (Ecuaciones (2.8) y (2.9)) se definen como:

$$\begin{aligned} \widehat{IWERM}_{\lambda} &= \min_{f} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{P_{te}(\boldsymbol{x}_{\text{tr, i}})}{P_{tr}(\boldsymbol{x}_{\text{tr, i}})} \ell\left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f\right) + \lambda R(\boldsymbol{\theta}) \right] \\ \widehat{\boldsymbol{\theta}}_{\lambda} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[\frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \frac{P_{\text{te}}(\boldsymbol{x}_{\text{tr, i}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{tr, i}})} \ell\left(\boldsymbol{x}_{\text{tr, i}}, y_{\text{tr, i}}, f\left(\boldsymbol{x}_{\text{tr, i}}; \boldsymbol{\theta}\right)\right) + \lambda R(\boldsymbol{\theta}) \right] \end{aligned} \tag{2.11}$$

Elecciones habituales para el término de regularización son la norma L2 (Ecuación (2.12)) o la norma L1 (Ecuación (2.13)):

$$R(\boldsymbol{\theta}) = \sum_{i=1}^{b} {\theta_i}^2 \tag{2.12}$$

$$R(\boldsymbol{\theta}) = \sum_{i=1}^{b} |\theta_i| \tag{2.13}$$

siendo b el número de parámetros del modelo (Sección 1.1.1).

3. Métodos de estimación de importancia

La idea principal sobre la que se basan los distintos métodos de estimación es asignar un peso, al que llamamos importancia, en cada una de las muestras de entrenamiento. El valor del peso para cada muestra dependerá, únicamente, de cómo de probable es que dicha instancia pueda aparecer dentro del conjunto de test.

De esta forma, se define la importancia como el cociente entre la probabilidad de que una muestra aparezca en el conjunto de test y la probabilidad de que aparezca en el conjunto de entrenamiento. Por tanto, la importancia, se puede expresar matemáticamente como:

$$w(x) = \frac{P_{\text{te}}(\boldsymbol{x}_{\text{tr}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{tr}})} \tag{3.1}$$

El principal problema es que, en la práctica, no es posible conocer los pesos de forma exacta, ya que la distribución de las muestras de test $(P_{\text{te}}(\boldsymbol{x}_{\text{tr}}))$ es desconocida, y por tanto, será necesario estimarlos. Este es el objetivo de este capítulo en el que se detallan los distintos métodos que permiten obtener estimaciones de la importancia $\widehat{w}(x)$.

Es importante mencionar que existen dos caminos distintos para obtener la estimación de la importancia. Por un lado, se puede estimar la función de densidad del conjunto de entrenamiento y test $\left(\widehat{P_{tr}x}\text{ y }\widehat{P_{te}x}\right)$ y hacer el cociente entre ellas o, por otro lado, se puede estimar directamente el valor de la importancia $\left(\widehat{w}(x)\right)$.

En este capítulo, se van a explorar los siguientes métodos del estado del arte que permiten hacer esta estimación [2]:

- Kernel Density Estimation (KDE)
- Kernel Mean Matching (KMM)
- Regresión logística
- Procedimiento de Kullback-Leibler
- Ajuste de importancia por mínimos cuadrados

Para explicar el funcionamiento de estos métodos, se supone que se dispone de n_{tr} muestras de entrenamiento, $\left(x_{tr,1}, y_1\right),\ \left(x_{tr,2}, y_2\right),\ \ldots,\ \left(x_{tr,n_{tr}}, y_{n_{tr}}\right) \sim P_{\text{tr}}(x)$ y también de n_{te} instancias de test $\left(x_{te,1}\right),\ \left(x_{te,2}\right),\ \ldots,\ \left(x_{te,n_{te}}\right) \sim P_{\text{te}}(x)$.

3.1. Kernel Density Estimation (KDE)

La estimación de densidad por kernel, también conocida como *Kernel Density Estimation (KDE)*, es una técnica no paramétrica para estimar funciones de densidad a partir de unas muestras $\left(\{x_i\}_{i=1}^{n}\right)$ [2]. Esta técnica, sin necesidad de asumir una forma específica para la distribución de probabilidad, proporciona una aproximación $\left(\widehat{p}(x)\right)$ para dicha distribución:

$$\widehat{p}(x) = \frac{1}{n(2\pi\ \sigma^2)^{\frac{d}{2}}} \sum_{i=1}^{n} K_\sigma(x, x_i) \tag{3.2}$$

De esta forma, se puede obtener una estimación para la importancia $\left(\widehat{w}(x)\right)$. Para ello, se utiliza la estimación de las funciones de densidad (Ecuación (3.2)) con las muestras de entrenamiento $\left(\{\boldsymbol{x}_{\text{tr, i}}\}_{i=1}^{n_{tr}}\right)$ y las de test $\left(\{\boldsymbol{x}_{\text{te, i}}\}_{i=1}^{n_{te}}\right)$ y se calcula el cociente entre ambas. Por tanto, la estimación de la importancia utilizando KDE se expresa como:

$$\begin{gathered} \widehat{w}(x) = \frac{\widehat{p}_{te}(x)}{\widehat{p}_{tr}(x)} \\ \widehat{p}_{tr}(x) \text{ el estimador (3.2) utilizando } \{\boldsymbol{x}_{\text{tr, i}}\}_{i=1}^{n_{tr}} \\ \widehat{p}_{te}(x) \text{ el estimador (3.2) utilizando } \{\boldsymbol{x}_{\text{te, i}}\}_{i=1}^{n_{te}} \end{gathered} \tag{3.3}$$

Para poder hacer la estimación de estas funciones de densidad (Ecuación (3.3)) es necesario seleccionar un kernel (K_σ). En el caso de KDE, el más utilizado suele ser el kernel gaussiano, definido en la Ecuación (3.4) [2].

$$K_\sigma(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right) \tag{3.4}$$

Además, también es necesario elegir la amplitud del kernel (σ), un parámetro que controla el grado de suavizado de la estimación de la densidad. Para hacerlo existen varias heurísticas como la regla de Scott [13, págs. 143-144], la regla de Silverman [14, págs. 47-48], otra que se basa en el método de K vecinos más próximos (KNN) utilizando K=50 [15] o validación cruzada.

Es importante destacar que este método sufre de un problema conocido como la maldición de la dimensionalidad o *curse of dimensionality*. Este problema se produce porque, a medida que aumenta la dimensión del espacio de características, la cantidad de datos necesarios para estimar adecuadamente las funciones de densidad $\left(\widehat{p}_{tr}(x)\right.$ y $\left.\widehat{p}_{te}(x)\right)$ crece exponencialmente [16]. Por tanto, este método no es adecuado en problemas de alta dimensión.

Al estimar los términos de la importancia (Ecuación (3.3)), se comete un error tanto en el numerador como en el denominador que, al dividirlos, se puede amplificar. Por este este motivo, los siguientes métodos buscan hacer la estimación de la importancia directamente que, al no estimar las funciones de densidad, también evita la maldición de la dimensionalidad.

3.2. Obtención de pesos por regresión logística

Una aproximación para estimar directamente la importancia consiste en utilizar un clasificador probabilistico. Para ello, es necesario utilizar una variable clasificadora (η) que permita diferenciar las muestras de entrenamiento y las de test [2]. Esta variable toma el valor -1 para las muestras de entrenamiento y 1 para las de test, de manera que se define como:

$$\eta = \begin{cases} -1 & \text{si es una muestra de } P_{\text{tr}}(x) \\ +1 & \text{si es una muestra de } P_{\text{te}}(x) \end{cases} \tag{3.5}$$

Por tanto, es posible definir las densidades de las muestras del conjunto de entrenamiento y test en función del valor de esta variable (η) como:

$$\begin{aligned} P_{\text{tr}}(x) &= p(x|\eta = -1) \\ P_{\text{te}}(x) &= p(x|\eta = 1) \end{aligned} \tag{3.6}$$

De esta forma, se puede expresar la importancia $(w(x))$ en función del valor de la variable clasificadora. Para conseguirlo, solo es necesario utilizar el teorema de Bayes en la Ecuación (3.6) llegando entonces a la siguiente expresión:

$$w(x) = \frac{P_{\text{te}}(x)}{P_{\text{tr}}(x)} = \frac{p(\eta = 1|x)p(\eta = -1)}{p(\eta = -1|x)p(\eta = 1)} = \frac{p(\eta = -1)}{p(\eta = 1)} \frac{p(\eta = 1|x)}{p(\eta = -1|x)} \tag{3.7}$$

Una vez se dispone de la expresión de la importancia en función de la variable clasificadora (η) (Ecuación (3.7)) es necesario explicar como se calculan sus distintos elementos. El cociente de las probabilidades de la variable clasificadora (η), que aparece en color azul, se estima como el ratio entre el número de muestras de entrenamiento y de test, es decir:

$$\frac{p(\eta = -1)}{p(\eta = 1)} \approx \frac{n_{tr}}{n_{te}} \tag{3.8}$$

Las probabilidades condicionadas $(p(\eta|x))$ del cociente que aparece en color rojo, se pueden aproximar discriminando las muestras de entrenamiento y test empleando un clasificador de regresión logística donde la variable clasificadora (η) es la que juega el papel de la variable de clase. Para hacerlo se utiliza un modelo llamado *logistic regression*, aunque es de clasificación y no de regresión, que es un ERM (Sección 2.1.1) con la entropía cruzada (Sección 1.1.3.4) como función de pérdida.

Con este clasificador se expresa la probabilidad condicionada como:

$$\widehat{p}(\eta|x) = \frac{1}{1 + \exp\left(-\eta \sum_{i=1}^{b} \zeta_i \rho_i(x)\right)} \tag{3.9}$$

siendo ρ las funciones base, b el número de estas funciones y ζ el valor que minimiza la log-verosimilitud regularizada negativa. Es posible obtener una estimación para este valor ($\widehat{\zeta}$) utilizando los datos de entrenamiento y test, y añadiendo un término de regularización $\lambda\zeta^T\zeta$ de la siguiente forma:

$$\begin{aligned}\hat{\zeta} = \underset{\zeta}{\operatorname{argmin}} \Bigg[&\sum_{i=1}^{n_{tr}} \log\left(1 + \exp\left(\sum_{j=1}^{b} \zeta_j \rho_j(\boldsymbol{x}_{\text{tr, i}})\right)\right) \\ &+ \sum_{i=1}^{n_{te}} \log\left(1 + \exp\left(-\sum_{j=1}^{b} \zeta_j \rho_j(\boldsymbol{x}_{\text{te, i}})\right)\right) + \lambda\zeta^T\zeta \Bigg]\end{aligned} \tag{3.10}$$

donde la función a minimizar es convexa lo que hace posible el obtener la solución utilizando métodos de optimización como, por ejemplo, el descenso del gradiente.

Si se utilizan todas estas aproximaciones (Ecuaciones 3.8 a 3.10) en la expresión de la importancia (Ecuación (3.7)) se obtiene una estimación de la misma. Esta estimación también se conoce como el modelo log-lineal y se expresa de la siguiente forma:

$$\widehat{w}(x) = \frac{n_{tr}}{n_{te}} \exp\left(\sum_{i=1}^{b} \zeta_i \rho_i(x)\right) \tag{3.11}$$

Este método no sufre de la maldición de la dimensionalidad, al no ser necesario hacer una estimación de las funciones de densidad. Sin embargo, el funcionamiento depende enormemente de la capacidad del clasificador logístico para diferenciar entre las muestras de entrenamiento y test. Además, la elección de las funciones base $\{\rho_i(x)\}_{i=1}^{b}$ es crucial para el rendimiento del modelo. Estas limitaciones han llevado al desarrollo de otros métodos que utilizan un problema de optimización para estimar la importancia. Estos métodos evitan las complicaciones asociadas con el clasificador aunque tienen la posible dificultad de resolver el problema para encontrar la solución óptima.

3.3. Kernel Mean Matching (KMM)

Otra aproximación que permite estimar directamente la importancia es el emparejamiento de medias en el núcleo, también conocido como *Kernel Mean Matching* (KMM). Este busca obtener la estimación $\left(\widehat{w}(x)\right)$ de forma que se minimice la distancia entre las esperanzas de las muestras de entrenamiento y test en un espacio de Hilbert con núcleo reproductor (RKHS) [2].

Este método permite obtener una estimación de la importancia utilizando un kernel que induzca un espacio RKHS para resolver el siguiente problema de optimización:

$$\begin{aligned} &\min_{w(x)} \left\| \mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[K_\sigma \left(\boldsymbol{x}_{\text{te}}, \cdot \right) \right] - \mathbb{E}_{\boldsymbol{x}_{\text{tr}}} \left[w \left(\boldsymbol{x}_{\text{tr}} \right) K_\sigma \left(\boldsymbol{x}_{\text{tr}}, \cdot \right) \right] \right\|_{\mathcal{H}}^2 \\ &\text{sujeto a:} \\ &\mathbb{E}_{\boldsymbol{x}_{\text{tr}}} \left[w \left(\boldsymbol{x}_{\text{tr}} \right) \right] = 1 \\ &w(x) \geq 0 \end{aligned} \tag{3.12}$$

donde $\| \cdot \|_{\mathcal{H}}$ representa la norma en el RKHS gaussiano.

En la práctica, dado que solo se dispone de las muestras, se reemplazan las esperanzas por medias empíricas en el problema de optimización anterior (Ecuación (3.12)) [2]. De esta forma, el problema de minimización es el siguiente:

$$\begin{aligned} &\min_{w(x)} \left\| \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} K_\sigma \left(\boldsymbol{x}_{\text{te, i}}, \cdot \right) - \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \left[w \left(\boldsymbol{x}_{\text{tr, i}} \right) K_\sigma \left(\boldsymbol{x}_{\text{tr, i}}, \cdot \right) \right] \right\|_{\mathcal{H}}^2 \\ &\text{sujeto a:} \\ &\frac{\sum_{i=1}^{n_{tr}} w \left(\boldsymbol{x}_{\text{tr, i}} \right)}{n_{tr}} = 1 \\ &w(x) \geq 0 \end{aligned} \tag{3.13}$$

El problema de optimización anterior (Ecuación (3.13)) siempre se reduce a un problema cuadrático que se puede resolver de una forma más eficiente. Por tanto, el problema de minimización que se resuelve para obtener la estimación de la importancia por KMM es el siguiente:

$$\begin{aligned} &\min_{w_i} \frac{\left[[w, \mathbb{1}] \, \mathcal{K} [w, \mathbb{1}]^T \right]}{n_{te} n_{tr}} \\ &\text{sujeto a:} \\ &\left\| \sum_{i=1}^{n_{tr}} w_i - n_{tr} \right\| \leq n_{tr} \varepsilon \\ &0 \leq w_i \leq B \quad \forall i = 1, \ldots, n_{tr} \end{aligned} \tag{3.14}$$

donde w tiene longitud n_{tr}, $\mathbb{1}$ es un vector de unos de longitud n_{te} y $\mathcal{K}$ es la matriz kernel dada por $\mathcal{K}(i, j) = K_\sigma \left(x_i, x_j \right)$. De esta forma, la matriz $\mathcal{K}$ es una matriz por bloques que contiene las distancias entre las muestras utilizando un kernel K_σ y, por tanto, se puede expresar como:

$$\mathcal{K} = \left(\begin{array}{ccc:ccc} K_\sigma\left(X_{tr,1}, X_{tr,1}\right) & \cdots & K_\sigma\left(X_{tr,1}, X_{tr,n_{tr}}\right) & K_\sigma\left(X_{tr,1}, X_{te,1}\right) & \cdots & K_\sigma\left(X_{tr,1}, X_{te,n_{te}}\right) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K_\sigma\left(X_{tr,n_{tr}}, X_{tr,1}\right) & \cdots & K_\sigma\left(X_{tr,n_{tr}}, X_{tr,n_{tr}}\right) & K_\sigma\left(X_{tr,n_{tr}}, X_{te,1}\right) & \cdots & K_\sigma\left(X_{tr,n_{tr}}, X_{te,n_{te}}\right) \\ \hdashline K_\sigma\left(X_{te,1}, X_{tr,1}\right) & \cdots & K_\sigma\left(X_{te,1}, X_{tr,n_{tr}}\right) & K_\sigma\left(X_{te,1}, X_{te,1}\right) & \cdots & K_\sigma\left(X_{te,1}, X_{te,n_{te}}\right) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ K_\sigma\left(X_{te,n_{te}}, X_{tr,1}\right) & \cdots & K_\sigma\left(X_{te,n_{te}}, X_{tr,n_{tr}}\right) & K_\sigma\left(X_{te,n_{te}}, X_{te,1}\right) & \cdots & K_\sigma\left(X_{te,n_{te}}, X_{te,n_{te}}\right) \end{array}\right) \tag{3.15}$$

En este problema (Ecuación (3.14)), la primera restricción permite controlar la desviación media de la importancia respecto del valor 1. Esto limita la capacidad del modelo para ajustarse a los datos intentando evitar que se produzca sobreajuste. Por otro lado, la segunda restricción limita el valor máximo de la importancia para evitar que se produzcan valores extremos que puedan afectar negativamente al rendimiento del modelo.

Este método no sufre de la maldición de la dimensionalidad, al no hacer una estimación de las funciones de densidad, por lo que es de esperar que funcione correctamente en problemas de alta dimensión. Sin embargo, KMM necesita resolver un problema de optimización que, en algunas situaciones, puede ser excesivamente costoso. Además, la elección del kernel y del valor de los parámetros (σ, ε y B) pueden afectar significativamente al resultado obtenido. Por este motivo, se han desarrollado otras alternativas para estimar directamente la importancia pero intentando evitar estos problemas. Para ello, se modifica el problema de optimización a resolver. Por ejemplo si, en lugar de minimizar las distancias en un espacio RKHS, se utiliza la divergencia de Kullback-Leibler se tiene el procedimiento que se explica a continuación.

3.4. Procedimiento de Kullback-Leibler (KLIEP)

El procedimiento de Kullback-Leibler, también conocido como *Kullback-Leibler Importance Estimation Procedure* (KLIEP), es un método que busca estimar la importancia $\widehat{w}(x)$ de forma que se minimice la divergencia de Kullback-Leibler (KL) entre las distribuciones de entrenamiento y test. Por tanto, este método no requiere de hacer la estimación de las funciones de densidad de ambas distribuciones evitando así el problema de la maldición de la dimensionalidad [2].

La divergencia de Kullback-Leibler es una medida utilizada para medir la diferencia que existe entre dos distribuciones de probabilidad P y Q [17]. Se puede expresar matemáticamente como (Ecuación (3.16)):

$$D_{KL}(P \mid\mid Q) = \sum_{x \in X} \left(P(x) \log \frac{P(x)}{Q(x)} \right) \tag{3.16}$$

Esta distribución (Ecuación (3.16)) es siempre positiva y solo toma el valor 0 cuando ambas distribuciones son idénticas. Además, no es simétrica, es decir, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

El algoritmo de KLIEP busca obtener la importancia utilizando una combinación lineal de los parámetros (Sección 1.1.1) de la siguiente forma (Ecuación (3.17)):

$$\widehat{w}(x) = \sum_{i=1}^{b} \alpha_i \rho_i(x) \tag{3.17}$$

siendo b el número de parámetros $\left(\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_b)^T\right)$ y $\{\rho_i(x)\}_{i=1}^{b}$ las funciones base.

El objetivo es obtener una estimación de la importancia $\widehat{w}(x)$ que minimice la divergencia de Kullback-Leibler entre la distribución real del conjunto de test $(P_{\text{te}}(x))$ y la distribución estimada $\left(\widehat{P_{te}}(x) = \widehat{w}(x) P_{\text{tr}}(x)\right)$, es decir se quiere resolver el siguiente problema:

$$\min_{w(x)} \left[D_{KL} \left(P_{\text{te}}(x) \,||\, \widehat{w}(x) P_{\text{tr}}(x) \right) \right] \tag{3.18}$$

El procedimiento consiste entonces en determinar los parámetros, α que permiten obtener la importancia estimada (Ecuación (3.17)), de forma que la divergencia de Kullback-Leibler (Ecuación (3.18)) sea mínima. Por tanto, el problema de optimización que se quiere resolver consiste en calcular el mínimo de $D_{KL} \left(P_{\text{te}}(x) \,||\, \widehat{w}(x) P_{\text{tr}}(x) \right)$, lo que se puede expresar como

$$\begin{aligned}
&\min_{w(x)} \left[D_{KL} \left(P_{\text{te}}(x) \,||\, \widehat{w}(x) P_{\text{tr}}(x) \right) \right] = \\
&\min_{w(x)} \left[\mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[\log \left(\frac{P_{\text{te}}(\boldsymbol{x}_{\text{te}})}{\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right) P_{\text{tr}}(\boldsymbol{x}_{\text{te}})} \right) \right] \right] = \\
&\min_{w(x)} \left[\mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[\log \left(\frac{P_{\text{te}}(\boldsymbol{x}_{\text{te}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{te}})} \right) \right] - \mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[\log \left(\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right) \right) \right] \right]
\end{aligned} \tag{3.19}$$

Como el primer término de la Ecuación (3.19) es constante, el problema queda reducido a calcular el mínimo del segundo (■), es decir:

$$\begin{aligned}
&\min_{w(x)} \left[-\mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[\log \left(\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right) \right) \right] \right] = \\
&\max_{w(x)} \left[\mathbb{E}_{\boldsymbol{x}_{\text{te}}} \left[\log \left(\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right) \right) \right] \right]
\end{aligned} \tag{3.20}$$

El problema de optimización de KLIEP a resolver, si se reemplaza la esperanza en la Ecuación (3.20) por la media empírica y la importancia estimada $\left(\widehat{w}\right)$ con la Ecuación (3.17), es el siguiente:

$$\begin{aligned}
&\max_{\{\alpha_i\}_{i=1}^{b}} \left[\sum_{i=1}^{n_{te}} \log \left(\sum_{j=1}^{b} \alpha_j \rho_j \left(\boldsymbol{x}_{\text{te, i}} \right) \right) \right] \\
&\text{sujeto a:} \\
&\sum_{i=1}^{n_{tr}} \sum_{j=1}^{b} \alpha_j \rho_j \left(\boldsymbol{x}_{\text{tr, i}} \right) = n_{tr} \\
&\alpha_i \geq 0 \text{ para } i = 1, 2, \ldots, b
\end{aligned} \tag{3.21}$$

El principal problema de este método es que su funcionamiento depende de la elección de las funciones base $\left(\{\rho_i(x)\}_{i=1}^{b}\right)$. Para resolverlo, el método habitual consiste en seleccionar distintas funciones base y elegir aquella que maximice el valor de la Ecuación (3.21). El modelo candidato suele ser un kernel gaussiano (Ecuación (3.4)) centrado en las muestras de test de forma que la estimación de la importancia (Ecuación (3.17)) se obtendría como [18]:

$$\widehat{w}(x) = \sum_{i=1}^{n_{te}} \alpha_i K_\sigma\left(x, \boldsymbol{x}_{\text{te, i}}\right) \tag{3.22}$$

3.5. Ajuste de importancia por mínimos cuadrados (LSIF)

El ajuste de importancia por mínimos cuadrados, también conocido como *Least-Squares Importance Fitting* (LSIF) es un procedimiento muy similar a KLIEP (Sección 3.4) pero que utiliza la pérdida cuadrática (Sección 1.1.3.1) para medir las diferencias entre las funciones de densidad en lugar de la divergencia de Kullback-Leibler [19]. En este método se busca modelar la importancia utilizando combinaciones lineales en los parámetros (Sección 1.1.1):

$$\widehat{w}(x) = \sum_{i=1}^{b} \alpha_i \rho_i(x) \tag{3.23}$$

de forma que el valor de los parámetros (α) minimice el siguiente error cuadrático:

$$\begin{aligned}
&\min_{w(x)}\left[\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right) - \frac{P_{\text{te}}(\boldsymbol{x}_{\text{tr}})}{P_{\text{tr}}(\boldsymbol{x}_{\text{tr}})}\right)^2\right]\right] = \\
&\min_{w(x)}\left[\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right) - w\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right]\right] = \\
&\min_{w(x)}\left[\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(w\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right] - \mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right) w\left(\boldsymbol{x}_{\text{tr}}\right)\right]\right] = \\
&\min_{w(x)}\left[\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(w\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right] - \mathbb{E}_{\boldsymbol{x}_{\text{te}}}\left[\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right)\right]\right]
\end{aligned} \tag{3.24}$$

En la Ecuación (3.24) se busca minimizar las diferencias entre la importancia estimada $\left(\widehat{w}(x)\right)$ y la importancia real $\left(w(x) = \frac{P_{\text{te}}(x)}{P_{\text{tr}}(x)}\right)$ utilizando el error cuadrático. Como el segundo término (que aparece en color rojo) es constante, lo podemos ignorar, ya que no afecta en la minimización. De esta forma, la expresión que se quiere minimizar es:

$$\min_{w(x)}\left[\frac{1}{2}\mathbb{E}_{\boldsymbol{x}_{\text{tr}}}\left[\left(\widehat{w}\left(\boldsymbol{x}_{\text{tr}}\right)\right)^2\right] - \mathbb{E}_{\boldsymbol{x}_{\text{te}}}\left[\widehat{w}\left(\boldsymbol{x}_{\text{te}}\right)\right]\right] \tag{3.25}$$

El problema de optimización a resolver, si se reemplazan las esperanzas en la Ecuación (3.25) por las medias empíricas y las importancias estimadas $\left(\widehat{w}\right)$ por su combinación lineal (Ecuación (3.23)), es el siguiente:

$$\begin{aligned}
&\min_{w(x)} \left[\frac{1}{2n_{tr}} \sum_{i=1}^{n_{tr}} \left(\widehat{w}\left(\boldsymbol{x}_{\text{tr, i}}\right) \right)^2 - \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \widehat{w}\left(\boldsymbol{x}_{\text{te, i}}\right) \right] = \\
&\min_{\alpha} \left[\frac{1}{2n_{tr}} \sum_{i=1}^{n_{tr}} \left(\sum_{j=1}^{b} \alpha_j \rho_j\left(\boldsymbol{x}_{\text{tr, i}}\right) \sum_{k=1}^{b} \alpha_k \rho_k\left(\boldsymbol{x}_{\text{tr, i}}\right) \right) - \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \left(\sum_{j=1}^{b} \alpha_j \rho_j\left(\boldsymbol{x}_{\text{te, i}}\right) \right) \right]
\end{aligned} \tag{3.26}$$

En el procedimiento de ajuste de importancia por mínimos cuadrados se busca obtener los parámetros (α) que minimizan la Ecuación (3.26). Cabe destacar que en esta misma expresión también es posible añadir un término de regularización $\left(\lambda \mathbb{1}_b^T \alpha\right)$ siendo λ el parámetro de regularización $(\lambda > 0)$. Habitualmente, este procedimiento, incluyendo el término de regularización, se expresa de una forma más sencilla como:

$$\begin{aligned}
&\min_{\alpha \in \mathbb{R}} \left[\frac{1}{2} \alpha^T \widehat{H} \alpha - \widehat{h}^T \alpha + \lambda \mathbb{1}_b^T \alpha \right] \\
&\text{sujeto a:} \\
&\alpha_i \geq 0
\end{aligned} \tag{3.27}$$

en donde $\widehat{H}$ y $\widehat{h}$ vienen dadas por:

$$\begin{aligned}
\widehat{H}_{j,k} &= \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \rho_j\left(\boldsymbol{x}_{\text{tr, i}}\right) \rho_k\left(\boldsymbol{x}_{\text{tr, i}}\right) \\
\widehat{h}_j &= \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \rho_j\left(\boldsymbol{x}_{\text{te, i}}\right)
\end{aligned} \tag{3.28}$$

Al igual que ocurría con KLIEP, el funcionamiento de este procedimiento depende de la selección de las funciones base (ρ).

Existe una variante de LSIF conocida como ajuste de importancia por mínimos cuadrados sin restricciones (uLSIF) que intenta obtener la importancia $\widehat{w}(x)$ eliminando las restricciones de no negatividad en los parámetros (α) [19]. Se va a denotar a los parámetros de esta variante como β en lugar de α con el objetivo de diferenciarlos de manera sencilla. En este caso el problema de optimización es muy similar a la de LSIF, eliminando la restricción sobre los parámetros (α) en la Ecuación (3.27). Es importante tener en cuenta que, al eliminar esta restricción también es necesario modificar el término de regularización $\left(\lambda \mathbb{1}_b^T \alpha\right)$ sustituyéndolo por uno cuadrático $\left({}^{\lambda}/_{2} \beta^T \beta\right)$ para evitar que los parámetros se compensen. De esta forma, el procedimiento de ajuste de importancia por mínimos cuadrados sin restricciones (uLSIF) busca los parámetros que minimizan:

$$\min_{\beta \in \mathbb{R}} \left[\frac{1}{2} \beta^T \widehat{H} \beta - \widehat{h}^T \beta + \frac{\lambda}{2} \beta^T \beta \right] \tag{3.29}$$

3.6. Comparación de los métodos de estimación

Por último, con el objetivo de expresar de forma clara y resumida los métodos explicados en este capítulo, se crea la Tabla 3.1 con las principales ventajas e inconvenientes de cada uno de ellos.

Comparación de los métodos de estimación de la importancia	
KDE	**Ventajas** ▪ Fácil de implementar **Inconvenientes** ▪ Estimación de las marginales ▪ Sufre la maldición de la dimensionalidad
Regresión logística	**Ventajas** ▪ Estimación directa de la importancia ▪ No sufre la maldición de la dimensionalidad **Inconvenientes** ▪ Depende del clasificador logístico ▪ Elección de las funciones base
KMM	**Ventajas** ▪ Estimación directa de la importancia ▪ No sufre la maldición de la dimensionalidad **Inconvenientes** ▪ Resolver problema de optimización (minimizar distancias en un RKHS) ▪ Elección del kernel y parámetros
KLIEP	**Ventajas** ▪ Estimación directa de la importancia ▪ No sufre la maldición de la dimensionalidad **Inconvenientes** ▪ Resolver problema de optimización (minimizar divergencia de Kullback-Leibler) ▪ Elección de las funciones base
LSIF y uLSIF	**Ventajas** ▪ Estimación directa de la importancia ▪ No sufre la maldición de la dimensionalidad **Inconvenientes** ▪ Resolver problema de optimización (minimizar el error cuadrático) ▪ Elección de las funciones base

Tabla 3.1: Ventajas e inconvenientes de los métodos de estimación de la importancia

4. Ejemplos de los métodos de estimación de importancia

En el Capítulo 3 se han visto los diferentes métodos que permiten estimar la importancia y este capítulo se centra en evaluar el funcionamiento de cada uno de ellos.

Los códigos necesarios para replicar los experimentos desarollados a lo largo de este capítulo, salvo aquellos utilizados para las representaciones gráficas de los resultados, se incluyen en el apéndice A. Estos códigos han sido implementados en Python.

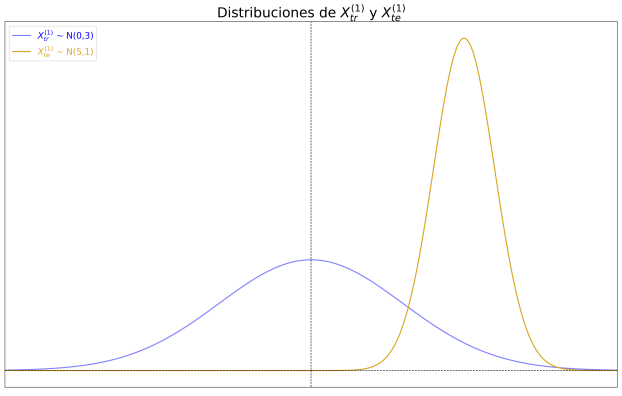
4.1. Generación de los datos

El problema que se va a plantear es de clasificación binaria en donde la clase (Y) será la variable respuesta que depende de dos covariables $\left(X^{(1)}, X^{(2)}\right)$ y está definida por la Ecuación (4.1).

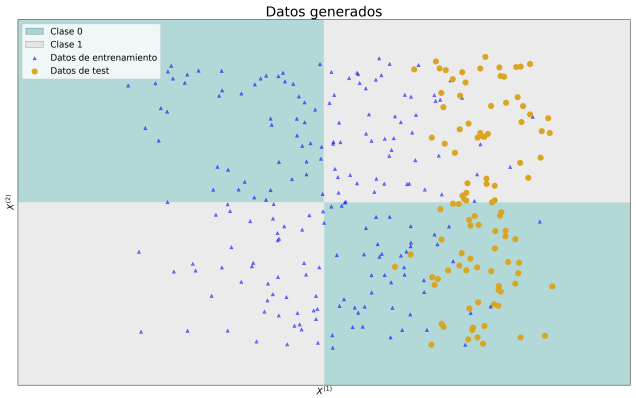
$$Y = \begin{cases} 0 & \text{si } X^{(1)}X^{(2)} \leq 0 \\ 1 & \text{si } X^{(1)}X^{(2)} > 0 \end{cases} \tag{4.1}$$

En primer lugar, se generan los datos que se utilizarán como ejemplo. Estos datos se dividen en dos conjuntos, entrenamiento y test. Para aplicar la suposición del *covariate shift* los datos de entrenamiento y test han de tener distribuciones marginales sobre las instancias diferentes.

Así, en el conjunto de entrenamiento $\left(x^{(1)}_{tr,1}, x^{(2)}_{tr,1}, y_{tr,1}\right), \left(x^{(1)}_{tr,2}, x^{(2)}_{tr,2}, y_{tr,2}\right), \ldots, \left(x^{(1)}_{n_{tr}}, x^{(2)}_{n_{tr}}, y_{n_{tr}}\right)$, las covariables explicativas, $X^{(1)}$ y $X^{(2)}$ siguen una $N(0,3)$ y $U(-1,1)$ respectivamente, mientras que en el conjunto de test $\left(x^{(1)}_{te,1}, x^{(2)}_{te,1}\right), \left(x^{(1)}_{te,2}, x^{(2)}_{te,2}\right), \ldots, \left(x^{(1)}_{n_{te}}, x^{(2)}_{n_{te}}\right)$, las distribuciones de las covariables $X^{(1)}$ y $X^{(2)}$ son una $N(5,1)$ y $U(-1,1)$. Los datos generados se muestran en la Figura 4.1.



(a) Distribuciones de $X^{(1)}$.



(b) Datos generados para clasificación.

Figura 4.1: Datos generados para evaluar los métodos de estimación de la importancia.

En los experimentos que se realizan en este capítulo se busca un clasificador que separe correctamente las clases en los datos del conjunto de test. Como el problema es de clasificación binaria, para un clasificador de la forma $\theta^T \rho(x)$, es posible asociar su valor a cada una de las clases del siguiente modo:

$$\begin{cases} \theta^T \rho(x) \leq 0 \Longrightarrow \text{Clase 0} \\ \theta^T \rho(x) > 0 \Longrightarrow \text{Clase 1} \end{cases} \tag{4.2}$$

En este caso, se consideran como funciones $\rho(x) = \left(1, x^{(1)}, x^{(2)}\right)$ y $\theta = [\theta_0, \theta_1, \theta_2]$. Es decir, se utiliza un término independiente y las dos covariables explicativas. Esta situación lleva a que el clasificador que separa las clases sea una línea recta como se ve en la Ecuación (4.3):

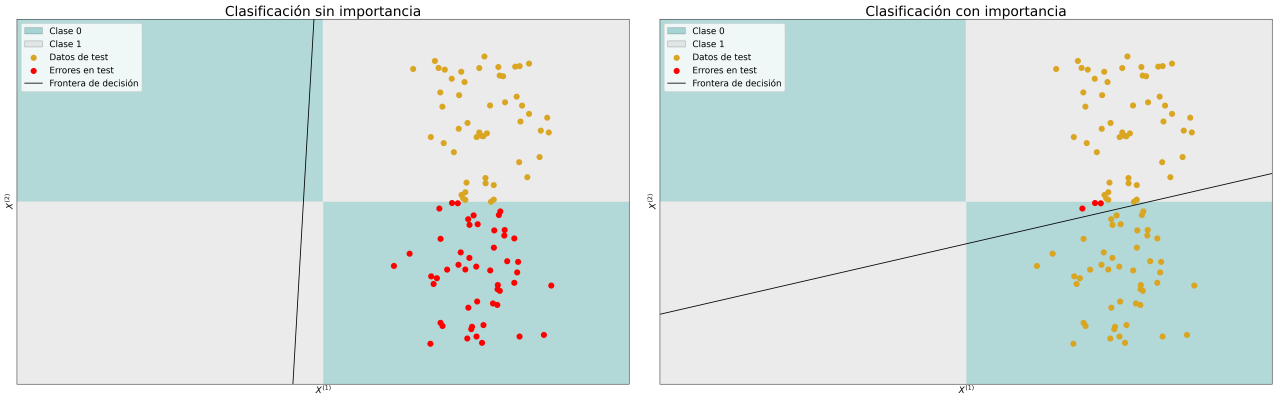
$$\begin{aligned} 0 &= \theta^T \rho(x) \\ 0 &= \theta_0 + \theta_1 X^{(1)} + \theta_2 X^{(2)} \\ X^{(2)} &= -\frac{\theta_0}{\theta_2} - \frac{\theta_1}{\theta_2} X^{(1)} \end{aligned} \tag{4.3}$$

Además, otro motivo para utilizar una línea recta como frontera entre las clases es que, aunque los datos de entrenamiento no son linealmente separables, los de test sí que lo son. Por otra parte, con funciones sencillas, como las lineales, los modelos tienden a generalizar mejor.

4.2. Necesidad de la importancia

Se comienza comprobando el motivo por el que es necesario utilizar la importancia en una situación donde la distribución de entrenamiento y test son distintas. Para hacerlo se utiliza primero un clasificador que no busca corregir el *covariate shift*, es decir, un clasificador que proporciona el mismo peso a todas las muestras del conjunto de entrenamiento y al que se denomina de ahora en adelante como "clasificador sin importancia". Dada la simetría de las distribuciones de este conjunto $\left(X_{tr}^{(1)} \sim N(0,3) \text{ y } X_{tr}^{(2)} \sim U(-1,1)\right)$, los datos deberían estar distribuidos de una forma similar entre los cuatro cuadrantes. Esto, añadido a la forma de las clases en este problema, hace que se esperen malos resultados dado que la región donde $X^{(1)} < 0$ no es de utilidad para aprender sobre el conjunto de test.

A continuación, se crea un clasificador con la importancia real de las muestras de entrenamiento (Ecuación (3.1)). En la práctica, este valor es desconocido pero aquí se puede calcular al conocer las distribuciones con las que se generan los datos. De esta forma, se pueden utilizar ambos clasificadores para obtener la recta que separa las clases en el conjunto de test. En la Figura 4.2a se representa la línea de separación obtenida con el clasificador sin importancia mientras que en la Figura 4.2b aparece el resultado del que emplea la importancia real. En ambas figuras, se colorean en rojo los puntos del conjunto de test que se clasifican incorrectamente, es decir, que la clase real no coincide con la predicha por el clasificador correspondiente.



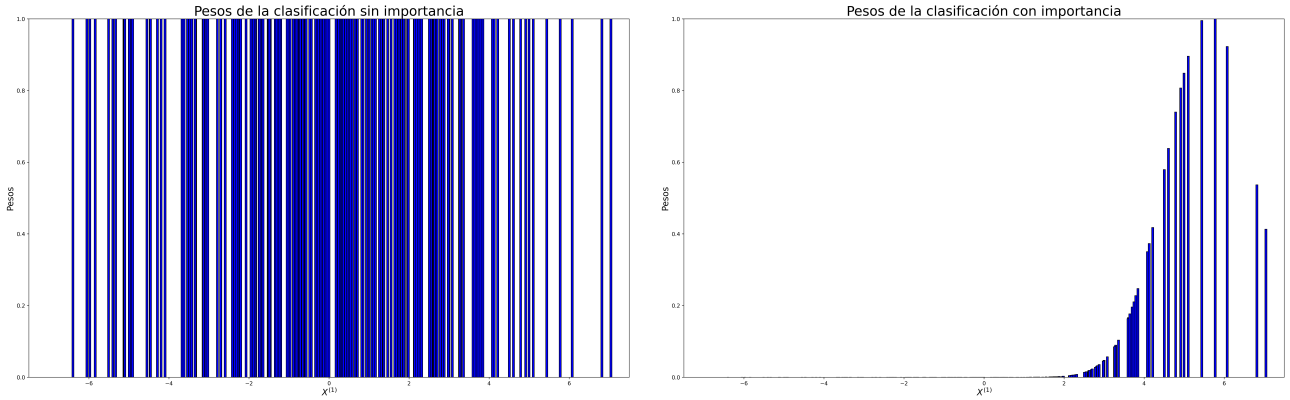
(a) Representación de la recta de clasificación sin importancia.

(b) Representación de la recta de clasificación obtenida al utilizar la importancia.

Figura 4.2: Resultados del clasificador sin importancia y del que usa la importancia real.

Se observa que, en este caso, el clasificador sin importancia (Figura 4.2a) hace que todos los datos del conjunto de test pertenezcan a la misma clase, la clase 1, ya que se encuentran en el mismo lado (a la derecha) de la línea de separación entre clases. El motivo de este mal resultado es porque se esta creando la clasificación para el conjunto de entrenamiento y la de test, en este caso, es distinta. En cualquier situación, asignar todos los datos a una sola clase es un muy mal resultado lo que coincide con lo esperado.

Por otro lado, se puede ver, en los resultados obtenidos al aplicar la importancia (Figura 4.2b), que el funcionamiento es mucho mejor puesto que se crea una división de clases que comete muchos menos errores. Si se representan los pesos aplicados a las muestras de entrenamiento en ambos casos (Figura 4.3) se puede ver claramente la diferencia y el motivo por el que los resultados son tan distintos.



(a) Pesos de las muestras de entrenamiento con el clasificador sin importancia.

(b) Pesos de las muestras de entrenamiento con el clasificador con importancia.

Figura 4.3: Comparación de los pesos utilizados con ambos clasificadores.

A continuación, con la idea de mostrar más claramente el efecto de los pesos de la Figura 4.3b se representan las muestras de entrenamiento coloreadas en función de su peso en la Figura 4.4. En ella se puede observar que las muestras que se encuentran en la región donde están las de test son las que tienen los pesos más altos.

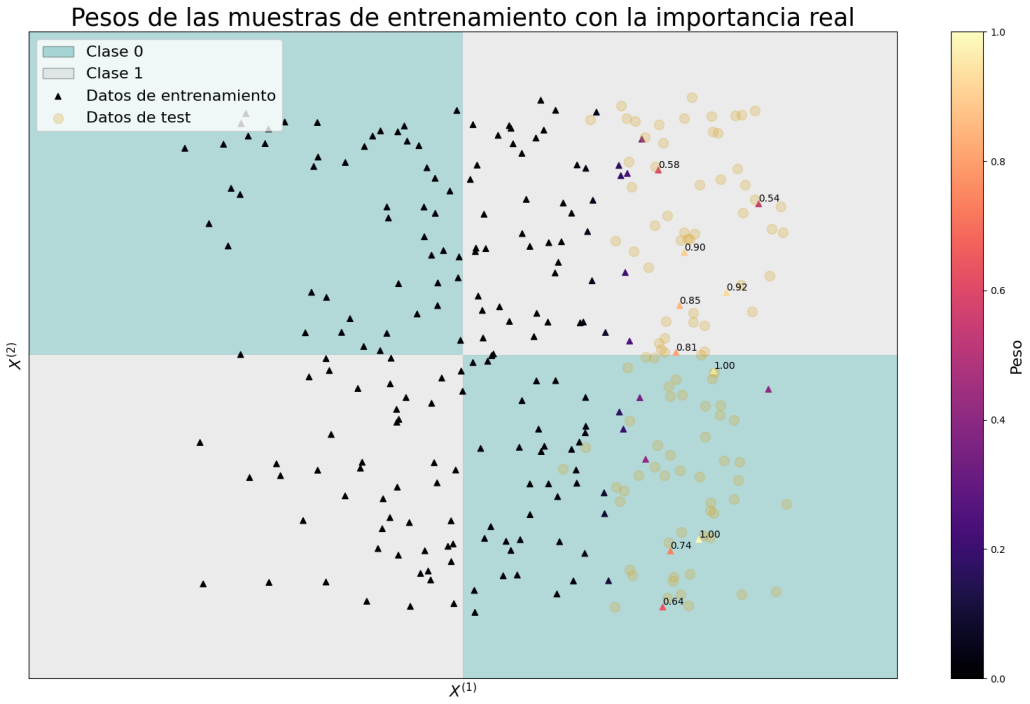


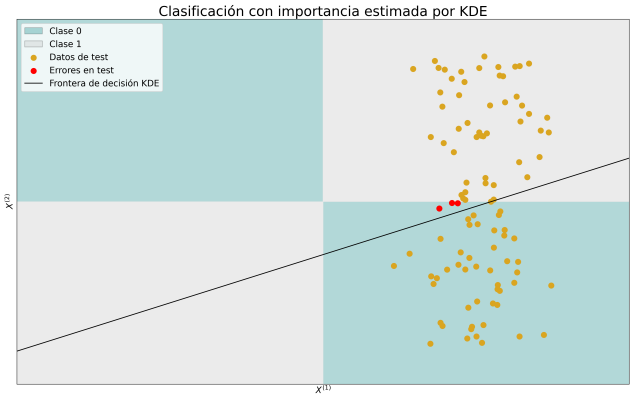
Figura 4.4: Importancia real aplicada en las muestras de entrenamiento.

Con este ejemplo se ve la mejora de los resultados que se produce al aplicar la importancia cuando nos encontramos en una situación donde las distribuciones varían. De ahí la búsqueda de métodos que se adapten a estos cambios. El problema es que, en una situación real, no se conocen las distribuciones y por eso existen los métodos de estimación de la importancia.

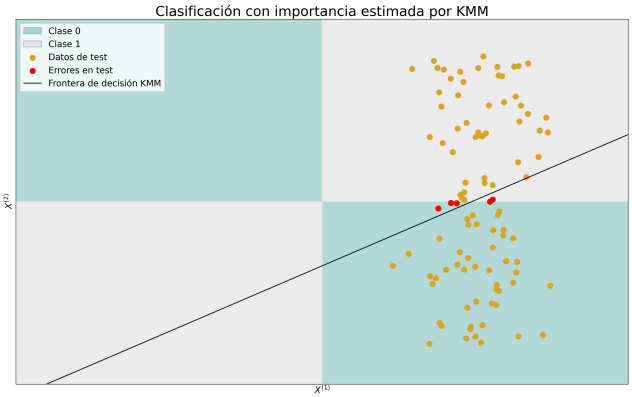
4.3. Métodos de estimación de la importancia

Una vez se ha comprobado el efecto que se produce al aplicar la importancia sobre unos datos donde las distribuciones de entrenamiento y test no coinciden se quiere evaluar el funcionamiento de los métodos que permiten estimarla.

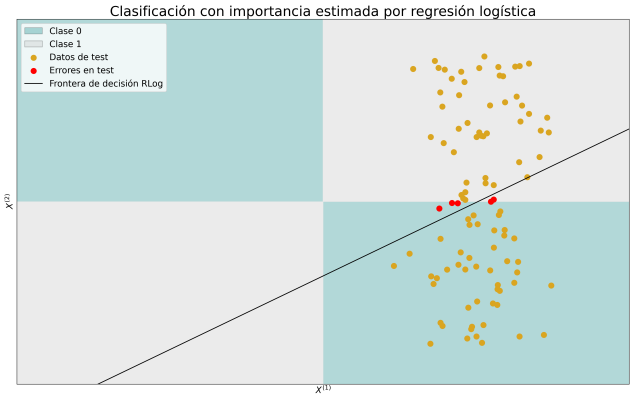
Para hacerlo, se consideran los métodos de estimación explicados en el Capítulo 3. Con cada uno de ellos se crea un clasificador a partir de las estimaciones. Con el objetivo de visualizar los resultados obtenidos se realiza una representación (Figura 4.5), para cada uno de ellos, de la recta que separa las clases. El clasificador se ha obtenido utilizando los pesos estimados con el método correspondiente. Obviamente, al ser estimaciones de la importancia es de esperar que todos los resultados sean similares al obtenido utilizando la importancia real (Figura 4.2b). Por tanto, también se espera que el error de clasificación en el conjunto de test sea pequeño.



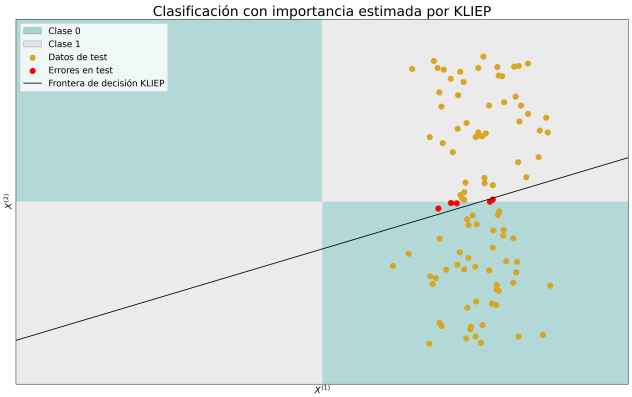
(a) Método de KDE.



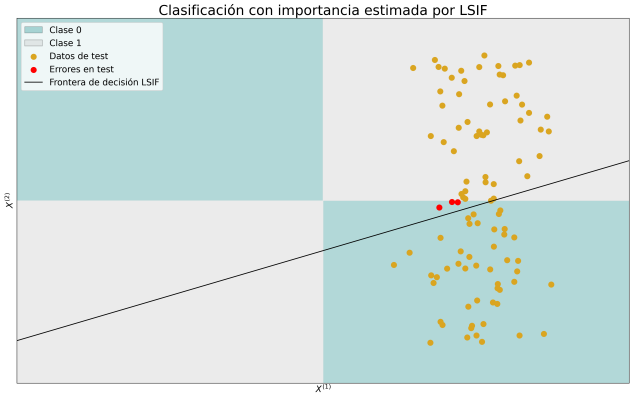
(b) Método de KMM.



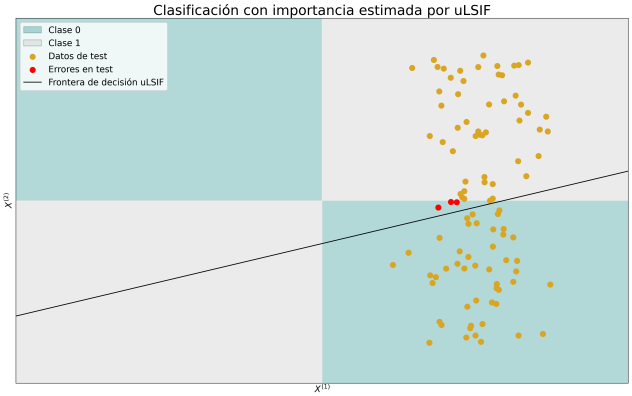
(c) Método de regresión logística.



(d) Procedimiento de KLIEP.



(e) Método de LSIF.



(f) Método de uLSIF.

Figura 4.5: Rectas de clasificación obtenidas con los métodos de estimación de la importancia.

Con estas representaciones se puede comprobar que todos los métodos proporcionan buenos resultados tal como se esperaba. También se puede observar que la recta utilizada como separación varía ligeramente en función del método. Por este motivo, se pretende hacer una comparación más exhaustiva de los resultados. Para hacerlo, se comienza por una representación gráfica conjunta en donde se representan todas las rectas obtenidas (Figura 4.5) en un solo gráfico (Figura 4.6).

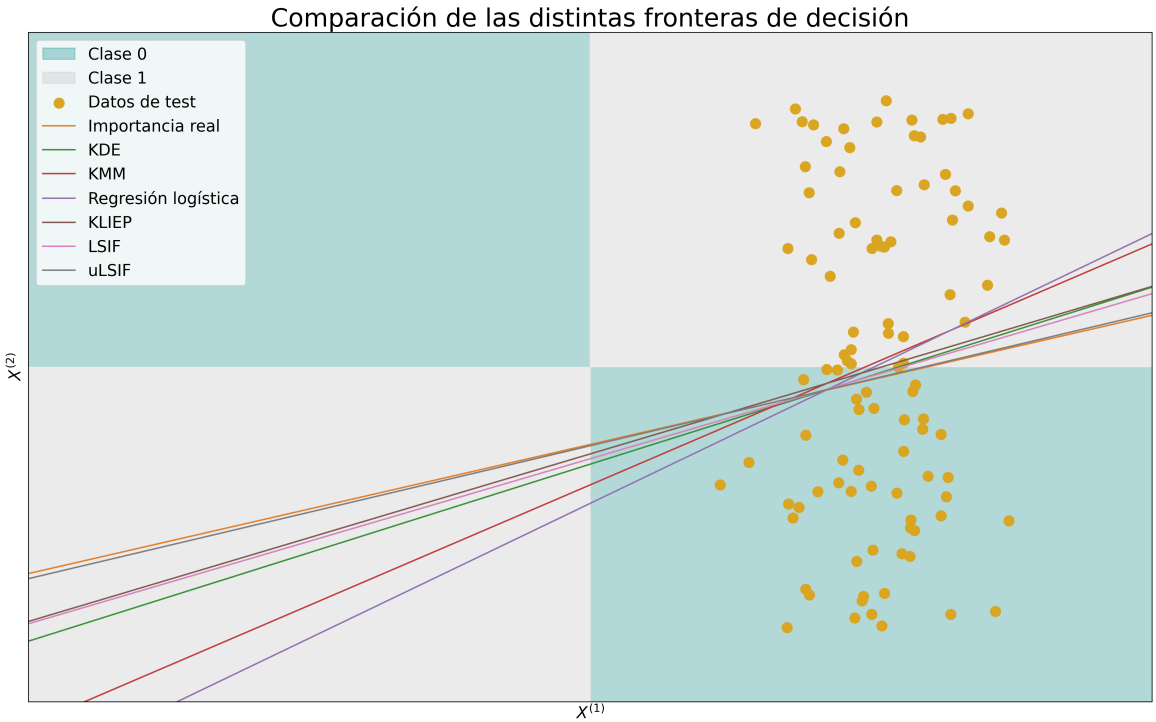


Figura 4.6: Representación conjunta de las rectas de clasificación obtenidas por los diferentes métodos de estimación de la importancia.

Se puede comprobar que todos los métodos hacen una división muy similar. Con esta representación es bastante dificil hacer una valoración de cuál es el método que se comporta mejor con unos datos generados de esta forma. Además, para hacerlo sería más adecuado utilizar más de un conjunto de datos para conocer el comportamiento de una forma más general, es decir, hacer múltiples repeticiones.

4.4. Comparación de los métodos de estimación de la importancia

Con el objetivo de conseguir una comparación más precisa y adecuada de los métodos de estimación de la importancia y comprobar si existen diferencias entre ellos se propone un nuevo experimento. Este consiste en crear 100 conjuntos de datos con las distribuciones de entrenamiento y test explicadas previamente en la Sección 4.1. Para cada conjunto de datos se crea un clasificador con los métodos de estimación de la importancia y también con la importancia real para así comprobar si existe una diferencia significativa entre ellos. Además, también se añade un clasificador sin importancia para evaluar las diferencias que se producen al no utilizar la importancia. Como medida de comparación entre todos estos clasificadores se utiliza el error de clasificación, explicado previamente en la Sección 1.1.3.3.

Si se realiza este experimento y se representan los errores de los clasificadores con los distintos conjuntos de datos se obtiene lo siguiente (Figura 4.7):

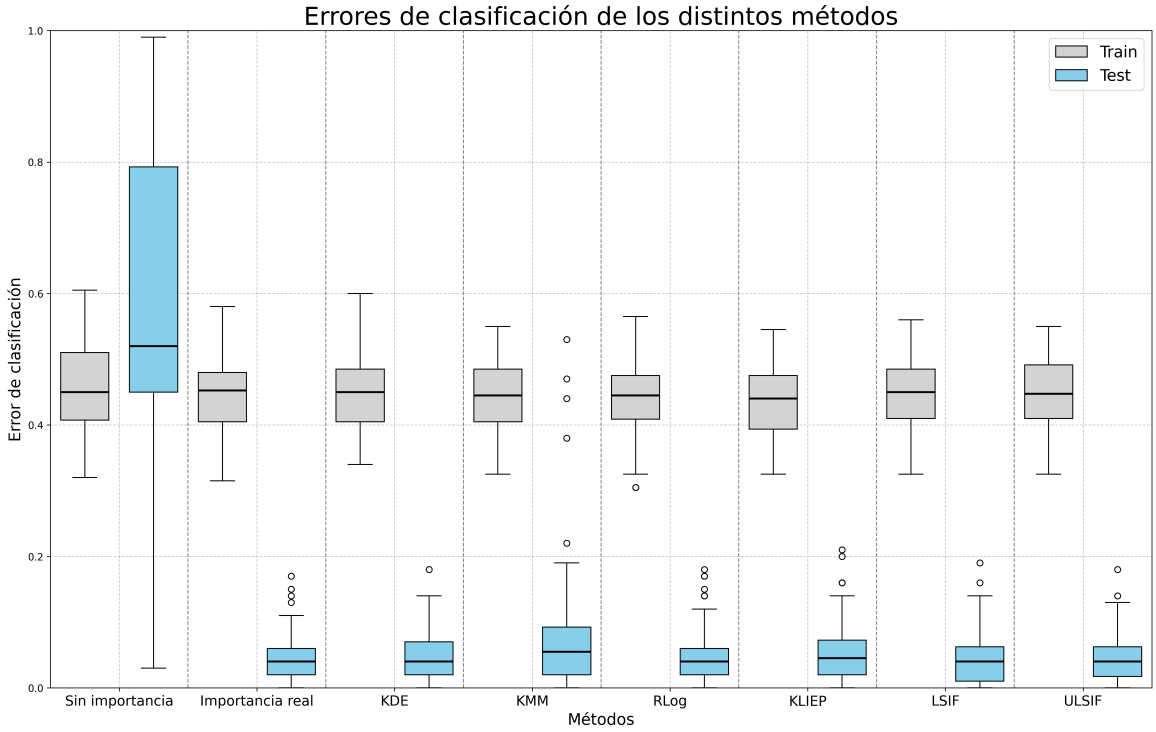


Figura 4.7: Errores de clasificación de los métodos en los conjuntos de entrenamiento y test.

En la Figura 4.7 se representa en color gris (■) los errores sobre el conjunto de entrenamiento y en azul (■) los errores en el conjunto de test. Se puede ver claramente que todos los métodos tienen malos resultados sobre los conjuntos de entrenamiento con errores de clasificación de aproximadamente el 0.45. Aunque no se ha mencionado previamente, esto es algo razonable porque, como el interés se centra aproximar distribuciones de test en lugar de entrenamiento, los clasificadores obtenidos pueden no generalizar bien en esas regiones.

Por otro lado, en los conjuntos de test, se aprecia que los métodos que utilizan la importancia tienen resultados bastante mejores. Además, el clasificador sin importancia, el que aparece más a la izquierda en la Figura 4.7, es verdaderamente inestable puesto que, con conjuntos de datos generados de la misma forma, sus errores varían enormemente, tomando valores cercanos al 0 y en otros conjuntos cercanos a 1.

Como el clasificador sin importancia toma valores en un rango muy diferente y con el objetivo de apreciar mejor las diferencias que existen en los métodos que utilizan la importancia se crea una nueva representación (Figura 4.8) en la que se elimina el clasificador sin importancia y también los resultados sobre el conjunto de entrenamiento al no ser de interés. Cabe destacar que el rango del error de clasificación (eje Y) se reduce respecto al utilizado en la Figura 4.7.

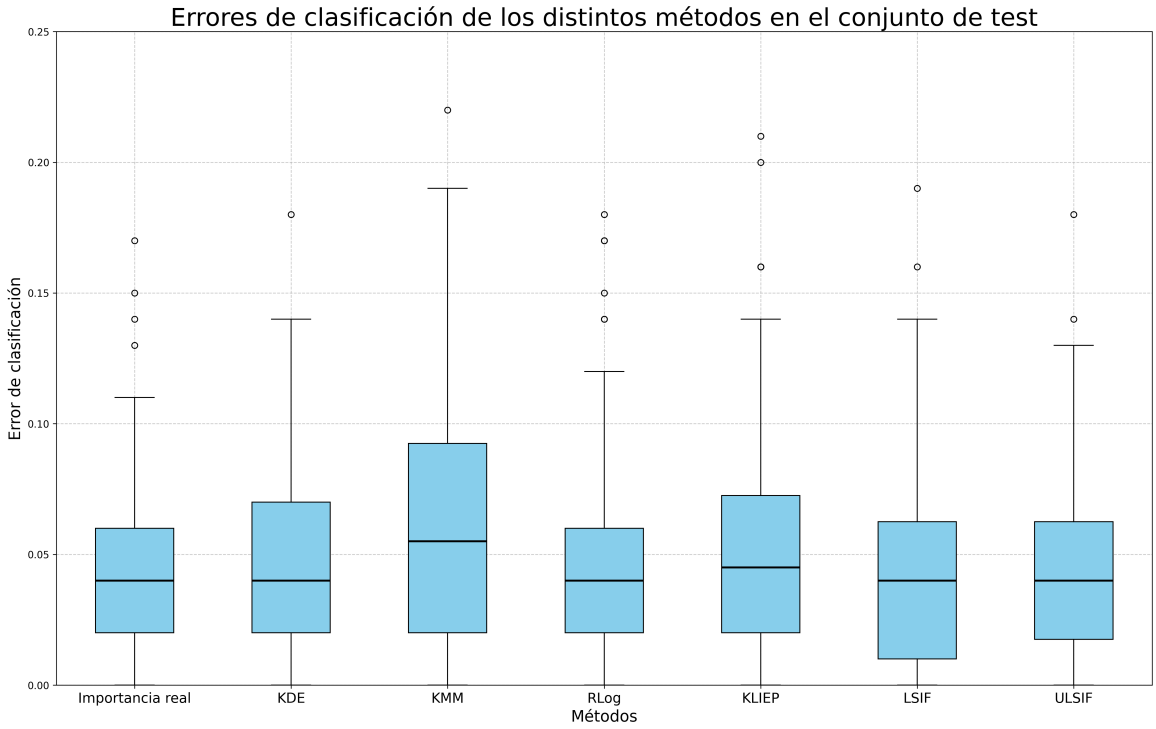


Figura 4.8: Errores de clasificación sobre el conjunto de test de los diferentes métodos de estimación de la importancia.

Se observa como todos los métodos de estimación tienen resultados parecidos a la importancia real. Sin embargo, hay que destacar el resultado de LSIF (Sección 3.5) ya que parece tener incluso mejores resultados que la importancia real sobre estos conjuntos de datos.

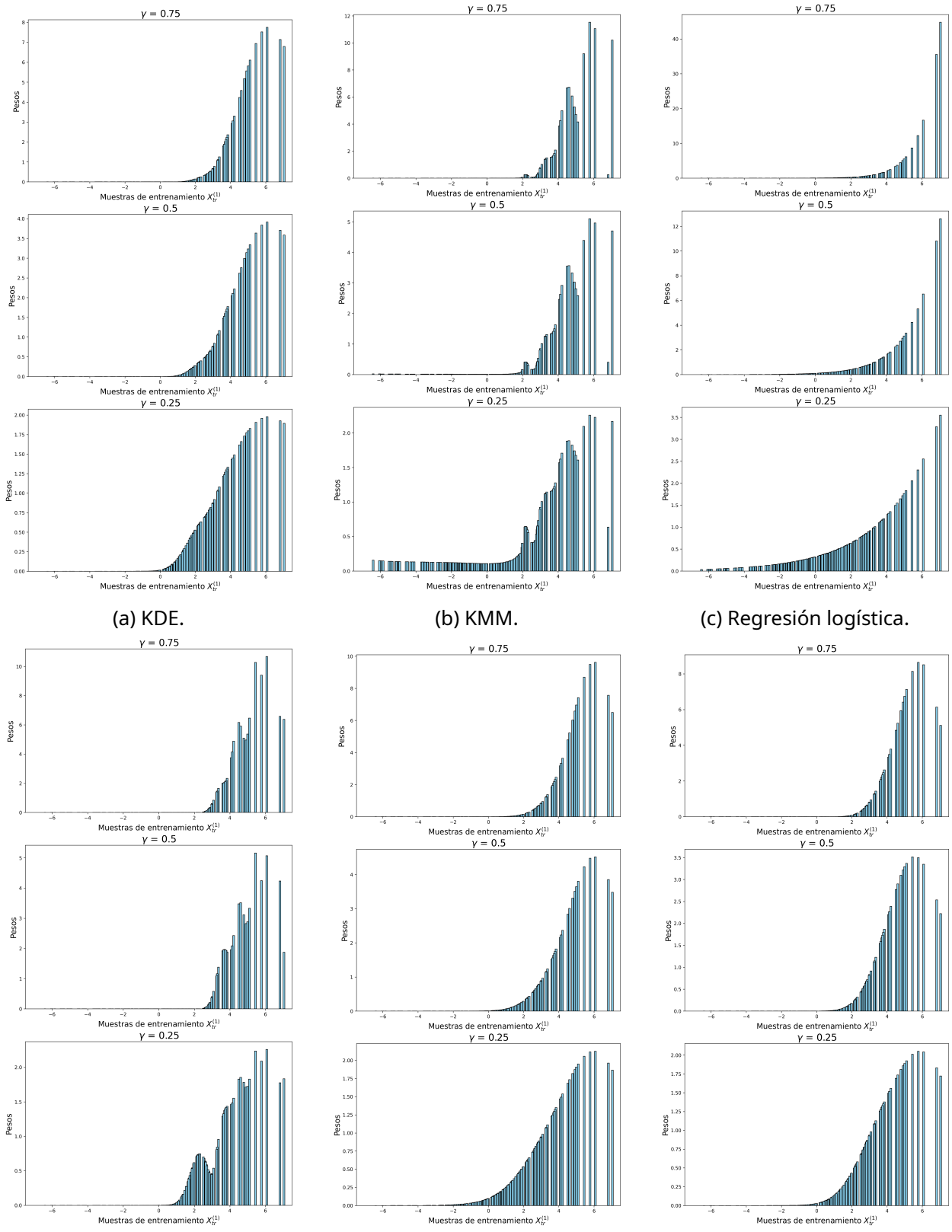
Por otra parte, también se puede ver que KMM parece ser el método que proporciona una estimación más inestable en este caso al cometer errores cercanos a 0.5 en algunos conjuntos. Este valor es muy elevado, el motivo puede ser que en esos casos no ha sido posible encontrar la solución óptima para el problema de optimización (Ecuación (3.14)) haciendo que el resultado sea el mismo que produciría un clasificador sin importancia.

4.5. Efecto de aplanar los pesos

Una vez se han comparado los métodos de estimación se quiere evaluar el efecto que tiene aplanar sus pesos, es decir, w^{γ}. Para hacerlo, se propone un nuevo experimento en el que se utilizan los mismos datos de las Secciones 4.2 y 4.3 pero con los siguientes valores de γ:

- $\gamma = 0{,}75$
- $\gamma = 0{,}5$
- $\gamma = 0{,}25$

En la Figura 4.9 se representan los pesos de los distintos métodos en función del valor de γ. Además, también hay que recordar que se conoce el resultado cuando $\gamma = 1$, los pesos aplicados en la Sección 4.3 y cuando $\gamma = 0$ que es el caso donde no se utiliza la importancia.



(a) KDE. (b) KMM. (c) Regresión logística.

(d) KLIEP. (e) LSIF. (f) uLSIF.

Figura 4.9: Efecto de aplanar el peso de la importancia en los métodos de estimación.

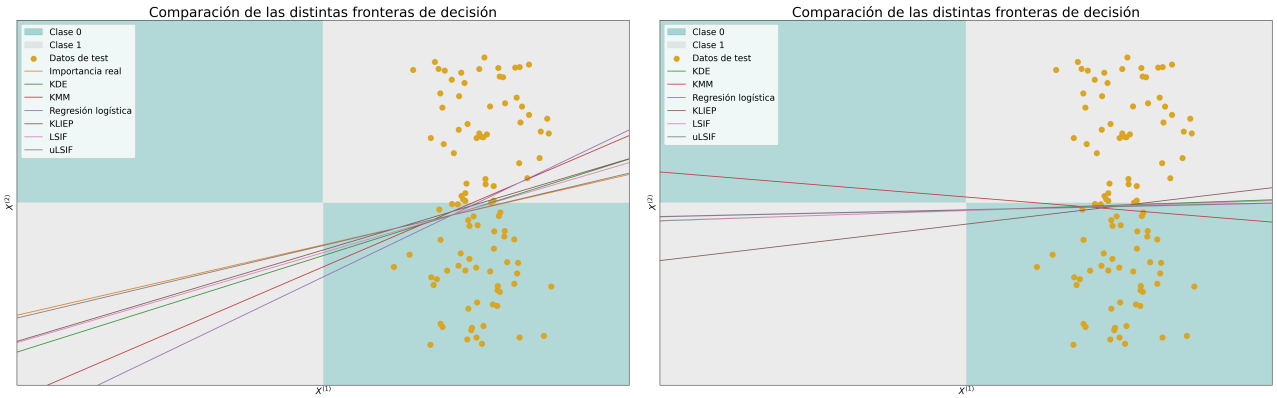
Una vez se observa la diferencia en los pesos en función del valor de γ se quiere comprobar si este efecto tiene un impacto en los resultados obtenidos con los clasificadores. Para ello, se crea un clasificador con cada uno de ellos y se evalúa el error de clasificación sobre un conjunto de test. Los resultados obtenidos se muestran en la Tabla 4.1.

γ	KDE	KMM	Regresión logística	KLIEP	LSIF	uLSIF
1	0.03	0.05	0.05	0.05	0.03	0.03
0.75	0.03	0.05	0.05	0.03	0.03	0.03
0.5	0.02	0.03	0.03	0.04	0.02	0.03
0.25	0.02	0.02	0.02	0.02	0.02	0.02

Tabla 4.1: Errores de clasificación sobre el conjunto de test para distintos valores de γ.

En la Tabla 4.1 se observa que el mejor resultado en todos los casos se obtiene con un $\gamma = 0{,}25$. Esto puede parecer sorprendente ya que proporciona pesos a muestras de entrenamiento alejadas de las de test. Esto se puede explicar por la alta variabilidad de los pesos para valores de γ mayores. Por ejemplo, con $\gamma = 0{,}75$ los pesos en el caso de la regresión logística (Figura 4.9c) varían entre 0 y 40, y para $\gamma = 1$ la variación es de 0 a 160 (resultados de la Sección 4.3). Esta alta variabilidad puede hacer que el modelo sea más sensible a las muestras con pesos extremadamente altos produciendo un sobreajuste lo que reduce la capacidad de generalización. Sin embargo, en este caso, como las muestras con pesos más elevados son las de mayor utilidad para predecir el conjunto de test, aunque el sobreajuste sea menor la diferencia en resultados es pequeña.

Para ilustrar la diferencia que se produce en los resultados en esta situación se representan las rectas de clasificación obtenidas para $\gamma = 0{,}25$, el mejor resultado, y para $\gamma = 1$ (Figura 4.6) en la Figura 4.10.



(a) Representación de la recta de clasificación obtenida con $\gamma = 1$.

(b) Representación de la recta de clasificación obtenida con $\gamma = 0{,}25$.

Figura 4.10: Representación conjunta de las rectas de clasificación obtenidas por los diferentes métodos de estimación de la importancia.

En la Figura 4.10 se comprueba que todos los métodos producen unas rectas de separación más horizontales al utilizar $\gamma = 0{,}25$ lo que explica la mejora en resultados ya que, en este problema, las clases en el conjunto de test se pueden separar perfectamente con una recta horizontal.

4.6. Modificación de la distribución de una de las covariables

Una vez se han comparado los distintos métodos bajo una situación en donde tenemos *covariate shift* se quiere comprobar que ocurre si se utilizan estos métodos en una situación donde las distribuciones de entrenamiento y test sí que son iguales.

Para ello, se propone un nuevo experimento en donde se modifica ligeramente la distribución de la variable $X_{tr}^{(1)}$ hasta llegar a hacer que las distribuciones de entrenamiento y test coincidan. Por otro lado, se decide no modificar las distribuciones relacionadas con $X^{(2)}$, es decir, que $X_{tr}^{(2)} \sim U(-1,1)$ y $X_{te}^{(2)} \sim U(-1,1)$. Con esta idea, se propone evaluar el resultado de los distintos métodos (sin importancia, con importancia y métodos de estimación) generando 10 conjuntos de datos con las siguientes distribuciones para la variable $X^{(1)}$:

(a) Misma situación que la utilizada previamente en la Sección 4.1

$$X_{tr}^{(1)} \sim N(0;3); \quad X_{te}^{(1)} \sim N(5;1)$$

(b) Modificación 1

$$X_{tr}^{(1)} \sim N(1{,}25;2{,}5); \quad X_{te}^{(1)} \sim N(5;1)$$

(c) Modificación 2

$$X_{tr}^{(1)} \sim N(2{,}5;2); \quad X_{te}^{(1)} \sim N(5;1)$$

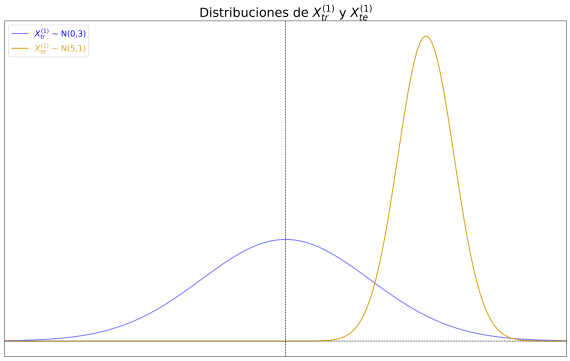
(d) Modificación 3

$$X_{tr}^{(1)} \sim N(3{,}75;1{,}5); \quad X_{te}^{(1)} \sim N(5;1)$$

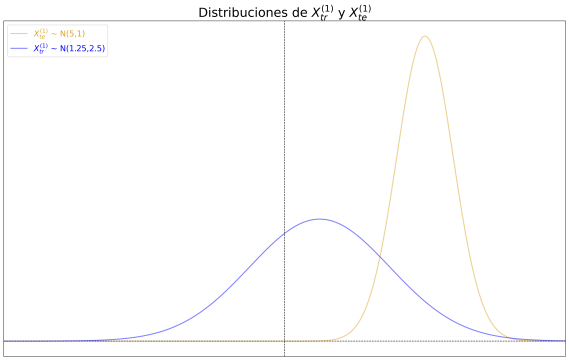
(e) Distribuciones coinciden completamente

$$X_{tr}^{(1)} \sim N(5;1); \quad X_{te}^{(1)} \sim N(5;1)$$

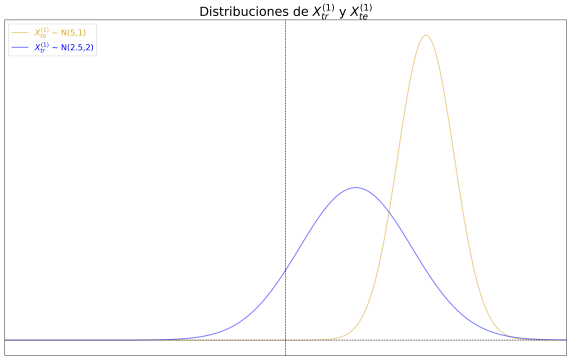
Con el objetivo de expresar de una forma más clara las distribuciones que se quieren utilizar y sus diferencias con la utilizada sobre el conjunto de test se representan todas las situaciones propuestas en la Figura 4.11.



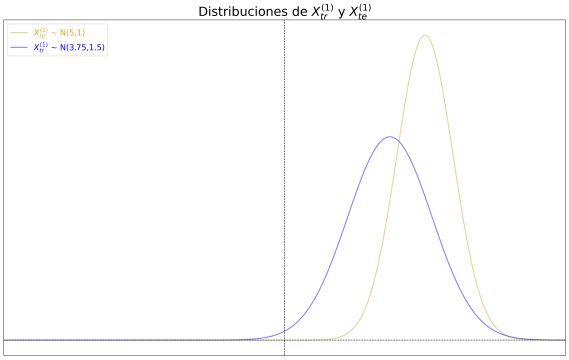
(a) $X_{tr}^{(1)} \sim N(0;3)$; $X_{te}^{(1)} \sim N(5;1)$.



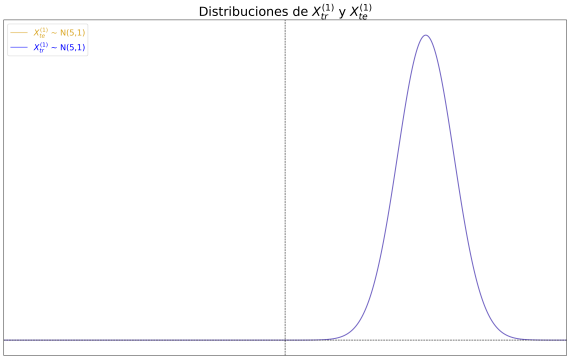
(b) $X_{tr}^{(1)} \sim N(1{,}25;2{,}5)$; $X_{te}^{(1)} \sim N(5;1)$



(c) $X_{tr}^{(1)} \sim N(2{,}5;2)$; $X_{te}^{(1)} \sim N(5;1)$



(d) $X_{tr}^{(1)} \sim N(3{,}75;1{,}5)$; $X_{te}^{(1)} \sim N(5;1)$



(e) $X_{tr}^{(1)} \sim N(5{,}1)$; $X_{te}^{(1)} \sim N(5;1)$

Figura 4.11: Distintas distribuciones utilizadas para comprobar el funcionamiento de los métodos.

Conociendo las distribuciones utilizadas y la distribución de las clases, es de esperar que el efecto de usar la importancia sea muy pequeño a partir de la situación (c). El motivo es que, aunque las distribuciones son distintas, la mayoría de puntos que se utilizarán para entrenar en ellas tendrán la característica de que $X^{(1)} > 0$. Esto hace que, aunque las distribuciones de entrenamiento y test sean diferentes, la distribución de clases en ellas sea la misma. Por tanto, a partir esta situación el error que cometen los métodos debería ser bajo excepto en algunas situaciones atípicas. Por otro lado, el resultado de la situación (a) debe ser similar a la Figura 4.7. Una vez explicado lo que se espera que ocurra teóricamente, se procede a hacer este experimento y se representan los errores obtenidos sobre los conjuntos de test en la Figura 4.12

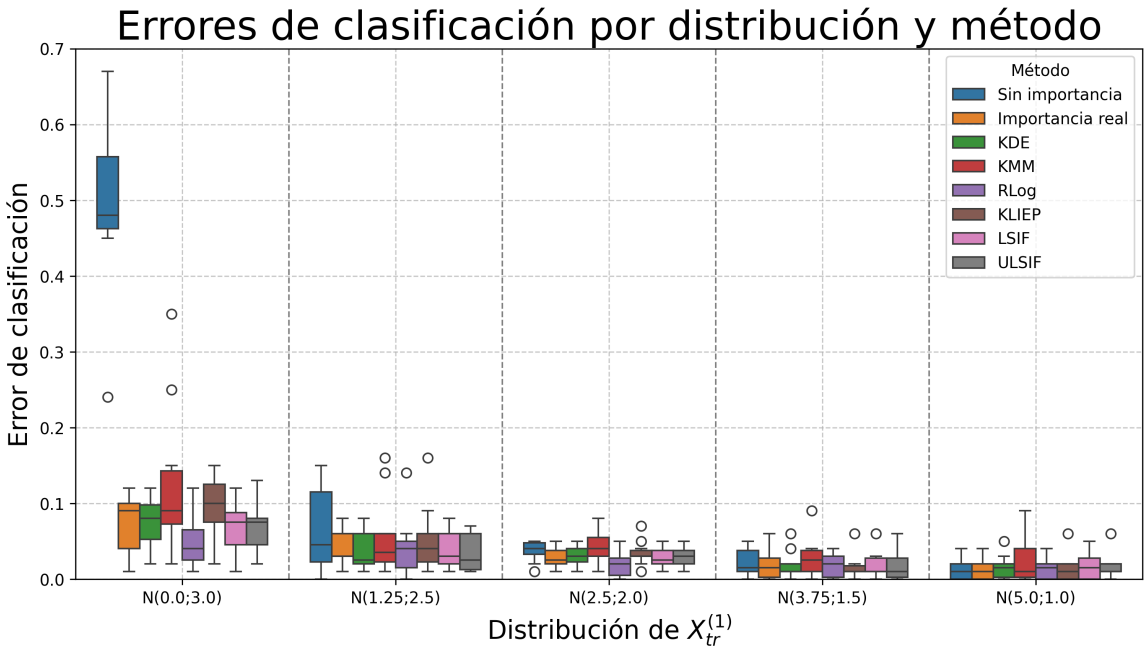


Figura 4.12: Errores de clasificación al variar la distribución de $X_{tr}^{(1)}$.

En la Figura 4.12 se puede ver que ocurre lo esperado, a partir de la situación (c), donde $X_{tr}^{(1)} \sim N(2{,}5; 2)$, los resultados de los métodos son casi idénticos. Por otro lado, al fijarse únicamente en los métodos donde se utiliza la importancia se observa que, a partir de la situación (b) en la que $X_{tr}^{(1)} \sim N(1{,}25; 2{,}5)$, cometen un error menor al 0.05 en la mayoría de los casos incluso llegando a 0 en algunos. Para poder apreciar mejor las diferencias que existen entre ellos se ha decidido crear, a partir de la Figura 4.12, una representación de las situaciones (c), (d) y (e) en la que se reduce el rango del eje Y para así poder observar mejor las diferenciar entre los métodos (Figura 4.13).

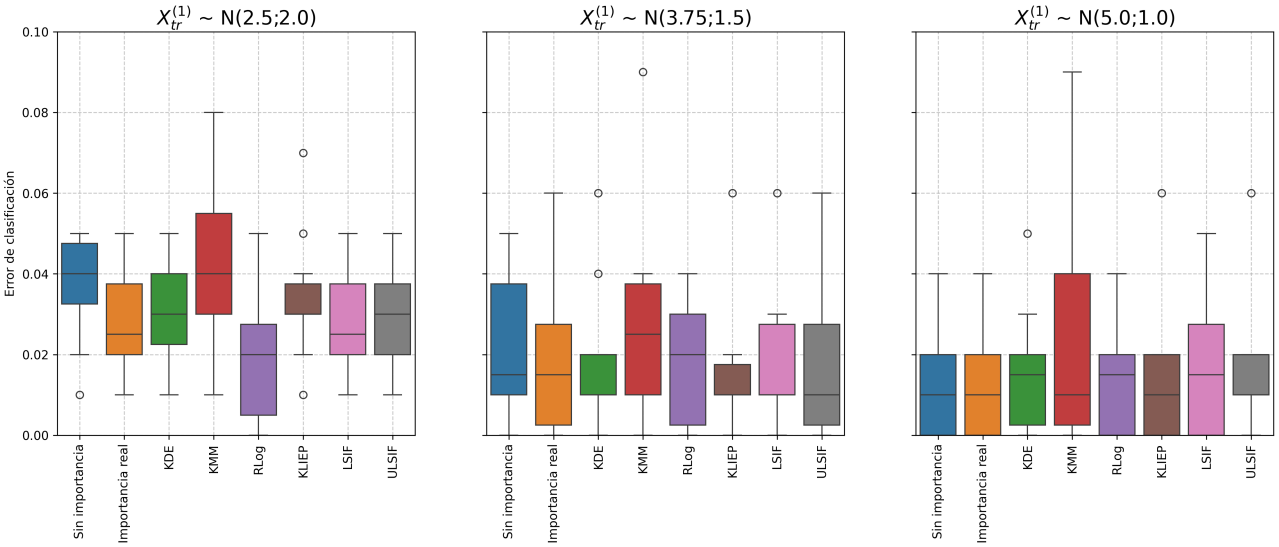


Figura 4.13: Errores de clasificación al variar la distribución de $X_{tr}^{(1)}$.

En la Figura 4.13 se puede ver que los métodos que utilizan la importancia tienen resultados muy similares a aquel que no la utiliza. Esto es algo esperable ya que, aunque las distribuciones son distintas, la diferencia entre ellas no es lo suficientemente grande como para que existan diferencias signficativas. De todas formas, se puede observar que, aunque los resultados son parecidos, el único caso en el que la importancia real (■) coincide exactamente con el método que no utiliza la importancia (■) es en la situación (e) que es en la que las distribuciones coinciden completamente.

5. Conclusiones

En este trabajo se ha analizado lo que ocurre cuando las distribuciones marginales de los datos de entrenamiento y test no coinciden pero las distribuciones condicionales sí lo hacen, lo que se conoce como *covariate shift*. En esta situación, se ha explicado que es necesario utilizar unos pesos para las muestras del conjunto de entrenamiento para corregir esta diferencia entre las distribuciones. Estos pesos se denominan como importancia, pero, en la práctica, no son conocidos y es por ello por lo que existen distintos métodos para estimarla.

Estos métodos de estimación siguen dos enfoques distintos, por un lado, algunos estiman las funciones de densidad de las distribuciones para posteriormente calcular la importancia, como es el caso de KDE, y otros intentan obtener la estimación directamente, mediante un clasificador, como es el caso de la regresión logística o, resolviendo un problema de optimización, como es el caso de KMM, KLIEP, LSIF y uLSIF.

Se ha experimentado para comprobar el efecto de estos métodos en la práctica. Para ello, se ha generado un conjunto de datos para así conocer las distribuciones que permiten calcular la importancia real y comprobar los resultados de los distintos métodos. En esta situación, se ha comprobado que, aunque algunos son algo inestables, los distintos métodos son capaces de obtener buenas estimaciones. Además, también se ha podido comprobar que utilizar una clasificación sin emplear la importancia no es capaz de generalizar bien en los datos del conjunto de test lo que permite apreciar la diferencia que se produce al utilizar la importancia.

Por otro lado, también se ha evaluado el efecto de aplicar la importancia de las muestras de entrenamiento en una situación donde las distribuciones marginales de entrenamiento y test coinciden. En este caso, en el que no sería necesario utilizar la importancia, se ha podido observar que los resultados obtenidos son muy similares a los que se tienen sin utilizarla. Esto es un resultado de especial interés ya que, en la práctica, puede ser difícil conocer si estas distribuciones coinciden o no. De esta forma, al saber que utilizar la importancia no tiene un efecto negativo en los resultados, su uso en un ámbito real no se ve perjudicado.

Por último, se ha comprobado el efecto que tiene aplanar el peso de las muestras de entrenamiento lo que hace que la varianza de las estimaciones se reduzca. En este caso, se ha podido ver que los resultados son mejores que al no aplanar la importancia. Esto se debe a que la varianza de las estimaciones originales llega a ser muy alta lo que provoca un sobreajuste en los clasificadores al centrarse demasiado en algunas de las muestras de entrenamiento. Esta situación hace que la capacidad de generalización se vea afectada y, por tanto, explica que los resultados sobre el conjunto de test sean ligeramente peores.

5.1. Líneas de trabajo futuras

En este trabajo se ha comprobado lo que ocurre cuando las distribuciones marginales de entrenamiento y test no coinciden y también cuando sí lo hacen. Sin embargo, una posible línea de investigación futura podría centrarse en el estudio de lo que sucede cuando los soportes de las distribuciones marginales no coinciden. El motivo es que, en este caso, la importancia real de las muestras de entrenamiento sería 0, ya que $P_{\text{te}}(\boldsymbol{x}_{\text{tr}}) = 0$, lo que resultaría en un clasificador completamente aleatorio. Esto se debe a que si todas las muestras de entrenamiento tienen un peso de 0, el clasificador no puede aprender nada de ninguna de ellas y, por tanto, la predicción sobre el conjunto de test será completamente aleatoria.

Además, también sería interesante comprobar lo que ocurre cuando la importancia de alguna muestra es mucho más grande que el resto. Normalmente esta situación se produce cuando se tiene una muestra de entrenamiento con una probabilidad muy baja de ocurrencia $(P_{\text{tr}}(\boldsymbol{x}_{\text{tr}}) \approx 0)$ pero la probabilidad de ocurrencia en el conjunto de test es alta $(P_{\text{te}}(\boldsymbol{x}_{\text{tr}}) \gg 0)$. El problema es que, en este caso, el clasificador se centrará en exceso en esta muestra produciendo sobreajuste lo no es adecuado y, por tanto, se espera que se tengan malos resultados.

Otra posible línea de investigación sería aplicar estos métodos de estimación en problemas reales. En este caso, se ha decidido no realizar esta tarea puesto que, en una situación real, no se conoce la importancia real de las muestras y, por tanto, sería imposible comprobar que tan buenas son las estimaciones conseguidas. Además, también es de esperar que los resultados no sean tan claros como lo pueden ser en los experimentos realizados debido a que, en un problema real, donde las distribuciones pueden ser mucho más complejas se puede dar la situación de que, aunque con unas buenas estimaciones para la importancia, los resultados de la clasificación no sean buenos por otros factores.

Bibliografía

[1] S. Shalev-Shwartz y S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[2] M. Sugiyama y M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[3] F. F. Rebollo y D. Barrojo, «Aprendizaje por Refuerzo,» *Aprendizaje Automático, Departamento de Informática, Escuela Politécnica Superior*, 2009.

[4] J. I. Segovia-Martín, S. Mazuelas y A. Liu, «Double-weighting for covariate shift adaptation,» *International Conference on Machine Learning*, 2023.

[5] M. P. Deisenroth, A. A. Faisal y C. S. Ong, *Mathematics for machine learning*. Cambridge University Press, 2020.

[6] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. .ºReilly Media, Inc.", 2022.

[7] C. M. Bishop y N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

[8] C. Huyen, *Designing machine learning systems*. .ºReilly Media, Inc.", 2022.

[9] S. Garg, Y. Wu, S. Balakrishnan y Z. Lipton, «A Unified View of Label Shift Estimation,» *Advances in Neural Information Processing Systems*, 2020.

[10] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama y G. Zhang, «Learning under Concept Drift: A Review,» *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[11] V. Vapnik, «Principles of risk minimization for learning theory,» *Proceedings of the 5th International Conference on Neural Information Processing Systems*, 1991.

[12] M. Kimura y H. Hino, *A Short Survey on Importance Weighting for Machine Learning*, 2024.

[13] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015, págs. 143-144.

[14] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018, págs. 47-48.

[15] J. Wen, C.-N. Yu y R. Greiner, «Robust Learning under Uncertain Test Distributions: Relating Covariate Shift to Model Misspecification,» *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[16] Z. Wang y D. W. Scott, «Nonparametric density estimation for high-dimensional data—Algorithms and applications,» *WIREs Computational Statistics*, 2019.

[17] J. Schusterbauer, *Intuitive Explanation of the Kullback-Leibler Divergence - Johannes Schusterbauer*.

[18] M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau y M. Kawanabe, «Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation,» *Advances in Neural Information Processing Systems*, 2007.

[19] T. Kanamori, S. Hido y M. Sugiyama, «A Least-squares Approach to Direct Importance Estimation,» *Journal of Machine Learning Research*, 2009.

A. Códigos

En este capítulo aparecen los códigos que se han desarrollado para utilizar los distintos métodos de estimación de la importancia del Capítulo 4. Es importante mencionar que solo se incluye el código que permite generar los datos y el que permite obtener las estimaciones de la importancia (los valores que aparecen son los utilizados para el experimento de las Secciones 4.2 y 4.3)

```
from adapt.instance_based import KLIEP, ULSIF
import cvxpy as cp
from matplotlib.patches import Patch, Rectangle
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.optimize import minimize
from scipy.stats import norm, gaussian_kde
from sklearn.linear_model import LogisticRegression
from sklearn.metrics.pairwise import rbf_kernel
from sklearn.neighbors import NearestNeighbors
```

Código A.1: Paquetes necesarios

A.1. Generación de datos

```
# Función para generar las distribuciones U(-1, 1)
def function(x):
    return np.random.uniform(-1, 1, size=len(x))

# Funcion para generar las variables X1, X2 y las etiquetas de clase
def generate_data(media_train, desviacion_train, n_elem_train,
                  media_test, desviacion_test, n_elem_test):
    # Datos de entrenamiento y test de la variable X1
    X1_train = np.random.normal(media_train, desviacion_train, n_elem_train)
    X1_test = np.random.normal(media_test, desviacion_test, n_elem_test)

    # Datos de entrenamiento y test de la variable X2
    X2_train = function(X1_train)
    X2_test = function(X1_test)

    # Creamos las etiquetas de nuestras clases
    clases_train = (X1_train * X2_train > 0).astype(int)
    clases_test = (X1_test * X2_test > 0).astype(int)

    return X1_train, X2_train, X1_test, X2_test, clases_train, clases_test
```

Código A.2: Código que permite generar los datos de entrenamiento y test y sus clases.

A.2. Clasificador sin importancia

```
clasificador_sinimportancia = LogisticRegression()
# Ajustamos el modelo utilizando los datos de entrenamiento (x,y) y las etiquetas de clase
clasificador_sinimportancia.fit(np.column_stack((X1_train, X2_train)), clases_train)
```

Código A.3: Código para ajustar un clasificador sin importancia (sin pesos para las muestras de entrenamiento).

Para todos los métodos se puede utilizar el clasificador para hacer predicciones de unos puntos. Por ejemplo para obtener la predicción de la clase de unos puntos X1, X2 se utiliza:

“nombre_clasificador.predict(np.column_stack((X1, X2)))“.

A.3. Importancia real

```
# Calculamos los pesos como Ptest(xtrain) / Ptrain(xtrain)
p_test = norm.pdf(X1_train, loc=media_test, scale=desviacion_test)
p_train = norm.pdf(X1_train, loc=media_train, scale=desviacion_train)
weights = p_test / p_train

# Ajustamos el modelo utilizando los datos de entrenamiento (x,y), las etiquetas de clase y
    los pesos
clasificador_importancia = LogisticRegression()
clasificador_importancia.fit(np.column_stack((X1_train, X2_train)), clases_train,
    sample_weight=weights)
```

Código A.4: Código para ajustar un clasificador utilizando la importancia real.

A.4. Métodos de estimación de la importancia

A.4.1. Kernel Density Estimation (KDE)

```
# 1. Calculamos hat_P_train
kde_train = gaussian_kde(X1_train, bw_method='scott') # bw_method="scott"/"silverman"/
    constante
hat_P_train = kde_train.evaluate(X1_train)

# 2. Calculamos el hat_P_test
kde_test = gaussian_kde(X1_test, bw_method='scott') # bw_method="scott"/"silverman"/constante
hat_P_test = kde_test.evaluate(X1_train)

# 3. Estimamos la importancia
hat_w_kde = hat_P_test/hat_P_train

# Ajustamos un clasificador con los pesos calculados (hat_w_kde)
clasificador_kde = LogisticRegression()
clasificador_kde.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_kde)
```

Código A.5: Código para ajustar un clasificador utilizando KDE

A.4.2. Kernel Mean Matching (KMM)

```python
def KMM(X1_train, X1_test):
    n_train = X1_train.shape[0]
    n_test = X1_test.shape[0]
    X = np.concatenate((X1_train, X1_test), axis=0)
    epsilon = 1 - 1 / (np.sqrt(n_train))
    B = 1000
    if X.shape[0] < 50:
        neighbour_ind = X.shape[0] - 2
    else:
        neighbour_ind = 50

    nbrs = NearestNeighbors(n_neighbors=(neighbour_ind + 1), algorithm='ball_tree').fit(X)
    distances, indices = nbrs.kneighbors(X)
    # Calculamos la distancia media a los vecinos más proximos
    sigma = np.average(distances[:, neighbour_ind])
    K = rbf_kernel(X, X, 1 / (2 * sigma ** 2))

    # Variables del problema de optimizacion
    weights = cp.Variable((n_train, 1))
    alpha_ = np.ones((n_test, 1))

    # Funcion Objetivo
    objective = cp.Minimize(cp.quad_form(cp.vstack([weights/n_train, -alpha_/n_test]), cp.
        psd_wrap(K)))
    # Restricciones
    constraints = [
        weights >= np.zeros((n_train, 1)),
        weights <= B * np.ones((n_train, 1)),
        cp.sum(weights) / n_train - 1 <= epsilon,
    ]

    problem = cp.Problem(objective,constraints)
    try:
        problem.solve(solver = cp.GUROBI)
    except cp.error.SolverError:
        try:
            problem.solve(solver = cp.MOSEK)
        except cp.error.SolverError:
            try:
                problem.solve(solver = cp.ECOS, feastol = 1e-6, reltol = 1e-3, abstol = 1e-6)
            except cp.error.SolverError:
                raise ValueError('No se ha encontrado la solución')

    weights = weights.value
    return weights

# Se utiliza la función para calcular los pesos
hat_w_kmm = KMM(X1_train.reshape(-1, 1), X1_test.reshape(-1, 1))
hat_w_kmm = hat_w_kmm.squeeze()

# Ajustamos un clasificador con los pesos calculados (hat_w_kmm)
clasificador_kmm = LogisticRegression()
clasificador_kmm.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_kmm)
```

Código A.6: Código para ajustar un clasificador utilizando KMM

A.4.3. Obtención de pesos por regresión logística

```
# Creamos la variable clasificadora (eta)
eta_train = -np.ones(n_elem_train)
eta_test = np.ones(n_elem_test)

# Juntamos los datos de entrenamiento y test
X = np.concatenate((X1_train, X1_test))
eta = np.concatenate((eta_train, eta_test))

# Creamos un clasificador de regresión logística
clasificador_logistico = LogisticRegression()
clasificador_logistico.fit(X.reshape(-1, 1), eta)

# Calculamos las probabilidades condicionadas p(y|x) (p(eta|x))
p_etatrain_x = clasificador_logistico.predict_proba(X.reshape(-1, 1))[:, 0]
p_etatest_x = clasificador_logistico.predict_proba(X.reshape(-1, 1))[:, 1]

# Calculamos la importancia
hat_w_rlog = (len(X1_train) / len(X1_test)) * (p_etatest_x / p_etatrain_x)
hat_w_rlog = hat_w_rlog[eta == -1] # Solo nos quedamos con los pesos de los datos de
    entrenamiento

# Ajustamos un clasificador con los pesos calculados (hat_w_rlog)
clasificador_rlog = LogisticRegression()
clasificador_rlog.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_rlog)
```

Código A.7: Código para ajustar un clasificador utilizando pesos estimados por regresión logística.

A.4.4. Procedimiento de Kullback-Leibler (KLIEP)

```
kliep = KLIEP(kernel="rbf", sigmas=np.arange(0.1, 10.1, 0.1), max_centers=20, cv=10, max_iter
    =2000, verbose=0)
hat_w_kliep = kliep.fit_weights(X1_train.reshape(-1, 1), X1_test.reshape(-1, 1))

# Ajustamos un clasificador con los pesos calculados (hat_w_kliep)
clasificador_kliep = LogisticRegression()
clasificador_kliep.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_kliep)
```

Código A.8: Código para ajustar un clasificador utilizando KLIEP.

A.4.5. Ajuste de importancia por mínimos cuadrados (LSIF)

```python
cp.settings.EIGVAL_TOL = 1e-4

def gaussian_kernel_matrix(X, centros, sigma):
    X_norm = np.sum(X**2, axis=1, keepdims=True)
    centers_norm = np.sum(centros**2, axis=1)
    dist_sq = X_norm + centers_norm - 2 * X.dot(centros.T)
    return np.exp(-dist_sq / (2 * sigma**2))

def LSIF(X1_train, X1_test, h_kernel=1.0, lambda_pen=0.001):
    n_train = X1_train.shape[0]
    n_test = X1_test.shape[0]

    # Utilizamos como funciones base el kernel gaussiano centrado en las muestras de test
    centers = X1_test
    rho_train = gaussian_kernel_matrix(X1_train, centers, h_kernel)
    rho_test  = gaussian_kernel_matrix(X1_test, centers, h_kernel)

    # Creamos las matrices H y h
    H = (rho_train.T @ rho_train) / n_train
    h = np.sum(rho_test, axis=0) / n_test

    # Aseguramos que el tamaño de los elementos es correcto
    assert rho_train.shape == (n_train, n_test), "Tamaño incorrecto de rho_train"
    assert rho_test.shape == (n_test, n_test), "Tamaño incorrecto de rho_test"
    assert H.shape == (n_test, n_test), "Tamaño incorrecto de la matriz H"
    assert h.shape == (n_test,), "Tamaño incorrecto de la matriz h"

    # Definimos el vector de parametros
    alpha = cp.Variable(n_test)
    # Se formula la función objetivo con una penalización lineal (Ecuacion 3.34)
    objective = cp.Minimize(0.5 * cp.quad_form(alpha, H) - h.T @ alpha + lambda_pen * cp.sum(
        alpha))
    # Sujeto a que los parametros (alpha) >= 0
    constraints = [alpha >= 0]

    # Resolvemos el problema de minimizacion
    problem = cp.Problem(objective, constraints)
    problem.solve()

    # Obtenemos la solución y calculamos los pesos como: w = alpha * rho
    weights = rho_train @ alpha.value
    weights = np.maximum(weights, 0)
    assert weights.shape == (n_train,), "Tamaño incorrecto de weights"

    return weights

# Se utiliza la función para calcular los pesos
hat_w_LSIF = LSIF(X1_train.reshape(-1, 1), X1_test.reshape(-1, 1), h_kernel=1.0, lambda_pen=0)

# Ajustamos un clasificador con los pesos calculados (hat_w_LSIF)
clasificador_lsif = LogisticRegression()
clasificador_lsif.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_LSIF)
```

Código A.9: Código para ajustar un clasificador utilizando LSIF.

A.4.6. Ajuste de importancia por mínimos cuadrados sin restricciones (uLSIF)

```
ulsif = ULSIF(Xt=X1_test, kernel="rbf", lambdas=np.arange(0.1, 10.1, 0.1), verbose=0)
hat_w_ulsif = ulsif.fit_weights(X1_train.reshape(-1, 1), X1_test.reshape(-1, 1))

# Ajustamos un clasificador con los pesos calculados (hat_w_ulsif)
clasificador_ulsif = LogisticRegression()
clasificador_ulsif.fit(np.column_stack((X1_train, X2_train)), clases_train, sample_weight=
    hat_w_ulsif)
```

Código A.10: Código para ajustar un clasificador utilizando uLSIF.