Clasificación Binaria en Problemas Desequilibrados mediante Equivalencia del Cociente de Verosimilitudes

Alexander Benítez Buenache

Tesis depositada en cumplimiento parcial de los requisitos para el grado de Doctor en

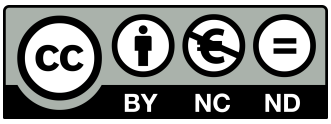
Multimedia y Comunicaciones

Universidad Carlos III de Madrid

Director:

Dr. Aníbal R. Figueiras Vidal

Leganés, Junio de 2021



AGRADECIMIENTOS

Cuando llega el momento de finalizar una etapa como esta es imposible no echar la vista atrás y recordar el camino recorrido. Por ello, quiero dar las gracias a todas y cada una de las personas que tanto me han ayudado durante este período.

Sin duda, las primeras palabras de agradecimiento deben ser para mi Director, el Profesor Dr. Aníbal R. Figueiras Vidal, por brindarme la oportunidad de formar parte de su grupo de investigación y haber podido realizar el trabajo que, finalmente, ha desembocado en el desarrollo de la presente Tesis. Aún recuerdo el día que le dije las palabras "Me gustaría especializarme en Aprendizaje Máquina" a principios de 2016. Desde entonces, he aprovechado cada explicación y cada consejo suyo para mejorar académica, profesional y personalmente.

Seguidamente, me gustaría agradecerle su ayuda a todas las personas con las que he coincidido y colaborado en el laboratorio 2.C.03, más conocido como "La Cueva": Lorena (fácilmente podría rellenar un párrafo entero para agradecerle tanta ayuda y consejo durante todos estos años), Óscar, Simón, Adil, Carlos (y todos sus chistes malos), Javier, Aitor, Estefanía y Pablo. Fueron casi tres años compartiendo con ellos el día a día, pasando allí grandes momentos y con infinitas anécdotas.

También quiero dar las gracias a mis compañeros de GMV, quienes, mientras finalizaba mi Tesis, han hecho que dar ese paso de la Universidad a la Empresa fuese mucho más sencillo. Especialmente, quiero darles las gracias a Antón, Paloma e Inma por los "cafecitos" y por toda su ayuda y apoyo desde el primer día en la empresa.

Por último, quiero destacar que durante la realización de una Tesis Doctoral es fundamental la gestión anímica y emocional, mayor incluso en tiempos de pandemia. Por ello, quiero concluir dando las gracias a aquellas personas que tanto me han ayudado durante estos años: a Cris (por su infinita paciencia y apoyo incondicional), a mis padres (por su esfuerzo durante tantos años para brindarme todos los medios necesarios para mi desarrollo académico) y al resto de mi familia y amigos (por cada momento de desconexión cuando más se necesitaba).

Gracias a todos.

CONTENIDOS PUBLICADOS Y PRESENTADOS

- Roca-Sotelo, S., Benítez-Buenache, A., and Figueiras-Vidal, A. R. (2016). An exploratory evaluation of Bayesian principled approaches to solve imbalanced problems. In *Workshop Advances and Applications of Data Science and Engineering*, pages 107−112. Madrid (Spain).

  - Rol: Co-autor.
  - Fuente incluida parcialmente en el Capítulo 3 de la Tesis.
  - El material de esta fuente incluido en la Tesis no está señalado por medios tipográficos ni referencias.

- Benítez-Buenache, A., and Figueiras-Vidal, A. R. (2017). *Likelihood-Ratio Equivalent Classification Problems and Principled Re-Balancing Techniques* (Master Thesis). Universidad Carlos III de Madrid, Leganés, Madrid (Spain).

  - Rol: Autor principal
  - Fuente incluida parcialmente en el Capítulo 3 de la Tesis.
  - El material de esta fuente incluido en la Tesis no está señalado por medios tipográficos ni referencias.

- Benítez-Buenache, A., Álvarez-Pérez, L., Mathews, V. J., and Figueiras-Vidal, A. R. (2019). Likelihood ratio equivalence and imbalanced binary classification. *Expert Systems with Applications*, 130:84−96. Disponible en: https://doi.org/10.1016/j.eswa.2019.03.050

  - Rol: Autor principal
  - Fuente incluida totalmente en el Capítulo 3 de la Tesis.
  - Todo material de esta fuente incluido en la Tesis está señalado por medios tipográficos y una referencia explícita: [Benítez-Buenache et al., 2019]

- Benítez-Buenache, A., Álvarez-Pérez, L., Mathews, V. J., and Figueiras-Vidal, A. R. (2020). Corrigendum to "Likelihood ratio equivalence and imbalanced binary classification" [Expert Systems with Applications, Volume 130 (2019), Pages 84–96]. *Expert Systems with Applications*, 146:113299. Disponible en: https://doi.org/10.1016/j.eswa.2020.113299

  - Rol: Autor principal
  - Fuente incluida totalmente en el Capítulo 3 de la Tesis.
  - Todo material de esta fuente incluido en la Tesis está señalado por medios tipográficos y una referencia explícita: [Benítez-Buenache et al., 2020]

- Benítez-Buenache, A., Álvarez-Pérez, L., and Figueiras-Vidal, A. R. (2021). On the design of Bayesian principled algorithms for imbalanced classification. Aceptado para su publicación en *Knowledge-Based Systems*. Disponible en: https://doi.org/10.1016/j.knosys.2021.106969

  - Rol: Autor principal
  - Fuente incluida totalmente en el Capítulo 4 de la Tesis.
  - Todo material de esta fuente incluido en la Tesis está señalado por medios tipográficos y una referencia explícita: [Benítez-Buenache et al., 2021]

RESUMEN

Los Problemas Singulares son aquellos cuyas características pueden comprometer el correcto funcionamiento de máquinas discriminativas convencionales, obteniendo resultados poco satisfactorios. Entre ellos destacan los problemas de clasificación desequilibrada, aquellos en los que existen grandes diferencias en las poblaciones de las clases o/y la política de costes penaliza en mayor medida la elección de determinadas clases, sesgando la salida de la máquina en favor de las clases predominantes. Por ello, se precisa la aplicación de métodos específicos que compensen el desequilibrio existente, permitiendo la detección de las clases minoritarias.

Particularizando para el caso binario, se lleva a cabo un estudio del estado del arte de los métodos de re-equilibrado existentes. La mayoría de las técnicas propuestas son puramente empíricas, sin un análisis completo de las implicaciones estadísticas que tiene su aplicación. A pesar de que su uso puede ofrecer buenos resultados bajo determinadas condiciones, cualquier cambio en dichas condiciones puede producir una degradación en las prestaciones. Por ello, se presenta una metodología fundamentada en la teoría estadística bayesiana con el objetivo de construir soluciones robustas. Esta metodología se basa en el principio de invarianza del cociente de verosimilitudes, estableciendo dos condiciones suficientes y necesarias: el uso de divergencias de Bregman como coste subrogado y métodos de re-equilibrado estadísticamente neutrales. Además, se proponen procedimientos fundamentados de clasificación en dos pasos y se describe detalladamente un proceso de diseño re-equilibrado basado en la combinación de métodos. Diversos experimentos avalan la metodología, estudiando sus efectos y limitaciones en problemas reales bajo distintas circunstancias: mayor o menor número de muestras disponibles y presencia de ruido.

Por último, se estudia en mayor profundidad el algoritmo SMOTE, uno de los métodos de re-equilibrado más comunes. Debido a la generación −por medio de los vecinos más próximos− filiforme de muestras, SMOTE presenta dificultades ante problemas de alta dimensionalidad. Por ello, se propone una alternativa, VoluSMOTE, para corregir o atenuar tales efectos por medio de una generación volumétrica.

ABSTRACT

Singular Problems are those whose characteristics compromise the correct operation of conventional discriminative machines, obtaining unsatisfactory results. Among them, imbalanced classification problems stand out, those in which there are large differences in the class populations or/and the cost policy penalizes to a greater extent the choice of certain classes, biasing the machine output in favor of the predominant classes. Therefore, the application of specific methods that compensate the imbalance is required, allowing the detection of the minority classes.

Particularly for the binary case, a state-of-the-art survey of the existing rebalancing methods is carried out. Most of the proposed techniques are purely empirical, without a complete analysis of the statistical implications of their application. Although their use may provide good results under certain conditions, any change in these conditions may lead to a degradation of their performance. Therefore, a principled methodology based on Bayesian statistical theory is presented with the aim of constructing robust solutions. This methodology is based on the likelihood ratio invariance principle, for which two sufficient and necessary conditions are established: the use of Bregman divergences as a surrogate cost and statistically neutral rebalancing methods. In addition, principled two-step classification procedures are proposed and a rebalanced design process based on the combination of methods is described in detail. Several experiments support the methodology, studying its effects and limitations in real problems under different circumstances: larger or smaller number of available samples and presence of noise.

Finally, the SMOTE algorithm, one of the most common rebalancing methods, is studied in more depth. Due to the filiform generation of samples −by means of the nearest neighbors−, SMOTE presents difficulties with high dimensionality problems. Therefore, an alternative, VoluSMOTE, is proposed to correct or mitigate such effects by volumetric generation.

"*La perfeccción es*
una pulida corrección de errores"

Mario Benedetti
1920 - 2009

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo se presentan los conceptos básicos que permiten enmarcar esta Tesis dentro de la disciplina del Aprendizaje Máquina. Para ello, se lleva a cabo una descripción temporal de la evolución de dicho campo, desde sus orígenes hasta el presente, haciendo mayor hincapié en aquellos aspectos que juegan un papel más importante en el desarrollo de la Tesis.

1.1. Origen del Aprendizaje Máquina

La toma de decisiones es algo inherente al ser humano. Durante toda su vida, las personas se ven obligadas a elegir entre distintas opciones en función de un contexto o una situación determinada. Por su parte, el uso de la tecnología busca la automatización de ciertas tareas o actividades para ayudar −o en algunos casos sustituir− a las personas a la hora de llevarlas a cabo. Por tanto, teniendo en cuenta ambas afirmaciones, parece algo lógico que, durante los últimos tiempos, se haya buscado automatizar la toma de decisiones, dando lugar a la aparición del Aprendizaje Máquina (ML, "Machine Learning"), también conocido como Aprendizaje Automático. Es decir, dotar a una máquina de la capacidad de "aprender" a tomar decisiones basadas en la información disponible.

A lo largo de las últimas décadas, la constante y gran evolución tecnológica ha permitido el diseño e implementación de numerosas aplicaciones basadas en el Aprendizaje Máquina. No obstante, su origen se remonta a mitad del siglo XX, cuando comenzó el estudio y diseño de los primeros modelos neuronales de aprendizaje [McCulloch and Pitts, 1943] [Hebb, 1949] basados en el exitoso trabajo realizado por Santiago Ramón y Cajal sobre el flujo de la información en el cerebro y las activaciones neuronales. En 1951, Marvin Minsky creó la SNARC ("Stocastic Neural Analog Reinforcement Calculator"), una red aleatoriamente conectada de 40 sinapsis hebbianas, lo que para muchos fue considerado el primer simulador de redes neuronales, pero que realizaba numerosas suposiciones estadísticas. Sin embargo, sería Frank Rosenblatt quien, basado en la teoría hebbiana, presentó la Regla del Perceptrón [Rosenblatt, 1958], la primera Máquina de Aprendizaje (LM, "Learning Machine") binaria sin la necesidad de llevar a cabo suposiciones estadísticas. El objetivo era entrenar una máquina discriminativa lineal mediante una función de activación dura, es decir, la salida es positiva o negativa en función de la pertenencia del patrón de entrada a una clase u otra, calculando de forma iterativa los pesos de las unidades (neuronas) que forman el Perceptrón.

Como alternativa al Perceptrón, surgiría el ADALINE (ADAptative LINear Element) [Widrow and Hoff, 1960], una red neuronal desarrollada por Bernard Widrow y Ted Hoff para resolver problemas de filtrado. Su principal diferencia respecto al Perceptrón es el uso que se hace de la salida durante el proceso de aprendizaje: mientras el Perceptrón únicamente valora el acierto o desacierto de la salida, el ADALINE tiene en cuenta el grado de error cometido.

Desafortunadamente, la aparición de sistemas adaptativos, junto con las limitaciones asociadas al modelo y las dificultades de extender el aprendizaje a arquitecturas multicapa, motivaron la errónea respuesta de Marvin Minsky y Seymour Papert [Minsky and Papert, 1969]. Su escepticismo acabaría provocando lo que se conoce como el primer "invierno"de la Inteligencia Artificial, un período de varias décadas en el cual se perdió el interés en la materia por parte de la comunidad científica.

No obstante, durante estos años surgió el algoritmo de Retro-Propagación (BP, "Back-Propagation"), un método de optimización basado en el descenso por gradiente mediante la regla de la cadena para la derivación que, con el paso de los años, jugaría un papel muy relevante en el diseño de las redes neuronales. Propuesto de forma teórica por Henry Kelley [Kelley, 1960] y Arthur Bryson [Bryson, 1961], no tuvo mucha aceptación inicialmente. De hecho, no fue hasta casi una década después cuando Seppo Linnainmaa [Linnainmaa, 1970] consiguió implementarlo en su Tesis de Máster. Fue Paul Werbos quien propuso su utilización en Redes Neuronales Profundas (DNNs, "Deep Neural Networks") −arquitecturas con varias capas ocultas− como método de aprendizaje [Werbos, 1974], lo que motivó la publicación de numerosos trabajos donde se aplicaba el algoritmo de Retro-Progragación [Parker, 1982] [LeCun, 1985]. A pesar de ello, su uso no se popularizaría hasta 1986, cuando David Rumelhart, Geoffrey Hinton y Ronald Williams, un grupo de científicos del MIT, definieron el Perceptrón Multi-Capa (MLP, "Multi-Layer Perceptron") [Rumelhart et al., 1986a] [Rumelhart et al., 1986b] tal y como lo conocemos. En la Figura 1.1 se muestra su estructura.

Sin duda, el MLP jugó y juega un papel muy importante dentro de las denominadas Redes Neuronales (NNs, "Neural Networks"). De hecho, a raíz de ello, surgieron distintas aproximaciones, entre las que destacan las Máquinas de Boltzmann (BMs, "Boltzmann Machines") [Hinton and Sejnowski, 1986] −así como su versión restringida, RBMs ("Restricted Boltzmann Machines") [Smolensky, 1986]− y las Redes de Funciones Radiales de Base (RBFNs, "Radial Basis Function Networks") [Broomhead and Lowe, 1988]. Las BMs utilizan una función de energía que se minimiza durante el entrenamiento de los pesos asociados a las interconexiones entre las unidades de entrada, ocultas y de salida. De esta forma, se obtienen las probabilidades de los posibles estados. Por su parte, las RBFNs obtienen su salida de acuerdo a la distancia a unos puntos −denominados centroides− a partir de una función de transformación, generalmente gaussiana.

Además, Alexander Waibel y Geoffrey Hinton, entre otros, consiguieron resolver

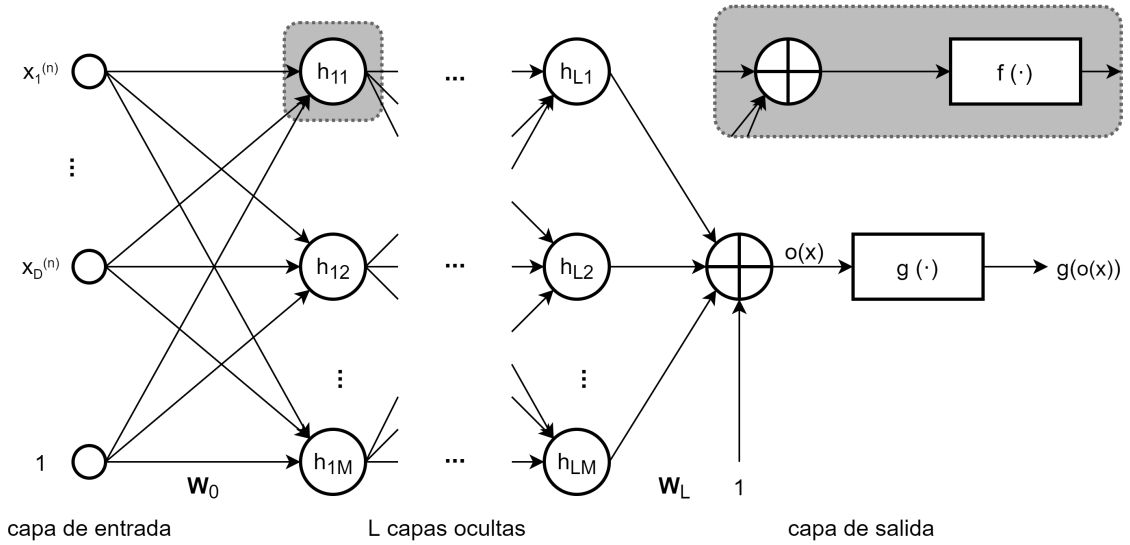


Figura 1.1: Representación de un MLP de L capas ocultas, con M neuronas en cada una. La capa inicial está formada por las D dimensiones de $\mathbf{x}$ y un sesgo; W_i, $i \in \{0, L\}$ son matrices de pesos para cada una de las interconexiones entre las neuronas de las distintas capas; $f(\cdot)$ y $g(\cdot)$ representan las funciones de activación de las capas ocultas y de la salida, respectivamente.

la falta de memoria en las NNs empleando Redes Neuronales con Retardo de Tiempo (TDNNs, "Time Delay Neural Networks") [Waibel et al., 1989] para el reconocimiento de fonemas.

Mientras tanto, comenzó a estudiarse cómo añadir información del contexto temporal −secuencias de datos− a este tipo de arquitecturas. Por ello, surgieron las Redes Neuronales Recurrentes (RNNs, "Recurrent Neural Networks") [Jordan, 1986] [Elman, 1990] basadas en la arquitectura propuesta por John Hopfield [Hopfield, 1982], donde cada capa tiene asociado un estado interno. Posteriormente, darían paso a las Redes de Memoria a Largo Plazo (LSTM, "Long Short-Term Memory") [Hochreiter and Schmidhuber, 1997] y a versiones bidireccionales −permiten tener en cuenta elementos de la secuencia tanto anteriores como posteriores− de las mismas [Schuster and Paliwal, 1997].

Cabe destacar la aparición de las Redes Convolucionales (CNNs, "Convolutional

Neural Networks") [Fukushima, 1980] [LeCun et al., 1989], cuyas capas están formadas por filtros convolucionales de una o más dimensiones. Sin duda, fue Yann LeCun quien mayor contribución hizo en esta línea de investigación, aplicando CNNs en tareas de clasificación de imágenes, concretamente en el reconocimiento automático de códigos postales manuscritos. Este trabajo desembocaría en el desarrollo de la bien conocida base de datos MNIST de dígitos manuscritos. Posteriormente, junto con su colega Yoshua Bengio, extendería su aplicación a problemas como el reconocimiento del habla y las series temporales [LeCun and Bengio, 1995].

Durante estos años también surgieron los Auto-Codificadores Profundos (DAEs, "Deep Auto-Encoders") [Ballard, 1987]. Se basan en la representación de los datos de entrada en un espacio de variables latentes ("Encoder") con el objetivo de reconstruir posteriormente la entrada utilizando dicha representación ("Decoder"). Son muy utilizados en el tratamiento de señales para reducción de ruido o de dimensionalidad y en la detección de anomalías, evaluando el error de reconstrucción cometido.

Por su parte, George Cybenko [Cybenko, 1989] y Kurt Hornik [Hornik et al., 1989] consiguieron demostrar que un MLP superficial −de una sola capa oculta− es un aproximador universal ajustable a cualquier aplicación. Sin embargo, como toda demostración de carácter universal, debe tomarse con cierta cautela, ya que se asumía un ilimitado número de unidades en la capa oculta para inferir la correspondencia entre un número finito de entradas y salidas. No obstante, comenzaron a aparecer limitaciones de los MLPs, como la escasa capacidad expresiva o la necesidad de ser reentrenados ante cambios en los datos de entrada.

Estas limitaciones propiciaron la aparición durante la década de 1990 de sistemas semilineales basados en núcleos ("kernels"). El objetivo de dichos métodos es construir una solución no lineal de algoritmos lineales por medio de núcleos de Mercer, $K(\phi(\mathbf{x}_i)^T, \phi(\mathbf{x}_j))$, donde $\phi(\mathbf{x})$ constituye una transformación no lineal de la observación $\mathbf{x}$. Entre ellos, destacan especialmente los Procesos Gaussianos (GPs, "Gaussian Processes") y las Máquinas de Vectores Soporte (SVMs, "Support Vector Machines").

Los GPs surgen por la integración del "truco" del núcleo en un filtro de Wiener [Wiener, 1949] para el tratamiento de señales [Ver Hoef and Cressie, 1993]. De esta forma, para un proceso estocástico, se lleva a cabo una asociación entre un conjunto finito de variables y una distribución gaussiana multivariante. Para un mayor detalle, se recomienda la lectura de [Williams and Rasmussen, 2006].

Por otro lado, la combinación del trabajo de clasificación de Vladimir Vapnik, basado en el principio de Máximo Margen [Vapnik, 1982], con el "truco"del núcleo conllevó la aparición de las famosas SVMs [Boser et al., 1992] por medio de la programación cuadrática. Suscitaron un gran interés y su uso e investigación centró gran parte de los estudios del Aprendizaje Máquina del momento. Para un estudio en mayor profundidad, se recomiendan las siguientes lecturas: [Vapnik, 1995], [Cortes and Vapnik, 1995], [Vapnik, 1998] y [Shawe-Taylor and Cristianini, 2004].

Como se ha comentado, los métodos de núcleos aparecieron para afrontar las limitaciones que existían en ese momento para el uso del Aprendizaje Profundo, el cual acabaría imponiéndose posteriormente debido a su gran capacidad expresiva y los avances informáticos que hicieron posible su diseño e implementación. No obstante, durante este período también surgieron nuevas líneas de investigación, como los conjuntos de LMs, detallados a continuación.

1.2. Conjuntos de Máquinas de Aprendizaje

Los conjuntos de LMs (MEs, "Machine Ensembles") surgen a partir de la Teoría de lo Probablemente Casi Correcto (PAC, "Probably Almost Correct") de Lee Valiant [Valiant, 1984] con el objetivo de diseñar clasificadores más robustos o consistentes. Para ello, se propone la agregación de arquitecturas más sencillas, denominadas aprendices. La idea principal radica en que cada uno de los aprendices se entrena bajo condiciones dispares, de manera que la salida obtenida por cada uno de ellos sea lo suficientemente distinta para poder generalizar tras llevar a cabo su agregación [Hansen and Salamon, 1990]. De esta forma, se consigue mejora a partir de

diversidad.

Dentro de los MEs, destacan dos tipos de arquitecturas. Por un lado, los comités, que generan la diversidad inicialmente en el entrenamiento para, posteriormente, llevar a cabo la agregación, diseñada de forma independiente. Dicha agregación puede resolverse de diversas maneras, siendo las más comunes el promediado de las salidas y el voto por mayoría (decidiendo en favor de la moda). Por otro lado, aparecen los consorcios, en los cuales se combina el diseño de los aprendices y su agregación durante el entrenamiento. Obviamente, este segundo tipo ofrece mejores prestaciones, aunque conlleva una mayor carga computacional.

1.2.1. Comités

Como ya se ha mencionado, el objetivo del uso de MEs es el de conseguir diversidad. Es decir, obtener una salida distinta con cada uno de los aprendices, los cuales han sido entrenados en condiciones distintas.

La forma más común de obtener diversidad es diversificando la información con la que se entrena cada uno de los aprendices. En esta línea de investigación, Leo Breiman fue, sin ninguna duda, uno de los nombres más destacados tras presentar los métodos de "Bagging" (**B**ootstrap **agg**regat**ing**) [Breiman, 1996] y "Switching" [Breiman, 2000]. En el primero de los casos, se aplica un re-muestreo tipo "Bootstrap" −re-muestreo con reemplazo de las muestras originales−, mientras que el "Switching" supone una alteración aleatoria de las etiquetas.

Otra alternativa es diversificar la arquitectura utilizada por cada uno de los aprendices. Aunque, en general, estos métodos no suponen un gran incremento de las prestaciones, cabe mencionar las Selvas Aleatorias (RFs, "Random Forests") [Breiman, 2001], en las cuales se diversifican árboles de decisión ramificados de forma probabilística y en las que cada uno de los aprendices tiene un subespacio de observación distinto.

Otra forma de diversificar la arquitectura utilizada es el método de apilamiento ("Stacking") [Wolpert, 1992]. Esta técnica se basa en dividir el conjunto de entre-

namiento en varios subconjuntos −por medio de la técnica del K-fold−, cada uno de los cuales produce M salidas al entrenarse M clasificadores con arquitecturas distintas entre sí con el subconjunto complementario de datos. Posteriormente, se utilizan todas las salidas generadas para entrenar un último clasificador como capa de agregación. Aunque se ha popularizado a la hora de afrontar retos y competiciones de Aprendizaje Máquina, su uso no ofrece una mejora clara de las prestaciones. De hecho, la diversidad de las salidas que cada uno de los aprendices genera no es alta.

Por último, también es posible diversificar la optimización que cada uno de los aprendices realiza durante su entrenamiento a través de distintas inicializaciones, costes o algoritmos. Sin embargo, este último tipo de métodos no ofrece grandes resultados.

1.2.2. Consorcios

Dentro de los consorcios −donde el diseño de los aprendices y su agregación se lleva a cabo simultáneamente− destacan, entre una gran cantidad de métodos existentes, las Mezclas de Expertos (MoEs, "Mixtures of Experts"), el Aprendizaje por Correlación Negativa (NCL, "Negative Correlation Learning") y, por supuesto, el "Boosting".

Las MoEs proponen como agregador un sistema de arbitraje que controle la contribución que cada uno de los aprendices aporta a la salida final. Se puede pensar como un sistema de puertas, donde el agregador pondera cada una de las salidas de los aprendices, sumando, posteriormente, todas ellas y construyendo así la salida. Fueron propuestas [Jacobs et al., 1991] para tareas de regresión y se llevaron a cabo distintas líneas de investigación para aumentar su capacidad expresiva [Jordan and Jacobs, 1994] [Jordan and Xu, 1995] [Olteanu and Rynkiewicz, 2008]. Sin embargo, la dificultad de adaptar dichas técnicas a tareas de clasificación [Omari and Figueiras-Vidal, 2013] lo hacen desaconsejable.

Por otro lado, aparecen los consorcios por medio de NCL [Liu and Yao, 1999a] [Liu and Yao, 1999b], cuyo objetivo es fomentar la diversidad de las salidas penalizando

−de forma parametrizable− la correlación entre ellas. Aunque sus prestaciones son realmente buenas en tareas de regresión, no es una técnica muy efectiva de cara al diseño de consorcios de clasificadores, como ya ocurría con las MoEs.

Por último, el "Boosting" es, sin ninguna duda, la familia de consorcios más importante. Se basa, principalmente, en el uso de aprendices "débiles" −término establecido por Robert Schapire [Schapire, 1990], quien, junto a su colega Yoav Freund, fue el gran promotor del "Boosting" − para obtener un clasificador robusto. Para ello, se entrenan de forma sucesiva los aprendices débiles enfatizando aquellas muestras que tienden a clasificarse erróneamente a partir de un coste exponencial. Originalmente presentado como AdaBoost (AB, "Adaptive Boosting") [Freund and Schapire, 1995] [Freund and Schapire, 1996a] [Freund and Schapire, 1997] para la obtención de salidas duras $o = \pm 1$, rápidamente surgieron numerosas variantes, destacando su versión para salidas blandas por medio de la minimización de una cota del coste exponencial, el Real AdaBoost (RAB) [Freund and Schapire, 1996b] [Schapire and Singer, 1999]. Una de las características que hacen tan atractivo el uso del "Boosting" es su robustez frente a la aparición de sobreajuste. Para más información, se recomienda la lectura de [Freund and Schapire, 2012].

Por último, mencionar la línea iniciada por Breiman, quien presentó el "Arcing"("**A**daptive **r**esampling and **c**ombin**ing**") [Breiman, 1998] [Breiman, 1999a] [Breiman, 1999b], una generalización conceptual del "Boosting".

1.3. El Aprendizaje Máquina en la actualidad

Todas las arquitecturas descritas anteriormente siguen utilizándose a día de hoy. De hecho, los avances informáticos −destacando la aparición y gran evolución de Unidades de Procesamiento Gráfico (GPUs, "Graphics Processing Units")− han permitido extender su uso, siendo el Aprendizaje Profundo (DL, "Deep Learning") el que ha acaparado gran parte de los últimos avances en el Aprendizaje Máquina [Bengio et al., 2007].

En este apartado, se enumeran algunas de las líneas de investigación con más interés en la actualidad, describiéndolas brevemente.

1.3.1. Aprendizaje Profundo

Como se ha dicho anteriormente, el avance del software ha ido de la mano del diseño de redes cada vez más complejas que han permitido el aprovechamiento de la gran capacidad expresiva del DL. De hecho, las DNNs han llegado a alcanzar mejores prestaciones que el ser humano en determinados problemas [Stallkamp et al., 2011].

En ocasiones, el objetivo es el desarrollo de versiones profundas de algoritmos descritos en apartados anteriores. Un ejemplo de ello son las Redes de Creencias Profundas (DBNs, "Deep Belief Networks") [Hinton and Salakhutdinov, 2006], formadas por RBMs. Esta variante prescinde de la capa de salida y de las interconexiones entre unidades dentro de la misma capa, simplificando así su entrenamiento. Posteriormente, los propios Ruslan Salakhutdinov y Geoffrey Hinton desarrollarían una nueva versión profunda de las BMs, las Máquinas de Boltzman Profundas (DBMs, "Deep Boltzman Machines") [Salakhutdinov and Hinton, 2009].

En otros casos, la investigación se ha centrado en aspectos concretos del diseño de estas redes. Entre ellos, destaca la técnica de "Drop-Out" [Srivastava et al., 2014], la cual consiste en introducir una probabilidad de desconectar aleatoriamente las interconexiones de un MLP. Cabe mencionar que se obtiene así un efecto de regularización, ya que dicha desconexión produce la no activación de dicha unidad, por lo que su peso no sería actualizado en ese caso. Otros trabajos se centran en la inicialización de los pesos [Glorot and Bengio, 2010], la optimización durante el entrenamiento sin el uso del Hessiano en RNNs [Martens and Sutskever, 2011], nuevos métodos de optimización −como Adam [Kingma and Ba, 2015] o RMSProp [Tieleman and Hinton, 2012]− o nuevas técnicas de decremento progresivo de la tasa de aprendizaje o el incremento del tamaño del lote ("batch") −conjunto de muestras que se utiliza para actualizar el gradiente durante el entrenamiento de la máquina de aprendizaje− [Smith et al., 2017], por citar algunos ejemplos.

La aplicación de estas redes generalmente es multidisciplinar, es decir, su uso no está supeditado a un determinado ámbito. En cambio, sí es muy frecuente que cada campo o disciplina cuente con una serie de algoritmos que se adaptan mejor a la resolución de sus problemas.

Un caso muy evidente de esto último lo encontramos en el Procesamiento del Lenguaje Natural (NLP, "Natural Language Processing"), un campo que, desde que Yoshua Bengio y sus colegas redujeran la alta dimensionalidad de las representaciones de palabras −a partir de las Incrustaciones de Palabras ("Word Embeddings")− por medio de modelos neuronales [Bengio et al., 2003], ha estado ligado al uso de técnicas de DL. Tareas como el etiquetado de la parte de la oración (POS, "Part-of-Speech Tagging") [Dos Santos and Zadrozny, 2014], el Reconocimiento de Entidades Nombradas (NER, "Named Entities Recognition") [Habibi et al., 2017], la traducción automática [Cho et al., 2014] o la generación de resúmenes automáticos −tanto extractivos [Nallapati et al., 2017] como abstractivos [Rush et al., 2015]− son algunos de los ejemplos que actualmente se basan en arquitecturas neuronales profundas. Típicamente, se han basado en el uso de las ya mencionadas RNNs o LSTMs bidireccionales, dando paso a nuevos enfoques como el uso de Transformadores ("Transformers") [Vaswani et al., 2017], redes que no requieren orden en el procesado de las secuencias de datos, facilitando así su paralelización durante el entrenamiento. Esto ha desembocado en la aparición de modelos pre-entrenados con grandes bases de datos, como es el caso de las Representaciones de Codificador Bidireccional de los Transformadores (BERT, "Bidirectional Encoder Representations from Transformers") [Devlin et al., 2018].

Otro de los campos actuales con mayor relevancia dentro del DL es la Visión Artificial ("Computer Vision"). Como ya se ha mencionado páginas atrás, desde finales de la década de 1980 se utilizaban CNNs para tareas de clasificación de imágenes. Las capacidades de procesamiento actual han permitido el uso de redes cada vez más complejas entrenadas con grandes bases de datos. Esto ha propiciado el extendido uso de las técnicas de Aprendizaje por Transferencia (TL, "Transfer Learning").

Propuesto originalmente por Lorien Pratt [Pratt, 1993], el TL consiste en utilizar los pesos obtenidos durante el entrenamiento de una red para resolver un problema distinto al original, es decir, re-entrenando el sistema con nuevos datos. Para ello, la técnica más común es la congelación de los pesos de la red pre-entrenada añadiendo varias capas totalmente conectadas al final con el objetivo de adaptar el sistema a los nuevos datos. Dicha técnica se ha popularizado, brindando así la posibilidad de aprovechar redes pre-entrenadas para diversas tareas como: clasificación, utilizando redes como AlexNet [Krizhevsky et al., 2012], VGG-Net [Simonyan and Zisserman, 2014] o ResNet [He et al., 2016]; detección de objetos por medio de redes de la familia YOLO [Redmon et al., 2016]; o segmentación de imágenes con redes como la U-Net [Ronneberger et al., 2015] o la V-Net [Milletari et al., 2016].

Sin embargo, el auge del DL no debe significar que técnicas clásicas que, desde hace décadas, se han utilizado con éxito en la Visión Artificial queden en el olvido. Esto ha provocado cierto debate en la comunidad científica, pero es evidente que ambos tipos de técnicas son viables e, incluso, pueden complementarse [O'Mahony et al., 2019].

Aunque se han descrito dos de los principales campos actuales, la aplicación de las técnicas de DL se ha extendido a todos los ámbitos, como en el Tratamiento de la Señal, en Sanidad, en Industria, en Finanzas, etcétera. Sin embargo, para no perder el hilo principal de la Tesis Doctoral, se recomienda a quien desee un mayor detalle la lectura del manual elaborado por Jürgen Schmidhuber [Schmidhuber, 2015].

1.3.2. Aprendizaje Máquina Adversario

En los últimos años, ha adquirido gran importancia el Aprendizaje Máquina Adversario ("Adversarial Machine Learning"), un nuevo campo dentro del ML que aparece como intersección con el campo de la seguridad informática, por lo que tampoco se detallará en exceso, ya que, pese a su relevancia actual, su estudio y aplicación están fuera del alcance de esta Tesis. Este emergente campo busca la utilización de las técnicas, algoritmos y arquitecturas del ML contra un oponente ("adversario").

Este tipo de técnicas comenzaron a utilizarse dentro del Aprendizaje Automático [Huang et al., 2011] y otras variantes estadísticas, como el Análisis de Riesgo Adversario (ARA, "Adversarial Risk Analysis") [Ríos-Insua et al., 2009], que basado en la Teoría de Juegos y el equilibrio de Nash, toma decisiones en función de las acciones de un adversario. Sin embargo, sería la publicación de las Redes Adversarias Generativas (GANs, "Generative Adversarial Networks") [Goodfellow et al., 2014] por parte de Ian Goodfellow lo que asentaría esta nueva línea de investigación. Las GANs están compuestas por dos modelos que se entrenan de manera simultánea, manteniendo cierto equilibrio entre la cooperación y la competición. Por un lado, un modelo generativo G, encargado de generar los datos sintéticos. Por otro lado, un modelo discriminativo D, encargado de estimar la probabilidad de que los datos provengan del conjunto de entrenamiento. Inicialmente, G comienza generando datos a partir de una entrada ruidosa, lo que supone una fácil tarea de clasificación entre datos reales y artificiales por parte de D. Sin embargo, el objetivo es que durante el entrenamiento, en el que ambos modelos comparten su información, G acabe generando muestras a partir de la distribución real de los datos, por lo que D será incapaz de identificar las muestras sintéticas. De esta forma, se pueden crear datos artificiales (incluyendo imágenes, sonidos, textos...) prácticamente indistinguibles de los datos reales. Destaca su aplicación en Visión Artificial −como la generación de imágenes de caras humanas artificiales por parte de Nvidia [Karras et al., 2019]−, pero han surgido variantes que permiten trabajar con datos tabulares, como la CTGAN ("Conditional Tabular Generative Adversarial Network") [Xu et al., 2019]. Gracias a ello, su uso se ha extendido en todo tipo de campos, incluyendo la reducción de sesgos [Zhang et al., 2018].

1.3.3. Problemas Singulares

Los Problemas Singulares, campo de investigación en el que se enmarca esta Tesis, son, sin duda, uno de los grandes retos actuales en el Aprendizaje Máquina. Los Problemas Singulares engloban aquellas tareas cuyas características provocan

que el uso de máquinas discriminativas convencionales produzca sesgos a la salida del sistema o, en definitiva, resultados no satisfactorios.

Estos problemas son objeto de gran interés desde hace un par de décadas y han surgido numerosas soluciones para resolverlos. No obstante, la gran mayoría de estas soluciones son puramente empíricas y no están fundamentadas en la teoría estadística. Este hecho supone que diversas soluciones obtengan buenos resultados bajo unas circunstancias específicas, pero que se degraden en gran medida ante la aparición de cambios en dichas circunstancias: cambios en la política de costes o el número de muestras disponibles, por ejemplo. Este efecto, se mostrará y estudiará en capítulos posteriores de la Tesis.

Algunos ejemplos son la clasificación con costes dependientes del ejemplo ("example-dependent-cost") [Elkan, 2001] [Bahnsen et al., 2015a] y la regresión ordinal −es decir, clasificación bajo un criterio de rango− [Anderson, 1984] [Gutiérrez et al., 2015]. En estos casos, la introducción de los correspondientes costes en el coste subrogado, que se minimiza durante el entrenamiento, como un factor de pesos o por medio de re-muestreo, no es estrictamente equivalente a minimizar el coste Bayesiano, el cual requiere otras formulaciones fundamentadas [Lázaro et al., 2018] [Lázaro and Figueiras-Vidal, 2019].

La familia de Problemas Singulares más conocida son los Problemas Desequilibrados −objeto de estudio de esta Tesis−, los cuales se describirán en detalle durante los siguientes capítulos. Estos problemas engloban todas las situaciones en las que alguna (o algunas) de las clases está en clara desventaja respecto al resto, ya sea por un tamaño menor de su población o por la política de costes del problema, pudiendo ser mucho más cara su elección.

1.4. Orientación y objetivos de la Tesis

Como se mostrará en el capítulo siguiente, los problemas de clasificación desequilibrada −pertenecientes a la familia de los Problemas Singulares− están presentes en

numerosas aplicaciones y, por regla general, precisan del uso de técnicas específicas para su resolución.

Las máquinas de clasificación convencionales −típicamente máquinas discriminativas, que incluyen MLPs, RBFNs, SVMs, y los correspondientes conjuntos de máquinas− son sensibles a los efectos del desequilibrio, ya que la obtención de los valores de sus parámetros se realiza por medio de algoritmos cuyo objetivo es el de optimizar medidas que no tienen en cuenta dichos efectos. Por ejemplo, funciones típicas de coste subrogado tienen una escasa representación de muestras minoritarias, por lo que la minimización de dicho coste lleva a pobres prestaciones para su(s) clase(s). En otros casos, la clase minoritaria puede ser identificada por el propio sistema como ruido. De hecho, ante la posible aparición de ruido, la clase minoritaria es mucho más vulnerable a la distorsión. Por tanto, la salida de una máquina convencional está sesgada en favor de la clase mayoritaria, obteniendo prestaciones muy pobres.

Por su parte, las máquinas generativas −las cuales buscan obtener una estimación de la verosimilitud $\hat{p}(\mathbf{x}|C_i)$, $i \in \{1, 2, ..., I\}$ de cada una de las I clases− son insensibles a los efectos del desequilibrio. No obstante, debido a la dificultad de obtener buenas estimaciones de dichas verosimilitudes, suelen tener prestaciones muy limitadas. Por ello, el estudio se centrará en las máquinas discriminativas, utilizando el MLP −y sus versiones profundas− como máquina de aprendizaje.

Por tanto, resulta necesaria la aplicación de diversas técnicas que puedan hacer frente a los ya mencionados efectos. Sin embargo, tras un estudio exhaustivo de las técnicas publicadas, se puede afirmar que la gran mayoría de procedimientos propuestos para tratar problemas desequilibrados tienen una naturaleza cualitativa, es decir, son modificaciones razonables del conjunto de datos de entrenamiento o/y del algoritmo de clasificación, pero no existe un análisis de por qué y cómo proporcionan sus resultados, lo que puede implicar incluso degradación. Aunque esto no disminuye su uso, no hay siempre una clara perspectiva de sus capacidades y limitaciones. Como establecen los autores de [[He and García, 2009], pp. 1279-1280] y [[Branco et

al., 2016], pp. 31-33], se necesita una mejor comprensión de los mecanismos de estos métodos para evitar errores y para fomentar los avances en este campo. Partiendo de esta perspectiva, en esta Tesis se sigue la línea de los pocos precedentes que analizan formalmente las aproximaciones de re-equilibrado y que han inspirado esta investigación, incluyendo [Domingos, 1999] en el marco del "meta-coste", el análisis basado en el teorema de equivalencia de [Elkan, 2001] y [Zadrozny et al., 2003], y el riguroso estudio presentado en [Castro and Braga, 2013].

Basada en todo lo anterior, esta Tesis se centra en la resolución de problemas desequilibrados para el caso binario, realizando un exhaustivo estudio y análisis del estado del arte para resolver este tipo de problemas. A partir de ello, se propone una metodología fundamentada −en la teoría bayesiana−, basada en la equivalencia del cociente de verosimilitudes mediante el uso de divergencias de Bregman y el re-equilibrado neutral −aquél que utiliza todas las muestras de la misma clase de igual manera estadísticamente para llevar a cabo el proceso de re-equilibrado−, las cuales se demuestra que son condiciones suficientes y necesarias en la resolución controlable de problemas desequilibrados.

Además, se propone un procedimiento de diseño completo del clasificador −tanto de la máquina de aprendizaje, como del proceso de re-equilibrado− para hacer frente al desequilibrio. Este proceso tendrá como objetivo maximizar las prestaciones de la máquina −en términos de detección de la clase minoritaria− para un punto de trabajo definido. Para ello, se plantea la combinación de distintos métodos de re-equilibrado, ajustando sus parámetros de diseño y el nivel de intensidad con el que se aplica cada uno de ellos. Asimismo, se estudia el comportamiento de redes profundas ante la presencia de desequilibrio, con el objetivo de conocer si su mayor capacidad expresiva favorece o puede llegar a comprometer la utilización de los métodos propuestos.

Para finalizar, se lleva a cabo una primera exploración en los métodos de generación de muestras artificiales para el re-equilibrado. El objetivo será conocer aún más en detalle las virtudes y defectos de la técnica más conocida y empleada para ello: SMOTE [Chawla et al., 2002], una técnica basada en la generación de muestras

sintéticas a partir de los K vecinos más próximos. Una vez analizada, con el objetivo de hacer frente a la aparición de estructuras filiformes (en forma de hilo) −lo que supone la mayor limitación de SMOTE−, se presenta (de manera preliminar) VoluSMOTE, una adaptación de SMOTE para generar muestras de forma volumétrica.

1.5. Contenido de la Tesis

El Capítulo 2 describirá detalladamente en qué consisten los problemas de clasificación desequilibrada, mostrando

- su relevancia;
- las dificultades asociadas y sus efectos en el aprendizaje;
- un estudio del estado del arte sobre las familias de métodos comúnmente empleadas para su resolución;
- y las métricas necesarias para evaluar correctamente sus prestaciones.

Por su parte, en el Capítulo 3 se presentará la metodología fundamentada propuesta, principal aportación y eje conductor de la Tesis. A partir de una introducción de la teoría bayesiana de clasificación, se describirá el método propuesto, basado en un re-equilibrado neutral y el uso de las divergencias de Bregman, así como una breve descripción de su extensión a SVMs y conjuntos. Además, se incluye un nuevo procedimiento de re-equilibrado informado −de manera contraria a los métodos de re-equilibrado neutrales, dota de mayor importancia o interés a ciertas muestras del conjunto de entrenamiento− fundamentado basado en un entrenamiento en dos pasos. El capítulo incluye una serie de experimentos que ilustran cada uno de los conceptos presentados a lo largo de él, pero por medio de un re-equilibrado completo ("full rebalancing"), es decir, compensando íntegramente el desequilibrio existente entre las clases.

En el Capítulo 4 se expondrá un procedimiento detallado de diseño del clasificador re-equilibrado. Para ello, se propone la combinación de distintos métodos de re-equilibrado, optimizando la aportación de cada uno de ellos en términos de su intensidad y sus parámetros. En este capítulo se demuestra que el re-equilibrado completo no es necesariamente la mejor opción. Experimentos con bases de datos en condiciones diversas (mayor y menor número de muestras observables disponibles o presencia de ruido aditivo) sirven para ilustrar tanto las fortalezas como las limitaciones (esperadas) del método propuesto.

En el Capítulo 5 se llevará a cabo un estudio detallado de SMOTE como referente a la hora de hacer frente al desequilibrio mediante la generación de muestras sintéticas. Para ello, se presentan tanto las ventajas asociadas a su uso −que lo convierten en un referente en la generación de muestras− como sus limitaciones, entre las que destaca la aparición de estructuras en forma de "tela de araña" que dificultan su correcto funcionamiento en problemas de alta dimensionalidad. Con el objetivo de solucionar dicha limitación se propone un nuevo algoritmo: VoluSMOTE. Se detalla su funcionamiento y varias pruebas preliminares cierran el capítulo, demostrando el posible potencial de la técnica propuesta.

Para finalizar, en el Capítulo 6 se presentarán las conclusiones extraídas de la Tesis, además de las líneas de investigación abiertas a partir de ella.

Capítulo 2

Problemas de clasificación desequilibrada

Una máquina de aprendizaje es capaz de tomar decisiones a partir del espacio observable del que dispone. La certeza de dicha aseveración ha sido demostrada a lo largo del capítulo anterior, donde se han presentado multitud de ejemplos de ello. De manera general, la máquina no tendrá dificultades para "aprender" a tomar esas decisiones. Sin embargo, hay determinados casos en los que las características propias de los datos influyen en las decisiones que toma la máquina y obligan a actuar de manera consecuente. Éstos son los denominados Problemas Singulares, entre los que destacan los problemas de clasificación desequilibrada, cuya versión binaria es objeto de estudio en esta Tesis. Los problemas desequilibrados son aquellos en los que hay clases mucho menos representadas que el resto. Esto ocurre cuando el tamaño de las poblaciones de las clases es muy distinto o/y cuando algunas clases se ven claramente desfavorecidas por una política de costes que penaliza en exceso su elección. En estos casos, las máquinas de aprendizaje tienden a sesgar −tal y como haría un humano en la misma situación− su salida en favor de las clases mayoritarias (las más probables) o las que conllevan un menor coste. Por ello, es fundamental corregir dicho comportamiento con el objetivo de paliar las dificultades existentes para detectar las clases minoritarias, las cuales son (generalmente) las de mayor interés.

Actualmente, esto supone un reto trascendental en numerosas aplicaciones de

diversos campos como:

- Medicina [Rao et al., 2006] [Mazurowski et al., 2008] [Mena and González, 2009] [Freitas, 2011] [Nahar et al., 2013] [Samant and Agarwal, 2019];

- Bio-Informática [Radivojac et al., 2004] [Batuwita and Palade, 2009] [Yu et al., 2013] [Triguero et al., 2015];

- procesado de imagen y extracción de documentos [Viola and Jones, 2004] [Tao et al., 2006] [Kwak, 2008] [Chen et al., 2011] [De la Torre et al., 2015] [Li et al., 2018] [Anne et al., 2018];

- procesos de producción [Liao, 2008] [Park et al., 2013] [Seiffert et al., 2014];

- Seguridad [Chan and Stolfo, 1998] [Phua et al., 2004] [Tavallaee et al., 2010] [Mehrotra et al., 2016] [Yang et al., 2019];

- Empresa y Finanzas [Liu et al., 1999] [Panigrahi et al., 2009] [Ngai et al., 2009] [Zhou, 2013] [Verbraken et al., 2014] [Bahnsen et al., 2015b] [Nami and Shajari, 2018];

- clasificación de textos [Manevitz and Yousef, 2001] [Tong and Koller, 2001];

- Meteorología [Tsai et al., 2009];

- Biología [González et al., 2013];

- y servicios de información en Internet, como buscadores y redes sociales [Rao and Pais, 2019] [Basak et al., 2019] [Fard et al., 2019];

por citar algunos ejemplos representativos. En definitiva, todo problema de detección de anomalías (o sucesos poco probables) es susceptible de verse afectado por los efectos del desequilibrio en las tareas de clasificación por medio de máquinas de aprendizaje.

A lo largo de este capítulo se presentan los aspectos más característicos de este tipo de problemas, comenzando por los efectos que tiene el desequilibrio en la resolución de las tareas de clasificación. Seguidamente, se realiza un estudio del estado del arte donde se revisan las técnicas que se han propuesto durante los últimos años para minimizar o atenuar las dificultades previamente descritas. Por último, se presentan algunas métricas adecuadas para la correcta evaluación de las prestaciones cuando existen grandes diferencias entre las clases, tanto en sus poblaciones como en sus costes asociados.

Un estudio en profundidad del desequilibrio, sus factores y las técnicas empleadas podría alargar esta Tesis en exceso. Por ello, se recomienda la lectura de los tutoriales [He and García, 2009], [Sun et al., 2009], [López et al., 2013], [He and Ma, 2013], [Krawczyk, 2016], [Branco et al., 2016] y [Haixiang et al., 2017] para un mayor detalle, ya que darán una perspectiva completa al lector interesado. Además, la presente Tesis centra sus esfuerzos en la resolución de problemas binarios de clasificación desequilibrada, pero existen extensiones del estudio a problemas multiclase, como las presentadas en [Wang and Yao, 2012] y [Fernández et al., 2013].

2.1. Efectos del desequilibrio

Por regla general, cuanto mayor es el grado de desequilibrio −es decir, la diferencia en las poblaciones de las clases o la política de costes−, mayores dificultades aparecen para realizar una correcta clasificación. Dichas dificultades o limitaciones no son provocadas (directamente) por el desequilibrio −de hecho, puede darse el caso de problemas altamente desequilibrados que sean, incluso, fáciles de resolver−; sin embargo, sí se ven acrecentadas ante la presencia del mismo. La mayor o menor dificultad del problema estará condicionada por la capacidad de caracterizar la clase minoritaria a partir de los datos de los que se dispone. A continuación, se detallan cuáles son los factores que más interfieren y dificultan la clasificación [López et al., 2013] [Stefanowski, 2016] ante la existencia de desequilibrio.

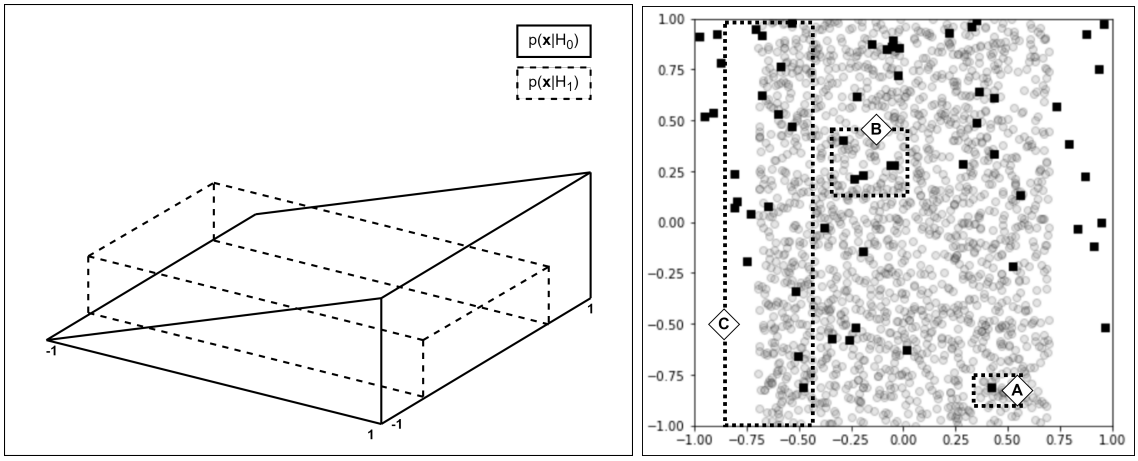
La función de verosimilitud $p(\mathbf{x}|C_i)$ describe cómo se distribuye la variable observable $\mathbf{x}$ para la clase i. Muchos clasificadores basados en el criterio de Máxima Verosimilitud ("Maximum Likelihood", ML) −un caso particular del criterio del Máximo *a Posteriori* (MAP) que asume que las clases son equiprobables− buscan estimar dichas funciones, eligiendo posteriormente la clase de acuerdo a

$$j^*_{ML} = \underset{j}{\text{argmáx}} \{p(\mathbf{x}|C_j)\} \tag{2.1}$$

Desafortunadamente, cuando se dispone de poca información y una de las clases está muy poco representada, la estimación de la verosimilitud de la clase minoritaria puede estar alejada de la distribución original de dicha clase. Este hecho conlleva una degradación de las prestaciones y es, sin duda, la gran limitación de las máquinas generativas, hecho que se describe en mayor detalle en el Capítulo 3.

Por otro lado, es muy común la aparición de pequeños grupos de muestras que pueden −erróneamente− considerarse subclases de la clase minoritaria (problema conocido como "small disjuncts"). La formación de esos grupos provoca que aquellas muestras que se mantienen aisladas puedan interpretarse como ruido o como muestras fuera de rango. Cuando esto ocurre, la clase minoritaria aparece poco (o nada) representada en regiones del espacio observable donde realmente −según su distribución original− sí tiene presencia, hecho que favorece la selección de la clase mayoritaria −más representada en dicha región− por parte del clasificador, sesgando así su salida.

Otro factor que juega un papel importante ante la presencia de desequilibrio es el solapamiento existente entre las clases. Es decir, en aquellas regiones donde exista presencia de ambas clases será donde el clasificador sea más sensible al desequilibrio. Un ejemplo claro: un problema cuyas clases son linealmente separables, pero en la que una de ellas está mucho más representada que la otra, seguirá siendo sencillo de resolver pese al alto desequilibrio que pueda existir; por contra, en un problema cuyas clases estén totalmente solapadas, un desequilibrio alto apenas permitirá detectar la clase minoritaria, ya que la región donde aparece estará poblada de un mayor número de muestras de la clase mayoritaria.



(a) **(b)**

Figura 2.1: Ejemplo de problema de clasificación desequilibrada. (a) Funciones de verosimilitud de las clases C_1 (línea continua) y C_0 (línea discontinua). (b) Diagrama de dispersión del problema con presencia de desequilibrio donde se ilustran varios efectos: (A) muestra posiblemente considerada como ruido; (B) pequeña subclase formada de la clase minoritaria ("small disjunct"); (C) Zona de muestras fronterizas.

Considerando la separación entre las clases, en [Kubat and Matwin, 1997] se distinguen tres tipos de muestras: seguras, ruidosas y fronterizas. En dicho trabajo, definen como muestras seguras aquellas que aparecen en zonas relativamente homogéneas respecto a su clase. Por otro lado, las muestras ruidosas son las que aparecen en la zona segura de la otra clase. Por último, las muestras fronterizas son aquellas que aparecen en las regiones próximas a la frontera entre ambas clases, es decir, donde se solapan. Obviamente, la presencia de una clase mayoritaria en zonas fronterizas dificulta la detección de las muestras de clase minoritaria, ya que en muchas ocasiones se pueden llegar a considerar ruidosas. Este hecho ha motivado la aparición de métodos centrados en el re-equilibrado únicamente de estas regiones fronterizas, algo que, como se muestra en el siguiente capítulo, puede suponer una degradación aún mayor de las prestaciones.

La mayoría de clasificadores son sensibles a la presencia de ruido en el espacio

de observación, algo que, por otro lado, es bastante común. Aunque dicho ruido puede aparecer en ambas clases, su presencia tiene mayores efectos sobre la clase minoritaria, ya que, al estar poco representada, toda estimación de la distribución de dicha clase estará totalmente alejada de la original. Además, acentuará los problemas del clasificador para generalizar ante la presencia de nuevas muestras observables.

Por último, cuando se realizan particiones o divisiones en el conjunto de datos −algo común a la hora de realizar la división entre conjuntos de entrenamiento, validación y test, o al fragmentar el conjunto de entrenamiento mediante K-fold− debe tenerse en cuenta la distribución de los datos y el desequilibrio existente en el problema original. Si esto no se hace correctamente, pueden aparecer regiones donde la clase minoritaria apenas esté representada debido a la baja presencia de muestras de dicha clase −fomentando la aparición de "small disjuncts"−, dando lugar a una vaga generalización a la hora de clasificar nuevos datos. Además, es importante preservar el grado de desequilibrio original, algo que se consigue haciendo la división mediante muestreo estratificado respecto a la etiqueta.

2.2. Técnicas de re-equilibrado

Una vez vistos en el apartado anterior los efectos negativos que tiene el desequilibrio en las tareas de clasificación, se revisan las técnicas de re-equilibrado existentes para hacer frente a estos problemas, cuyo principal objetivo es atenuar dichos efectos.

Desde finales de la década de 1990, han surgido numerosas técnicas y algoritmos para reducir las dificultades asociadas a estos problemas. Estos procedimientos se dividen en dos principales familias de métodos: las técnicas de pre-procesado y los métodos que modifican los algoritmos de aprendizaje, aunque también es posible combinaciones de ambas. Las primeras sirven para reducir el grado de desequilibrio original de los datos, creando un problema asociado más sencillo de clasificar que permita posteriormente resolver el problema original. Por su parte, los métodos relativos a los algoritmos de aprendizaje buscan minimizar la tendencia de las máquinas

de aprendizaje a decidir en favor de la clase mayoritaria, ya sea modificando su arquitectura o proponiendo nuevos enfoques. A continuación se describen detalladamente ambas familias de métodos.

2.2.1. Re-equilibrado mediante técnicas de pre-procesado

Las técnicas de pre-procesado proponen la transformación de los datos originales con el objetivo de modificar sus poblaciones o establecer la importancia de cada una de las muestras durante el entrenamiento. Dentro de esta familia de métodos, aparece una nueva taxonomía, diferenciando tres tipos de técnicas: los llamados métodos sensibles a costes ("cost-sensitive") o ponderación, el re-muestreo y la generación de datos sintéticos.

Los métodos cost-sensitive buscan incrementar el peso de las muestras minoritarias durante el entrenamiento [Domingos, 1999] [Elkan, 2001] [Kukar and Kononenko, 1998] [Zadrozny et al., 2003] por medio de una ponderación de las clases. Es decir, el error cometido con las muestras minoritarias será penalizado en mayor medida que el de las mayoritarias durante el entrenamiento de la máquina. Para ello, partiendo de una máquina discriminativa y su coste subrogado $C\left(t^{(n)},\, o(\mathbf{x}^{(n)})\right)$ −error cometido por el clasificador para la muestra $\mathbf{x}^{(n)}$, siendo t la etiqueta para dicha muestra y o la salida del clasificador−, se asigna un criterio de ponderación que penalice más el error a la hora de detectar las muestras minoritarias. Para N_1 muestras de la clase minoritaria (o positiva) y N_0 muestras de la clase mayoritaria (o negativa), el proceso de optimización durante el entrenamiento buscará los mejores parámetros $\mathbf{w}$ del clasificador según

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmín}} \left(w_{(+)} \sum_{n_1}^{N_1} C\left(t=1,\, o\left(x^{(n_1)}\right)\right) + w_{(-)} \sum_{n_0}^{N_0} C\left(t=-1,\, o\left(x^{(n_0)}\right)\right) \right) \tag{2.2}$$

siendo $w_{(+)}$ y $w_{(-)}$ los pesos asociados a la clase minoritaria y mayoritaria, respectivamente, $x^{(n_1)}$ cada una de las muestras positivas y $x^{(n_0)}$ las muestras negativas. De esta forma, el clasificador compensará el bajo número de muestras minoritarias con

un mayor peso $w_{(+)}$, aumentando así su probabilidad de detección.

En otras ocasiones, se utilizan técnicas de selección de muestras [Kubat and Matwin, 1997] y de re-muestreo, las cuales buscan modificar las poblaciones originales de los datos utilizados para el entrenamiento del clasificador. Para conseguirlo, se utilizan las técnicas de sub-muestreo −eliminan muestras de la clase mayoritaria− o/y sobre-muestreo −repetición de muestras de la clase minoritaria− con el objetivo de enfatizar la clase minoritaria durante el entrenamiento a partir de las muestras originales. Es muy común la aplicación de este tipo de métodos de forma aleatoria, con criterios de selección de muestras basados en una distribución binomial negativa [Hido et al., 2009]. También hay extensiones a problemas multiclase [Abdi and Hashemi, 2015].

Ahora bien, el sub-muestreo o sobre-muestreo aleatorio puede provocar efectos negativos en el clasificador [Batista et al., 2004]. La eliminación de forma aleatoria de muestras de la clase mayoritaria puede suponer la pérdida de información relevante de dicha clase. Esto puede reducir las prestaciones de la máquina, ya que las muestras eliminadas pueden ser importantes (o críticas) para ajustar correctamente la frontera de decisión [He and Ma, 2013]. Por su parte, el sobre-muestreo aleatorio aumenta la probabilidad de que el clasificador sobreajuste respecto a la clase minoritaria debido a la repetición de muestras [Fernández et al., 2013]. Por ejemplo, una muestra ruidosa (o fuera de margen) de la clase minoritaria que se repita varias veces puede provocar que el clasificador asigne −erróneamente− mayor importancia a la clase minoritaria en esa región. Este fenómeno se incrementa cuando la tasa de repetición es elevada [Branco et al., 2016].

Debido a lo anterior, surgen otros métodos de sub-muestreo no aleatorios. Entre ellos destacan algunas técnicas clásicas como el método de los vecinos más cercanos condensados [Hart, 1968] y su versión editada [Wilson, 1972] o las conexiones de Tomek [Tomek, 1976]. El método de los vecinos más próximos condensados reduce la clase mayoritaria (o negativa) a aquellas muestras que pueden clasificarse incorrectamente como positivas mediante su vecino más próximo −o mediante sus 3 vecinos

más próximos en su versión editada−. Por su parte, la técnica de las conexiones de Tomek busca pares de muestras compuestos por ambas clases a partir de su vecino más próximo, eliminando las muestras de la clase mayoritaria de cada par. Otras aproximaciones más recientes se basan en el solapamiento existente entre las clases [Vuttipittayamongkol and Elyan, 2020]. Además, aparece la posibilidad de llevar a cabo combinaciones de métodos, incluso de sub-muestreo y sobre-muestreo según distintas estrategias, seleccionando el grado de re-equilibrado adecuado [Estabrooks et al., 2004] o filtrando aquellas muestras de la clase mayoritaria que dificultan la clasificación, como en el método conocido como SPIDER ("Selective Preprocessing of Imbalanced Data") [Stefanowski and Wilk, 2008].

Sin embargo, las técnicas de pre-procesado más utilizadas son aquellas que generan datos artificiales de la clase minoritaria. Dentro de esta familia de métodos aparece la clásica técnica de generación a partir de un modelo de densidad de probabilidad con ventanas de Parzen [Parzen, 1962], el cual ha sido aplicado con éxito para aprendizaje ruidoso. Pero destaca, sin ninguna duda, el algoritmo SMOTE ("Synthetic Minority Over-sampling Technique") [Chawla et al., 2002], un método tan sencillo como eficiente. Se basa en la generación de muestras sintéticas sobre las conexiones de cada muestra de la clase minoritaria y sus K vecinos más próximos. Para ello, se elige una muestra minoritaria de forma aleatoria, se calculan sus K vecinos más próximos y aleatoriamente se elige uno de esos vecinos. En la línea que separa la muestra original y el vecino seleccionado, se genera la nueva muestra sintética en un lugar también aleatorio. Este proceso se repite durante un determinado número de iteraciones −generalmente el número de muestras generadas se elige en función del grado de re-equilibrado elegido− hasta conseguir el nuevo conjunto de entrenamiento. La Figura 2.2 muestra un ejemplo de su funcionamiento. Como se verá en el siguiente capítulo, se trata de un método aproximadamente neutral, ya que todas las muestras tienen la misma probabilidad de ser seleccionadas para la generación. Sin embargo, su mayor limitación aparece ante problemas con un espacio de observación de alta dimensionalidad, no sólo porque el cálculo de las distancias se complica

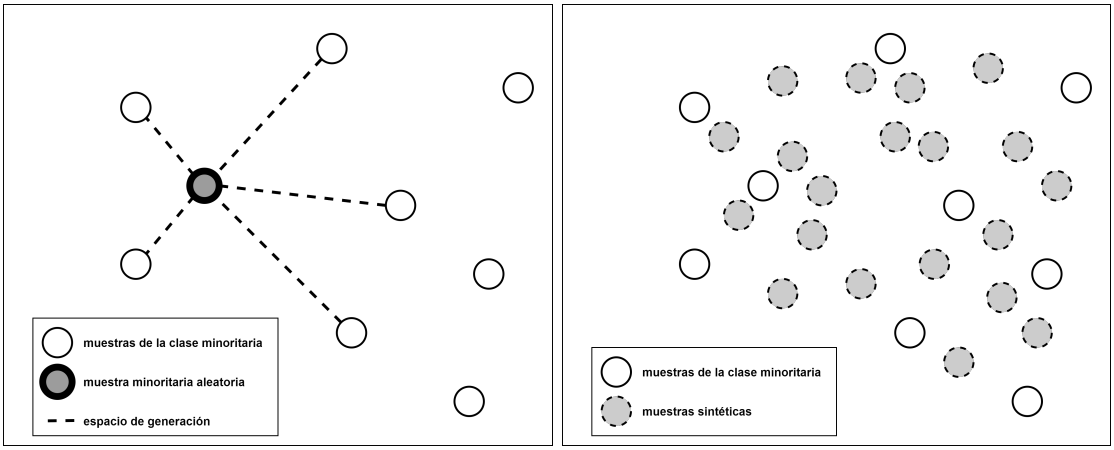


Figura 2.2: Funcionamiento de SMOTE. (a) Cálculo de los $K = 5$ vecinos más próximos para una muestra minoritaria aleatoria y sus espacios de generación. (b) Muestras sintéticas generadas tras varias iteraciones.

cuando el número de dimensiones es elevado, sino porque se produce un efecto de "tela de araña". Además, es importante destacar que su uso se limita a variables numéricas, ya que la generación de datos sintéticos categóricos (o discretos) requiere otras consideraciones, como una codificación previa de los datos.

El método SMOTE es el más extendido para re-equilibrar por medio de la generación de datos sintéticos. De hecho, ha servido de inspiración para la aparición de numerosos métodos basados en su funcionamiento [Barua et al., 2014] [Hu et al., 2009] [Ramentol et al., 2012] [Bunkhumpornpat et al., 2009] [Bunkhumpornpat et al., 2012]. Dentro de estas versiones alternativas de SMOTE, destacan los métodos de Borderline-SMOTE [Han et al., 2005] y ADASYN ("ADAptive SYNthetic sampling") [He et al., 2008]. Ambas técnicas definen un subconjunto de muestras minoritarias "en peligro"de acuerdo a su cercanía a muestras de la clase mayoritaria, llevando a cabo el re-equilibrado únicamente a partir de estas muestras. No obstante, como se verá en el Capítulo 3, este tipo de alternativas son en su mayoría empíricas y no fundamentadas, por lo que su aplicación puede conllevar, incluso, una degradación

de las prestaciones.

Por otro lado, existen otras alternativas respecto a la generación de datos, como las réplicas ruidosas de la clase minoritaria [Lee, 1999] [Lee, 2000] o extensiones del Aprendizaje Activo [Settles, 2010] para el pre-procesado de los datos de entrenamiento [Abe, 2003] [Bordes et al., 2005] [Ertekin et al., 2007]. Este último grupo de algoritmos emplean procedimientos para seleccionar progresivamente muestras con el objetivo de mejorar las prestaciones y reducir los tiempos de entrenamiento.

2.2.2. Re-equilibrado mediante algoritmos de aprendizaje

El re-equilibrado también puede conseguirse modificando los parámetros de diseño de algunos algoritmos. Por ejemplo, la forma propuesta en [Veropoulos, 1999] para SVMs simplemente otorga mayor peso para las muestras minoritarias a las variables de holgura que cuantifican la desviación sobre la clasificación correcta ideal. Se han aplicado modificaciones similares en conjuntos de "Boosting" [Fan et al., 1999] [Sun et al., 2007] [Ting, 2000]. También es posible ponderar los términos de la solución de la SVM [Imam et al., 2006]. Estas aproximaciones son básicamente lo mismo que una ponderación de muestras. Por su parte, en [Masnadi-Shirazi and Vasconcelos, 2010] y [Masnadi-Shirazi and Vasconcelos, 2011] se proponen algoritmos de aprendizaje modificados basados en análisis estadísticos.

Entre los algoritmos orientados a la resolución de problemas desequilibrados destacan las SVM de una clase ("One-class SVM"), como las propuestas en [Manevitz and Yousef, 2001], [Kowalczyk and Raskutti, 2002] y [Fard et al., 2019]. Estos métodos aparecen para ser efectivos en situaciones de gran desequilibrio y son comúnmente utilizados para detección de ruido, ya que están basados en la detección de una única clase −la mayoritaria−, descartando aquellas muestras −minoritarias− que consideran fuera de rango ("outliers").

Se han propuesto algunas modificaciones de núcleos de SVM para enfrentarse a las dificultades relacionadas con el desequilibrio [Fung and Mangasarian, 2005] [Hong et al., 2007] [Wu and Chang, 2005] [Yang et al., 2009]. También se han modificado

SVMs difusas para ello, siendo [Batuwita, and Palade, 2010] un ejemplo interesante.

Por último, hay que destacar las cada vez más empleadas GANs. Este tipo de arquitectura utiliza dos modelos −el generador y el discriminador− que compiten/cooperan entre ellos compartiendo información en fase de entrenamiento para generar nuevos datos sintéticos de tal manera que, una vez completado el proceso de aprendizaje, el discriminador sea incapaz de distinguir los datos reales de los sintéticos. De forma general, este proceso se emplea para la generación de imágenes [Ali-Gombe and Elyan, 2019], pero puede aprovecharse este esquema de aprendizaje para adaptar las redes a otros tipos de datos, como las CTGANs ("Conditional Tabular Generative Adversarial Networks") [Xu et al., 2019] para datos tabulares.

2.3. Métricas de evaluación para problemas desequilibrados

La selección de métricas adecuadas es uno de los aspectos clave a la hora de evaluar las prestaciones de las máquinas de aprendizaje, pero esta tarea tiene una relevancia aún mayor cuando se trabaja con problemas desequilibrados. La característica principal de estos problemas es la predominancia de una de las clases en el conjunto de datos, algo que debe tenerse en cuenta para evaluar la calidad de la salida obtenida por la máquina. Por ello, a continuación se presentan las métricas más utilizadas a la hora de evaluar las prestaciones en este tipo de Problemas Singulares.

Por convenio, se establece la clase positiva C_1 como la minoritaria y la clase negativa C_0 como mayoritaria. Además, con el objetivo de facilitar el entendimiento de las posteriores métricas, en la Tabla 2.1 se muestra la matriz de confusión. Esta matriz proporciona de manera directa el número de muestras correcta e incorrectamente clasificadas en función de las etiquetas reales del problema y la decisión. Esto permite identificar los siguientes grupos: muestras positivas correctamente clasificadas/detectadas (TP, "True Positive"), falsos positivos (FP, "False Positive"), muestras positivas clasificadas como negativas (FN, "False Negative") y muestras

de la clase negativa correctamente clasificadas (TN, "True Negative").

	Decidir positivo: D_1	Decidir negativo: D_0
Etiqueta positiva: $t = C_1$	TP	FN
Etiqueta negativa: $t = C_0$	FP	TN

Tabla 2.1: Matriz de confusión.

La presencia de una clase mayoritaria supone que la evaluación del sistema por medio de la tasa de acierto ("Accuracy")

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.3}$$

sea totalmente desaconsejable. El motivo es evidente: eligiendo siempre en favor de la clase mayoritaria se obtendría una alta tasa de acierto −igual al porcentaje de datos de la clase mayoritaria−, pero donde nunca se detectaría la clase minoritaria −que suele ser la de mayor interés− debido al elevado valor de TN con respecto al resto de grupos.

A partir de esto, surgen otras métricas relacionadas con las características del clasificador. Son clásicas la probabilidad de falsa alarma y la probabilidad de detección, las cuales provienen de la tecnología radar.

La probabilidad de falsa alarma P_{FA} −también conocida como probabilidad de error de tipo I− representa la tasa de falsos positivos y se define como

$$P_{FA} = Pr(D_1 |\, t = C_0) = \int_{X_{D_1}} p(\mathbf{x}|C_0)\, d\mathbf{x} = \frac{FP}{FP + TN} \tag{2.4}$$

donde X_{D_1} representa las regiones del espacio observable donde se elige la clase positiva.

Por su parte, la probabilidad de detección P_D −también conocida como Sensibilidad− representa la tasa de muestras positivas detectadas correctamente

$$P_D = Pr(D_1 |\, t = C_1) = \int_{X_{D_1}} p(\mathbf{x}|C_1)\, d\mathbf{x} = \frac{TP}{TP + FN} \tag{2.5}$$

Estas dos probabilidades serán las que darán una mayor información acerca del buen funcionamiento del sistema respecto a la detección de la clase positiva. De igual manera se pueden definir sus análogas respecto a la clase negativa: la probabilidad de pérdida P_M −también conocida como probabilidad de error de tipo II y complementaria a la probabilidad de detección− y la probabilidad de clasificar correctamente las muestras negativas P_{D_N} −también conocida como Especificidad y complementaria a la probabilidad de falsa alarma−, definidas como:

$$P_M = 1 - P_D = Pr(D_0|\, t = C_1) = \int_{X_{D_0}} p(\mathbf{x}|C_1)\, d\mathbf{x} = \frac{FN}{TP + FN} \tag{2.6a}$$

$$P_{D_N} = 1 - P_{FA} = Pr(D_0|\, t = C_0) = \int_{X_{D_0}} p(\mathbf{x}|C_0)\, d\mathbf{x} = \frac{TN}{FP + TN} \tag{2.6b}$$

donde X_{D_0} representa las regiones del espacio observable donde se elige la clase negativa.

Las características de un problema desequilibrado marcan el objetivo de maximizar la probabilidad de detección sin comprometer los niveles de falsa alarma, que deben ser bajos. La curva ROC ("Receiver Operating Characteristic") −que se detallará desde un punto de vista estadístico en el siguiente capítulo− establece el compromiso existente entre ambas probabilidades a medida que se exploran todos los posibles valores del umbral de decisión. Esto permite fijar un punto de trabajo −típicamente un valor de falsa alarma− con el objetivo de diseñar el clasificador optimizando las prestaciones en dicho punto, algo que no permiten otras técnicas. Debe resaltarse que, en la práctica, todos los diseños operarán en un cierto punto de trabajo. La Figura 2.3 muestra un ejemplo de curva ROC.

A partir de la curva ROC, surge el área bajo la curva (ROC-AUC, "Area Under ROC Curve"), utilizada en numerosos estudios. Sin embargo, no se recomienda el uso de esta métrica para comparar las prestaciones entre distintos clasificadores. El motivo es que no tiene en cuenta ningún punto de trabajo y puede resultar demasiado optimista en situaciones en las que hay un alto nivel de detección a costa de un elevado número de falsos positivos, ya que otorga el mismo valor a todas las regiones de la

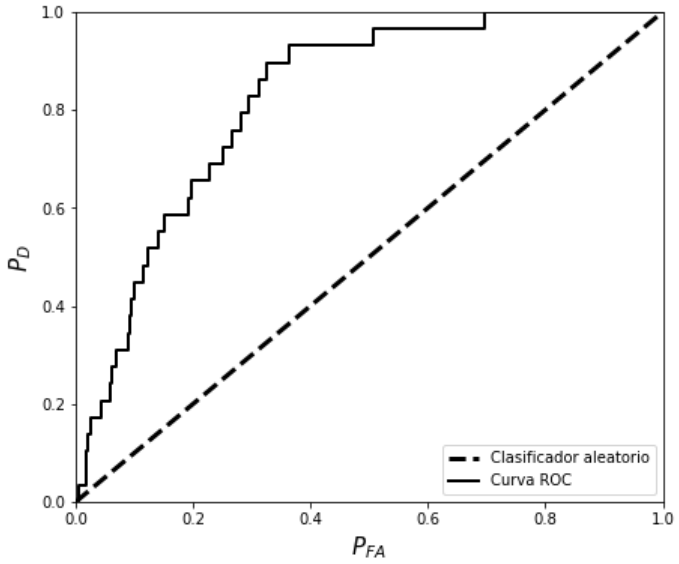


Figura 2.3: Ejemplo de Curva ROC. Cada punto de la curva está relacionado con un umbral de decisión. El clasificador debe estar entre el clasificador aleatorio (línea discontinua) y el ideal ($P_{FA} = 0,\ P_D = 1$).

curva ROC a la hora de calcular el área descrita por la misma.

Otra alternativa para evaluar las prestaciones en función de los casos detectados y los niveles de falsos positivos es la dupla de métricas compuesta por la Precisión ("Precision") −también conocida como Valor Positivo Predicho (PPV, "Positive Predictive Value")− y la Sensibilidad ("Recall") −también conocida como Exhaustividad, que coincide con la probabilidad de detección anteriormente descrita−, definidas como

$$Precisión = \frac{TP}{TP + FP} \tag{2.7a}$$

$$Sensibilidad = \frac{TP}{TP + FN} \tag{2.7b}$$

De igual manera que ocurre con la curva ROC con las probabilidades de detección y falsa alarma, el compromiso entre la Precisión y la Sensibilidad puede representarse por una curva que relacione ambas métricas y cuyos puntos estarán definidos por el valor del umbral de decisión. Por tanto, ambas métricas también permitirían realizar

el diseño del clasificador a partir de un punto de trabajo.

Por su parte, el Valor-F es una métrica que combina armónicamente la Precisión y la Sensibilidad por medio de

$$Valor - F = \frac{(1+\beta^2) \cdot Sensibilidad \cdot Precisión}{\beta^2 \cdot Precisión + Sensibilidad} \tag{2.8}$$

donde β (cuyo valor suele ser unitario) indica la importancia de la Precisión respecto a la Sensibilidad. Aunque su uso a la hora de evaluar problemas desequilibrados es muy común, esta métrica presenta dos grandes inconvenientes: por un lado, suele ser más sensible a cambios en la Precisión que en la Sensibilidad provocando la selección de modelos sub-óptimos [López et al., 2013] y, por otro lado, su valor está asociado a un punto de trabajo implícitamente fijado.

Como se ha dicho anteriormente, para este tipo de problemas no es aconsejable el uso de la tasa de acierto. Por ello, surge la tasa de acierto equilibrada (BA, "Balanced Accuracy") definida como

$$BA = \frac{P_D + P_{D_N}}{2} \tag{2.9}$$

cuyo uso está cada vez más extendido, ya que promedia los niveles de detección de ambas clases, aunque no permite valorar el comportamiento general del clasificador al estar vinculada a un punto de trabajo implícitamente fijado. No obstante, una vez elegido el punto de trabajo del sistema para optimizar su diseño, puede utilizarse como indicador de las prestaciones finales del clasificador.

Con el mismo objetivo de maximizar de manera equilibrada la tasa de acierto de ambas clases, surge la Media Geométrica (GM, "Geometric Mean")

$$GM = \sqrt{\frac{TP}{TP+FN} \cdot \frac{TN}{FP+TN}} \tag{2.10}$$

Desafortunadamente, la naturaleza simétrica de la distribución de GM sobre la Sensibilidad y la Especificidad dificulta el contraste de los distintos modelos según el acierto de cada clase [López et al., 2013]. Por ello, aparece la versión ajustada AGM

("Adjusted Geometric Mean"), definida como

$$AGM = \begin{cases} \frac{GM + P_{D_N} \cdot (FP+TN)}{1+FP+TN}, & si\ P_D > 0 \\ 0, & si\ P_D = 0 \end{cases} \tag{2.11}$$

Para finalizar el capítulo, se debe destacar la importancia de fijar un punto de trabajo a la hora de realizar el diseño de un clasificador −algo que se detallará en el siguiente capítulo− y la utilización de las métricas adecuadas para evaluar las prestaciones ante la presencia de desequilibrio.

Capítulo 3

Introducción al re-equilibrado fundamentado

En este capítulo, se presenta la metodología fundamentada para resolver problemas de clasificación binaria desequilibrada; metodología basada en la equivalencia del cociente de verosimilitudes, eje fundamental del trabajo realizado en esta Tesis y cuyo desarrollo teórico ha sido publicado en [Benítez-Buenache et al., 2019] [Benítez-Buenache et al., 2020] [1].

La mayoría de métodos de re-equilibrado presentados en el capítulo anterior son puramente empíricos y funcionan correctamente bajo determinadas circunstancias, pero ante cambios de las condiciones del problema sufren grandes limitaciones que, incluso, pueden llegar a degradar las prestaciones respecto al diseño directo (desequilibrado). Por contra, los métodos fundamentados, basados en la teoría estadística −de Thomas Bayes en este caso−, permiten construir soluciones robustas que mantienen buenas prestaciones ante posibles cambios en las condiciones del problema que se desea resolver.

[1] [Benítez-Buenache et al., 2020] es una corrección menor publicada por los propios autores. En [Benítez-Buenache et al., 2019] se definió un método enfatizado como método neutral, pero dicho método no garantiza la invarianza del cociente de verosimilitudes. Dicha denominación no implica ningún cambio en los experimentos realizados y ha sido directamente corregida en esta Tesis.

3.1. Breve introducción a la teoría de clasificación Bayesiana

En este apartado se revisa la teoría Bayesiana de clasificación en que se basa la metodología propuesta. Para ello, la descripción seguirá el formato empleado en el texto clásico [Van Trees, 1968].

Se considera que la variable aleatoria observable $\mathbf{x}$ pertenece a una de las dos clases, C_1 (positiva) o C_0 (negativa). Además, se considera que las probabilidades *a priori* $\{P_i\}$ y las verosimilitudes $\{p(\mathbf{x}|C_i)\}$, $i \in \{0,1\}$ son conocidas y se define la política de costes $\{c_{ji}\}$, $i,j \in \{0,1\}$ para el problema de clasificación. Esta política de costes indica el coste de seleccionar j cuando i es cierto, siendo $c_{ji} > c_{ii}$, es decir, el coste de equivocarse es mayor que el de acertar. Minimizando el coste medio de clasificación se llega al test de cociente de verosimilitudes

$$q_L(\mathbf{x}) = \frac{p(\mathbf{x}|C_1)}{p(\mathbf{x}|C_0)} \underset{C_0}{\overset{C_1}{\gtrless}} \frac{c_{10}-c_{00}}{c_{01}-c_{11}} \frac{P_0}{P_1} = Q_C Q_P = Q \tag{3.1}$$

donde $Q_P = P_0/P_1$ y $Q_C = (c_{10}-c_{00})/(c_{01}-c_{11})$, así como su producto Q, son valores no negativos. Por tanto, la clasificación se lleva a cabo comparando el cociente de verosimilitudes $q_L(\mathbf{x})$ con un umbral Q. Las prestaciones de este clasificador Bayesiano estarán definidas por la política de costes y las probabilidades de falsa alarma y detección −mencionadas en el capítulo anterior− y que se definen como sigue:

$$P_{FA} = Pr(decidir\ C_1|C_0\ es\ cierta) = \int_Q^\infty p(q_L|C_0)\, dq_L \tag{3.2a}$$

$$P_D = Pr(decidir\ C_1|C_1\ es\ cierta) = \int_Q^\infty p(q_L|C_1)\, dq_L \tag{3.2b}$$

respectivamente, donde $\{p(q_L|C_i)\}$ son las verosimilitudes para la variable aleatoria $q_L(\mathbf{x})$ (función de la variable aleatoria $\mathbf{x}$).

A partir de (3.1) se pueden resolver todos los problemas que tengan el mismo cociente de verosimilitudes, aunque sus costes o/y probabilidades *a priori* sean distintas. De hecho, los cambios en ambos factores únicamente afectan al valor de Q.

La Característica de Operación de Neyman-Pearson (NPOC) representa la solución analítica del problema por medio de la relación existente entre las probabilidades de detección P_D y falsa alarma P_{FA}. Variando Q desde $Q \to \infty$ ($P_D = 0$ para $P_{FA} = 0$) hasta $Q = 0$ ($P_D = 1$ para $P_{FA} = 1$) obtenemos esta curva, estrictamente creciente y convexa. Además, esta curva nos permite seleccionar un valor de P_{FA} y obtener el valor máximo alcanzable de P_D, ya que constituye la solución óptima para cualquier valor de Q. El valor de Q para dicho punto de trabajo P_{FAW} se obtiene por medio de la pendiente de la tangente a la curva en ese punto.

A partir de las verosimilitudes y las probabilidades *a priori* es sencillo obtener las probabilidades *a posteriori* según

$$Pr(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P_i}{p(\mathbf{x})} = \frac{p(\mathbf{x}|C_i)P_i}{p(\mathbf{x}|C_0)P_0 + p(\mathbf{x}|C_1)P_1} \tag{3.3}$$

Haciendo uso de la expresión anterior, se puede obtener una alternativa del clasificador de Bayes como

$$\frac{Pr(C_1|\mathbf{x})}{Pr(C_0|\mathbf{x})} \underset{C_0}{\overset{C_1}{\gtrless}} Q_C \tag{3.4a}$$

o, análogamente,

$$Pr(C_1|\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} \frac{Q_C}{Q_C + 1} \tag{3.4b}$$

ya que $Pr(C_0|\mathbf{x}) = 1 - Pr(C_1|\mathbf{x})$.

Considerando la clase C_1 como positiva/minoritaria y C_0 como negativa/mayoritaria −manteniendo la notación utilizada a lo largo de la Tesis−, el desequilibrio aparece cuando $Q \gg 1$, es decir, cuando $Q_P \gg 1$ y/o $Q_C \gg 1$. Puede resolverse fácilmente el problema aplicando (3.1), siempre y cuando $q_L(\mathbf{x})$ sea conocido. Esta afirmación también es cierta para máquinas generativas, es decir, máquinas que funcionan por medio de las estimaciones $\{\widehat{p}(\mathbf{x}|C_i)\}$ y $\{\widehat{P}_i\}$ obtenidas a partir del conjunto de muestras de entrenamiento. Su uso para la estimación de verosimilitudes está extendido. Sin embargo, sus prestaciones son generalmente peores que las máquinas discriminativas, ya que la estimación de $q_L(\mathbf{x})$ se hace estimando cada una de las verosimilitudes por separado y después, dividiéndolas, un proceso propenso a grandes errores numéricos.

Para definir el grado de desequilibrio del problema se utiliza el cociente de desequilibrio IR ("Imbalance Ratio"), el cual se expresa como

$$IR = Q_P \cdot Q_C = \frac{P_0}{P_1}\left(\frac{c_{10} - c_{00}}{c_{01} - c_{11}}\right) = Q \tag{3.5}$$

siendo Q_P y Q_C el cociente de probabilidades *a priori* y de la política de costes, respectivamente.

Una vez presentada la teoría estadística para resolver problemas de clasificación binaria por medio de (3.1), en los próximos apartados se sientan las bases para implementar métodos de re-equilibrado fundamentados.

3.2. Bases del re-equilibrado fundamentado

Las máquinas discriminativas de clasificación no lineales con transformaciones entrenables −como los MLPs− se diseñan utilizando una aproximación indirecta. La decisión en un problema de clasificación sigue la forma

$$o(\mathbf{x})_{\mathbf{w}^*} \underset{C_0}{\overset{C_1}{\gtrless}} \eta \tag{3.6}$$

donde η es la frontera de decisión y $o(\mathbf{x})_{\mathbf{w}^*}$ es la salida de una máquina cuyos parámetros $\mathbf{w}$ se optimizan por medio de

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{argmín}} \sum_n c\left(t^{(n)}, o(\mathbf{x})_{\mathbf{w}^*}\right) \tag{3.7}$$

siendo $\{t^{(n)}, \mathbf{x}^{(n)}\}$, $n = 1, ..., N$ las muestras etiquetadas disponibles para el entrenamiento y $c(t, o)$ el coste subrogado de entrenamiento, el cual se incrementa cuanto mayor es la disimilitud entre t y o. Se recomienda el uso de una tangente hiperbólica como función de activación de la salida para asegurar que la salida sea $-1 \leq o \leq 1$ si las etiquetas toman los valores $t = +1/-1$ para las clases C_1/C_0, respectivamente; como se hará aquí. Además de los riesgos de sub/sobreajuste comunes durante el aprendizaje, este clasificador es sensible al desequilibrio, es decir, sus prestaciones se pueden ver seriamente comprometidas cuando $Q \gg 1$, ya que las muestras

minoritarias tienen una muy reducida contribución al coste subrogado de (3.7), y, consecuentemente, la salida o no las considerará de manera apropiada.

Por ello, se presentan dos condiciones necesarias para poder resolver problemas desequilibrados por medio de (3.1) utilizando este tipo de clasificadores:

- Re-equilibrado neutral, es decir, la construcción de un problema menos desequilibrado que no modifique (sustancialmente) el cociente de verosimilitudes. En otras palabras, el proceso de re-equilibrado debe mantener el cociente de verosimilitudes invariante. En el siguiente apartado se profundiza en este tema, presentando aquellos métodos y procesos considerados neutrales.

- La aplicación de una máquina y, en particular, de un coste subrogado que haga posible una correcta estimación de $q_L(\mathbf{x})$. Esto puede realizarse utilizando divergencias de Bregman [Bregman, 1967] como coste subrogado, tal y como se detalla a continuación.

Entre las máquinas con transformaciones entrenables, los MLPs tienen teóricamente capacidades de representación ilimitadas [Cybenko, 1989] [Hornik et al., 1989]. Las extremadamente potentes DNNs, como versión profunda del MLP, también pertenecen a esta categoría. Para este tipo de máquinas, emplear divergencias de Bregman como coste subrogado es una condición suficiente y necesaria para obtener estimaciones de las probabilidades *a posteriori* −y, consecuentemente, del cociente de verosimilitudes− a partir de la salida de estas máquinas [Cid-Sueiro et al., 1999] [Cid-Sueiro and Figueiras-Vidal, 2001].

Para el caso binario, una divergencia de Bregman es una función $c_B(t, o)$ que cumple

$$\frac{\partial\, c_B(t,o)}{\partial\, o} = -g(o)(t-o) \tag{3.8}$$

donde $g(o) > 0$, es decir, debe ser una función de la salida estrictamente positiva.

La estimación de la etiqueta t por medio de una función $o_B(\mathbf{x})$ de la variable observable $\mathbf{x}$ para minimizar el coste subrogado $c(t, o)$ requiere minimizar el coste

medio

$$\int_{\mathbf{x}} \int_{t} c(t,o)\, p(\mathbf{x},t)\, d\mathbf{x}dt = \int_{\mathbf{x}} \left[\int_{t} c(t,o)\, p(t|\mathbf{x})\, dt \right] p(x)d\mathbf{x} \tag{3.9}$$

y, aceptando que $c(t,o)$ es no negativo, es suficiente con minimizar la integral interior, ya que $p(\mathbf{x})$ no puede ser negativo:

$$o_B(\mathbf{x}) = \underset{o}{\text{argmín}} \int c(t,o)\, p(t|\mathbf{x})\, dt \tag{3.10}$$

Obviamente, siempre se puede escribir

$$\frac{\partial\, c(t,o)}{\partial\, o} = -g(t,o)(t-o) \tag{3.11}$$

definiendo de forma adecuada $g(t,o)$. Asumiendo que la integral de (3.11) es (absolutamente) convergente, la solución es

$$\int g(t,o)\, (t-o)\, p(t|\mathbf{x})\, dt = 0 \tag{3.12}$$

Si $c(t,o)$ es una divergencia de Bregman, $g(t,o) = g(o)$, (3.12) se convierte en

$$g(o_B) \int (t - o_c)\, p(t|\mathbf{x})\, dt = 0 \tag{3.13}$$

y, como $g(o_B) \neq 0$, se tiene

$$\int o_B\, p(t|\mathbf{x})\, dt = o_B(\mathbf{x}) = \int t\, p(t|\mathbf{x})\, dt = E\{t|\mathbf{x}\} \tag{3.14}$$

donde E indica la esperanza estadística. En otro caso, $o_B(x) \neq E\{t|\mathbf{x}\}$.

Por tanto, se comprueba que las divergencias de Bregman como coste subrogado son una condición suficiente y necesaria para obtener una estimación de la esperanza *a posteriori* de la etiqueta a partir de la salida de la máquina

$$o_B(\mathbf{x}) = E\{t|\mathbf{x}\} \tag{3.15}$$

Algunos ejemplos de divergencias de Bregman incluyen varios costes subrogados popularmente utilizados, como el error cuadrático $(t-o)^2$ o la entropía (simétrica), la cual, para $t \pm 1$ y $-1 \leqslant o \leqslant 1$, tiene la forma

$$c_E(t,o) = -(1+t)log_e(1+o) - (1-t)log_e(1-o) \tag{3.16}$$

En la práctica, una máquina cuyos parámetros entrenables minimizan la versión muestral de (3.7) generará una estimación de la esperanza *a posteriori* a la salida. Esta estimación será mejor cuanto mayor sea el conjunto de muestras de entrenamiento y mayor capacidad de representación tenga la máquina.

Para un clasificador binario con etiquetas $t = \{\pm 1\}$, la esperanza *a posteriori* viene dada por

$$E\{t|\mathbf{x}\} = (+1)\, Pr(C_1|\mathbf{x}) + (-1)\, Pr(C_0|\mathbf{x}) = 2\, Pr(C_1|\mathbf{x}) - 1 \tag{3.17}$$

dando lugar al estimador

$$\widetilde{Pr}(C_1|\mathbf{x}) = \frac{1}{2}(o_B(\mathbf{x}) + 1) \tag{3.18}$$

obtenido a partir de la salida de la máquina de clasificación. A partir de (3.3), se tiene

$$Pr(C_1|\mathbf{x}) = \frac{q_L(\mathbf{x})}{q_L(\mathbf{x}) + Q_P} \tag{3.19}$$

e incluyendo Q_C, que puede tomarse como una Q_P, se llega a

$$\widetilde{q}_L(\mathbf{x}) = \widetilde{Q}\, \frac{\widetilde{Pr}(C_1|\mathbf{x})}{1 - \widetilde{Pr}(C_1|\mathbf{x})} = \widetilde{Q}\, \frac{1 + o_B(\mathbf{x})}{1 - o_B(\mathbf{x})} \tag{3.20}$$

donde $\widetilde{Q} = \widetilde{Q}_P\, \widetilde{Q}_C$, siendo $\widetilde{Q}_P$ y $\widetilde{Q}_C$ los cocientes de las poblaciones y los costes de ambas clases del problema asociado −en general, la situación correspondiente a un proceso de re-equilibrado− utilizado para obtener $\widetilde{Pr}(C_1|\mathbf{x})$.

La estimación de $\widetilde{q}_L(\mathbf{x})$ se obtiene para una versión re-equilibrada de manera neutral, siendo también válida para el problema desequilibrado original. Por tanto, la decisión del clasificador se obtiene comparando (3.20) con el umbral de la parte derecha de (3.1). Además, es posible establecer un criterio de decisión basado en la salida de la máquina de clasificación como

$$o_B(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} \frac{\widehat{Q} - \widetilde{Q}}{\widehat{Q} + \widetilde{Q}} = \eta \tag{3.21}$$

donde $\widehat{Q}$ es el umbral para el problema desequilibrado original, es decir, $\widehat{Q} = \widehat{Q}_P\, \widehat{Q}_C$, siendo $\widehat{Q}_P$ una estimación del cociente de las poblaciones originales (por ejemplo,

a partir de las frecuencias relativas) y $\widehat{Q}_C$ se calcula con los costes originales. Se destaca el hecho de que la regla de clasificación proviene de la salida de la máquina entrenada bajo condiciones de re-equilibrado, cambiando el umbral de decisión.

La fórmula (3.21) proporciona la solución óptima (aproximada) para el problema desequilibrado original utilizando la salida de la máquina que se ha entrenado para resolver el problema de clasificación re-equilibrado de manera neutral. La calidad de esta solución dependerá de la calidad de la estimación $\widetilde{q}_L$ del cociente de verosimilitudes definido en (3.20). De forma general, esta estimación será mejor cuanto mayor sea la capacidad expresiva de la máquina, el número de muestras sea mayor y en las regiones del espacio de observación mejor representadas de manera muestral. Además, un número limitado de dimensiones también favorecerá que la estimación sea mejor. En cualquier caso, esta aproximación estima el cociente de verosimilitudes y no cada una de las verosimilitudes de forma separada, como en los métodos generativos.

Llegado a este punto, el diseñador tiene completa libertad para seleccionar el valor de $\widehat{Q}$, es decir, la política de costes o, desde otra perspectiva, el punto de trabajo en la NPOC, la cual se produce cambiando el valor del umbral desde 1 hasta -1 en (3.21). Si fuese necesario, se puede obtener $\widehat{Pr}(C_1|\mathbf{x})$, la probabilidad *a posteriori* de la clase minoritaria para el problema desequilibrado, utilizando (3.19) y (3.20).

El análisis anterior confirma que las dos condiciones mencionadas al comienzo del apartado −uso de costes de Bregman y re-equilibrado neutral− son necesarias y suficientes para entrenar una máquina que proporcione una estimación fundamentada del cociente de verosimilitudes de un problema desequilibrado de clasificación binaria por medio de un problema re-equilibrado asociado. El proceso, por tanto, es el siguiente:

1) Aplicar un re-equilibrado neutral apropiado con el objetivo de obtener un problema de clasificación más sencillo con el mismo cociente de verosimilitudes.

2) Entrenar una máquina de clasificación utilizando una divergencia de Bregman

como coste subrogado que permita estimar de manera fundamentada la probabilidad *a posteriori* según (3.19) y, consecuentemente, el cociente de verosimilitudes según (3.20).

3) Resolver el problema desequilibrado original comparando el cociente de verosimilitudes con el umbral que corresponda. Una buena opción para ello es seleccionar un punto de trabajo en la NPOC −estimada por medio del movimiento del valor del umbral desde ∞ hasta 0. De manera alternativa, también se puede utilizar la salida de la máquina, según (3.21).

Siguiendo los pasos anteriores puede estimarse la NPOC teórica, la cual representa la solución analítica al problema de clasificación. La estimación de la NPOC a partir de la salida de la máquina se define como curva Característica de Operación de la Máquina (MOC, "Machine Operating Characteristic"). En general, no puede asegurarse que la MOC descrita por una máquina entrenada en otras condiciones (sin neutralidad ni costes de Bregman) sea una estimación de la NPOC. El proceso definido anteriormente compensa el re-equilibrado para obtener la solución del problema original (desequilibrado) de manera fundamentada.

Cuando cualquiera de las dos condiciones no aplica, el proceso de re-equilibrado no tiene una base fundamentada. La consecuencia de esto es una degradación de las prestaciones, que en algunas ocasiones pueden llegar a ser peores que la resolución directa del problema desequilibrado. Los siguientes apartados ilustran la importancia de ambas condiciones.

3.3. Re-equilibrado neutral e informado

En el Capítulo 2 se han presentado las distintas familias de métodos de re-equilibrado que conforman el estado del arte para hacer frente a los problemas de clasificación desequilibrada. Ahora, se presenta una nueva taxonomía de dichos métodos, clasificándolos entre neutrales e informados. Se denominan métodos neutrales

de re-equilibrado aquellos que mantienen invariante el cociente de verosimilitudes, mientras que los métodos informados no aseguran que dicha condición se cumpla. A continuación se detallan ambos tipos de métodos, presentando los algoritmos más representativos de cada uno de ellos.

3.3.1. Métodos de re-equilibrado neutral

Como ya se ha mencionado a lo largo de este capítulo, una de las dos condiciones suficientes y necesarias de la metodología fundamentada propuesta es la aplicación de métodos neutrales de re-equilibrado, es decir, aquellos que mantienen invariante el cociente de verosimilitudes. Para ello, todas las muestras de la misma clase deben ser tratadas de igual manera, evitando así la distorsión del cociente de verosimilitudes que puede aparecer al enfatizar el proceso de re-equilibrado en determinadas regiones. Seguidamente se presentan algunos de estos métodos.

Debido a su sencilla aplicación, la asignación de pesos (o ponderación) a cada una de las clases es una técnica muy extendida a la hora de re-equilibrar un problema de clasificación. Sin embargo, para realizarlo de manera neutral, los pesos deben asignarse uniformemente a todas las muestras de la misma clase, para no alterar su densidad de probabilidad. De hecho, esta técnica es equivalente a modificar la probabilidad *a priori* de la clase, lo cual supone un cambio en el umbral de clasificación.

Por otro lado, aparecen los métodos de re-muestreo, los cuales deben seleccionar las muestras con igual probabilidad para cada clase para considerarse neutrales (en promedio). Es decir, todas las muestras de una clase deben tener la misma probabilidad de ser eliminadas −en caso de sub-muestreo− o repetidas −para el sobre-muestreo−. Además, son equivalentes a la ponderación de muestras, tal y como observó por primera vez Leo Breiman en [Breiman et al., 1984]. No obstante, para ser efectivamente neutrales requieren la aplicación de conjuntos con diversidad en el re-muestreo de las poblaciones, algo que aumentará la calidad de los resultados de clasificación y que atenuará el efecto de algunas versiones extremas de re-muestreo (o generación), cuyas diferencias prácticas se detallan en [Dal Pozzolo et al., 2015],

[Japkowicz and Stephen, 2002], [Ling and Li, 1998] y [Wallace et al., 2011]. Las técnicas de "Bootstrap"también se consideran dentro de esta categoría de métodos de re-muestreo neutral.

Lo mismo ocurre con los métodos de generación, que son neutrales (en promedio) si la probabilidad de generar nuevas muestras y los mecanismos de generación son iguales para todas las muestras de una misma clase. Nuevamente, las prestaciones aumentarán con el uso de conjuntos que diversifiquen la generación de muestras entre sus aprendices. Dentro de este grupo aparece la técnica clásica de generación mediante ventanas de Parzen [Parzen, 1962], ya que la ventana es la misma −en forma y varianza− para todas las muestras. No obstante, como método (aproximadamente) neutral destaca SMOTE [Chawla et al., 2002], ya que todas las muestras tienen la misma probabilidad de ser seleccionadas (así como sus K vecinos más próximos) para la generación de las nuevas muestras sintéticas.

Por último, se destaca la posibilidad de combinar estos tres tipos de procedimientos, reduciendo así el riesgo de deformaciones menores de la verosimilitud y manteniendo las ventajas de la diversidad, como se verá más adelante. No obstante, no puede asegurarse de manera general cuál es la mejor combinación, ya que es un aspecto de diseño dependiente del problema.

3.3.2. Métodos de re-equilibrado informado

Los métodos de re-equilibrado informados, que pretenden mejorar las prestaciones, no son neutrales, es decir, no aseguran la invarianza del cociente de verosimilitudes. La principal motivación de este tipo de métodos es dar mayor importancia, aumentando la intensidad del re-equilibrado, a regiones del espacio observable que se consideran más críticas a efectos del re-equilibrado. Estas regiones −generalmente cercanas a la frontera entre ambas clases− se determinan por medio de un clasificador sencillo, como un K-NN (K vecinos más próximos) o un esquema similar, que se aplica directamente sobre las muestras de entrenamiento. Algunos ejemplos que incluyen métodos de re-muestreo informado directo son [Japkowicz, 2000], [Jo and

Japkowicz, 2004], [Kubat and Matwin, 1997] y [Laurikkala, 2001].

No muy distintas de este tipo de técnicas son algunas versiones informadas de SMOTE, que definen los criterios de selección de muestras minoritarias de cara a la generación según diferentes estrategias.

MWMOTE ("Majority Weighted Minority Oversampling Technique") [Barua et al., 2014] identifica las muestras minoritarias más difíciles de clasificar asignando pesos según su distancia euclídea respecto a las muestras de la clase mayoritaria más cercanas.

Por su parte, MSMOTE [Hu et al., 2009] clasifica las muestras minoritarias en tres grupos: seguras, fronterizas y ruidosas. Después, seleccionando aleatoriamente una muestra segura, genera la muestra sintética en el espacio existente entre ella y la muestra fronteriza más cercana.

Borderline-SMOTE [Han et al., 2005], quizás (junto con ADASYN) la versión alternativa de SMOTE más extendida, genera muestras sintéticas únicamente a partir de las muestras (de la clase minoritaria) que considera "en peligro". Este tipo de muestras son aquellas para las que al menos la mitad de sus m vecinos más próximos son de la clase mayoritaria. Además, excluye aquellas muestras cuyos m vecinos más próximos son de la clase mayoritaria, al considerarlas como casos ruidosos. Por tanto, cuanto más bajo sea el valor del parámetro m, más restrictivo (e informado) será el método, siendo su comportamiento más parecido a SMOTE cuanto mayor es dicho parámetro m.

En cuanto a ADASYN [He et al., 2008], la selección de la muestra que se utilizará para llevar a cabo la generación se realiza de acuerdo con una distribución obtenida de manera proporcional al número de muestras de la clase mayoritaria entre sus K vecinos más próximos, a diferencia de SMOTE, cuya selección aleatoria se ejecuta respecto a una distribución uniforme. De esta manera, todas las muestras minoritarias que estén cercanas al menos a una muestra de la clase mayoritaria podrán ser seleccionadas para la generación, siendo mayor su probabilidad cuanto más alejadas estén de otras muestras minoritarias.

Otras versiones de SMOTE incluyen pre- y post-procesado de muestras [Batista et al., 2004] [Ramentol et al., 2012] y modificaciones informadas de las características de generación [Bunkhumpornpat et al., 2009] [Bunkhumpornpat et al., 2012]. Además, existen métodos híbridos que combinan distintos diseños bajo condiciones diferentes [Provost and Fawcett, 2001].

Hay muchas pruebas experimentales de las mejoras en las prestaciones que pueden aportar estos métodos informados. Sin embargo, tienen numerosas limitaciones intrínsecas que suponen ciertos riesgos implícitos en su aplicación directa. Son procedimientos que pueden dar grandes resultados para casos concretos, pero que ante el más mínimo cambio en las condiciones del problema (poblaciones, muestras disponibles o políticas de costes) pueden verse altamente degradados, ya que el problema asociado que resuelven no es equivalente (en su cociente de verosimilitudes) al problema original, pues la verosimilitud ha sido modificada. Por tanto, este tipo de métodos no pueden utilizarse en la metodología fundamentada que se ha presentado en este capítulo. Asimismo, la curva MOC obtenida a la salida de estas máquinas no es una estimación de la NPOC, de tal forma que si se selecciona un punto de trabajo modificando el umbral de dicha curva MOC, se corre el riesgo de obtener resultados deficientes.

3.3.3. Diversidad y conjuntos en el re-equilibrado fundamentado

Por último, debe mencionarse que el re-equilibrado fundamentado es totalmente compatible con la aplicación de diversidad, siendo sus ventajas extensibles a la resolución de problemas desequilibrados. La forma de llevarlo a cabo es aprovechar los aspectos aleatorios de los métodos de re-equilibrado −por ejemplo, cada vez que SMOTE genera una muestra elige una muestra minoritaria de forma aleatoria, así como el vecino más próximo utilizado para generar la muestra sintética− para construir distintas poblaciones re-equilibradas para el entrenamiento de cada uno de los aprendices.

El proceso de agregación se lleva a cabo a partir de las salidas de cada uno de los

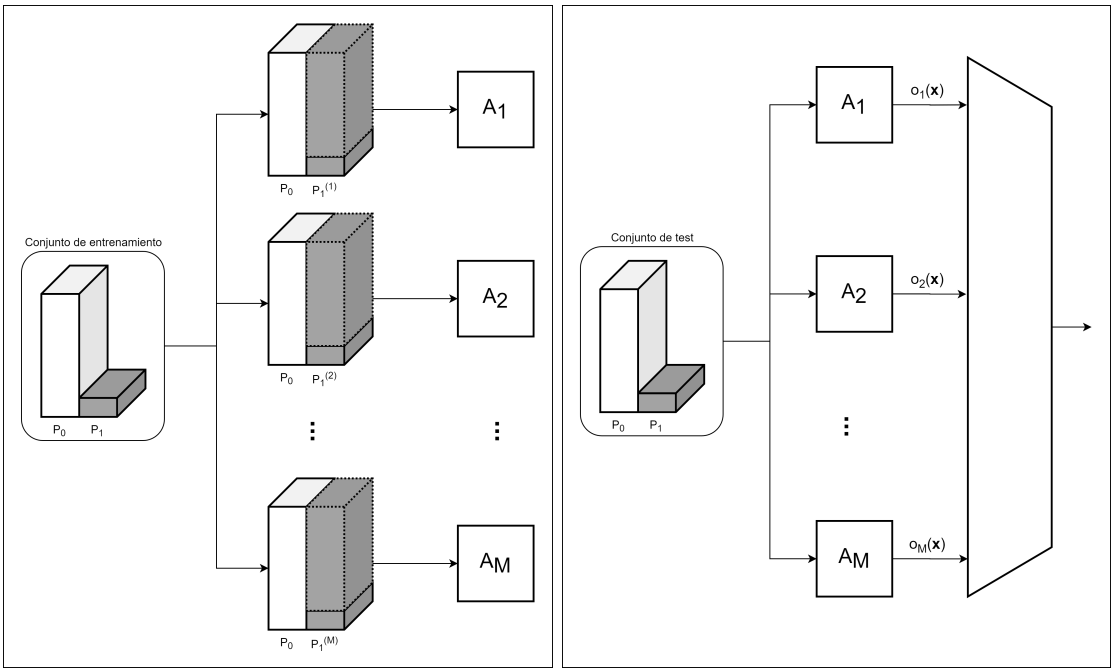


Figura 3.1: Diversidad en el re-equilibrado. (a) Por medio de algoritmos de generación, se crean M poblaciones distintas de la clase minoritaria para el entrenamiento, utilizadas para entrenar M aprendices. (b) Se utilizan los M aprendices para clasificar el conjunto (desequilibrado) de test, realizando el proceso de agregación a partir de sus salidas.

aprendices. No obstante, debido a la relación que se ha demostrado entre la salida y las estimaciones fundamentadas de la probabilidad *a posteriori* según (3.19) y del cociente de verosimilitudes según (3.20), surgen distintas alternativas para llevar a cabo la agregación de M aprendices:

- Promedio directo de las salidas obtenidas según

$$o_A(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} o_m(\mathbf{x}) \tag{3.22a}$$

siendo $o_m(\mathbf{x})$ la salida del aprendiz m, $m = \{1, ..., M\}$.

- Promedio directo de la estimación *a posteriori* según

$$\widetilde{Pr}_A(C_1|\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\widetilde{Pr}_m(C_1|\mathbf{x}) \tag{3.22b}$$

  siendo $\widetilde{Pr}_m(C_1|\mathbf{x})$ la estimación de la probabilidad *a posteriori* del aprendiz m, $m = \{1, ..., M\}$. Equivale al anterior (ver (3.18)).

- Promedio directo del cociente de verosimilitudes según

$$\widetilde{q_L}_A(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\widetilde{q_L}_m(\mathbf{x}) \tag{3.22c}$$

  siendo $\widetilde{q_L}_m(\mathbf{x})$ la estimación del cociente de verosimilitudes del aprendiz m, $m = \{1, ..., M\}$.

- En caso de utilizar un punto de trabajo determinado, obtener el umbral de decisión asociado. A partir de la salida de cada aprendiz, obtener la salida dura y realizar voto por mayoría para seleccionar la clase correspondiente.

La Figura 3.1 ilustra dicho proceso, donde los M aprendices que forman el conjunto se entrenan con M poblaciones re-equilibradas distintas. Una vez entrenados dichos aprendices, se utilizan para clasificar los nuevos datos haciendo uso del conjunto y aplicando la agregación como se ha descrito anteriormente.

3.3.4. Efectos de un re-equilibrado informado

Con el objetivo de ilustrar los efectos del re-equilibrado informado, se presenta un problema sintético cuya NPOC puede obtenerse analíticamente.

Dadas las verosimilitudes de las clases

$$p(\mathbf{x}|C_1) = \begin{cases} \frac{1}{4}(1 + x_2), & x_1, x_2 \in [-1, 1] \\ 0, & \text{en otro caso} \end{cases} \tag{3.23a}$$

$$p(\mathbf{x}|C_0) = \begin{cases} \frac{1}{2,8}, & x_1 \in [-0{,}7, 0{,}7], x_2 \in [-1, 1] \\ 0, & \text{en otro caso} \end{cases} \tag{3.23b}$$

puede obtenerse la expresión analítica de la NPOC con la relación existente entre la probabilidad de detección P_D y la probabilidad de falsa alarma P_{FA}, según

$$P_{FA} = Pr(D_1|H_0) = \int_{-0{,}7}^{0{,}7} \int_{U}^{1} \frac{1}{2{,}8}\, d_{x_2}\, d_{x_1} = \frac{1}{2} - \frac{U}{2} \rightarrow U = 1 - 2P_{FA} \tag{3.24a}$$

$$P_D = Pr(D_1|H_1) = \int_{-0{,}7}^{0{,}7} \int_{U}^{1} \frac{1}{4}(1 + x_2)\, d_{x_2}\, d_{x_1} + 2\int_{0{,}7}^{1} \int_{-1}^{1} \frac{1}{4}(1 + x_2)\, d_{x_2}\, d_{x_1} = \\ - 0{,}175U^2 - 0{,}35\, U + 0{,}525 + 2 \cdot 0{,}15 \tag{3.24b}$$

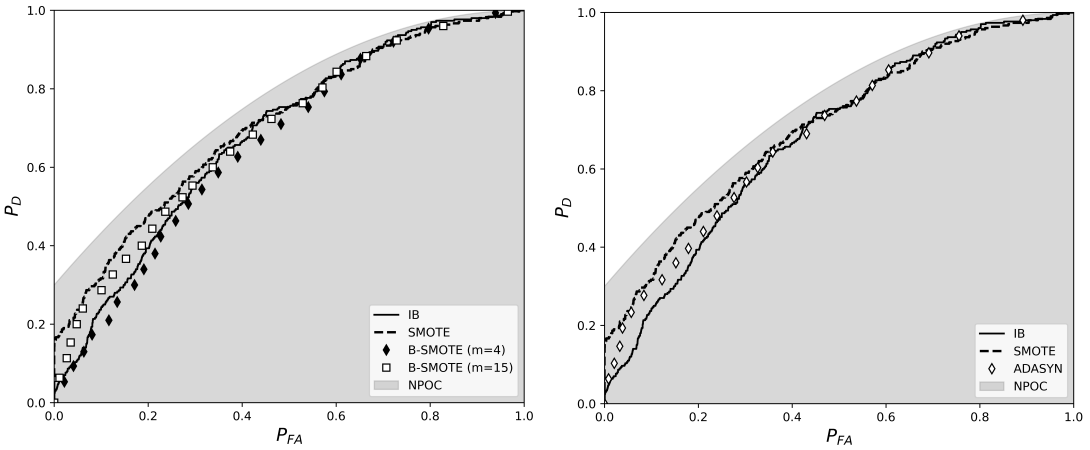
$$\Rightarrow P_D = 0{,}825 - 0{,}35\,(1 - 2P_{FA}) - 0{,}175\,(1 - 2P_{FA})^2 \tag{3.24c}$$

La curva descrita por esta expresión de la NPOC se considerará como referencia.

A partir de (3.23a) y (3.23b) se generan los conjuntos de entrenamiento y test, ambos con $IR = 64$ y $N_1 = 300$ y $N_0 = 19200$ muestras para las clases C_1 (positiva/minoritaria) y C_0 (negativa/mayoritaria), respectivamente.

Se emplea un conjunto de 11 clasificadores, donde cada aprendiz es un MLP compuesto por una única capa oculta con 6 neuronas, un número apropiado dada la baja complejidad de este ejemplo. Cada máquina se entrena por medio del algoritmo RMSProp, con una tasa de aprendizaje de $\mu = 10^{-3}$. El coste subrogado es el error cuadrático (cumpliendo la condición de Bregman). Para obtener las ventajas de la diversidad, cada aprendiz se entrena con un conjunto de datos distinto −cuando se emplea generación de datos− y se promedian las salidas obtenidas por cada uno de ellos.

En cuanto a los métodos de re-equilibrado, se emplearán aquí distintas técnicas de generación de datos sintéticos: SMOTE como método (aproximadamente) neutral y Borderline-SMOTE y ADASYN como métodos informados. Para todos ellos se aplica re-equilibrado completo con $\widetilde{Q} = 1$. Se selecciona $K = 3$ (para el número de vecinos más próximos) y, en el caso de Borderline-SMOTE, se consideran los valores $m = 4$



(a) **(b)**

Figura 3.2: Curvas MOC de los clasificadores con re-equilibrado informado. El borde del área sombreada corresponde con la NPOC. El resto de curvas son IB (problema original sin re-equilibrado), SMOTE y dos versiones de B-SMOTE en (a) y ADASYN en (b).

y $m = 15$ para evaluar el comportamiento respecto al valor de dicho parámetro. Se recuerda que cuanto mayor sea m, más parecido es respecto a SMOTE.

Para evaluar las prestaciones de cada uno de los métodos, se obtienen sus curvas MOC ("Machine Operating Characteristic"), ordenando las muestras según el valor de su salida y desplazando el umbral desde 1 a -1. Cada curva representará, por tanto, una aproximación a la NPOC tras aplicar cada uno de los métodos de re-equilibrado anteriores.

La Figura 3.2 muestra las curvas MOC obtenidas con el conjunto de test tras aplicar cada proceso de generación de muestras en el conjunto de entrenamiento, además de la NPOC teórica (3.24c). Se observa una degradación de todas las curvas respecto a la NPOC para valores bajos de P_{FA}. Si se evalúa el comportamiento de SMOTE, puede advertirse una clara mejora (un incremento de la capacidad de detección en torno al 0.1) respecto al diseño directo (IB) que va diluyéndose a medida que aumenta el nivel de falsa alarma hasta juntarse ambas MOCs alrededor de $P_{FA} = 0.5$. En cambio, cuando se evalúan los métodos informados, se aprecia una degradación

de sus prestaciones respecto a SMOTE para $0 < P_{FA} < 0.5$. Cabe destacar el caso de Borderline-SMOTE para $m = 4$, ya que se degrada, incluso, respecto al diseño directo en el que no se lleva a cabo ningún proceso de re-equilibrado. Mejora cuando la selección de muestras "en peligro" es menos restrictiva con $m = 15$, donde se observa una menor degradación con respecto a SMOTE. Lo mismo ocurre con ADASYN, el cual es menos restrictivo a la hora de seleccionar las muestras que forman parte de la generación de datos sintéticos, obteniendo una MOC muy similar a la descrita por Bordeline-SMOTE con $m = 15$. Estas degradaciones demuestran que las prestaciones obtenidas al aplicar un re-equilibrado informado pueden ser insatisfactorias.

En [Benítez-Buenache et al., 2019] se realiza el mismo proceso con una versión del problema con mayor solapamiento de las verosimilitudes. Los resultados obtenidos muestran el mismo comportamiento en las curvas MOC.

3.4. Efectos de no aplicar costes de Bregman

Para este ejemplo se selecciona la base de datos "Electricity" del repositorio OpenML [Harries et al., 2009]. Se trata de una base datos grande y (prácticamente) equilibrada, con la cual será sencillo obtener una estimación de la NPOC sin necesidad de re-equilibrado. La etiqueta indica si hay una subida o bajada del precio de la electricidad y está compuesta por $N_0 = 26075$ (para subidas) y $N_1 = 19237$ (para bajadas) muestras de un espacio observable de 8 dimensiones. A partir de la base de datos, se realiza una división estratificada −para mantener el IR original− del 80/20 % para los conjuntos de entrenamiento y test, respectivamente.

Se estima la NPOC para el conjunto de test tal y como se ha explicado anteriormente (es decir, ordenando las salidas del conjunto de test) tras diseñarse un clasificador a partir de los datos de entrenamiento. Una vez estimada, se llevan a cabo dos procesos de sub-muestreo aleatorios con el objetivo de obtener dos problemas desequilibrados con una cantidad distinta de datos disponibles. El sub-muestreo se realiza para que ambos casos tengan $IR \approx 50$, obteniendo los siguientes conjuntos

de datos:

- Problema 1: Cuenta, finalmente, con $N_1 = 261$ muestras positivas y $N_0 = 13037$ muestras negativas para el conjunto de entrenamiento.
- Problema 2: Con $N_1 = 33$ y $N_0 = 1629$ muestras positivas y negativas, respectivamente.

Para ambos problemas se utiliza el conjunto de test original para evaluar las prestaciones.

Se utilizan SMOTE ($K = 3$) y ponderación como métodos para un re-equilibrado completo $\widetilde{Q} = 1$. Para la técnica de ponderación se aplica, por tanto, un peso $w_{(+)} = IR$ para las muestras minoritarias −manteniendo $w_{(-)} = 1$ como peso de la clase mayoritaria− en el coste subrogado durante el proceso de entrenamiento de la máquina.

En cuanto a la máquina de aprendizaje, se emplea un MLP con una única capa oculta de 7 neuronas, parámetro seleccionado por tanteo para resolver este problema. Como se ha mencionado, el objetivo de este ejemplo es ilustrar los efectos de aplicar un coste subrogado que no sea de Bregman. Para ello, se define la combinación convexa

$$c(t, o) = \alpha(t - o)^2 + (1 - \alpha)|t - o| \tag{3.25}$$

con $0 \leq \alpha \leq 1$, como coste subrogado para el entrenamiento del MLP. Cuando $\alpha = 1$, se emplea el error cuadrático, el cual es un coste de Bregman. Por otro lado, cuando $\alpha \neq 1$ no se trata de un coste de Bregman, y cuanto más disminuye α más alejado está del error cuadrático, hasta llegar al error absoluto cuando $\alpha = 0$.

En la Figura 3.3 se presentan los resultados obtenidos. En todos los casos, resulta obvio el aumento de la degradación a medida que disminuye α, evidenciando la necesidad de utilizar un coste de Bregman como coste subrogado para un re-equilibrado fundamentado. Además, estas degradaciones son aún mayores cuando el número de muestras disponible es más limitado (conjunto de entrenamiento más

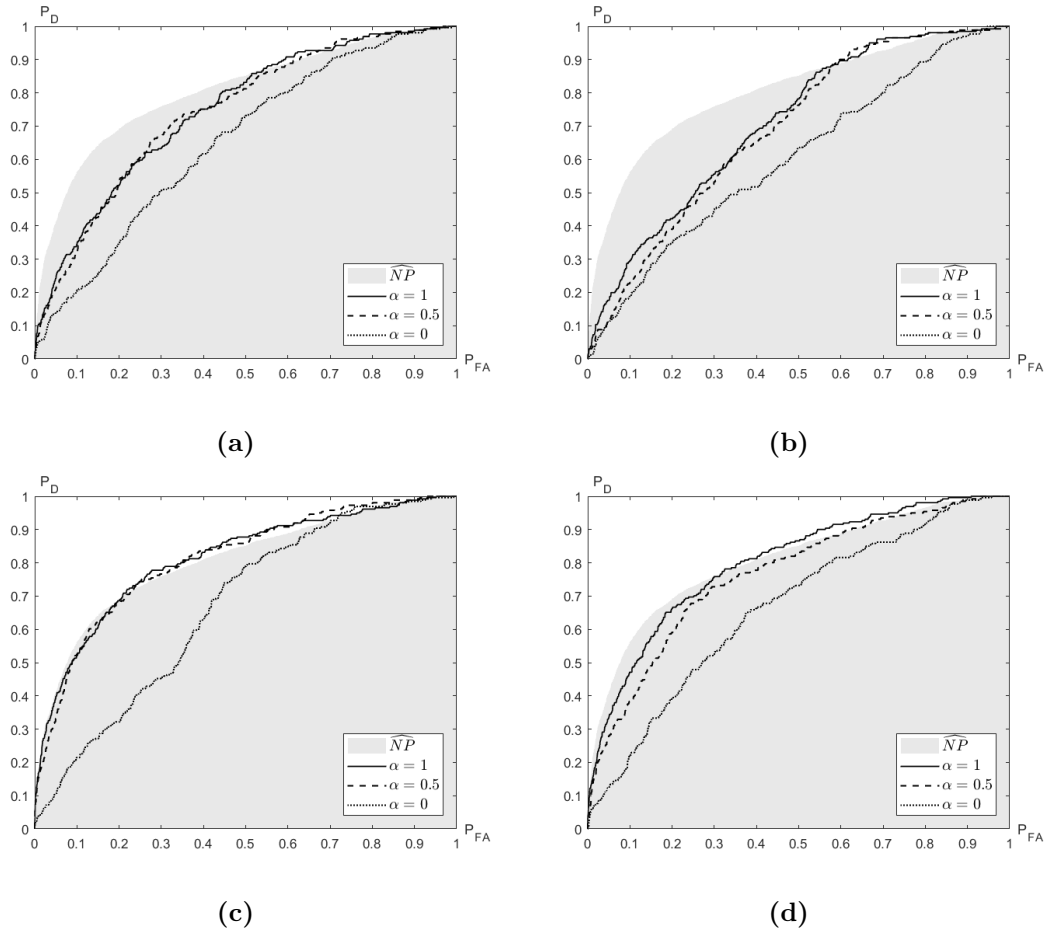


Figura 3.3: Curvas MOC para los clasificadores con distintos valores de α para el coste subrogado (3.25) para cada problema y método de re-equilibrado. El borde de la zona sombreada corresponde a la estimación de la NPOC para el problema original [Harries et al., 2009]. Re-equilibrado por ponderación para el Problema 1 (a) y para el Problema 2 (b). Re-equilibrado con SMOTE para el Problema 1 (c) y para el Problema 2 (d).

pequeño del Problema 1) y, para este caso, SMOTE ofrece mejores prestaciones que el re-equilibrado basado en costes. Por lo tanto, los resultados obtenidos son acordes a lo esperado, avalando la teoría que se defiende en esta Tesis.

3.5. Observaciones sobre el re-equilibrado mediante SVMs y conjuntos

Las Máquinas de Vectores Soporte (SVMs) están diseñadas bajo un coste bisagra, que no es una divergencia de Bregman. Por tanto, la aproximación fundamentada expuesta no puede aplicarse a SVMs, que requiere un análisis completamente distinto. Pueden considerarse estructuras lineales en los parámetros entrenables, y esto conduce a una solución de la forma

$$\sum_{n^*} \alpha_{n^*} t^{(n^*)} k(\mathbf{x}, \mathbf{x}^{(n^*)}) = \boldsymbol{\beta}^T \mathbf{k}(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} U \tag{3.26a}$$

para la SVM básica. $k(.,.)$ es un núcleo de Mercer y los pesos no nulos (posiblemente dispersos) $\{\alpha_{n^*}\}$ se obtienen por medio de un proceso de optimización que maximiza el margen entre las muestras de entrenamiento en torno a la frontera de clasificación, siendo $\{\mathbf{x}^{(n^*)}\}$ los vectores soporte. El criterio de clasificación en (3.26a) no está basado en el cociente de verosimilitudes. La parte izquierda de la fórmula no es necesariamente una función estrictamente creciente desde 0 hasta ∞. No obstante, es posible escribir

$$f(\boldsymbol{\gamma}^T \mathbf{k}(\mathbf{x})) \underset{C_0}{\overset{C_1}{\gtrless}} f(U') \tag{3.26b}$$

donde f es una función no lineal, estrictamente creciente y, por tanto, invertible, f: $(min\{\boldsymbol{\gamma}^T k(x)\}, max\{\boldsymbol{\gamma}^T k(x)\}) \rightarrow (0, \infty)$. La fórmula (3.26b) es posible si la capacidad de representación del núcleo es suficientemente alta, como ocurre normalmente. Desde esta perspectiva, (3.26a) se asemeja a (3.26b). Aplicando f^{-1} en ambas partes de (3.26b) se llega a

$$\boldsymbol{\gamma}^T \mathbf{k}(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} U' \tag{3.26c}$$

y γ podría relacionarse con β, así como U' con U, si f fuese conocida.

Un potencial problema es la selección del núcleo $k(.,.)$ de forma que (3.26b) pueda aproximarse mediante una función f. Un análisis en mayor profundidad de dicha conexión proporcionaría probablemente información sobre la selección o/y el

diseño del núcleo, incluyendo la forma y modificaciones que se han propuesto para problemas desequilibrados [Fung and Mangasarian, 2005] [Hong et al., 2007] [Wu and Chang, 2005] [Yang et al., 2009]. Aún así, el carácter lineal de (3.26a) significa que las SVMs son clasificadores estables y, consecuentemente, tienen una sensibilidad reducida respecto a las dificultades producidas por el desequilibrio [Japkowicz and Stephen, 2002]. Sin embargo, a pesar de esta ventaja, la imposibilidad de aplicar la equivalencia del cociente de verosimilitudes persiste.

De manera similar a las SVMs, algunos conjuntos de máquinas tampoco pueden incluir la equivalencia del cociente de verosimilitudes. Un ejemplo de ello son los conjuntos de "Boosting", cuya agregación de los aprendices se realiza mediante un coste que no cumple las condiciones de Bregman, lo que implica que el cociente de verosimilitudes no es aplicable. Para un estudio más exhaustivo, se recomienda la lectura de [Galar et al., 2012].

3.6. Re-equilibrado en dos pasos

Como se ha demostrado con los ejemplos anteriores, un re-equilibrado informado puede distorsionar el cociente de verosimilitudes. No obstante, es posible compensar estas distorsiones, aunque sigue habiendo cierto riesgo en ello. La ponderación muestral permite determinar un peso para cada una de las muestras, independientemente de la clase a la que pertenezca. De esta manera, a la hora de diseñar el clasificador, se incorpora dicho peso en el coste subrogado a minimizar por medio de

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmín}} \sum_n e(\mathbf{x})\, c\left(t^{(n)}, o(\mathbf{x})_{\mathbf{w}^*}\right) \tag{3.27}$$

siendo $e(\mathbf{x})$ una función de énfasis dependiente de la variable observable. Esta compensación puede llevarse a cabo en regiones donde la estimación de la NPOC haya sido suficientemente buena. Si el punto de trabajo seleccionado está fuera de estas regiones, la estimación de la NPOC en dicho punto estará totalmente alejada de lo teórico, obteniendo resultados incluso peores que con la aplicación directa de

procedimientos informados.

Como consecuencia de lo anterior, surge la idea de un proceso en dos pasos para aplicar esquemas de re-equilibrado informado. El primer paso utilizará un re-equilibrado neutral para hacer la estimación de la NPOC, permitiendo seleccionar un punto de trabajo razonable. Una vez elegido dicho punto, se puede aplicar un esquema de re-equilibrado informado apropiado. Además, si fuese necesario, se puede incluir una compensación del cociente de verosimilitudes.

3.6.1. Compensación del re-equilibrado informado

En este apartado se presenta un método para compensar los efectos negativos del re-equilibrado informado para obtener estimaciones precisas de la NPOC. Este análisis permitirá el uso de métodos de re-equilibrado informados que, primero, estimarán (a través de la MOC) las muestras cercanas a un punto de trabajo para, después, aplicar un segundo paso de re-equilibrado informado enfatizando dichas muestras. Como resultado, se obtendrá una nueva estimación de la NPOC, que será mejor en las zonas cercanas al punto de trabajo seleccionado, con el objetivo de mejorar las prestaciones en dicho punto.

De manera general, el efecto de un método de re-equilibrado informado en el diseño de un clasificador es ponderar las muestras de la clase minoritaria en el proceso de optimización del problema por medio de un factor $e_1(\mathbf{x})$ (para re-muestreo o generación). La función de ponderación $e_1(\mathbf{x})$ depende de la muestra de entrenamiento $\mathbf{x}$ y será mayor cuando $\mathbf{x}$ se encuentre en la región seleccionada por el método aplicado. Para definir la región, primero ha de seleccionarse un valor de falsa alarma como punto de trabajo P_{FAW} en la estimación de la NPOC (proporcionada tras la aplicación de un método de re-equilibrado neutral como primer paso). Además, se incluye en este análisis otro posible factor $e_0(\mathbf{x})$ para las muestras de la clase mayoritaria. Se asume que estos factores son distintos de cero para cualquier valor de $\mathbf{x}$. Por tanto,

el cociente de verosimilitudes teórico para el problema re-equilibrado es

$$q'_L(\mathbf{x}) = \frac{p(\mathbf{x}|C_1)e_1(\mathbf{x})/a_1}{p(\mathbf{x}|C_0)e_0(\mathbf{x})/a_0} = \frac{q_L(\mathbf{x})q_E(\mathbf{x})}{Q_A} \tag{3.28}$$

donde $q_E(\mathbf{x}) = e_1(\mathbf{x})/e_0(\mathbf{x})$, $Q_A = a_1/a_0$, y a_1 y a_0 son constantes de normalización. Se hace notar que los $e_i(\mathbf{x})$ modifican los perfiles de verosimilitud y que, por tanto, se consideran las funciones resultantes como nuevas verosimilitudes, normalizándolas para asegurar el volumen unitario.

En consecuencia, siempre que se utilice una divergencia de Bregman como coste subrogado, se puede estimar $q'_L(\mathbf{x})$ usando (3.20) como

$$\hat{q}'_L(\mathbf{x}) = \frac{\widehat{q}_L(\mathbf{x})q_E(\mathbf{x})}{\widehat{Q}_A} = \tilde{Q}_2(\mathbf{x})\frac{1+o_{B2}(\mathbf{x})}{1-o_{B2}(\mathbf{x})} \tag{3.29}$$

siendo $o_{B2}(\mathbf{x})$ la salida del clasificador en el segundo paso.

Si −como es habitual− se entrena el segundo clasificador bajo un criterio MAP (mínima probabilidad de error), una vez ponderadas las muestras, resulta obvio que $\widetilde{Q}_2(\mathbf{x}) = q_E(\mathbf{x})$, de tal modo que el test resulta:

$$\frac{1+o_{B2}(\mathbf{x})}{1-o_{B2}(\mathbf{x})} \underset{C_0}{\overset{C_1}{\gtrless}} \frac{\widehat{Q}}{\widehat{Q}_A} \tag{3.30}$$

Como en el caso neutral, se puede trabajar directamente con $o_{B2}(\mathbf{x})$ para obtener el test equivalente:

$$o_{B2}(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} \frac{\widehat{Q}-\widehat{Q}_A}{\widehat{Q}+\widehat{Q}_A} = \eta_2 \tag{3.31}$$

El test anterior permite estimar la NPOC ordenando las salidas por su valor y moviendo el umbral η_2 entre 1 y -1.

La selección apropiada de los pesos $e_i(\mathbf{x})$ y la posición del punto de trabajo estarán condicionadas por la estimación previa de la NPOC (aplicando el re-equilibrado neutral).

Por último, como ya se ha mencionado, la implementación del sistema anterior requiere la estimación de los parámetros de normalización a_1 y a_0. Una manera

sencilla es utilizar estimaciones basadas en muestras para un determinado esquema de ponderación, como

$$\hat{a}_i = \frac{1}{N_i} \sum_{n_i} e_i(\mathbf{x}^{(n_i)}) \tag{3.32a}$$

para $i = \{0, 1\}$, donde $\mathbf{x}^{(n_i)}$ son las muestras de entrenamiento y N_i el número de muestras de la clase i.

Este esquema es válido tanto para métodos de ponderación como para re-muestreo o/y generación de datos sintéticos. De hecho, para los dos últimos la aplicación de conjuntos será muy ventajosa a causa de la diversidad. No obstante, en caso de utilizar un proceso de generación,

$$\hat{a}_i = \frac{1}{N_i'} \sum_{n_i'} e_i'(\mathbf{x}^{(n_i')}) \tag{3.32b}$$

donde $e_i'(\cdot)$ excluye la tasa de generación, $\{\mathbf{x}^{(n_i')}\}$ es la muestra $\mathbf{x}^{(n_i)}$ del conjunto de entrenamiento y aquellas generadas a partir de ella, y N_i' es el número total de muestras de entrenamiento para la clase i. La fórmula (3.32a) es aceptable si las muestras generadas están suficientemente cercanas al conjunto de entrenamiento original. Una estimación errónea de los factores de normalización puede afectar a las prestaciones del clasificador, por lo que se debe prestar atención a este proceso.

3.6.2. Procedimiento de re-equilibrado en dos pasos

La estimación del cociente de verosimilitudes $q_L'(\mathbf{x})$ es más precisa donde la densidad de las muestras utilizadas (incluyendo sus pesos) sea mayor. Sin un buen clasificador previo capaz de determinar las muestras cercanas a la frontera de trabajo deseada, existe un riesgo elevado de modificar la densidad de las muestras de manera inapropiada. Dicho clasificador debe estar diseñado bajo las condiciones de un coste subrogado de Bregman y re-equilibrado neutral. Esta es la base del procedimiento de re-equilibrado en dos pasos que se presenta en este apartado.

Por tanto, los pasos necesarios son los siguientes:

Paso 1:

1.1. Se diseña un clasificador auxiliar, con salida $o_{B1}(\mathbf{x})$, bajo las condiciones necesarias −coste subrogado de Bregman y re-equilibrado neutral−. Este clasificador mantiene inalterado el cociente de verosimilitudes y, por tanto, proporcionará una estimación precisa de la NPOC.

1.2. Se selecciona el punto de trabajo P_{FAW} y la ventana de operación (simétrica o asimétrica) en torno a dicho punto.

1.3. Se determinan las funciones de ponderación $e_i(\mathbf{x})$ a partir de la estimación de la NPOC y el punto de trabajo seleccionado.

Paso 2:

2.1. Se aplican las $e_i(\mathbf{x})$ y se entrena el segundo clasificador con un coste subrogado de Bregman.

2.2. Se obtiene el valor de η_2^* para conseguir P_{FAW}.

Operación:

Se aplica $o_{B2}(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} \eta_2^*$

Conviene destacar la principal diferencia existente entre este proceso y otros procesos de re-equilibrado informado, como pueden ser Borderline-SMOTE o ADASYN. Mientras que los citados métodos informados seleccionan sus muestras de acuerdo a la posición de las muestras minoritarias con respecto a la clase mayoritaria, el proceso descrito da mayor peso a aquellas muestras cercanas al punto de trabajo, manteniendo el objetivo fundamental de llevar a cabo la mejor estimación posible de la NPOC en dicho punto P_{FAW}. Asimismo, durante el proceso de diseño hay total libertad para experimentar con distintas formas de ponderación informada y utilizarlas en función de su rendimiento. De este modo, es posible utilizar formas de ponderación paramétricas tales como las que tienen perfiles similares a las empleadas eficazmente

para mejorar el "Boosting" [Ahachad et al., 2017] y clasificación con SDAE ("Stacked Denoising Auto-Encoding") [Alvear-Sandoval and Figueiras-Vidal, 2018], extender versiones previas de énfasis [Gómez-Verdejo et al.(2006)] [Gómez-Verdejo et al., 2008], y determinar los parámetros a través de un proceso de validación cruzada.

Por último, se destaca que este proceso de re-equilibrado en dos pasos proporciona diseños que son adecuados únicamente para el punto de trabajo seleccionado. Esto supone que ante un cambio en la selección del punto de trabajo debe realizarse un nuevo diseño. Por tanto, al contrario que con los procesos neutrales de re-equilibrado en un paso, la solución obtenida no sería consistente para otras condiciones de trabajo.

3.6.3. Versión enfatizada del re-equilibrado en dos pasos

Basándose en el proceso de re-equilibrado en dos pasos anterior, se propone una alternativa simplificada. El objetivo nuevamente es el de enfatizar de manera informada las muestras correspondientes a la región elegida en torno a punto de trabajo. La diferencia radica en que este énfasis se aplicará a todas las muestras cuya salida se encuentre en los rangos establecidos, independientemente de la clase a la que pertenezcan. Por tanto, el segundo clasificador enfatizará las muestras de dicha región, con el objetivo de obtener una mejor estimación de la NPOC en torno al punto de trabajo.

Así, los pasos a seguir son los siguientes:

Paso 1:

1.1. Mismo proceso que el método anterior: se re-equilibra de manera neutral con costes de Bregman y se fija un punto de trabajo P_{FAW}.

1.2. Se establece una ponderación de la forma

$$\widehat{Q}_A \begin{cases} > 1 & \text{para muestras dentro del margen en torno a } P_{FAW} \\ = 1, & \text{fuera de ese margen} \end{cases}$$

Paso 2:

2.1. Se realiza un nuevo re-equilibrado $\widetilde{Q}_2$ y ponderación sobre todas las muestras según la $\widehat{Q}_A$ fijada en el paso previo.

2.2. Se diseña el segundo clasificador utilizando un coste subrogado de Bregman.

Operación:

Se aplica

$$o_{B2}(\mathbf{x}) \underset{C_0}{\overset{C_1}{\gtrless}} \frac{\widehat{Q} - \widetilde{Q}_2\widehat{Q}_A}{\widehat{Q} + \widetilde{Q}_2\widehat{Q}_A} = \eta_2' \tag{3.33}$$

En cuanto al énfasis, hay varios aspectos de diseño que serán dependientes del problema, como su perfil, la intensidad y la ventana en torno al punto de trabajo donde se aplica.

A raíz de este método, surge la posibilidad de realizar un enfatizado neutral. Para ello, habría que realizar una neutralización uniforme del énfasis en torno a P_{FAW},

- o definiendo entornos que igualen el cociente N_{ei}/N_i, es decir, que haya la misma proporción de muestras de una clase que de otra;
- o enfatizando de manera distinta, de forma que se compense la diferencia.

En todo caso, para una mayor flexibilidad, permitir formas no neutrales como la anterior parece la mejor idea.

3.7. Prueba de concepto con re-equilibrado completo

Para finalizar, se lleva a cabo una prueba de concepto con un problema del mundo real que ilustre todo el proceso descrito a lo largo del capítulo. Con ese fin, se emplea la base de datos "BNG: PageBlocks"[Holmes et al., 2014]. El problema a resolver es la clasificación de los elementos de las páginas de un documento utilizando 10 variables numéricas como el ancho, la altura, la longitud o el número de píxeles negros,

entre otras. A pesar de que el problema original es multiclase, se trabaja únicamente con las clases "Bloque 1"(el bloque mayoritario con $N_0 = 265174$ muestras) como negativa, y "Bloque 5"($N_1 = 6238$ muestras) como positiva, resultando un problema desequilibrado ($IR = 42.5$) de clasificación binaria. Para obtener los conjuntos de entrenamiento y test, se divide de manera estratificada para mantener el IR original con una proporción de 75/25 %.

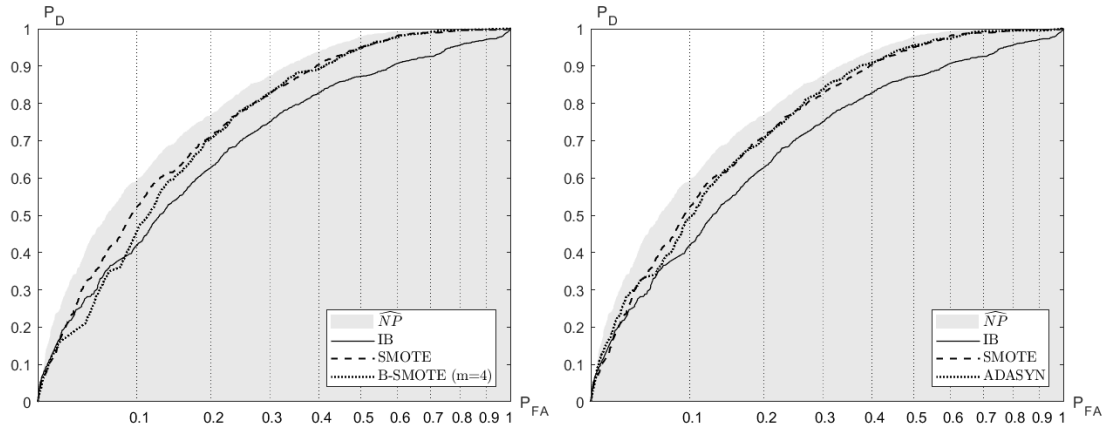
Para todas las pruebas que siguen a continuación, se emplea la misma arquitectura de máquina de aprendizaje, formada por un conjunto de 5 MLPs de 27 neuronas en su única capa oculta. El coste subrogado es el error cuadrático, cumpliendo así la condición de Bregman.

Se utiliza el conjunto de entrenamiento re-equilibrado (por completo) con SMOTE ($K = 3$) para estimar la NPOC sobre el conjunto de test, estableciendo así la curva de referencia para el resto del experimento. Una vez estimada la NPOC, se reduce a una tercera parte el número de muestras disponibles del conjunto de entrenamiento para las siguientes pruebas.

3.7.1. Re-equilibrado en el primer paso: neutral e informado

En un primer paso, se aplica SMOTE ($K = 3$) como re-equilibrado neutral y Borderline-SMOTE ($K = 3$ y $m = 4$) y ADASYN ($K = 3$) como métodos informados. Con todos ellos se aplica re-equilibrado completo ($\widetilde{Q} = 1$).

La Figura 3.4 muestra las curvas MOC obtenidas por cada uno de los diseños de re-equilibrado, además de la estimación de la NPOC realizada con el conjunto de entrenamiento completo y el diseño directo sin re-equilibrado. Los resultados cumplen con lo esperado: SMOTE, como método neutral, obtiene la mejor estimación de la NPOC para valores de falsa alarma pequeños. A medida que se incrementa P_{FA} las diferencias entre el método neutral y los informados se reducen. Además, se observa que las prestaciones de Borderline-SMOTE para niveles bajos de P_{FA} es incluso peor que el diseño con el conjunto de datos original desequilibrado. Aunque esto no ocurre con ADASYN, su desventaja respecto a SMOTE también es clara. La



(a) **(b)**

Figura 3.4: Curvas MOC de los diseños de re-equilibrado del primer paso. El borde del área sombreada representa la estimación de la NPOC. Las máquinas de las que se muestran resultados han sido entrenadas: sin re-equilibrado (IB), con re-equilibrado por medio de SMOTE, con Borderline-SMOTE en (a) y con ADASYN en (b).

ventaja observable de ADASYN respecto a Borderline-SMOTE se atribuye al menor número de muestras excluidas por ADASYN a la hora de seleccionar las muestras "en peligro", 82, mientras que Borderline-SMOTE excluye 485. Por supuesto, el diseño directo (IB) ofrece pobres resultados.

3.7.2. Re-equilibrado en dos pasos

En el segundo paso se aplica un proceso de re-equilibrado por medio de SMOTE ($K = 3$), pero sólo sobre las muestras cuya salida se encuentre en un rango en torno al umbral asociado al punto de trabajo seleccionado. A partir de los resultados obtenidos en el paso anterior, se fija el punto de trabajo en $P_{FAW} = 0.15$ para el segundo paso.

Una vez fijado dicho punto, es necesario establecer la ventana en torno a P_{FAW} en términos de simetría y tamaño, una tarea no trivial y que supone un gran esfuerzo a la hora de obtener el rango más adecuado. Tras realizar un minucioso estudio,

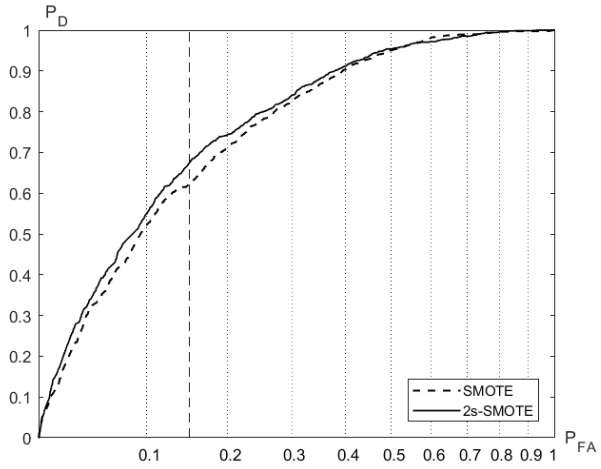


Figura 3.5: 2s-SMOTE es la curva MOC obtenida tras aplicar SMOTE con las muestras que en el primer paso se encuentran entre $P_{FA} = 0$ y $P_{FA} = 0.3$. El punto de trabajo se fija en $P_{FAW} = 0.15$. Se incluye la MOC obtenida en el primer paso para facilitar la comparación.

se determina como ventana óptima la comprendida entre $P_{FA} = 0$ y $P_{FA} = 0.3$. La Figura 3.5 muestra la curva MOC obtenida con el re-equilibrado informado (sobre las muestras que en el primer paso se encontraban en dicha ventana), donde claramente se aprecia una mejora respecto al primer paso.

Como se mencionó en los apartados anteriores, este tipo de diseños en dos pasos no son totalmente consistentes frente a cambios en las condiciones del problema y, en caso de que esto ocurra, es necesario volver a entrenar el segundo clasificador. Para ilustrarlo, se fija un nuevo punto de trabajo en $P_{FAW} = 0.1$, imponiendo así una reducción de 0.05 en la probabilidad de falsa alarma. De nuevo, se encuentran dificultades para obtener una ventana adecuada para el problema, por lo que se decide explorar incluso ventanas asimétricas. Tras un prolongado proceso de búsqueda de la ventana, se establecen dos rangos de umbrales: uno simétrico para $P_{FA} \in [0, 0.2]$ y otro asimétrico para $P_{FA} \in [0, 0.25]$.

En la Figura 3.6 se muestran los resultados obtenidos para el nuevo punto de trabajo con ambas ventanas (simétrica y asimétrica). Se observa una clara mejora de

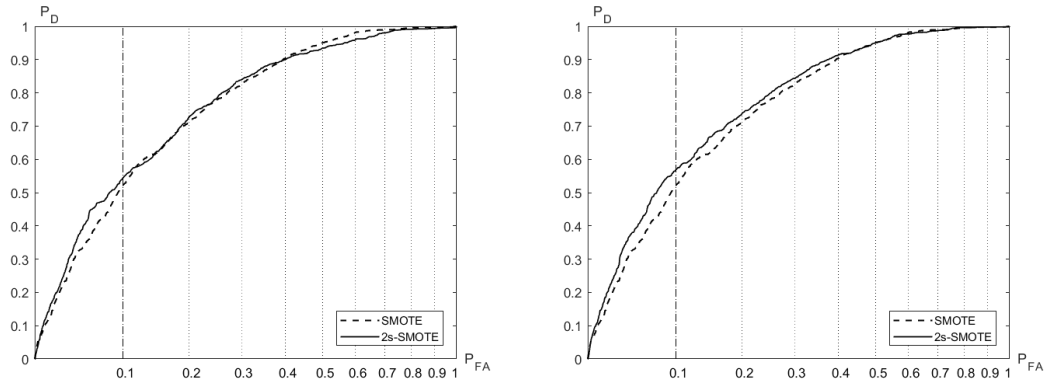
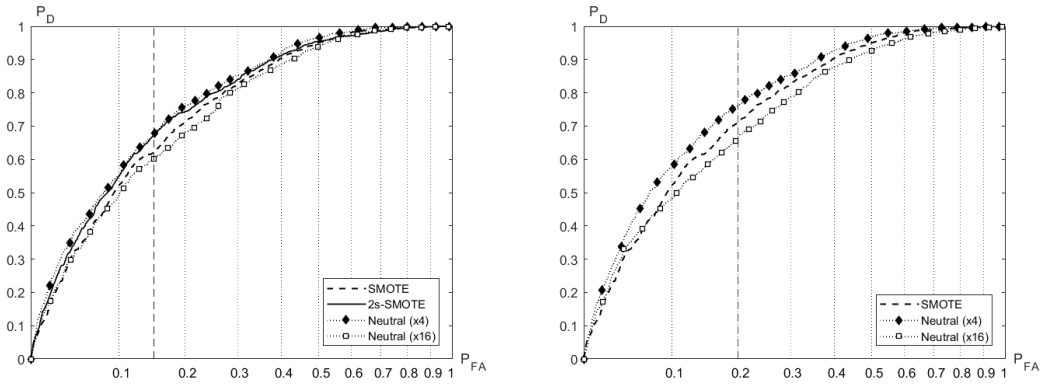


Figura 3.6: 2s-SMOTE es la curva MOC para el segundo re-equilibrado con SMOTE aplicado en las muestras que se encuentran en el primer paso en los intervalos: (a) $P_{FA} \in [0,\ 0.2]$ (simétrica) y (b) $P_{FA} \in [0,\ 0.25]$ (asimétrica). El punto de trabajo se sitúa en $P_{FAW} = 0.1$. Se incluye la MOC obtenida en el primer paso para facilitar la comparación.

las prestaciones respecto al primer paso, siendo aún mayor en el caso de la ventana asimétrica.

Por último, se repite el segundo paso de la clasificación empleando una versión enfatizada de re-equilibrado. Nuevamente, se parte de los resultados del primer paso del apartado anterior y se re-equilibra con SMOTE ($K = 3$) en el segundo paso para las muestras comprendidas en una determinada ventana en torno a P_{FAW}, pero en este caso se incluye un peso para todas las muestras (de ambas clases) situadas en dicho rango de umbrales. Para evaluarlo, se establecen dos niveles de intensidad: $\widehat{Q}_A = \{4, 16\}$.

La Figura 3.7a muestra los resultados para el punto de trabajo $P_{FAW} = 0.15$ y una ventana $P_{FA} \in [0.1,\ 0.2]$. Además, se incluyen las curvas MOC del primer paso con SMOTE y el segundo paso obtenido en la anterior prueba. Se observa como la versión enfatizada con un peso $\widehat{Q}_A = 4$ (que pondera las muestras de ambas clases) tiene unas prestaciones muy similares a las del diseño del segundo paso sin ponderación. Por contra, se observa que $\widehat{Q}_A = 16$ supone un empeoramiento en la



(a) **(b)**

Figura 3.7: Curvas MOC para el segundo paso de re-equilibrado con la versión enfatizada. (a) Aplicación de SMOTE y dos niveles de ponderación (4 y 16) sobre las muestras que en el primer paso estaban en $P_{FA} \in [0.1, 0.2]$, siendo $P_{FAW} = 0.15$ el punto de trabajo. Las curvas MOC de SMOTE del primer paso y la versión en dos pasos se incluyen para facilitar la comparación. b) Resultados correspondiente al mismo proceso para $P_{FAW} = 0.2$ y enfatizando las muestras en $P_{FA} \in [0.15, 0.25]$.

estimación de la NPOC respecto al primer paso. Este efecto era de esperar, ya que un énfasis demasiado elevado puede llegar a deformar seriamente las verosimilitudes y, por ello, producir una peor estimación del cociente de verosimilitudes.

Para finalizar, la Figura 3.7b demuestra que el método es efectivo para diferentes puntos de trabajo.

Capítulo 4

Diseño completo mediante algoritmos fundamentados

En el Capítulo 3 se ha presentado una metodología fundamentada para resolver problemas desequilibrados de clasificación binaria. Las dos condiciones suficientes y necesarias que fundamentan la metodología son el uso de divergencias de Bregman como coste subrogado y un re-equilibrado neutral. Para demostrar su validez y correcto funcionamiento, se han presentado varios ejemplos ilustrativos y una prueba de concepto con una base de datos real. Sin embargo, por sencillez, únicamente se utilizaba un método de re-equilibrado completo ($\widetilde{Q} = 1$), lo que no es necesariamente la mejor opción. En este capítulo, se extiende el estudio con el objetivo de maximizar las prestaciones de los diseños resultantes, optimizando los parámetros del clasificador y combinando distintas técnicas de re-equilibrado. Además, se analizan las posibles limitaciones que pueden aparecer ante circunstancias adversas como la presencia de ruido −algo común en los datos obtenidos por medio de sensores−, un número muy limitado de datos disponibles para entrenar el sistema o la alta dimensión del espacio de observación.

El contenido de este capítulo ha sido recientemente aceptado para su publicación [Benítez-Buenache et al., 2021].

4.1. Selección del punto de trabajo

La estimación de la NPOC por medio de la metodología fundamentada presentada en el Capítulo 3 permite la selección de un punto de trabajo P_{FAW}, con el objetivo de maximizar la probabilidad de detección en dicho punto. El criterio de selección del punto de trabajo puede variar en función del tipo de problema que se quiere resolver. No obstante, en general se debe trabajar con valores de falsa alarma pequeños, ya que un elevado número de falsos positivos puede suponer un alto sobrecoste, por ejemplo, el tiempo de un operario en solucionar una falsa incidencia o la necesidad de pruebas sanitarias adicionales tras un diagnóstico médico erróneo. Por tanto, la selección de P_{FAW} puede estar condicionada por niveles de calidad impuestos previamente, por el coste o beneficio asociado a las métricas de detección y falsa alarma o, simplemente, por el criterio del analista durante el diseño. Asimismo, un diseño en dos pasos −como los propuestos en el capítulo anterior− puede ayudar a su elección, basando la selección del punto de trabajo en la estimación de la NPOC obtenida en el primer paso.

Por otro lado, algunas características del problema pueden dificultar la selección del punto de trabajo. Es destacable el número de muestras disponibles: Un número muy reducido −del orden de decenas− de la clase minoritaria supone una resolución en la estimación de la NPOC muy baja, situación en la que típicamente aparecen curvas MOC escalonadas. Esto se produce cuando los valores con distinto efecto que puede tomar el umbral η en (3.21) están muy limitados debido al número tan reducido de muestras.

Por todo ello, se propone llevar a cabo la selección de los distintos parámetros de diseño $\mathbf{d}^*$ de la máquina de aprendizaje (arquitectura y parámetros no entrenables del clasificador, intensidad del re-equilibrado y métodos de re-equilibrado utilizados) por medio de

$$\mathbf{d}^* = \underset{\mathbf{d}}{\text{argmáx}}\{P_D | P_{FA} = P_{FAW}\} \tag{4.1}$$

De esta forma, el diseño primará la calidad de la estimación de la NPOC en el punto

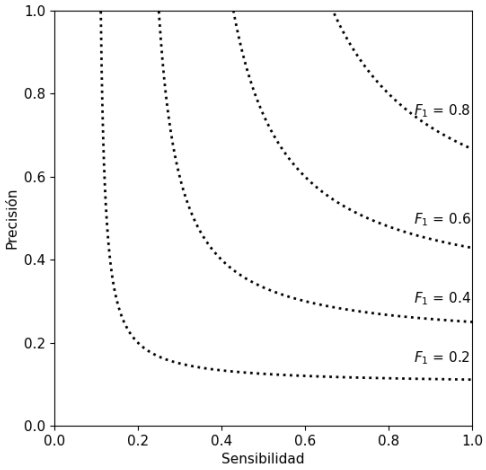


Figura 4.1: Representación de varias cotas del Valor-F1 en función de la relación entre Precisión y Sensibilidad obtenidos.

de trabajo elegido.

Sin embargo, el diseñador puede utilizar otras métricas alternativas a la hora de diseñar su clasificador para resolver problemas desequilibrados, como las indicadas al final del Capítulo 2. Como se mencionó entonces, es recomendable utilizar métricas a partir de la salida blanda del sistema para poder seleccionar el umbral adecuado para conseguir las prestaciones deseadas o requeridas. El Valor-F, comúnmente utilizado en problemas desequilibrados, no aporta suficiente información por sí solo, ya que sus valores intermedios se consiguen con combinaciones de niveles de Precisión y Sensibilidad −se recuerda que el Valor-F es su media armónica− muy dispares. Una evidencia de ello es la Figura 4.1, donde se representan las distintas cotas que puede tener dicha métrica en función de los valores de Precisión y Sensibilidad.

Por último, cabe mencionar que el uso del área bajo la curva (AUC) −tanto de la MOC como de la curva Precisión/Sensibilidad− puede dar una falsa seguridad al diseñador, ya que otorga la misma importancia a todas las zonas de la curva cuando, realmente, la región de interés es aquella que proporciona un nivel reducido de falsos positivos.

4.2. Diseño de la máquina de aprendizaje

De acuerdo con la metodología fundamentada del capítulo anterior, la única condición en lo que respecta al diseño de la máquina de aprendizaje es el uso de un coste subrogado de Bregman. Por ello y por la gran capacidad expresiva del Aprendizaje Profundo [Bengio, 2009], se eligen las redes neuronales profundas (DNNs) como máquina de aprendizaje. Además, en los experimentos posteriores se incluye un estudio de los efectos del aumento del número de capas ocultas en las prestaciones y la robustez del diseño.

El diseño de la red neuronal supone la búsqueda de una arquitectura adecuada y de los parámetros intrínsecos de la misma, conocidos como parámetros no entrenables o hiperparámetros. Aunque no es el objetivo de esta Tesis, a continuación se enumeran brevemente una serie de recomendaciones para un buen diseño del clasificador.

Obviamente, el primer paso es seleccionar la arquitectura de la red (número, tipo y tamaño de las capas ocultas) de acuerdo a las características del problema. Esta Tesis se centra en problemas de clasificación binaria cuyas variables son numéricas, por lo que la red únicamente estará formada por capas densas totalmente conectadas. Sin embargo, en caso de extenderse a problemas de imagen o secuencias de datos, se incluirían capas convolucionales (con los filtros correspondientes) o capas de Memoria a Largo Plazo (LSTM, "Long Short-Term Memory") [Hochreiter and Schmidhuber, 1997], respectivamente. Además, cuando la complejidad del problema aumenta resulta razonable incorporar otro tipo de capas de normalización o regularización, como las capas de normalización del lote ("batch") −en el siguiente párrafo se explica en qué consiste el aprendizaje por lotes− o las capas de "Drop-Out"[Srivastava et al., 2014] −se recuerda que consiste en desconectar de manera probabilística las interconexiones entre las neuronas de un MLP−, estableciendo una probabilidad de desconexión adecuada para evitar el sobreajuste. En cuanto a la función de activación, la única recomendación es el uso de una tangente hiperbólica en la capa de salida para asegurar que la salida o de la red cumpla $-1 \leqslant o \leqslant 1$ y, de esta forma,

ser consistente con la teoría descrita en el capítulo anterior.

El aprendizaje por lotes, conocido como modo "batch", permite el entrenamiento de la red por subconjuntos reducidos del conjunto de entrenamiento. De esta manera, la actualización del gradiente en el proceso de optimización de la red se realiza por lotes de muestras hasta que todo el conjunto de entrenamiento ha intervenido en dicha optimización, lo que se conoce como una época. El tamaño del lote y el número de épocas son parámetros de diseño. No obstante, el número de épocas debe ser suficiente para asegurar la convergencia de la red, pero tampoco excesivamente elevado, ya que se ha de evitar el sobreajuste (o sobre-entrenamiento). En cuanto al tamaño del lote, cuando se resuelve un problema desequilibrado, se aconseja que sea superior al IR del problema original, ya que de ese modo se asegura que cada actualización del gradiente cuente con al menos una muestra de la clase minoritaria, evitando actualizaciones totalmente sesgadas por la clase mayoritaria.

Por último, la selección de un optimizador basado en el descenso por gradiente adecuado, como Adam [Kingma and Ba, 2015] o RMSProp [Tieleman and Hinton, 2012], supone utilizar una tasa de aprendizaje determinada. Esto es la magnitud con la que se actualiza el gradiente. Por tanto, la selección de un valor adecuado es determinante a la hora de entrenar la red: un valor muy alto favorece una optimización más rápida, pero puede producir saltos recurrentes en torno a un mismo mínimo; por otra parte, un valor muy reducido puede implicar que el proceso de optimización se limite a la obtención de un mínimo local.

Como es posible apreciar, el número de parámetros no entrenables de la red es elevado, y muy alto el número de posibles combinaciones. Por ello, se emplean técnicas para la selección de los parámetros que optimizan las prestaciones de la red [Feurer and Hutter, 2013]. La técnica más extendida es la Validación Cruzada ("Cross Validation", CV). Consiste en dividir el conjunto de entrenamiento en K_0 subconjuntos −denominados "folds"− y una rejilla ("grid") con los valores que puede tomar cada parámetro. Se establece un criterio de selección, eligiendo una métrica a optimizar, por ejemplo, la probabilidad de detección en el punto de trabajo. En

cada vuelta, se establece uno de los "folds" como validación y se entrena con la parte complementaria, repitiendo el procedimiento hasta que se utilizan todas las posibles combinaciones de parámetros y todos los subconjuntos se han empleado para validación. Posteriormente, se promedia el resultado obtenido para cada conjunto de validación, seleccionando la combinación de parámetros que ofrece la mejor métrica (en promedio). La mayor limitación de una búsqueda de los parámetros por medio de una rejilla de valores es la necesidad de utilizar todas las posibles combinaciones de parámetros. Para evitarlo, aparece la optimización bayesiana de hiperparámetros, con la cual se construye un modelo probabilístico que selecciona los siguientes valores a explorar a partir de los que mejores prestaciones ofrecen según un cierto criterio o métrica.

4.3. Diseño del re-equilibrado por medio de métodos neutrales

Una vez diseñada la máquina de aprendizaje, el diseño se centra en la búsqueda del re-equilibrado óptimo. El re-equilibrado completo ($\widetilde{Q} = 1$) no es siempre la mejor opción [Khoshgoftaar et al., 2007], por lo que será necesaria una exploración de distintas técnicas e intensidades de re-equilibrado.

Cada una de las técnicas de re-equilibrado descritas en los dos capítulos anteriores tiene ventajas y desventajas, que en general dependen de las características del problema. Por ello, partiendo siempre de la premisa de la aplicación de técnicas neutrales, se propone combinar distintas técnicas en un esquema de re-equilibrado fundamentado. El objetivo de este proceso es establecer el valor del cociente de re-equilibrado ("Balanced Ratio", BR) de manera que se atenúe el IR original del problema por medio de un factor $\alpha > 1$ según

$$\frac{1}{BR} = \frac{\alpha_W \cdot \alpha_R \cdot \alpha_G}{IR} = \frac{\alpha}{IR} \tag{4.2}$$

donde α se define como la intensidad del re-equilibrado para los distintos métodos

neutrales empleados, incluyendo ponderación de muestras minoritarias por medio de α_W, re-muestreo con α_R o generación de muestras minoritarias con α_G. Cuando alguna de ellas no se aplica, su valor es unitario. El mejor valor para BR depende del problema, por lo que será necesario explorar distintos valores de α para obtener las mejores prestaciones. Para ello, puede utilizarse el proceso de Validación Cruzada expuesto en el apartado anterior, maximizando la probabilidad de detección en el punto de trabajo o utilizando algún otro de los criterios o métricas descritas anteriormente.

Este esquema brinda la posibilidad de aprovechar las ventajas que ofrece la diversidad en técnicas que utilizadas por sí solas no disponen. Un ejemplo es el uso de ponderación de muestras según la clase a la que pertenecen, una técnica que no ofrece las ventajas de utilizar conjuntos, pero que combinada con re-muestreo o generación de muestras sí opera en condiciones de diversidad.

Sin embargo, como se mencionó en los capítulos anteriores, el re-muestreo puede ser arriesgado en términos de sobreajuste al utilizar sobre-muestreo (repitiendo muestras irrelevantes o, incluso, muestras fuera de rango) o eliminando información relevante por medio del sub-muestreo. Por tanto, se propone la combinación de la generación de muestras de la clase minoritaria con la ponderación por clases. Para ello, se define la tasa de generación ("generation rate", g_r) como el cociente entre el número de muestras generadas y el número original de muestras de la clase minoritaria, en porcentaje. Por tanto, se puede controlar la intensidad de la generación por medio de

$$\alpha_G = \frac{g_r}{100} + 1 \tag{4.3}$$

Por otro lado, α_W indica el cociente de pesos entre las clases positiva y negativa, de tal manera que

$$\alpha_W = \frac{w_{(+)}}{w_{(-)}} \tag{4.4}$$

donde $w_{(+)}$ y $w_{(-)}$ son los pesos para las clases positiva y negativa, respectivamente. El hecho de no aplicar re-muestreo supone que α_R sea unitario. Por tanto, la intensidad

de re-equilibrado de (4.2) se controla según

$$\frac{1}{BR} = \frac{1}{IR}\left(\frac{g_r}{100} + 1\right)\frac{w_{(+)}}{w_{(-)}} \tag{4.5}$$

4.4. Experimentos

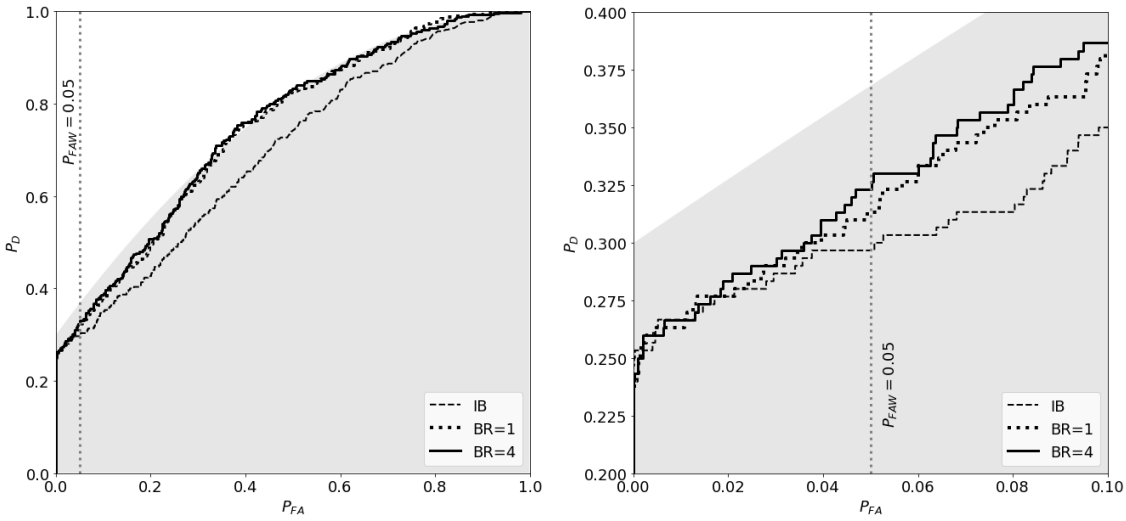
Tras definir el proceso de diseño del clasificador para un problema desequilibrado en términos de la máquina de aprendizaje y del re-equilibrado, en este apartado se presentan una serie de experimentos que ilustran el proceso que se ha propuesto.

4.4.1. Problema sintético

Se comienza con el mismo problema sintético considerado en el capítulo anterior, definiendo las verosimilitudes según (3.23a) y (3.23b) y la NPOC según (3.24c). Para este experimento se generan $N = 65000$ muestras sintéticas: $N_1 = 1000$ de la clase positiva y $N_0 = 64000$ de la clase negativa. Así, el problema tiene $IR = 64$. Se construyen los subconjuntos de entrenamiento (70 %) y test (30 %), manteniendo el IR original, por medio de una división estratificada de los datos.

Se fija el punto de trabajo en $P_{FAW} = 0.05$, un valor lo suficientemente bajo para evitar una explosión del número de falsos positivos para el problema original.

Se elige un MLP como máquina de aprendizaje, explorando arquitecturas superficiales y profundas, con el error cuadrático medio como coste subrogado. Para su diseño, se seleccionan la tasa de aprendizaje, el número de capas ocultas y neuronas por medio de una validación cruzada 5-fold, eligiendo la combinación de parámetros que maximiza la probabilidad de detección en el punto de trabajo para el problema original (desequilibrado). El diseño resultante es una red neuronal de tres capas ocultas con 6 neuronas cada una y una tasa de aprendizaje $\mu = 10^{-3}$ para el optimizador RMSProp. Además, se utiliza un conjunto formado por 5 aprendices con promediado de las salidas como método de agregación. El tamaño del conjunto se selecciona de manera separada, asegurando obtener ventaja de la diversidad.



(a) **(b)**

Figura 4.2: Curvas MOC obtenidas para el problema sintético original (línea discontinua), re-equilibrado completo (línea punteada) y el diseño óptimo propuesto (línea continua). El área sombreada representa la NPOC. (a) muestra las curvas MOC completas, mientras que (b) se centra en la zona en torno al punto de trabajo.

Para el re-equilibrado se emplea SMOTE ($K = 3$) como método de generación y ponderación de pesos. Para determinar las mejores intensidades de re-equilibrado, se utiliza nuevamente una validación cruzada 5-fold para asegurar las mejores prestaciones en el punto de trabajo. Los valores explorados son $g_r \in \{0, 100, 500, 1000, 2000, 3500, 5000, 6400, 10000\}$ (en porcentaje) para la tasa de generación y $BR \in \{1, 2, 4, 8\}$ para el cociente de re-equilibrado. Para conseguirlo, se fija el peso de las muestras de la clase mayoritaria en $w_{(-)} = 1$ y se varía el peso de las muestras minoritarias $w_{(+)}$ hasta alcanzar el valor de BR deseado. De esta forma, es posible que, tras el proceso de generación, haya un número mayor de muestras de la clase positiva (originalmente la clase minoritaria) que de la negativa, pero penalizadas con un peso menor. Tras el proceso de CV, las mejores prestaciones se obtienen para $g_r = 500\,\%$ y $BR = 4$ (conseguido con $w_{(+)} = 8/3$).

Los resultados obtenidos sobre el conjunto de test se muestran en la Figura 4.2,

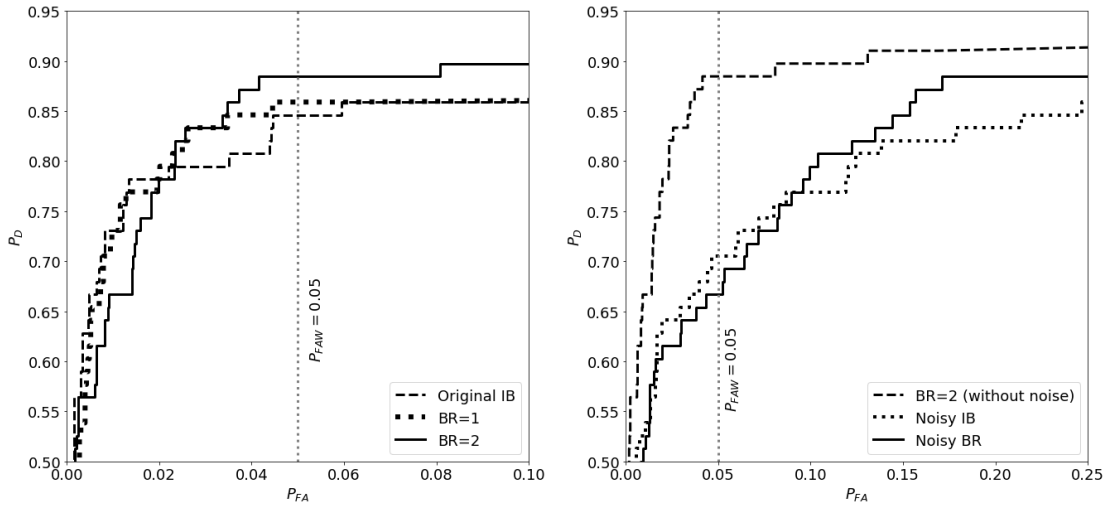
donde aparecen representadas las curvas MOC (se recuerda que es la estimación de la NPOC teórica a partir de las salidas de la máquina) para los diseños entrenados con el problema original (desequilibrado), el diseño de re-equilibrado anterior ($BR = 4$) y un diseño con re-equilibrado completo ($BR = 1$ con SMOTE, es decir, $g_r = 6400\,\%$). Las ventajas del diseño de re-equilibrado fundamentado son claras en torno al punto de trabajo: la probabilidad de detección aumenta 0.027 respecto al diseño directo (IB), y 0.010 respecto al re-equilibrado completo con SMOTE.

4.4.2. Problema de diagnóstico médico

Para el siguiente experimento se aborda un problema de diagnóstico médico, concretamente la base de datos "Mammography"[Woods et al., 1993], en el que hay que clasificar mamografías como "malignas" o "benignas". La base de datos cuenta con $N = 11183$ muestras descritas por 6 variables numéricas. El problema tiene originalmente un desequilibrio con $IR = 42$, ya que se dispone de $N_0 = 10923$ muestras benignas y $N_1 = 260$ muestras malignas, la clase que se desea detectar. Como es habitual, se dividen ambas clases en subconjuntos de entrenamiento/test (70/30 %) manteniendo el IR original por medio de una división estratificada respecto a la etiqueta.

Tras algunas pruebas preliminares, se fija el punto de trabajo en $P_{FAW} = 0.05$, un valor bajo de falsa alarma que proporciona unos niveles de detección aceptables. Una vez fijado, se diseña la máquina de aprendizaje y el re-equilibrado con el objetivo de maximizar P_D en el punto de trabajo seleccionado.

Para la máquina de aprendizaje se exploran arquitecturas superficiales y profundas, con distintos números de neuronas y la tasa de aprendizaje del optimizador RMSProp. Tras una validación cruzada 5-fold (maximizando P_D en P_{FAW}) se obtiene que el mejor diseño es un MLP de una sola capa oculta con 10 neuronas y una tasa de aprendizaje $\mu = 10^{-3}$ utilizando el RMSProp. Además, se emplea un conjunto de 21 aprendices, promediando sus salidas como método de agregación, con el objetivo de obtener las ventajas de la diversidad.



(a) **(b)**

Figura 4.3: Curvas MOC obtenidas para la base de datos "Mammography.ºriginal y una versión ruidosa. En (a) se utiliza el problema original, mostrando: el diseño directo desequilibrado (línea discontinua), el re-equilibrado completo (punteada) y el diseño óptimo propuesto (continua). En (b) se utiliza la versión ruidosa del problema, representando: el diseño óptimo del problema original (línea discontinua), el diseño directo (punteada) y el diseño re-equilibrado (continua).

Como en el caso anterior, se exploran distintas intensidades de re-equili-brado a partir de la tasa de generación $g_r \in \{0, 100, 500, 1000, 1500, 4200, 5000\}$ (en porcentaje) de muestras con SMOTE ($K = 3$) y cociente de re-equilibrado $BR \in \{1, 2, 4\}$, ajustado por ponderación. Tras un proceso de validación cruzada 5-fold, se obtienen los mejores resultados para $g_r = 500\,\%$ y $BR = 2$, lo que supone compensar la clase mayoritaria con un peso $w_{(+)} = 3.5$.

En la Figura 4.3a se presentan los resultados obtenidos para el diseño directo (desequilibrio original), re-equilibrado completo ($BR = 1$) y el diseño determinado por CV. En dichas curvas MOC, se observa una clara mejora de las prestaciones en el punto de trabajo: la probabilidad de detección aumenta 0.038 con respecto al diseño directo y 0.026 respecto al re-equilibrado completo.

Adicionalmente, con el objetivo de conocer la robustez de los métodos empleados frente a la presencia de ruido −fenómeno común que puede aparecer en sistemas de medición por sensores, como el problema bajo estudio−, se modifican los datos de entrenamiento originales por medio de ruido aditivo gaussiano $G(0,\, 1.5)$, es decir, de media nula y varianza 1.5.

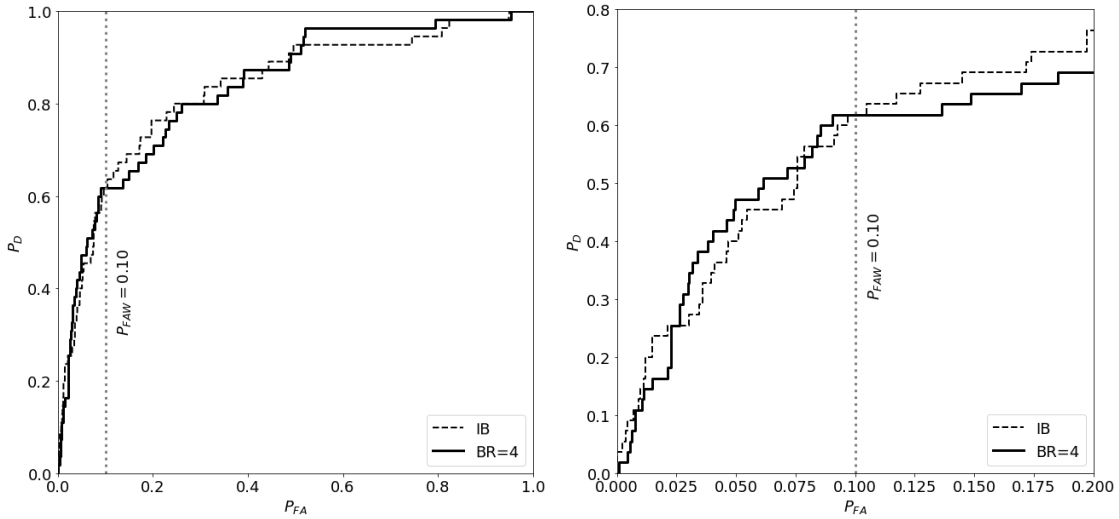
Utilizando la misma máquina de aprendizaje, se exploran nuevamente distintas intensidades de re-equilibrado y se selecciona el mejor caso por validación cruzada 5-fold, resultando $g_r = 0\,\%$ y $BR = 1$, es decir, no realizar generación y re-equilibrar por completo por medio de solo ponderación. Los resultados obtenidos se muestran en la Figura 4.3b, donde se observa una clara degradación respecto al diseño de re-equilibrado en ausencia de ruido. Este efecto es fácilmente explicable: tal y como se dijo en el Capítulo 3, el objetivo de aplicar el método fundamentado es el de obtener una buena estimación del cociente de verosimilitudes, algo que la presencia de ruido claramente dificulta.

4.4.3. Problemas desequilibrados con bajo número de muestras

En el experimento anterior, se ha comprobado que la presencia de ruido puede degradar la estimación del cociente de verosimilitudes. En este apartado, se llevan a cabo varios experimentos con un número de muestras reducido con la intención de analizar el efecto que tiene el número de muestras disponibles en la estimación de la NPOC.

4.4.3.1. Base de datos “Wine-quality-4”

La base de datos “Wine-quality-4” aparece en numerosos estudios sobre problemas de clasificación desequilibrada. Dicha base de datos es una versión de la original [Cortez et al., 2009], en la cual se desea detectar si la calidad de un vino es inferior a 4 (en una escala del 0 al 10) en base a 11 variables numéricas resultantes de pruebas físicoquímicas. Por tanto, el problema de regresión original se transforma en un problema binario de clasificación desequilibrada con $IR = 26$. La gran limitación de



(a) **(b)**

Figura 4.4: Curvas MOC para el diseño directo/desequilibrado (IB, línea discontinua) y el diseño re-equilibrado ($BR = 4$, línea continua). (a) representa las curvas MOC completa, mientras que (b) se centra en la zona en torno al punto de trabajo.

esta base de datos es el bajo número de muestras de la clase minoritaria, ya que el problema cuenta con $N_1 = 183$ muestras y $N_0 = 4715$ de las clases positiva (calidad inferior a 4) y negativa, respectivamente.

Ambas clases se dividen en los subconjuntos de entrenamiento/test con una proporción 70/30 %, manteniendo el IR original.

Como en los casos anteriores, se obtiene el mejor algoritmo por medio de dos procesos de validación cruzada 5-fold para el diseño de la máquina de aprendizaje y la intensidad de re-equilibrado, maximizando la probabilidad de detección en el punto de trabajo, fijado en $P_{FAW} = 0.10$. Los mejores resultados se consiguen con:

- un conjunto de 21 MLPs con 14 neuronas en su capa oculta y una tasa de aprendizaje de $\mu = 10^{-3}$ para el optimizador RMSProp;
- y una tasa de generación $g_r = 200\,\%$ y $BR = 4$, ponderando la clase minoritaria con $w_{(+)} = 2.167$.

Base de datos	IR	N	D	P_{FAW}	H *	μ *
Abalone19	129.53	4177	10	0.15	22	10^{-2}
Ozone-level	33.74	2536	72	0.15	10	10^{-3}
Yeast4	28.09	1484	8	0.15	16	10^{-2}
Yeast6	41.4	1484	8	0.15	6	10^{-2}

Tabla 4.1: Características de las bases de datos empleadas: Cociente de desequilibrio (IR), número de muestras (N), número de dimensiones (D) y punto de trabajo seleccionado (P_{FAW}). Los parámetros intrínsecos de la máquina (número de neuronas de la capa oculta (H) y la tasa de aprendizaje (μ) del optimizador RMSProp) también se muestran. * indica los parámetros obtenidos por validación cruzada 5-fold.

La Figura 4.4 muestra las curvas MOC obtenidas para el diseño directo (desequilibrado) y el diseño con re-equilibrado, donde se observa que el re-equilibrado no ofrece ninguna mejora en la detección en el punto de trabajo fijado. Como ocurría en el caso anterior con la presencia de ruido, un limitado número de muestras de la clase minoritaria no permite realizar una buena estimación del cociente de verosimilitudes, algo que se pretende corroborar en los siguientes experimentos.

4.4.3.2. Otras bases de datos de referencia

El experimento anterior muestra las limitaciones del re-equilibrado fundamentado cuando el número de muestras disponibles de la clase minoritaria es muy limitado. Para consolidar dicha conclusión, se utilizan cuatro bases de datos de los repositorios UCI [Dua and Graff, 2019] −de la Universidad de California− y KEEL ("Knowledge Extraction based on Evolutionary Learning") [Alcalá-Fernández et al., 2011] −de la Universidad de Granada−, de uso público y ampliamente conocidas. La Tabla 4.1 muestra sus características, además del punto de trabajo seleccionado P_{FAW} y los parámetros intrínsecos del MLP empleado, seleccionados nuevamente por medio de una validación cruzada 5-fold. Además, cada base de datos se divide en los conjuntos

Base de datos	$P_{D_{IB}}$	$g_r(\,\%)$ *	BR *	$w_{(+)}$ *	$P_{D_{RB}}$
Abalone19	0.295 ± 0.060	5000	1	2.54	0.614 ± 0.036
Ozone-level	0.732 ± 0.052	0	2	16.79	0.722 ± 0.063
Yeast4	0.755 ± 0.064	500	4	1.18	0.825 ± 0.087
Yeast6	0.888 ± 0.000	1000	1	3.80	0.888 ± 0.000

Tabla 4.2: Probabilidades de detección para el diseño directo (IB) y para los diseños de re-equilibrado fundamentado (RB). También se muestran los parámetros de diseño (indicados por *): tasa de generación (g_r), cociente de re-equilibrado (BR) y peso con el que se pondera la clase minoritaria ($w_{(+)}$).

de entrenamiento/test con una proporción 75/25 %, manteniendo el IR original de cada una de ellas.

Como en los casos anteriores, se lleva a cabo el proceso de diseño de re-equilibrado fundamentado por medio de una validación cruzada 5-fold de las intensidades de re-equilibrado. La Tabla 4.2 muestra los mejores valores obtenidos, además de las probabilidades de detección para el diseño directo (desequilibrado) y tras aplicar el diseño con re-equilibrado fundamentado. En este caso no se muestran las curvas MOC, ya que el reducido número de muestras de la clase minoritaria en los conjuntos de test hace que dichas curvas sean muy escalonadas y, por tanto, poco descriptivas.

Se puede observar que en las bases de datos “Abalone19” y “Yeast4” la capacidad de detección parece mejorar considerablemente con el re-equilibrado, pero esto se debe al reducido número de muestras minoritarias en el conjunto de test. De hecho, la mejora que muestra la Tabla 3 respecto a estas bases de datos supone una detección de dos y una muestras minoritarias más para “Abalone19” y “Yeast4”, respectivamente.

Por otro lado, el proceso de re-equilibrado no ofrece ninguna mejora para las bases de datos “Ozone-level” y “Yeast6”. Destaca el caso de “Ozone-level”, donde incluso llegan a degradarse las prestaciones. El motivo es la alta dimensionalidad del espacio

de observación (72 dimensiones), lo que reduce considerablemente las capacidades de SMOTE como esquema de generación. Este tema se tratará con más detalle en el Capítulo 5, donde se propone una alternativa para solucionarlo. En cuanto a "Yeast6", el re-equilibrado no altera el resultado del diseño directo. Hay que destacar que se trata de un problema relativamente sencillo, por lo que el re-equilibrado no tiene efecto al disponer de pocas muestras minoritarias.

Al final del capítulo se lleva a cabo un sencillo análisis estadístico que complementa los resultados presentados.

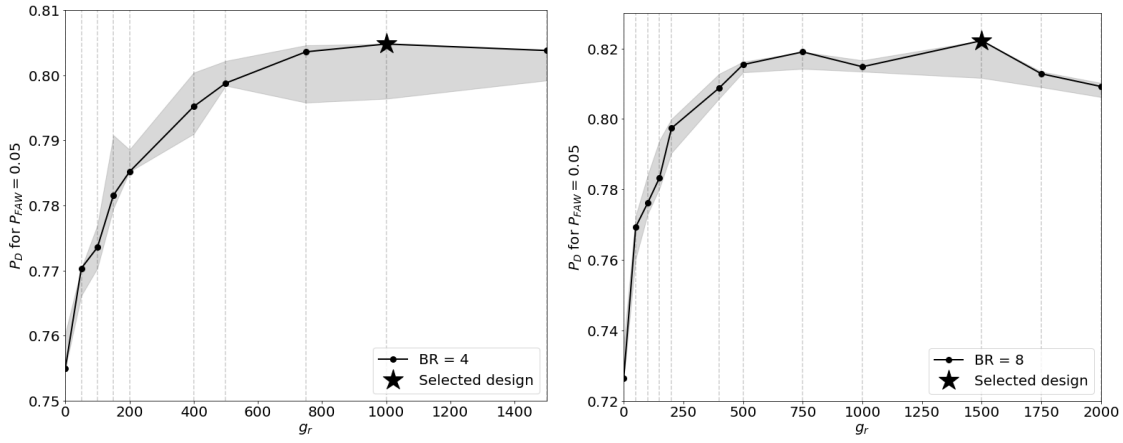
4.4.4. Un problema desequilibrado con alto número de muestras

Por último, se considera la base de datos "BNG: PageBlocks" [Holmes et al., 2014] como problema real con un número suficiente de muestras. Se trata de la misma base de datos utilizada en el Capítulo 3, pero en este caso se trabaja con todas las muestras disponibles ($N_0 = 265174$ y $N_1 = 6238$ muestras para las clases negativa (Bloque 1) y positiva (Bloque 5), respectivamente). Como es habitual, ambas clases se dividen en los conjuntos de entrenamiento/test con una proporción 70/30 %, manteniendo el $IR = 42.5$ original. Además, se fija el punto de trabajo en $P_{FAW} = 0.05$.

Como en los experimentos anteriores, se exploran arquitecturas superficiales y profundas del MLP. No obstante, en este caso se muestran ambos diseños para compararlos y conocer los efectos del aumento del número de capas ocultas. Para ello, se emplean los siguientes diseños:

- un MLP con una capa oculta de 34 unidades como arquitectura superficial;
- y una DNN compuesta por tres capas ocultas de 20 unidades cada una como arquitectura profunda.

En ambos casos, se emplean conjuntos de 21 aprendices, entrenando 300 épocas con un tamaño de "batch" de 128 muestras y el RMSProp como optimizador con una tasa de aprendizaje de $\mu = 10^{-2}$.



(a) **(b)**

Figura 4.5: Proceso de validación cruzada 5-fold de la intensidad de re-equilibrado sobre "PageBlocks" para ambas arquitecturas: (a) superficial y (b) profunda. El área sombreada es el rango de resultados que se han obtenido para todos los valores explorados de BR. La línea continua corresponde al mejor valor de BR y el diseño seleccionado se indica por medio de ★.

De nuevo, se seleccionan las intensidades de re-equilibrado para ambas arquitecturas (superficial y profunda) por medio de una validación cruzada 5-fold. Para ello se exploran los siguientes valores: la tasa de generación con SMOTE ($K = 3$) $g_r \in \{0, 50, 100, 150, 200, 400, 500, 750, 1000, 1500, 1750, 2000\}$ (en porcentaje) y $BR \in \{1, 2, 4, 8, 16\}$. En la Figura 4.5 se muestran los resultados del proceso de validación cruzada. Para la máquina superficial, la mejor opción se obtiene con los parámetros $g_r = 1000\,\%$ y $BR = 4$, lo que supone ponderar la clase minoritaria con $w_{(+)} = 0.966$. En cuanto a la arquitectura profunda, $g_r = 1500\,\%$ y $BR = 8$ proporcionan los mejores resultados, ponderando la clase minoritaria con $w_{(+)} = 0.332$. En ambos casos, el re-equilibrado completo no solo no es la mejor opción, sino que se compensa una excesiva generación de muestras con un bajo peso de las muestras minoritarias. Además, se observa cómo los resultados de la validación cruzada son más estables para la red profunda.

Tras obtener los mejores parámetros de re-equilibrado para cada arquitectura, se

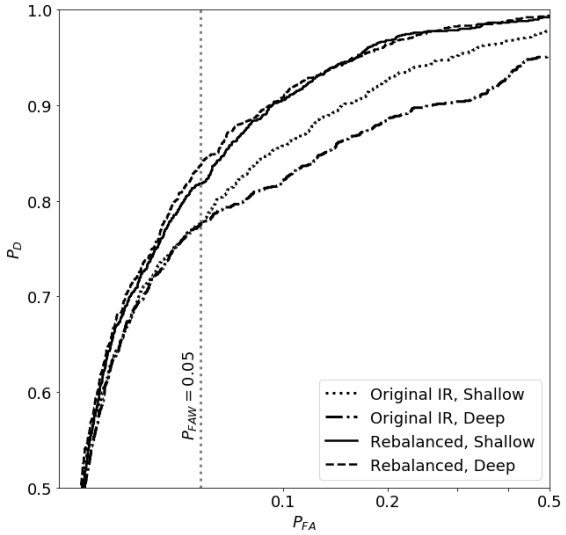


Figura 4.6: Curvas MOC obtenidas para el diseño directo ("Original") y el diseño re-equilibrado ("Rebalanced") con los MLP superficial ("Shallow") y profundo ("Deep"). El punto de trabajo está fijado en $P_{FAW} = 0.05$.

emplean ambas para resolver el problema. La Figura 4.6 muestra las prestaciones para el diseño directo (desequilibrado) y re-equilibrado de cada arquitectura. Puede apreciarse claramente que las mejores prestaciones en el punto de trabajo se obtienen para el diseño con DNN y re-equilibrado. No obstante, resulta sorprendente el efecto que tiene aumentar el número de capas: reduce la probabilidad de detección para niveles de falsa alarma superiores al punto de trabajo en el diseño directo y mejora al utilizar el proceso de re-equilibrado fundamentado. Esto, sin duda, avala la metodología fundamentada presentada en esta Tesis.

Debido al tamaño de la base de datos, no se ha llevado a cabo una exploración muy extensa acerca del número de capas y número de neuronas de las mismas para obtener un diseño óptimo, ya que conllevaría un alto coste computacional. Sin embargo, se estudia el efecto del número de neuronas en el diseño de tres capas ocultas, utilizando $H \in \{10, 20, 30\}$. Además, con el objetivo de analizar también las ventajas que proporciona la diversidad, se aplica el diseño de re-equilibrado óptimo para cada arquitectura con un único aprendiz y con el conjunto de 21 máquinas anterior. El

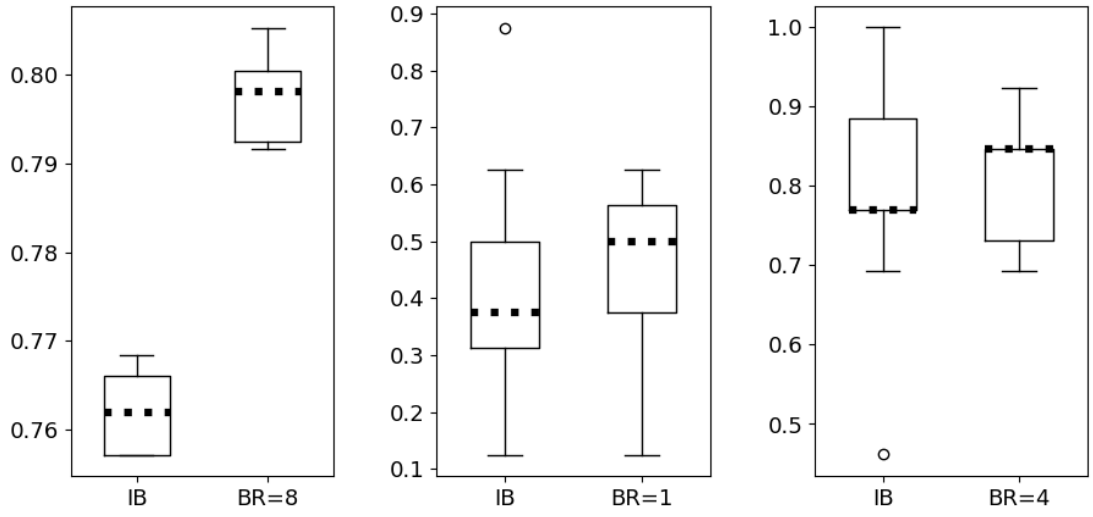
Diseño	Número de aprendices	
	1 aprendiz	21 aprendices
10-34-1	0.756 ± 0.007	0.818 ± 0.003
10-10-10-10-1	0.724 ± 0.003	0.801 ± 0.006
10-20-20-20-1	0.766 ± 0.002	**0.834 ±0.002**
10-30-30-30-1	0.758 ± 0.001	0.828 ± 0.001

Tabla 4.3: Probabilidad de detección (media ± desviación típica para 10 ejecuciones) en el punto de trabajo $P_{FAW} = 0.05$ para la arquitectura superficial y distintos tamaños de las capas de una DNN de tres capas ocultas. Los resultados se muestran para una única máquina y para conjuntos de 21 aprendices.

experimento se repite con 10 ejecuciones, promediando la probabilidad de detección obtenida en el punto de trabajo. Los resultados se muestran en la Tabla 4.3, donde se observa que el conjunto con aprendices de tres capas ocultas de 20 neuronas cada una es el que ofrece mejores prestaciones. Además, analizando esos resultados, se pueden extraer varias conclusiones. En primer lugar, utilizar un número suficiente de neuronas (en este caso, a partir de 20) en la arquitectura profunda mejora las prestaciones del diseño superficial, tanto en media como en desviación típica. Además, una búsqueda más exhaustiva del número de neuronas por capa (con la posibilidad de distinto número de neuronas en cada capa) podría mejorar considerablemente las prestaciones, pero, como se ha dicho anteriormente, esto sería un proceso muy costoso. Por último, se aprecia el efecto positivo que ofrecen los conjuntos, mejorando de manera muy significativa los resultados para todos los diseños estudiados, demostrando así las ventajas de aplicar diversidad.

4.4.5. Análisis estadístico de los resultados

Para finalizar el capítulo, se realiza un análisis estadístico para evaluar la estabilidad de los resultados de los diferentes diseños. Para ello, se emplean tres bases



(a) **(b)** **(c)**

Figura 4.7: Análisis con diagrama de caja de la probabilidad de detección en el punto de trabajo P_{FAW} para las bases de datos: (a) "PageBlocks", (b) "Abalone19", (c) "Yeast4".

de datos representativas: "PageBlocks" −base de datos real con un alto número de muestras−, "Abalone19" −base de datos relativamente pequeña con un alto IR− y "Yeast4" −base de datos pequeña con un IR más bajo−. Para este análisis, se realizan 11 particiones distintas de entrenamiento/test para cada una de las bases de datos, manteniendo el IR original de cada una de ellas en ambos subconjuntos de muestras.

El análisis consiste en obtener la probabilidad de detección para cada uno de los diseños utilizando las distintas particiones de entrenamiento/test y obteniendo sus diagramas de cajas. Los diagramas de cajas son una forma de analizar las diferencias estadísticamente. Están compuesto por una caja y los denominados bigotes. La caja representa los valores más bajo y más alto de los cuartiles de los resultados obtenidos, indicando con una línea punteada intermedia el valor de la mediana. Por su parte, los bigotes muestran el rango de dichos resultados. Por último, los puntos marcados

fuera de los bigotes representan resultados atípicos o fuera de rango.

Esos diagramas se muestran en la Figura 4.7, donde se representa el diseño directo (IB) y el re-equilibrado fundamentado (BR) con los parámetros e intensidades obtenidas en los apartados previos para cada base de datos.

La mayor ventaja se aprecia en "PageBlocks", cuyo número suficiente de muestras minoritarias permite realizar una buena estimación del cociente de verosimilitudes y, por tanto, conseguir mejores prestaciones. Además, este aumento de las prestaciones es estadísticamente estable.

Lo contrario ocurre cuando se emplean bases de datos pequeñas con un número muy reducido de muestras minoritarias. Se pueden observar ligeras mejoras, como es el caso de "Abalone19", pero con una estabilidad muy baja. Peor es el caso de "Yeast4", donde incluso se obtienen peores resultados.

Capítulo 5

Una incursión en la generación de muestras

A lo largo de esta Tesis se ha llevado a cabo un estudio detallado sobre los efectos que tiene el desequilibrio entre las clases a la hora de afrontar un problema de clasificación binaria. Además, tras citar las técnicas más utilizadas para atenuar los efectos del desequilibrio, se ha presentado una metodología fundamentada en los principios bayesianos que garantiza robustez, incluso ante cambios en las condiciones del problema. Por ello, se ha resaltado la importancia de emplear técnicas de re-equilibrado neutrales con el objetivo de mantener inalterado el cociente de verosimilitudes del problema original. Manteniendo siempre en mente dicha condición, en este capítulo se realiza una exploración en el ámbito de la generación de muestras. Para ello, se analiza aún más en detalle el algoritmo de SMOTE, que es, sin ninguna duda, la técnica de generación más extendida, presentando sus puntos fuertes y sus principales desventajas. Tras ese análisis, se propone un nuevo algoritmo: una variante de SMOTE centrada en reducir algunas de las limitaciones mostradas por el algoritmo original.

5.1. Virtudes y defectos de SMOTE

Han pasado cerca de dos decenios desde que Nitesh V. Chawla presentara la técnica de SMOTE [Chawla et al., 2002], un algoritmo de generación de muestras para mejorar la clasificación con poblaciones desequilibradas. Seguramente, cuando fue presentado no se esperaba el gran impacto que tendría, pero, desde entonces, se ha convertido claramente en una referencia de esta familia de técnicas y su uso se ha mantenido a lo largo de los años a pesar del gran avance tecnológico vivido durante esos veinte años.

Probablemente, el principal motivo de su éxito reside en su sencillez. Se recuerda brevemente su funcionamiento: de manera iterativa se selecciona una muestra aleatoria de la clase minoritaria y, a partir de uno de sus K vecinos más próximos (elegido de manera aleatoria), se genera una muestra sintética en el segmento que separa ambas muestras. El proceso no necesita ningún entrenamiento previo −algo que sí ocurre con otras técnicas de re-equilibrado más recientes basadas en GANs−, ya que únicamente requiere el cálculo de los vecinos más próximos. Por ello, tanto SMOTE como sus variantes tienen complejidad computacional proporcional al cuadrado del número de datos de entrada [Bunkhumpornpat et al., 2012], por lo que sus tiempos de ejecución son asumibles. Además, los parámetros de diseño son únicamente dos: el número de vecinos más próximos K y la tasa de generación. Esto contrasta con técnicas más complejas, las cuales incorporan un mayor número de parámetros, dificultando así la búsqueda de un buen diseño.

Una ventaja muy determinante de SMOTE es su neutralidad estadística. SMOTE asigna a todas las muestras de la clase minoritaria la misma probabilidad de contribuir a la generación de las muestras sintéticas, ya que la selección de muestras aleatoria (tanto de la muestra original como de su vecino más próximo) que se realiza se lleva a cabo de acuerdo a una distribución uniforme. Como se ha visto a lo largo de los capítulos anteriores, la neutralidad, junto con el uso de costes de Bregman, son las dos condiciones suficientes y necesarias para asegurar la invarianza en el cocien-

te de verosimilitudes $q_L(\mathbf{x})$. Otras técnicas informadas, entre las que se encuentran variantes de SMOTE, asignan mayor importancia a una selección de muestras −por ejemplo, Borderline-SMOTE utiliza las muestras más cercanas a la frontera entre las clases−, deformando así el cociente de verosimilitudes del problema original.

Los experimentos realizados en esta Tesis han demostrado sobradamente la eficacia de SMOTE para el re-equilibrado. Las muestras sintéticas generadas por medio de dicha técnica para el entrenamiento del clasificador posibilitan la consideración de la clase minoritaria por parte del algoritmo de clasificación, lo que permite una mejor discriminación entre ambas clases −originalmente sesgada en favor de la mayoritaria−, aumentando la probabilidad de detección para valores reducidos de falsa alarma.

No obstante, SMOTE tiene dos principales limitaciones: por un lado, su aplicación no es directa en presencia de variables discretas; y, además, sus efectos disminuyen a medida que aumenta el número de dimensiones.

5.1.1. Generación de datos discretos y categóricos

La presencia de variables categóricas dificulta el cálculo de los vecinos más próximos. El cálculo de la distancia únicamente tendría sentido si los valores que la variable categórica puede tomar estuviesen jerarquizados, es decir, tuviesen un orden. Podría pensarse en codificar estas variables previamente, pero esto supondría la obtención de variables discretas. El problema de las variables discretas se encuentra a la hora de generar las muestras sintéticas en un punto intermedio del segmento que separa la muestra original y su vecino más próximo seleccionado. Dicho de otro modo, SMOTE genera valores intermedios, por lo que no puede aplicarse directamente sobre datos discretos. Este problema fue descrito por los propios autores de SMOTE en el artículo en el que se presentó [Chawla et al., 2002]. Para solventarlo, proponían dos variaciones de SMOTE: SMOTE-NC ("Synthetic Minority Oversampling TEchnique-Nominal Continuous") para problemas con variables tanto continuas como discretas y SMOTE-N para problemas en los que todas sus variables son discretas.

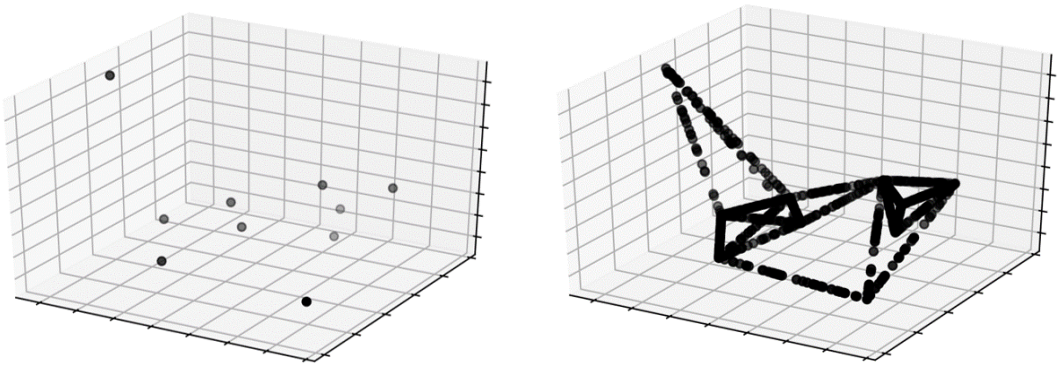
Para el cálculo de los vecinos más próximos de la muestra $\mathbf{x}^{(n)}$, SMOTE-NC utiliza la distancia euclídea según

$$d_E\left(\mathbf{x}^{(n)}, \mathbf{x}^{(n')}\right) = \sqrt{\sum_{i=1}^{D_C}\left(x_i^{(n)} - x_i^{(n')}\right)^2 + \sum_{D_D}(Med)^2} \tag{5.1}$$

donde D_C es el número de variables continuas y Med es la mediana de las desviaciones típicas de todas las variables continuas en las muestras de la clase minoritaria, añadida en el cálculo de la distancia euclídea D_D veces, siendo D_D el número de variables discretas cuyo valor es distinto entre $\mathbf{x}^{(n)}$ y $\mathbf{x}^{(n')}$. Una vez calculados los vecinos más próximos, las variables continuas se generan siguiendo el mismo proceso que la versión original de SMOTE, mientras que las variables categóricas o discretas se generan por un sistema de voto por mayoría a partir de los vecinos más próximos. No obstante, como indican los propios autores, una varianza muy alta en las variables continuas puede suponer un mayor solapamiento con la clase mayoritaria, aumentando así los niveles de falsos positivos.

Por su parte, SMOTE-N se propuso para generar muestras sintéticas cuyo espacio de observación está compuesto únicamente por variables discretas o categóricas. Para ello, proponen el uso de una versión modificada de la Métrica basada en la Diferencia de Valores (VDM, "Value Difference Metric") propuesta por [Cost and Salzberg, 1993] para la búsqueda de los vecinos más próximos y un proceso de voto por mayoría para la generación de las nuevas muestras.

Técnicas más recientes de generación basadas en GANs proponen algoritmos más complejos para abordar este problema. Por ejemplo, la CTGAN ("Conditional Tabular Generative Adversarial Network") [Xu et al., 2019] utiliza un Generador condicional para la producción de las variables categóricas. El objetivo de este sistema es que todas las posibles categorías que puede tomar una variable categórica formen parte del entrenamiento, eligiendo una categoría para una de estas variables de manera uniforme. Posteriormente, el Discriminador compara la muestra generada y una muestra real con la misma categoría en la variable seleccionada.



(a) **(b)**

Figura 5.1: Ejemplo que ilustra la generación de muestras filiforme de SMOTE. (a) Población original de la clase minoritaria. (b) Conjunto de entrenamiento de la clase minoritaria tras la aplicación de SMOTE con $K = 3$.

5.1.2. SMOTE y el espacio observable de alta dimensionalidad

A medida que aumenta el número de dimensiones del espacio de observación, mayores dificultades tiene SMOTE en la generación de datos sintéticos, reduciendo el beneficio obtenido en las prestaciones del clasificador. Son conocidas las limitaciones de la distancia euclídea en espacios de observación de alta dimensionalidad [Aggarwal et al., 2001], algo que puede limitar el potencial de SMOTE. En ocasiones, basta con aumentar el valor de K, es decir, aumentar el número de vecinos más próximos, de forma que de cada muestra minoritaria emergen un mayor número de segmentos. Algunos estudios, como [Blagus and Lusa, 2012], apuntan a una selección de variables previa a la aplicación de SMOTE para lidiar con la limitación del número de dimensiones.

No obstante, el espacio de generación de las muestras sintéticas de SMOTE está limitado al segmento que separa la muestra original y cada vecino más próximo, por lo que sólo se generan muestras de manera filiforme, es decir, en forma de hilos. Como resultado de ello, se obtienen conjuntos de entrenamiento en forma de tela de araña. La Figura 5.1 muestra este fenómeno para un conjunto de muestras de 3 dimensiones.

De esta forma, de todo el volumen delimitado por las muestras originales, únicamente se generan datos en los segmentos que las separan. Obviamente, cuando el conjunto de muestras minoritarias es denso −muchas muestras ocupando un espacio reducido− este tipo de generación puede ser representativo de la verosimilitud del problema. Sin embargo, cuando las muestras son dispersas y el número de dimensiones es elevado, este esquema de generación puede no ser representativo.

5.2. Una propuesta: VoluSMOTE

Como se ha mencionado en el apartado anterior, debido a la generación filiforme de muestras sintéticas de SMOTE, no se cubre todo el volumen delimitado por las muestras de la clase minoritaria. Por ello y bajo la premisa de distribuir y aumentar el espacio de generación de las muestras sintéticas, se propone una alternativa basada en SMOTE pero con una generación volumétrica: VoluSMOTE ("VOLUmetric Synthetic Minority Oversampling TEchnique"). Para ello, de forma iterativa se elegirá aleatoriamente −de manera uniforme para mantener la neutralidad− una muestra de la clase minoritaria $x^{(n)}$ y se generará la muestra artificial $x^{(n,k,i)}$ a partir de sus K vecinos más próximos según

$$x^{(n,k,i)} = \alpha\, w^{(0,i)}\, x^{(n,0)} + \sum_{k=1}^{K} (1-\alpha)\, w^{(k,i)}\, x^{(n,k)} \tag{5.2}$$

donde $x^{(n,0)}$ es la propia muestra (original); $x^{(n,k)}$ cada uno de sus K vecinos más próximos; $w^{(k,i)}$ son pesos aleatorios obtenidos según $U(0,\ 1)$; y α es un factor de control de la distancia respecto a la muestra original de la muestra generada.

La combinación convexa del parámetro α se incorpora para tener un mayor control sobre el espacio de generación de las muestras artificiales. Para $\alpha = 1$, se anula la contribución de los K vecinos más próximos, lo que supone una repetición de las muestras minoritarias seleccionadas, con el riesgo de sobreajuste que eso conlleva. Por otro lado, $\alpha = 0$ supone que la generación de las muestras sintéticas se lleve a cabo únicamente con los K vecinos más próximos, obviando la muestra original. Sin

duda, esta configuración podría utilizarse en entornos ruidosos, ya que cuando una muestra fuera de margen se seleccionase aleatoriamente no contribuiría directamente a la generación de las nuevas muestras. Evidentemente, este será un parámetro a explorar, siendo posible tomar valores $\alpha \in [0, 1)$ o, incluso, que el valor de α no sea constante, de manera que para cada iteración i se tome un valor aleatorio de acuerdo a una distribución uniforme $\alpha_i \sim U(0, 1)$.

Además, se normalizan los pesos que ponderan la aportación de cada muestra para, de este modo, acotar el espacio de generación. Esta normalización asegura:

$$\alpha \, w^{(0,i)} + \sum_{k=1}^{K} (1 - \alpha) \, w^{(k,i)} = 1 \tag{5.3}$$

Por tanto, el proceso del algoritmo propuesto para cada iteración i es el siguiente:

1) Se selecciona una muestra aleatoriamente −según una distribución uniforme para asegurar la neutralidad estadística− y se obtienen sus K vecinos más próximos.

2) Se generan los pesos según $w^{(k,i)} \sim U(0, 1)$, $k \in [0, K]$ para la muestra original y sus vecino más próximos.

3) El peso de la muestra original se multiplica por el parámetro α fijado, mientras que cada uno de los pesos de los vecinos más próximos se multiplica por $(1-\alpha)$ para controlar la cercanía de la muestra generada respecto a la original. Se recuerda que dicho parámetro α puede fijarse previamente o seleccionarse de manera uniforme α_i para cada iteración.

4) Se normalizan los pesos según (5.3).

5) Se genera la nueva muestra a partir de (5.2).

La Figura 5.2 ilustra las diferencias en el modo de generación de VoluSMOTE respecto a SMOTE. Se observa cómo desaparece la forma de tela de araña producida por SMOTE, generando un conjunto de muestras más distribuido por el espacio

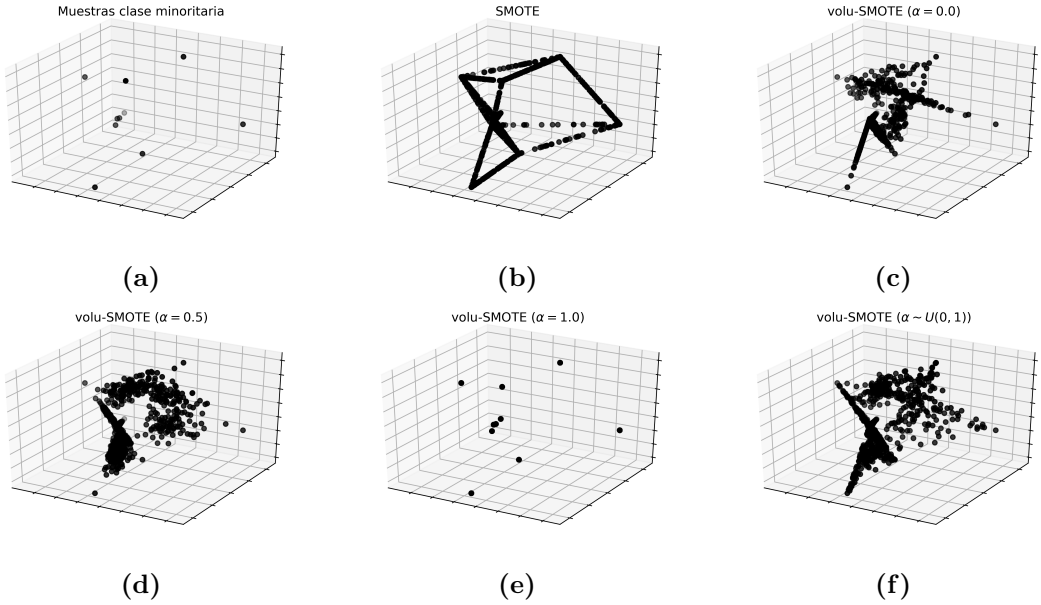


Figura 5.2: Diagramas de dispersión de la clase minoritaria en un problema de 3 dimensiones. (a) Población original de la clase minoritaria. (b) Conjunto de entrenamiento de la clase minoritaria tras la aplicación de SMOTE con $K = 3$. Conjuntos de entrenamiento de la clase minoritaria generados con VoluSMOTE para distintos valores de α: (c) $\alpha = 0$, (d) $\alpha = 0.5$, (e) $\alpha = 1$, (f) $\alpha_i \sim U(0, 1)$.

delimitado por las muestras originales. Además, se observa el efecto que tiene el factor α en el conjunto de datos resultante.

Por último, hay que destacar que VoluSMOTE es un método neutral, ya que todas las muestras tienen la misma probabilidad de ser seleccionadas inicialmente y forman parte del proceso de generación de los nuevos datos artificiales. Por tanto, es un método que cumple la condición de neutralidad estadística enunciada en la metodología fundamentada que se presenta en esta Tesis y, por ello, todo lo expresado en los capítulos anteriores en lo referente al proceso de re-equilibrado mediante SMOTE resulta aplicable al método propuesto.

5.3. Algunos experimentos preliminares y su discusión

Una vez descrito el algoritmo VoluSMOTE, se lleva a cabo un primer análisis exploratorio para comprobar la validez y el potencial de la técnica propuesta. Para las primeras pruebas se han utilizado las bases de datos descritas a continuación.

En primer lugar, "Mammography" [Woods et al., 1993], ya empleada en el capítulo anterior, es una base de datos compuesta por $N = 11183$ muestras descritas por 6 variables numéricas con $IR = 42$. Se ha dividido de manera estratificada (para mantener el IR original) en los subconjuntos de entrenamiento/test con una proporción 70/30 %.

"Ozone-level" [Dua and Graff, 2019], también considerada en el capítulo anterior. Se trata de una base de datos con un reducido número de muestras ($N = 2536$) en un espacio de observación de 72 dimensiones y con un desequilibrio de $IR = 33.73$. Los subconjuntos de entrenamiento y test están formados por el 75 y 25 % de los datos, respectivamente. En el capítulo anterior se observó que el bajo número de muestras de la clase minoritaria y su alta dimensionalidad provocaban que la metodología fundamentada no fuese capaz de realizar una buena estimación del cociente de verosimilitudes. Se incluye en este estudio para analizar si la técnica propuesta es capaz de solventar dicha dificultad.

Por último, se experimenta con "Protein Homology" [Caruana and Joachims, 2004], una base de datos propuesta como reto en la KDD Cup 2004, cuyo objetivo es clasificar proteínas según si son homólogas o no a una secuencia nativa. Está formada por $N = 145751$ muestras en un espacio observable de 74 dimensiones. Además, es una base de datos muy desequilibrada ($IR = 111.46$). Nuevamente, se crean los conjuntos de entrenamiento y test de manera estratificada con una proporción de 75/25 %. De manera adicional, se crea una versión reducida de la base de datos, seleccionando aleatoriamente el 10 % de las muestras del conjunto de entrenamiento (el 7.5 % del total y conservando el IR original), manteniendo el mismo conjunto de test.

Base de datos	P_{FAW}	Neuronas	μ	Épocas	"Batch"
Mammography	0.05	$6-10-1$	10^{-3}	200	64
Ozone-level	0.1	$72-10-1$	10^{-3}	300	32
Protein Homology	0.01	$74-64-1$	10^{-3}	150	512

Tabla 5.1: Configuración de los parámetros no entrenables de la máquina de aprendizaje empleada para cada base de datos: punto de trabajo seleccionado (P_{FAW}), número de neuronas de cada capa, tasa de aprendizaje del optimizador RMSProp (μ), número de épocas y tamaño del lote ("batch").

En cuanto al diseño del clasificador, en todos los casos se emplea un MLP superficial como máquina de aprendizaje, cuyos parámetros no entrenables se detallan en la Tabla 5.1. Para cada uno de los problemas, se fija un punto de trabajo P_{FAW}. El número de neuronas y la tasa de aprendizaje (μ) del optimizador RMSProp se obtienen por medio de una validación cruzada 5-fold bajo el criterio de maximizar la probabilidad de detección en el punto de trabajo. Se utiliza el error cuadrático (coste de Bregman) como coste subrogado y el resto de parámetros (número de épocas y tamaño del "batch") se obtienen a partir de pruebas iniciales hasta asegurar la convergencia del proceso de entrenamiento. Además, para garantizar neutralidad estadística y obtener las ventajas que ofrece la diversidad, se utilizan conjuntos de 11 aprendices.

Para conocer el potencial de VoluSMOTE, se evalúan sus prestaciones mediante la probabilidad de detección en el punto de trabajo. Además, las métricas obtenidas se comparan tanto con el diseño directo (entrenado con los datos desequilibrados originales) como con otros métodos de generación de muestras. En concreto, se utilizan SMOTE, cuyo funcionamiento ya ha sido reiteradamente descrito a lo largo de la Tesis, y Parzen [Parzen, 1962]. Se recuerda que el método de Parzen se basa en la estimación de la verosimilitud de la clase minoritaria por la acumulación de ventanas (o núcleos), comúnmente gaussianas, para posteriormente generar nuevas muestras

sintéticas a partir de la distribución estimada. Para el uso de Parzen, el aspecto de diseño más crítico es la elección de un ancho (o varianza) de la ventana adecuado. No obstante, por simplificar el estudio, se emplea la Regla de Scott [Scott, 1992] −los núcleos gaussianos incluyen una matriz de covarianza igual a la muestral multiplicada por el factor $N^{(-1/(D+4))}$, siendo N el número de muestras y D el número de dimensiones− para seleccionar este parámetro de manera automática, ya que una búsqueda de los parámetros óptimos podría extender en exceso el estudio.

Como se ha dicho, un mayor número K de vecinos más próximos puede mitigar los problemas de SMOTE a la hora de afrontar problemas desequilibrados de alta dimensionalidad. Por ello, se han explorado distintos valores de K tanto para SMOTE como para VoluSMOTE, concretamente $K \in [2, 6]$. Adicionalmente, se han explorado también los valores $K \in \{10, 15, 20\}$ para los problemas con mayor número de dimensiones. Asimismo, se exploran distintas intensidades de generación de muestras con los valores $BR \in \{1, 2, 4, 8, 16\}$ para el cociente de re-equilibrado. Particularmente para VoluSMOTE, se han tomado valores $\alpha \in \{0, 0.25, 0.5, 0.75\}$ y $\alpha_i \sim U(0, 1)$.

Los resultados obtenidos se muestran en la Tabla 5.2, donde se ha promediado la probabilidad de detección de 100 ejecuciones. Dicha tabla muestra un resumen con los mejores resultados obtenidos para cada diseño (el diseño directo IB y los diseños re-equilibrados mediante Parzen, SMOTE y VoluSMOTE). Para un estudio en detalle de los resultados obtenidos y la búsqueda de los mejores parámetros se recomienda atender a las tablas presentes en los Apéndices de la Tesis. Como puede observarse, los resultados son prometedores. A continuación se discuten los resultados para cada base de datos por separado.

En la base de datos “Mammography”, el problema con menor número de dimensiones, se aprecia una clara diferencia en las prestaciones tras aplicar SMOTE o VoluSMOTE con respecto al diseño directo o la generación mediante ventanas de Parzen. Sin embargo, como era de esperar, no existen grandes diferencias entre SMOTE y el método propuesto. Esto se debe a que las limitaciones de SMOTE respecto

B. Datos	IB	Parzen (BR)	SMOTE (BR, K)	VoluSMOTE (BR, K, α)
Mamm.	0.833 ± 0.012	0.837 ± 0.011 (BR=16)	0.882 ± 0.007 (BR=2, K=4)	0.884 ± 0.003 (BR=2, K=3, $\alpha = 0$)
Ozone	0.451 ± 0.089	0.469 ± 0.068 (BR=4)	0.431 ± 0.048 (BR=4, K=20)	0.476 ± 0.087 (BR=16, K=20, $\alpha = 0.5$)
Pr. Hom.	0.890 ± 0.008	0.910 ± 0.007 (BR=16)	0.914 ± 0.006 (BR=16, K=15)	0.917 ± 0.007 (BR=4, K=2, $\alpha = 0.25$)
Pr. Hom. (red.)	0.852 ± 0.009	0.865 ± 0.006 (BR=16)	0.865 ± 0.006 (BR=2, K=20)	0.878 ± 0.005 (BR=4, K=20, $\alpha = 0.75$)

Tabla 5.2: Resultados obtenidos para las bases de datos "Mammography" (Mamm.), "Ozone-level" (Ozone) y las versiones completa y reducida de "Protein Homology" (Pr. Hom. y Pr. Hom. (red.), respectivamente). Se muestra la probabilidad de detección promedio (en términos de media y desviación típica para 100 ejecuciones) en el punto de trabajo P_{FAW} de cada problema. Se presentan el diseño directo (IB, desequilibrado) y los diseños re-equilibrados mediante Parzen, SMOTE y VoluSMOTE. Entre paréntesis se indican los valores de los parámetros explorados que han proporcionado los mejores resultados.

al número de dimensiones se ven atenuadas en este problema de sólo 6 dimensiones.

Por su parte, los resultados para la base de datos "Ozone-level" no son concluyentes debido a la gran inestabilidad de las prestaciones obtenidas −desviaciones típicas muy elevadas− provocada por el ínfimo número de muestras de la clase minoritaria. Sin embargo, parece que VoluSMOTE puede mitigar las dificultades de SMOTE

cuando el número de dimensiones es elevado. En este caso, aplicar SMOTE empeora las prestaciones obtenidas para el diseño directo, algo que no ocurre de forma tan evidente con VoluSMOTE. Aún así, se observa que se mantiene una de las limitaciones expuestas a lo largo de la tesis: un número muy reducido de muestras minoritarias dificulta la estimación del cociente de verosimilitudes, reduciendo el potencial de los métodos de re-equilibrado.

Para la versión completa de la base de datos "Protein Homology", se atisba una ligera mejora al utilizar VoluSMOTE respecto al resto de métodos de re-equilibrado. Dicha ventaja es muy reducida, ya que el elevado número de muestras disponibles de la clase minoritaria hacen que, pese al desequilibrio, incluso el diseño directo obtenga una alta probabilidad de detección para un nivel de falsa alarma tan exigente como el fijado en $P_{FAW} = 0.01$. Las diferencias entre SMOTE y el método propuesto son mínimas, algo que era de esperar, ya que un número tan elevado de muestras minoritarias supone la aparición de grupos densos de muestras, lo que reduce las limitaciones provocadas por la generación filiforme por parte de SMOTE. Estas diferencias son más notables cuando se reduce el número de muestras disponibles. En la versión reducida de "Protein Homology" se obtiene una clara mejora al utilizar VoluSMOTE respecto al resto de métodos de re-equilibrado. En este caso, el número de muestras −aunque menor− es suficiente para realizar una buena estimación del cociente de verosimilitudes, pero no tan alto como para que SMOTE evite la limitación debida a la generación en forma de "tela de araña", algo que sí se consigue con VoluSMOTE.

Los resultados obtenidos son prometedores. No obstante, antes de finalizar el capítulo, conviene recordar que se trata de una primera exploración de la técnica propuesta, con posiblemente mucho trabajo y estudio futuro aún por realizar para asegurar su validez y consistencia. Aún así, puede decirse que la idea propuesta tiene potencial para mejorar los resultados obtenidos por SMOTE en algunos problemas desequilibrados de clasificación con pocas muestras y muchas dimensiones, tal y como se previó en la discusión inicial de fortalezas y debilidades de SMOTE.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Aportaciones de la Tesis

La principal contribución de la presente Tesis al conocimiento sobre el Aprendizaje Máquina y su aplicación a problemas reales es la propuesta de una metodología fundamentada −mediante relaciones con la teoría bayesiana− para tratar los tan frecuentes como importantes problemas desequilibrados −aquellos en los que las diferencias en el tamaño de las poblaciones de cada clase o/y una política de costes sensiblemente distintos producen dificultades para darles solución−, en situaciones binarias.

Dicha metodología se basa en:

- la utilización de **divergencias de Bregman** como costes subrogados en el entrenamiento de las máquinas discriminativas con transformaciones entrenables −las de mayor capacidad expresiva; incluye los Perceptrones Multicapa, tanto superficiales como profundos− permite obtener de la salida de dichas máquinas estimaciones de la probabilidad *a posteriori* de una de las clases −se utiliza la minoritaria−; con lo que se establece la conexión con la teoría bayesiana, y, a través de ésta, es posible tener en cuenta la correspondencia uno a uno que existe entre la citada probabilidad *a posteriori* y el cociente de verosimilitudes

de las clases del problema, para establecer el procedimiento fundamentado;

- como quiera que la correspondencia entre la probabilidad *a posteriori* y el cociente de verosimilitudes implica que, de aplicarse modos de re-equilibrado que no alteren el **cociente de verosimilitudes, éste permanecerá invariante** en el problema re-equilibrado resultante; por lo que, estimando ese cociente a partir de la estimación de la probabilidad *a posteriori* de la clase minoritaria para dicho problema re-equilibrado, mediante la relación inversa se recuperará una estimación de la probabilidad *a posteriori* de la clase minoritaria para el problema original, el desequilibrado.

Debe resaltarse que la contextualización bayesiana que se lleva a cabo tiene como beneficio adicional que permite, mediante la variación de los umbrales de los test bayesianos −para el cociente de verosimilitudes o para la probabilidad *a posteriori*−, estimar la característica de operación −curva que fija la probabilidad de detección de un positivo con la de falsa alarma, o dar un negativo por positivo− óptima, o de Neyman-Pearson. Naturalmente, será necesario fijar un punto de trabajo para la aplicación práctica del algoritmo fundamentado que se diseñe para resolver un cierto problema, para lo que se tendrá en cuenta la política de costes y los tamaños de las poblaciones de muestras de las clases −estos tamaños, a efectos de estimar las probabilidades *a priori* de las clases mediante frecuencias relativas−, o bien se fijará un cierto valor para la probabilidad de falsa alarma −a partir de consideraciones prácticas sobre el problema a resolver−. Nótese que esta probabilidad marca una diferencia trascendental con respecto a los diseños no fundamentados para re-equilibrado: los últimos no permiten establecer relación alguna entre la característica de operación de la máquina diseñada y la de Neyman-Pearson, mientras que los diseños fundamentados dan lugar a una **característica de operación máquina (MOC) que estima la característica de operación de Neyman-Pearson (NPOC)**.

Naturalmente, la propuesta metodológica se completa especificando qué procesos de re-equilibrado dejan invariante el cociente de verosimilitudes, o, dicho de otro

modo, **qué procesos de re-equilibrado son neutrales**. Lo son:

- la **ponderación** −o énfasis− uniforme de todas las muestras de cada clase;

- el **re-muestreo** (sub o sobre) de muestras de una clase, también de manera uniforme para todas ellas; que es, en sentido estricto, una neutralidad estadística: que se conseguirá en grado suficiente mediante la utilización de conjuntos de aprendices obtenidos mediante dicho re-muestreo, con la ventaja añadida de la potencial mejora de prestaciones que cabe esperar de esta diversificación;

- la **generación** de muestras a partir de las de cada clase −generalmente, de la clase minoritaria−, de nuevo con requisito de uniformidad; y con las mismas observaciones −carácter estadístico y utilización de conjuntos− que para el re-muestreo;

- la **combinación** de cualesquiera de los anteriores.

Contribuciones directamente relacionadas con la recién expuesta aportación principal son las siguientes:

- Señalar los **riesgos** de pobres prestaciones −y aún de degeneración, i.e, de empeorarlas respecto a diseños directos sobre el problema desequilibrado− que conlleva la utilización de los conocidos (curiosamente) como **re-equilibrados informados** (de cualquiera de los tipos anteriores: ponderación, re-muestreo o generación), en los que, típicamente, se presta mayor atención y tratamiento a las muestras más próximas a la frontera y menor a las lejanas o a las que son, plausiblemente, muestras fuera de margen ("outliers"). Tales riesgos provienen de que, en general, dichos procedimientos informados no garantizan la invarianza del cociente de verosimilitudes y, por tanto, que un retorno al problema original proporcione los resultados esperados; y ello, aunque ocasionalmente pueden proporcionar prestaciones satisfactorias. En esta Tesis, la

manifestación real de dichos riesgos se comprueba de manera experimental para algunos procedimientos publicados que han sido empleados repetidamente, sin consciencia de esa limitación.

- Proponer **combinaciones de los anteriores métodos básicos**, con coeficientes que pueden validarse por métodos tradicionales, como la mejor opción para conseguir las más altas prestaciones −determinando también la intensidad del re-equilibrado conjunto que resulte más conveniente, ya que en absoluto puede decirse que un re-equilibrado completo ("full rebalancing"), el que iguala el "peso" de ambas clases, sea la opción preferible−. La razón para ello es que cada uno de los métodos básicos tiene sus puntos fuertes y débiles, que no coinciden: lo que significa que una combinación apropiada puede aprovecharlos o reducirlos. Brevemente:

  - la ponderación es sencilla, pero no proporciona diversidad;
  - el re-muestreo implica riesgo de eliminar muestras trascendentales o reforzar muestras irrelevantes −incluso muestras fuera de margen−, pero permite diversificar;
  - la generación no incurre en riesgos de eliminación o refuerzo, y también sirve para diversificar; no obstante, ha de administrarse su intensidad, porque en caso contrario daría lugar a deformaciones de las verosimilitudes y, con ello, a un empeoramiento práctico de las prestaciones.

  Teniendo lo anterior en cuenta, en el Capítulo 4 se evalúa la posibilidad de más interés: la combinación de ponderación y generación −en particular, del excelente método conocido como SMOTE−, comprobando que hay mejoras en buena parte de un conjunto de problemas representativos de diferentes características.

- En el trabajo experimental que se acaba de mencionar, se comprueba también que las situaciones en la que la metodología fundamentada que se propone

muestra (cierta) **debilidad** son esperables, dada su condición: problemas con

- muy pequeñas poblaciones
- muy alta dimensionalidad
- muestras muy ruidosas

(las dos primeras, ligadas entre sí). Nada extraño, y sí totalmente previsible: las estimaciones en que se apoya el método (de la probabilidad *a posteriori*, del cociente de verosimilitudes) se empobrecen bajo tales circunstancias.

- En respuesta al deseo de introducir procedimientos de re-equilibrado que tengan en cuenta la importancia relativa de las diferentes muestras para el objetivo de clasificación, se propone también un procedimiento **informado, pero fundamentado**. Se dan dos pasos:

  - en el primero, mediante un re-equilibrado neutral, se determinan las muestras que pueden considerarse más críticas, según su proximidad al umbral de decisión del test que se aplique;
  - en el segundo, se enfatizan las muestras teniendo en cuenta su nivel crítico, y se aplica una formulación que permite recuperar la solución del problema original a partir de la hallada para el seudoproblema que corresponde a un cociente de verosimilitudes alterado por el factor que implican las funciones de énfasis elegidas.

  Unos primeros experimentos (con énfasis muy sencillos) evidencian la validez y utilidad del procedimiento propuesto. En esta dirección, queda todo un camino a explorar, con diferentes y generales funciones de énfasis.

Desde una perspectiva general, se propone en la Tesis una denominación para el conjunto de tipos de problemas que ofrecen dificultades a una buena resolución

cuando se aplican algoritmos convencionales, y se revisan rápidamente las relaciones entre ellos. Se denominan, pues, **problemas singulares** a

- los específicamente tratados en la Tesis: desequilibrados;

- los problemas de clasificación cuyos costes (de asignación de una muestra de una clase a la que se decida) dependen también del valor de la muestra, o **clasificación con costes dependientes del ejemplo**. Pensando en finanzas, negocio o medicina fácilmente se encuentran problemas relevantes: concesión de crédito, detección de fraude, diagnóstico, etcétera. Los costes dependientes del ejemplo constituyen una forma que puede dar lugar a desequilibrio; además, en la práctica, estos problemas presentan también desequilibrio en su sentido tradicional −revísense los ejemplos citados más arriba−. La posibilidad de utilizar las herramientas fundamentadas aquí propuestas por el desequilibrio en estos problemas se discutirá más adelante, en el subapartado de direcciones de investigación abiertas;

- y, entre otros, los **problemas multietiqueta**, en los que cada muestra ha de clasificarse según diversas perspectivas; un ejemplo, la detección de diversos tipos de objetos en una imagen. También se volverá sobre éstos como dirección abierta. Otro caso análogo es la clasificación **multitarea**.

Se volverá sobre estos problemas con la discusión de las direcciones de investigación abiertas por los trabajos de esta Tesis.

Una aportación complementaria −ya que su empleo puede extenderse a otros muchos escenarios, como al entrenamiento de familias de redes profundas− es la propuesta rápidamente evaluada en el Capítulo 5: VoluSMOTE, una variante de SMOTE que busca reducir la principal debilidad de este método de generación. En efecto: SMOTE presenta la gran ventaja de generar muestras manteniendo una baja dimensionalidad local −lo hace en las conexiones con sus vecinos más próximos−;

como quiera que está universalmente aceptado que las poblaciones de muestras que corresponden a situaciones reales tienen una baja dimensionalidad intrínseca, parece ser esta característica la que justifica su eficacia y éxito. Pero, al tiempo, la generación se hace de modo que la población resultante tiende a ocupar una estructura filiforme, cuyo aspecto es el de una "tela de araña": lo que es poco realista. VoluSMOTE corrige este comportamiento mediante la combinación de cada muestra y sus vecinos más próximos, generando por SMOTE a partir de ella. Los primeros experimentos, que se exponen en la Tesis, avalan que puede proporcionar mejoras respecto a SMOTE en aplicaciones prácticas.

Como resumen de todo lo que antecede: en esta Tesis se ha construido un ámbito completo para la resolución de problemas de clasificación desequilibrada, de trascendental valor: se apoya en los cimientos más sólidos, la Teoría de Bayes; así, permite un control razonado, y además tiende puentres entre las máquinas discriminativas y las generativas estadísticas.

6.2. Direcciones de investigación abiertas

Para no alargar indebidamente este subapartado, sólo se incluirán orientaciones cuyos trabajos se hayan iniciado −aunque no hayan proporcionado aún resultados completos−. Se prescinde, pues, de otras vías que simplemente sean posibilidades concebidas.

6.2.1. Problemas multiclase y ordinales

Evidentemente, formular el equivalente de lo expuesto en esta Tesis a situaciones multiclase es la extensión natural.

Ha de aclararse que, como se puede comprobar en los artículos de carácter tutorial o de revisión sobre multiclase que ya se han citado, hay una fuerte controversia acerca

de las ventajas relativas de las aproximaciones básicas para resolver estos problemas. En los estudios iniciados por el equipo de investigación que realizó esta Tesis, la aproximación monolítica (una sola máquina o un conjunto de aprendices con tantas salidas como clases, con activaciones softmax −también conocidas como activaciones de Potts−) se estima como descartable, puesto que sólo permite un esquema de re-equilibrado, y el compromiso entre lo que conviene para distinguir diferentes pares de clases no se puede resolver.

Distintas son las cosas cuando se binariza un problema multiclase, i.e., se obtiene la solución mediante la combinación de los resultados de una colección de problemas binarios convenientemente elegidos, lo que además incluye diversidad. Tres modos hay de llevarlo a cabo: enfrentar pares de clases de todas las formas posibles −uno contra uno ("One vs. One")−, enfrentar cada una de las clases contra todas las demás −uno contra el resto ("One vs. Rest")−, y agrupar clases en dos bloques que se enfrentan entre sí, siendo los agrupamientos tales que permiten deducir de sus resultados la clase vencedora −por códigos de salida correctores de errores (ECOC, "Error Correcting Output Codes")−. De nuevo por razones de brevedad no se incluye la discusión de los puntos fuertes y débiles −en representatividad, tamaño y dificultad de diseño−, pero se invita a consultar lo que se expone en los textos de carácter monográfico [Kuncheva, 2004] y [Rokach, 2010]. Pero puede anticiparse, como resumen de los trabajos en curso, que esta Tesis:

- posibilita realizar agregaciones fundamentadas de los resultados de los aprendices, superando los habituales promediado y votación;

- facilita el re-equilibrado de cualesquiera de los problemas binarios que se plantean, con lo que se potencia la opción de "One vs. Rest" (aunque debe aclararse que, en este caso como en los demás, no ha de optimizarse el re-equilibrado de cada dicotomía, sino el conjunto de esos re-equilibrados) y, muy en particular, de la opción "Error Correcting Output Codes", la de mayor potencia pero de difícil diseño salvo para un número modesto de clases. Aquí, aunque no tenga

relación directa con la Tesis, debe mencionarse que los trabajos en curso han conducido a proponer como método sistemático de diseño de códigos la utilización de las secuencias de Walsh, de mayor compacidad que los tradicionales con secuencias de Rademacher y de prestaciones comparables.

6.2.2. Extensiones a otros problemas singulares

- Las más avanzadas son las aplicadas a clasificación con costes dependientes del ejemplo, que ya se han presentado páginas atrás. Se ha remitido ya un artículo [Mediavilla-Relaño et al., 2020] −contiene una breve revisión bibliográfica− para casos binarios, en el que se sigue una vía ya conocida, pero aplicada fuera del contexto bayesiano: primero se estima la probabilidad *a posteriori*, utilizando el re-equilibrado fundamentado de esta Tesis; después, se aplica el test de Bayes utilizando dicha estimación. Los resultados mejoran los conseguidos por otros autores en la mayoría de bases de datos reales examinadas en la literatura.

  Además, se ha diseñado un procedimiento en una sola fase basado en considerar los efectos de los costes de clasificación como debidos a énfasis sobre las muestras de las clases y aplicar la formulación que se expone en esta Tesis. Los resultados son satisfactorios, y ya se prepara un artículo para remisión a revista internacional.

- También se están iniciando trabajos sobre clasificación ordinal: aquella en que las clases constituyen una escala o "ranking", siendo los costes de clasificación mayores para saltos mayores; véase [Gutiérrez et al., 2015] para mayor detalle. Dichos trabajos parten de que, obviamente, se trata de problemas en los que se combinan las múltiples clases con tales formas de costes; y se procede en consecuencia. Aún no se dispone de resultados significativos.

- Por último, se encuentran recién comenzadas las tareas encaminadas a la resolución fundamentada de problemas de clasificación con etiquetas múltiples.

Concretamente, se ha formulado una forma algorítmica para su resolución consistente en la aplicación consecutiva de la relación entre probabilidad conjunta y probabilidad condicional, $Pr(A, B) = Pr(A|B)Pr(B)$, procediendo blanco a blanco para obtener problemas binarios en cada uno de ellos, que requieren la inyección a la entrada de blancos previos: en la práctica, las decisiones −blandas o duras− de las máquinas previas. Nótese que cabe diversificar combinando las secuencias. En este proceso tienen cabida las técnicas propuestas en la Tesis, con lo que se manejan procedimientos fundamentados, a diferencia de lo que se ha propuesto hasta hoy, en esencia metodologías empíricas: véase la monografía [Herrera et al., 2016]. Debe decirse que, una vez completados los trabajos sobre multiclase, cabe aquí secuenciar por bloques, y no etiqueta a etiqueta.

Que la lista se cierre aquí en modo alguno significa que no existan otras rutas para extender los resultados. En particular, para los conocidos como problemas multitarea −véase una aplicación de estos en [García-Laencina et al., 2013]−, que pueden considerarse análogos a los multietiqueta trabajando con densidades de probabilidad de variables continuas en lugar de con probabilidades de variables discretas −el texto [Husmeier, 2012] aclara cómo hacerlo−. Versiones más flexibles −como la secuenciación paralela a la de clasificación multietiqueta− darían lugar a diseños más sencillos, pero potencialmente ventajosos (a causa de trabajar con aprendices de salida única, y por la posibilidad de ordenaciones diversas).

6.2.3. Otras extensiones

Se pasa ahora a otras clases de extensiones.

- La construcción de conjuntos con algoritmos de “Boosting” sigue ciertos principios analíticos −así, el Real Adaboost [Freund and Schapire, 1996a] [Freund and Schapire, 1996b] [Freund and Schapire, 1997] puede derivarse de la minimización de una cierta cota del error de clasificación−, pero muchas variantes son

empíricas. Si se tiene en cuenta que lo fundamental en estos procesos iterativos de construcción radica en enfatizar las muestras que mayor resistencia ofrezcan a una correcta clasificación, la metodología presentada en esta Tesis para tratar con tales énfasis hace posible una construcción fundamentada: se elige una forma general para el énfasis de las muestras de cada clase −plausiblemente teniendo en cuenta su error y su proximidad a la frontera, combinándolos según parámetros validables− y se obtiene una estimación de la probabilidad *a posteriori* de la clase 1, por ejemplo; que se va agregando paso a paso con la salida de las agregaciones previas, y con el resultado se enfatizan las muestras para el paso siguiente. La formulación correspondiente ha sido desarrollada completamente por el equipo de investigación en cuyo seno se ha preparado esta Tesis, incluyendo la consideración de desequilibrio y de costes dependientes del ejemplo; y se va a iniciar de inmediato la evaluación de sus prestaciones. Debe destacarse que la inclusión de desequilibrio y de costes dependientes del ejemplo es inmediata con esta formulación, mientras que precisa de complejas modificaciones si la construcción por "Boosting" del conjunto se acomete mediante otros procedimientos. Además, queda abierta la ruta hacia diseños fundamentados de "Boosting" para problemas multiclase, muy en particular usando binarización.

- Puede decirse que VoluSMOTE es un método razonable y sencillo de remediar el más grave inconveniente que presenta el método SMOTE original −que, por otra parte, ha acreditado su eficacia en muchas aplicaciones−: su estructura en "tela de araña". No obstante, ni con mucho se puede afirmar que es la única posibilidad de evitar la citada limitación: lo esencial es rellenar la "tela de araña" sin deteriorar la baja dimensionalidad local, y eso se puede conseguir mediante otros procedimientos que sitúen muestras entre los hilos de la "tela", como, por poner un ejemplo sencillo, generar las muestras mediante una ventana de Parzen modificada para evitar un aumento sensible de la dimensionalidad. Con ello se pone de manifiesto que VoluSMOTE es un paso inicial

en la exploración de métodos de generación de muestras que compartan las propiedades de mayor importancia: neutralidad, sencillez y baja dimensionalidad local. El interés de este ámbito de trabajo se debe no sólo a su interés para funciones de re-equilibrado fundamentado y asociadas: hay otras aplicaciones de importancia, como la generación de muestras para la construcción de redes profundas de la familia de los llamados clasificadores o estimadores por apilamiento de autocodificadores con reducción de ruido ("Stacked Denoising Auto-Encoders"), propuestos inicialmente por [Vincent et al., 2010] y cuyas modificaciones han demostrado extraordinario potencial: véase, por ejemplo, [Sánchez-Morales, 2019]. Por otro lado, estos procedimientos suponen aportaciones a la muy deseable conexión entre las máquinas discriminativas y las generativas (modelos estadísticos), asunto del que se habla justo a continuación.

- Como acaba de mencionarse, tender puentes entre las máquinas discriminativas y las generativas es un propósito que está mereciendo una atención rápidamente creciente, actualmente muy intensa: aprovechar los puntos fuertes de ambas clases de máquinas mediante su asociación o su interacción −ya se han mencionado técnicas de mejora de máquinas discriminativas mediante el sucesivo intercambio de muestras y resultados [Goodfellow et al., 2014]− constituye un objetivo de importancia innegable. Pues bien, para tal tarea, la relevancia de las divergencias de Bregman es obvia, al ligar salidas de máquinas discriminativas con probabilidades a posteriori; y su potenciación mediante la formulación básica presentada en esta Tesis es, en consecuencia, también de alto valor.

  Hay muchas formas en que el manejo simultáneo de máquinas discriminativas y máquinas generativas produce beneficio. Una, cuya exploración se ha iniciado, es como sigue: ante las limitaciones que las soluciones convencionales de selección de variables, como la medida de su información mutua con la salida, plantean para la reducción de dimensiones sin transformaciones −no debe olvidarse que es un problema NP-completo−, ya hace dos decenios que empezaron

a considerarse soluciones, llamadas métodos de subespacios, para conseguirla en el propio proceso de diseño de máquinas discriminativas. Lo más sencillo: construir diversos aprendices con distintos subconjuntos de las variables disponibles, y fusionar sus salidas; lo que se conoce como métodos de los subespacios aleatorios, y se expone, entre otros artículos, en [Ho, 1998]. Pues bien: complementar cada uno de los aprendices máquina con un modelo estadístico que represente la contribución a la decisión bayesiana del resto de las variables es posible gracias a la visión probabilística de las máquinas discriminativas que nace con las divergencias de Bregman y se amplía es esta Tesis. En particular, tiene especial atractivo el empleo de ventanas de Parzen como modelo probabilístico complementario, ya que su complejidad es accesible en los casos que conducen a recurrir a los métodos de subespacios aleatorios: pocas muestras y muchas dimensiones −que restan fiabilidad a los modelos semiparamétricos, como las mezclas de distribuciones−.

- Por último, se debe señalar que la formulación aquí introducida sirve de base para atacar de modo fundamentado en Bayes la explicabilidad de las decisiones de máquinas discriminativas en los ámbitos en que resulta ya no recomendable, sino hasta exigible: por ejemplo, salud y finanzas. Aunque la formulación básica para este fin se ha establecido, e incluso se han llevado a cabo experimentos preliminares, no se pasa al detalle por tratarse de un tema que requiere una buena dosis de confidencialidad.

Como se ha podido ver en este capítulo, puede asegurarse que las aportaciones de la Tesis son valiosas, y también la importancia de las muchas direcciones de investigación −y futuras aplicaciones− que ya ha permitido abrir.

Apéndice A

Resultados obtenidos en los experimentos de VoluSMOTE

En la Tabla 5.2 se muestra un resumen −para facilitar la lectura− con los mejores resultados obtenidos en los experimentos realizados en el Capítulo 5. Dicha tabla muestra el promedio tras 100 ejecuciones de la probabilidad de detección obtenida aplicando cada una de las técnicas de re-equilibrado (Parzen, SMOTE y VoluSMOTE) y el diseño directo (desequilibrado) en el punto de trabajo de cada base de datos, así como los parámetros con los que se han obtenido. Sin embargo, el proceso de obtención de los resultados ha sido el siguiente:

1) Explorar todas las combinaciones de parámetros y obtener la probabilidad de detección (en promedio) tras 20 ejecuciones −5 ejecuciones en el caso de la base de datos "Protein Homology" debido al tamaño de dicha base de datos−.

2) Seleccionar los mejores parámetros para cada técnica según los resultados del paso anterior, así como sus valores próximos.

3) Obtener la probabilidad de detección (en promedio) tras 100 ejecuciones y selección de los mejores resultados que se muestran en la Tabla 5.2.

A continuación, se presentan los resultados obtenidos en el paso 1) tras aplicar las técnicas de re-equilibrado para cada una de las bases de datos.

	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
Parzen	0.8038 ± 0.0091	0.8076 ± 0.0099	0.8192 ± 0.0098	0.8307 ± 0.0154	0.8416 ± 0.0124

Tabla A.1: Resultados obtenidos para "Mammography" tras aplicar Parzen. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

SMOTE	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8551 ± 0.0091	0.8788 ± 0.0103	0.8608 ± 0.0163	0.8480 ± 0.0129	0.8358 ± 0.0086
$K = 3$	0.8628 ± 0.0082	0.8814 ± 0.0089	0.8673 ± 0.0147	0.8615 ± 0.0174	0.8358 ± 0.0065
$K = 4$	0.8653 ± 0.0075	0.8801 ± 0.0093	0.8743 ± 0.0131	0.8576 ± 0.0176	0.8391 ± 0.0137
$K = 5$	0.8621 ± 0.0106	0.8788 ± 0.0124	0.8692 ± 0.0174	0.8487 ± 0.0143	0.8358 ± 0.0051
$K = 6$	0.8673 ± 0.0109	0.8801 ± 0.0116	0.8621 ± 0.0193	0.8487 ± 0.0104	0.8384 ± 0.0094

Tabla A.2: Resultados obtenidos para "Mammography" tras aplicar SMOTE. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8416 ± 0.0142	0.8826 ± 0.0061	0.8762 ± 0.0153	0.8608 ± 0.0173	0.8391 ± 0.0103
$K = 3$	0.8554 ± 0.0130	0.8839 ± 0.0027	0.8769 ± 0.0153	0.8666 ± 0.0173	0.8551 ± 0.0135
$K = 4$	0.8570 ± 0.0136	0.8833 ± 0.0038	0.8762 ± 0.0153	0.8608 ± 0.0177	0.8493 ± 0.0106
$K = 5$	0.8506 ± 0.0144	0.8820 ± 0.0065	0.8737 ± 0.0163	0.8538 ± 0.0142	0.8480 ± 0.0093
$K = 6$	0.8387 ± 0.0139	0.8794 ± 0.0062	0.8544 ± 0.0163	0.8519 ± 0.0124	0.8487 ± 0.0086
VoluSMOTE ($\alpha = 0.25$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8560 ± 0.0119	0.8750 ± 0.0068	0.8743 ± 0.0149	0.8564 ± 0.0160	0.8403 ± 0.0110
$K = 3$	0.8615 ± 0.0089	0.8807 ± 0.0058	0.8750 ± 0.0166	0.8506 ± 0.0116	0.8467 ± 0.0131
$K = 4$	0.8650 ± 0.0150	0.8839 ± 0.0027	0.8826 ± 0.0083	0.8589 ± 0.0181	0.8512 ± 0.0110
$K = 5$	0.8644 ± 0.0116	0.8820 ± 0.0086	0.8673 ± 0.0182	0.8506 ± 0.0116	0.8474 ± 0.0055
$K = 6$	0.8503 ± 0.0169	0.8801 ± 0.0061	0.8608 ± 0.0182	0.8506 ± 0.0116	0.8474 ± 0.0089

Tabla A.3: Resultados obtenidos para "Mammography" tras aplicar VoluSMOTE con $\alpha \in \{0, 0.25\}$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.5$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8506 ± 0.0101	0.8750 ± 0.0068	0.8743 ± 0.0154	0.8532 ± 0.0159	0.8391 ± 0.0094
$K = 3$	0.8608 ± 0.0093	0.8730 ± 0.0080	0.8730 ± 0.0161	0.8564 ± 0.0174	0.8448 ± 0.0145
$K = 4$	0.8589 ± 0.0107	0.8756 ± 0.0071	0.8820 ± 0.0086	0.8596 ± 0.0183	0.8564 ± 0.0174
$K = 5$	0.8608 ± 0.0081	0.8801 ± 0.0093	0.8608 ± 0.0177	0.8525 ± 0.0148	0.8487 ± 0.0125
$K = 6$	0.8641 ± 0.0098	0.8782 ± 0.0075	0.8660 ± 0.0208	0.8576 ± 0.0166	0.8455 ± 0.0063
VoluSMOTE ($\alpha = 0.75$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8570 ± 0.0101	0.8794 ± 0.0117	0.8641 ± 0.0153	0.8506 ± 0.0147	0.8378 ± 0.0093
$K = 3$	0.8615 ± 0.0104	0.8750 ± 0.0089	0.8685 ± 0.0156	0.8660 ± 0.0187	0.8487 ± 0.0143
$K = 4$	0.8538 ± 0.0102	0.8743 ± 0.0104	0.8692 ± 0.0174	0.8576 ± 0.0161	0.8423 ± 0.0082
$K = 5$	0.8512 ± 0.0124	0.8717 ± 0.0107	0.8628 ± 0.0199	0.8474 ± 0.0113	0.8455 ± 0.0159
$K = 6$	0.8564 ± 0.0076	0.8698 ± 0.0101	0.8608 ± 0.0223	0.8455 ± 0.0143	0.8391 ± 0.0063

Tabla A.4: Resultados obtenidos para "Mammography" tras aplicar VoluSMOTE con $\alpha \in \{0.5, 0.75\}$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha_i \sim U(0, 1)$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8621 ± 0.0106	0.8762 ± 0.0093	0.8730 ± 0.0161	0.8544 ± 0.0136	0.8358 ± 0.0051
$K = 3$	0.8653 ± 0.0086	0.8794 ± 0.0062	0.8679 ± 0.0167	0.8570 ± 0.0168	0.8435 ± 0.0149
$K = 4$	0.8653 ± 0.0086	0.8826 ± 0.0045	0.8782 ± 0.0137	0.8538 ± 0.0153	0.8435 ± 0.0095
$K = 5$	0.8628 ± 0.0108	0.8814 ± 0.0089	0.8717 ± 0.0181	0.8538 ± 0.0153	0.8512 ± 0.0130
$K = 6$	0.8615 ± 0.0118	0.8814 ± 0.0055	0.8596 ± 0.0196	0.8519 ± 0.0137	0.8461 ± 0.0121

Tabla A.5: Resultados obtenidos para "Mammography" tras aplicar VoluSMOTE con $\alpha_i \sim U(0, 1)$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
Parzen	0.4583 ± 0.0783	0.4527 ± 0.0663	0.4583 ± 0.0523	0.4416 ± 0.0832	0.4472 ± 0.0886

Tabla A.6: Resultados obtenidos para "Ozone-level" tras aplicar Parzen. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

SMOTE	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.3694 ± 0.0403	0.3472 ± 0.0425	0.3694 ± 0.0589	0.3583 ± 0.0511	0.3750 ± 0.0605
$K = 3$	0.4027 ± 0.0552	0.3888 ± 0.0496	0.3694 ± 0.0440	0.3666 ± 0.0477	0.4027 ± 0.0579
$K = 4$	0.3888 ± 0.0608	0.3833 ± 0.0606	0.3972 ± 0.0771	0.4000 ± 0.0715	0.4083 ± 0.0615
$K = 5$	0.3833 ± 0.0580	0.3805 ± 0.0589	0.3749 ± 0.0605	0.3972 ± 0.0589	0.4055 ± 0.0585
$K = 6$	0.3666 ± 0.0732	0.3333 ± 0.0527	0.3694 ± 0.0505	0.3777 ± 0.0647	0.3944 ± 0.0722
$K = 10$	0.4027 ± 0.0425	0.4138 ± 0.0371	0.3861 ± 0.0644	0.4055 ± 0.0499	0.3888 ± 0.0527
$K = 15$	0.4194 ± 0.0620	0.4416 ± 0.0371	0.4138 ± 0.0540	0.4250 ± 0.0615	0.4277 ± 0.0726
$K = 20$	0.4250 ± 0.0403	0.4277 ± 0.0433	0.4472 ± 0.0447	0.4222 ± 0.0812	0.4333 ± 0.0737

Tabla A.7: Resultados obtenidos para "Ozone-level" tras aplicar SMOTE. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.4277 ± 0.0558	0.3888 ± 0.0496	0.4055 ± 0.0635	0.3805 ± 0.0615	0.3944 ± 0.0461
$K = 3$	0.3805 ± 0.0403	0.3638 ± 0.0644	0.3916 ± 0.0480	0.4166 ± 0.0412	0.4222 ± 0.0643
$K = 4$	0.4027 ± 0.0460	0.3805 ± 0.0440	0.3722 ± 0.0499	0.4111 ± 0.0643	0.4305 ± 0.0647
$K = 5$	0.4194 ± 0.0540	0.4333 ± 0.0415	0.4277 ± 0.0659	0.4388 ± 0.0552	0.4277 ± 0.0767
$K = 6$	0.4388 ± 0.0630	0.4361 ± 0.0686	0.4055 ± 0.0558	0.4027 ± 0.0387	0.4444 ± 0.0582
$K = 10$	0.4083 ± 0.0589	0.3972 ± 0.0730	0.4361 ± 0.0708	0.4361 ± 0.0771	0.4861 ± 0.0722
$K = 15$	0.4000 ± 0.0623	0.4055 ± 0.0611	0.4166 ± 0.0621	0.4277 ± 0.0825	0.4527 ± 0.0663
$K = 20$	0.4666 ± 0.0538	0.4527 ± 0.0708	0.4583 ± 0.0605	0.4527 ± 0.0950	0.4527 ± 0.0608

Tabla A.8: Resultados obtenidos para "Ozone-level" tras aplicar VoluSMOTE con $\alpha = 0$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.25$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.3777 ± 0.0415	0.3777 ± 0.0451	0.4055 ± 0.0558	0.3888 ± 0.0464	0.4000 ± 0.0598
$K = 3$	0.3805 ± 0.0615	0.3805 ± 0.0403	0.3805 ± 0.0440	0.4083 ± 0.0563	0.4027 ± 0.0699
$K = 4$	0.4111 ± 0.0408	0.4027 ± 0.0579	0.3944 ± 0.0524	0.4000 ± 0.0571	0.3944 ± 0.0654
$K = 5$	0.4111 ± 0.0643	0.4222 ± 0.0538	0.4111 ± 0.0618	0.4083 ± 0.0535	0.4500 ± 0.0580
$K = 6$	0.4388 ± 0.0630	0.4472 ± 0.0480	0.4388 ± 0.0700	0.4333 ± 0.0777	0.4361 ± 0.0998
$K = 10$	0.4500 ± 0.0783	0.4555 ± 0.0693	0.4388 ± 0.0700	0.4611 ± 0.0914	0.4055 ± 0.0558
$K = 15$	0.4333 ± 0.0571	0.4388 ± 0.0630	0.4583 ± 0.0742	0.4472 ± 0.0620	0.4277 ± 0.0767
$K = 20$	0.4555 ± 0.0647	0.4416 ± 0.0668	0.4722 ± 0.0734	0.4361 ± 0.0686	0.4583 ± 0.0960

Tabla A.9: Resultados obtenidos para "Ozone-level" tras aplicar VoluSMOTE con $\alpha = 0.25$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.5$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.4055 ± 0.0500	0.3611 ± 0.0372	0.3750 ± 0.0425	0.3916 ± 0.0371	0.4027 ± 0.0677
$K = 3$	0.3638 ± 0.0480	0.3944 ± 0.0524	0.3888 ± 0.0633	0.4000 ± 0.0647	0.4194 ± 0.0620
$K = 4$	0.3916 ± 0.0511	0.4027 ± 0.0493	0.3861 ± 0.0511	0.4194 ± 0.0540	0.4083 ± 0.0589
$K = 5$	0.3972 ± 0.0589	0.4305 ± 0.0387	0.4333 ± 0.0623	0.4444 ± 0.0745	0.4555 ± 0.0571
$K = 6$	0.4333 ± 0.0598	0.4277 ± 0.0558	0.4388 ± 0.0654	0.4472 ± 0.0734	0.4333 ± 0.0671
$K = 10$	0.4111 ± 0.0408	0.4166 ± 0.0595	0.4388 ± 0.0783	0.4583 ± 0.0654	0.4138 ± 0.0568
$K = 15$	0.4277 ± 0.0558	0.4638 ± 0.0771	0.4694 ± 0.0644	0.4527 ± 0.0589	0.4388 ± 0.0580
$K = 20$	0.4388 ± 0.0743	0.4444 ± 0.0702	0.4666 ± 0.0689	0.4500 ± 0.0822	0.4805 ± 0.0640

Tabla A.10: Resultados obtenidos para "Ozone-level" tras aplicar VoluSMOTE con $\alpha = 0.5$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.75$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.3555 ± 0.0509	0.3666 ± 0.0444	0.3777 ± 0.0544	0.3666 ± 0.0689	0.4305 ± 0.0654
$K = 3$	0.3833 ± 0.0493	0.3694 ± 0.0363	0.3583 ± 0.0668	0.3833 ± 0.0678	0.4027 ± 0.0699
$K = 4$	0.3916 ± 0.0371	0.4055 ± 0.0355	0.4138 ± 0.0713	0.3944 ± 0.0580	0.4361 ± 0.0730
$K = 5$	0.4000 ± 0.0571	0.4194 ± 0.0620	0.4222 ± 0.0666	0.4138 ± 0.0644	0.4250 ± 0.0686
$K = 6$	0.3944 ± 0.0580	0.4333 ± 0.0623	0.4111 ± 0.0753	0.4305 ± 0.0699	0.4361 ± 0.0791
$K = 10$	0.4361 ± 0.0363	0.4611 ± 0.0529	0.4388 ± 0.0803	0.4611 ± 0.0611	0.4361 ± 0.0686
$K = 15$	0.4027 ± 0.0579	0.4027 ± 0.0630	0.4555 ± 0.0544	0.4694 ± 0.0620	0.4277 ± 0.0659
$K = 20$	0.4250 ± 0.0505	0.4611 ± 0.0500	0.4472 ± 0.0620	0.4527 ± 0.0535	0.4638 ± 0.0708

Tabla A.11: Resultados obtenidos para "Ozone-level" tras aplicar VoluSMOTE con $\alpha = 0.75$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha_i \sim U(0, 1)$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.3750 ± 0.0387	0.3694 ± 0.0505	0.3750 ± 0.0523	0.3916 ± 0.0480	0.4166 ± 0.0775
$K = 3$	0.3638 ± 0.0595	0.3388 ± 0.0524	0.3888 ± 0.0527	0.3694 ± 0.0440	0.4250 ± 0.0883
$K = 4$	0.3500 ± 0.0529	0.3777 ± 0.0484	0.3861 ± 0.0595	0.3972 ± 0.0589	0.4277 ± 0.0611
$K = 5$	0.3694 ± 0.0535	0.4000 ± 0.0544	0.4055 ± 0.0433	0.4055 ± 0.0396	0.4250 ± 0.0730
$K = 6$	0.3750 ± 0.0523	0.4138 ± 0.0480	0.4111 ± 0.0711	0.4305 ± 0.0699	0.4222 ± 0.0711
$K = 10$	0.4083 ± 0.0563	0.4194 ± 0.0668	0.4555 ± 0.0544	0.4500 ± 0.0700	0.4388 ± 0.0783
$K = 15$	0.4333 ± 0.0693	0.4305 ± 0.0425	0.4083 ± 0.0640	0.4500 ± 0.0803	0.4500 ± 0.0803
$K = 20$	0.4500 ± 0.0678	0.4527 ± 0.0640	0.4472 ± 0.0620	0.4444 ± 0.0765	0.4777 ± 0.0593

Tabla A.12: Resultados obtenidos para "Ozone-level" tras aplicar VoluSMOTE con $\alpha_i \sim U(0, 1)$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 20 ejecuciones) en términos de media y desviación típica.

	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
Parzen	0.9012 ± 0.0039	0.9024 ± 0.0057	0.9055 ± 0.0079	0.9074 ± 0.0027	0.9074 ± 0.0061
SMOTE ($K = 2$)	0.9030 ± 0.0041	0.9117 ± 0.0037	0.9135 ± 0.0039	0.9129 ± 0.0059	0.9129 ± 0.0045
SMOTE ($K = 3$)	0.9086 ± 0.0092	0.9086 ± 0.0037	0.9092 ± 0.0041	0.9111 ± 0.0096	0.9086 ± 0.0041
SMOTE ($K = 4$)	0.9117 ± 0.0057	0.9092 ± 0.0041	0.9104 ± 0.0043	0.9141 ± 0.0035	0.9129 ± 0.0083
SMOTE ($K = 5$)	0.9086 ± 0.0050	0.9129 ± 0.0059	0.9148 ± 0.0063	0.9092 ± 0.0084	0.9148 ± 0.0057
SMOTE ($K = 6$)	0.9067 ± 0.0045	0.9117 ± 0.0069	0.9111 ± 0.0053	0.9123 ± 0.0060	0.9098 ± 0.0049
SMOTE ($K = 10$)	0.9104 ± 0.0033	0.9129 ± 0.0049	0.9123 ± 0.0031	0.9154 ± 0.0024	0.9098 ± 0.0059
SMOTE ($K = 15$)	0.9104 ± 0.0033	0.9104 ± 0.0047	0.9135 ± 0.0051	0.9092 ± 0.0057	0.9141 ± 0.0035
SMOTE ($K = 20$)	0.9117 ± 0.0041	0.9123 ± 0.0046	0.9135 ± 0.0043	0.9104 ± 0.0043	0.9061 ± 0.0063

Tabla A.13: Resultados obtenidos para "Protein Homology" tras aplicar Parzen y SMOTE. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.9074 ± 0.0087	0.9067 ± 0.0094	0.9104 ± 0.0058	0.9111 ± 0.0053	0.9160 ± 0.0040
$K = 3$	0.9061 ± 0.0046	0.9061 ± 0.0015	0.9061 ± 0.0084	0.9055 ± 0.0063	0.9104 ± 0.0085
$K = 4$	0.9086 ± 0.0063	0.9061 ± 0.0024	0.9098 ± 0.0012	0.9067 ± 0.0045	0.9024 ± 0.0074
$K = 5$	0.9092 ± 0.0031	0.9043 ± 0.0067	0.9043 ± 0.0051	0.9049 ± 0.0062	0.9104 ± 0.0051
$K = 6$	0.9049 ± 0.0023	0.9018 ± 0.0030	0.9018 ± 0.0030	0.9018 ± 0.0085	0.9080 ± 0.0102
$K = 10$	0.9018 ± 0.0035	0.9049 ± 0.0049	0.8987 ± 0.0040	0.9098 ± 0.0035	0.9074 ± 0.0055
$K = 15$	0.8962 ± 0.0069	0.9067 ± 0.0035	0.9037 ± 0.0032	0.9000 ± 0.0041	0.9061 ± 0.0053
$K = 20$	0.9030 ± 0.0090	0.8919 ± 0.0043	0.9049 ± 0.0035	0.9067 ± 0.0049	0.9037 ± 0.0088

Tabla A.14: Resultados obtenidos para "Protein Homology" tras aplicar VoluSMOTE con $\alpha = 0$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.25$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.9104 ± 0.0085	0.9141 ± 0.0112	0.9216 ± 0.0041	0.9129 ± 0.0049	0.9129 ± 0.0079
$K = 3$	0.9000 ± 0.0041	0.9086 ± 0.0046	0.9135 ± 0.0073	0.9067 ± 0.0085	0.9074 ± 0.0027
$K = 4$	0.9061 ± 0.0015	0.9129 ± 0.0030	0.9148 ± 0.0050	0.9098 ± 0.0040	0.9117 ± 0.0041
$K = 5$	0.9030 ± 0.0057	0.9098 ± 0.0053	0.9086 ± 0.0050	0.9092 ± 0.0024	0.9117 ± 0.0066
$K = 6$	0.9055 ± 0.0077	0.9049 ± 0.0030	0.9043 ± 0.0033	0.9055 ± 0.0041	0.9043 ± 0.0058
$K = 10$	0.9061 ± 0.0059	0.9030 ± 0.0084	0.9024 ± 0.0069	0.9037 ± 0.0045	0.9061 ± 0.0079
$K = 15$	0.9000 ± 0.0060	0.9018 ± 0.0094	0.9012 ± 0.0039	0.9074 ± 0.0051	0.9061 ± 0.0024
$K = 20$	0.8981 ± 0.0070	0.8969 ± 0.0069	0.9062 ± 0.0046	0.9080 ± 0.0076	0.9049 ± 0.0040

Tabla A.15: Resultados obtenidos para "Protein Homology" tras aplicar VoluSMOTE con $\alpha = 0.25$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.5$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.9123 ± 0.0037	0.9111 ± 0.0045	0.9129 ± 0.0081	0.9086 ± 0.0031	0.9148 ± 0.0066
$K = 3$	0.9055 ± 0.0069	0.9080 ± 0.0045	0.9074 ± 0.0070	0.9154 ± 0.0046	0.9135 ± 0.0099
$K = 4$	0.9135 ± 0.0043	0.9135 ± 0.0055	0.9117 ± 0.0053	0.9074 ± 0.0039	0.9141 ± 0.0056
$K = 5$	0.9117 ± 0.0069	0.9055 ± 0.0071	0.9129 ± 0.0056	0.9104 ± 0.0043	0.9117 ± 0.0063
$K = 6$	0.9037 ± 0.0053	0.9061 ± 0.0081	0.9092 ± 0.0074	0.9098 ± 0.0053	0.9074 ± 0.0064
$K = 10$	0.9024 ± 0.0031	0.9012 ± 0.0058	0.9067 ± 0.0035	0.9098 ± 0.0068	0.9067 ± 0.0045
$K = 15$	0.8987 ± 0.0023	0.9000 ± 0.0057	0.9098 ± 0.0059	0.9043 ± 0.0101	0.9043 ± 0.0019
$K = 20$	0.8987 ± 0.0040	0.9030 ± 0.0071	0.9043 ± 0.0073	0.9043 ± 0.0080	0.9049 ± 0.0045

Tabla A.16: Resultados obtenidos para "Protein Homology" tras aplicar VoluSMOTE con $\alpha = 0.5$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.75$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.9067 ± 0.0053	0.9111 ± 0.0053	0.9092 ± 0.0100	0.9074 ± 0.0080	0.9135 ± 0.0058
$K = 3$	0.9043 ± 0.0043	0.9123 ± 0.0079	0.9092 ± 0.0079	0.9104 ± 0.0070	0.9123 ± 0.0050
$K = 4$	0.9061 ± 0.0050	0.9092 ± 0.0060	0.9067 ± 0.0065	0.9179 ± 0.0079	0.9129 ± 0.0035
$K = 5$	0.9067 ± 0.0065	0.9074 ± 0.0064	0.9061 ± 0.0050	0.9086 ± 0.0041	0.9086 ± 0.0041
$K = 6$	0.9074 ± 0.0110	0.9111 ± 0.0059	0.9111 ± 0.0023	0.9055 ± 0.0079	0.9074 ± 0.0073
$K = 10$	0.9024 ± 0.0057	0.9012 ± 0.0064	0.9061 ± 0.0074	0.9166 ± 0.0070	0.9154 ± 0.0041
$K = 15$	0.9055 ± 0.0066	0.9037 ± 0.0049	0.9043 ± 0.0055	0.9092 ± 0.0024	0.9049 ± 0.0053
$K = 20$	0.9030 ± 0.0031	0.9055 ± 0.0053	0.9024 ± 0.0031	0.9049 ± 0.0068	0.9061 ± 0.0046

Tabla A.17: Resultados obtenidos para "Protein Homology" tras aplicar VoluSMOTE con $\alpha = 0.75$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha_i \sim U(0, 1)$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.9141 ± 0.0053	0.9135 ± 0.0039	0.9092 ± 0.0069	0.9129 ± 0.0059	0.9123 ± 0.0079
$K = 3$	0.9086 ± 0.0050	0.9129 ± 0.0023	0.9030 ± 0.0057	0.9111 ± 0.0035	0.9104 ± 0.0033
$K = 4$	0.9160 ± 0.0035	0.9141 ± 0.0056	0.9092 ± 0.0041	0.9086 ± 0.0050	0.9092 ± 0.0046
$K = 5$	0.9111 ± 0.0030	0.9086 ± 0.0050	0.9092 ± 0.0063	0.9160 ± 0.0049	0.9074 ± 0.0067
$K = 6$	0.9141 ± 0.0083	0.9067 ± 0.0040	0.9055 ± 0.0050	0.9098 ± 0.0045	0.9043 ± 0.0043
$K = 10$	0.9037 ± 0.0035	0.9111 ± 0.0030	0.9092 ± 0.0050	0.9061 ± 0.0069	0.9067 ± 0.0065
$K = 15$	0.9074 ± 0.0103	0.9141 ± 0.0079	0.9080 ± 0.0071	0.9074 ± 0.0055	0.9074 ± 0.0075
$K = 20$	0.9111 ± 0.0062	0.9117 ± 0.0071	0.9074 ± 0.0061	0.9111 ± 0.0076	0.9104 ± 0.0064

Tabla A.18: Resultados obtenidos para "Protein Homology" tras aplicar VoluSMOTE con $\alpha_i \sim U(0, 1)$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
Parzen	0.8444 ± 0.0031	0.8506 ± 0.0060	0.8512 ± 0.0035	0.8586 ± 0.0074	0.8685 ± 0.0069
SMOTE ($K = 2$)	0.8524 ± 0.0040	0.8512 ± 0.0030	0.8512 ± 0.0023	0.8481 ± 0.0045	0.8555 ± 0.0040
SMOTE ($K = 3$)	0.8537 ± 0.0041	0.8567 ± 0.0015	0.8580 ± 0.0019	0.8586 ± 0.0053	0.8629 ± 0.0088
SMOTE ($K = 4$)	0.8586 ± 0.0045	0.8561 ± 0.0050	0.8567 ± 0.0071	0.8549 ± 0.0047	0.8592 ± 0.0024
SMOTE ($K = 5$)	0.8549 ± 0.0085	0.8543 ± 0.0023	0.8524 ± 0.0049	0.8512 ± 0.0065	0.8574 ± 0.0059
SMOTE ($K = 6$)	0.8580 ± 0.0061	0.8592 ± 0.0050	0.8574 ± 0.0030	0.8561 ± 0.0050	0.8567 ± 0.0037
SMOTE ($K = 10$)	0.8604 ± 0.0056	0.8629 ± 0.0046	0.8592 ± 0.0031	0.8598 ± 0.0041	0.8561 ± 0.0046
SMOTE ($K = 15$)	0.8611 ± 0.0064	0.8623 ± 0.0063	0.8623 ± 0.0024	0.8580 ± 0.0051	0.8641 ± 0.0055
SMOTE ($K = 20$)	0.8660 ± 0.0041	0.8635 ± 0.0053	0.8635 ± 0.0053	0.8660 ± 0.0102	0.8641 ± 0.0080

Tabla A.19: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar Parzen y SMOTE. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8530 ± 0.0037	0.8543 ± 0.0071	0.8567 ± 0.0037	0.8555 ± 0.0049	0.8549 ± 0.0027
$K = 3$	0.8500 ± 0.0050	0.8604 ± 0.0035	0.8580 ± 0.0019	0.8574 ± 0.0045	0.8567 ± 0.0041
$K = 4$	0.8555 ± 0.0049	0.8524 ± 0.0059	0.8598 ± 0.0057	0.8611 ± 0.0047	0.8586 ± 0.0023
$K = 5$	0.8604 ± 0.0045	0.8604 ± 0.0053	0.8660 ± 0.0031	0.8604 ± 0.0023	0.8611 ± 0.0047
$K = 6$	0.8537 ± 0.0046	0.8592 ± 0.0037	0.8641 ± 0.0058	0.8617 ± 0.0040	0.8629 ± 0.0031
$K = 10$	0.8611 ± 0.0033	0.8598 ± 0.0050	0.8623 ± 0.0046	0.8679 ± 0.0071	0.8648 ± 0.0053
$K = 15$	0.8641 ± 0.0039	0.8691 ± 0.0031	0.8722 ± 0.0015	0.8709 ± 0.0012	0.8697 ± 0.0035
$K = 20$	0.8790 ± 0.0035	0.8740 ± 0.0023	0.8771 ± 0.0053	0.8814 ± 0.0037	0.8777 ± 0.0031

Tabla A.20: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar VoluSMOTE con $\alpha = 0$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.25$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8512 ± 0.0040	0.8555 ± 0.0049	0.8549 ± 0.0043	0.8543 ± 0.0030	0.8561 ± 0.0041
$K = 3$	0.8549 ± 0.0039	0.8604 ± 0.0040	0.8567 ± 0.0057	0.8549 ± 0.0070	0.8592 ± 0.0037
$K = 4$	0.8512 ± 0.0035	0.8580 ± 0.0051	0.8555 ± 0.0056	0.8555 ± 0.0053	0.8586 ± 0.0023
$K = 5$	0.8611 ± 0.0055	0.8586 ± 0.0035	0.8586 ± 0.0035	0.8660 ± 0.0031	0.8648 ± 0.0059
$K = 6$	0.8574 ± 0.0040	0.8598 ± 0.0024	0.8629 ± 0.0046	0.8623 ± 0.0063	0.8623 ± 0.0031
$K = 10$	0.8641 ± 0.0043	0.8623 ± 0.0041	0.8629 ± 0.0024	0.8666 ± 0.0030	0.8660 ± 0.0053
$K = 15$	0.8672 ± 0.0027	0.8722 ± 0.0037	0.8716 ± 0.0063	0.8740 ± 0.0074	0.8771 ± 0.0053
$K = 20$	0.8753 ± 0.0088	0.8765 ± 0.0033	0.8802 ± 0.0045	0.8796 ± 0.0051	0.8820 ± 0.0040

Tabla A.21: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar VoluSMOTE con $\alpha = 0.25$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.5$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8524 ± 0.0035	0.8518 ± 0.0039	0.8549 ± 0.0027	0.8493 ± 0.0040	0.8530 ± 0.0031
$K = 3$	0.8500 ± 0.0050	0.8518 ± 0.0039	0.8574 ± 0.0023	0.8561 ± 0.0041	0.8580 ± 0.0019
$K = 4$	0.8549 ± 0.0033	0.8549 ± 0.0061	0.8574 ± 0.0059	0.8580 ± 0.0039	0.8586 ± 0.0035
$K = 5$	0.8500 ± 0.0071	0.8567 ± 0.0046	0.8580 ± 0.0019	0.8592 ± 0.0063	0.8604 ± 0.0030
$K = 6$	0.8586 ± 0.0040	0.8580 ± 0.0043	0.8604 ± 0.0035	0.8611 ± 0.0033	0.8654 ± 0.0046
$K = 10$	0.8617 ± 0.0023	0.8672 ± 0.0043	0.8635 ± 0.0035	0.8685 ± 0.0024	0.8672 ± 0.0043
$K = 15$	0.8728 ± 0.0023	0.8703 ± 0.0027	0.8759 ± 0.0012	0.8728 ± 0.0040	0.8722 ± 0.0050
$K = 20$	0.8790 ± 0.0053	0.8759 ± 0.0053	0.8783 ± 0.0031	0.8783 ± 0.0031	0.8771 ± 0.0065

Tabla A.22: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar VoluSMOTE con $\alpha = 0.5$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha = 0.75$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8469 ± 0.0050	0.8512 ± 0.0030	0.8487 ± 0.0019	0.8524 ± 0.0030	0.8561 ± 0.0037
$K = 3$	0.8555 ± 0.0023	0.8537 ± 0.0063	0.8549 ± 0.0019	0.8561 ± 0.0063	0.8555 ± 0.0012
$K = 4$	0.8518 ± 0.0027	0.8561 ± 0.0050	0.8543 ± 0.0023	0.8530 ± 0.0031	0.8592 ± 0.0050
$K = 5$	0.8518 ± 0.0047	0.8524 ± 0.0035	0.8580 ± 0.0043	0.8580 ± 0.0070	0.8574 ± 0.0023
$K = 6$	0.8549 ± 0.0051	0.8567 ± 0.0031	0.8524 ± 0.0045	0.8567 ± 0.0060	0.8611 ± 0.0019
$K = 10$	0.8586 ± 0.0053	0.8604 ± 0.0049	0.8586 ± 0.0023	0.8611 ± 0.0055	0.8635 ± 0.0030
$K = 15$	0.8685 ± 0.0041	0.8703 ± 0.0047	0.8716 ± 0.0024	0.8777 ± 0.0041	0.8771 ± 0.0023
$K = 20$	0.8753 ± 0.0057	0.8820 ± 0.0035	0.8796 ± 0.0055	0.8790 ± 0.0049	0.8814 ± 0.0074

Tabla A.23: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar VoluSMOTE con $\alpha = 0.75$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

VoluSMOTE ($\alpha_i \sim U(0, 1)$)	$BR = 1$	$BR = 2$	$BR = 4$	$BR = 8$	$BR = 16$
$K = 2$	0.8500 ± 0.0050	0.8493 ± 0.0049	0.8567 ± 0.0046	0.8543 ± 0.0049	0.8524 ± 0.0023
$K = 3$	0.8555 ± 0.0059	0.8543 ± 0.0059	0.8555 ± 0.0053	0.8543 ± 0.0040	0.8592 ± 0.0024
$K = 4$	0.8543 ± 0.0059	0.8555 ± 0.0040	0.8555 ± 0.0035	0.8561 ± 0.0031	0.8586 ± 0.0059
$K = 5$	0.8574 ± 0.0053	0.8592 ± 0.0015	0.8567 ± 0.0041	0.8654 ± 0.0046	0.8623 ± 0.0066
$K = 6$	0.8555 ± 0.0045	0.8598 ± 0.0041	0.8598 ± 0.0041	0.8635 ± 0.0035	0.8592 ± 0.0041
$K = 10$	0.8654 ± 0.0037	0.8641 ± 0.0043	0.8672 ± 0.0043	0.8672 ± 0.0039	0.8641 ± 0.0064
$K = 15$	0.8728 ± 0.0045	0.8716 ± 0.0031	0.8777 ± 0.0053	0.8709 ± 0.0053	0.8771 ± 0.0035
$K = 20$	0.8728 ± 0.0065	0.8802 ± 0.0071	0.8808 ± 0.0074	0.8783 ± 0.0046	0.8765 ± 0.0039

Tabla A.24: Resultados obtenidos para "Protein Homology" (reducida) tras aplicar VoluSMOTE con $\alpha_i \sim U(0, 1)$. Se muestra la probabilidad de detección en el punto de trabajo (promediando 5 ejecuciones) en términos de media y desviación típica.

Bibliografía

[Abdi and Hashemi, 2015] Abdi, L., and Hashemi, S. (2015). To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Trans. Knowledge and Data Engineering*, 28:238−251.

[Abe, 2003] Abe, N. (2003). Invited talk: Sampling approaches to learning from imbalanced datasets: Active learning, cost sensitive learning and beyond. *Workshop Learning from Imbalanced Data Sets II*, in *Proc. 20th Intl. Conf. Machine Learning*, Washington, DC: IEEE Press.

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Proc. Intl. Conf. on Database Theory*, pages 420−434. Berlin (Germany): Springer.

[Ahachad et al., 2017] Ahachad, A., Álvarez-Pérez, L., and Figueiras-Vidal, A. R. (2017). Boosting ensembles with controlled emphasis intensity. *Pattern Recognition Letters*, 88:1−5.

[Alcalá-Fernández et al., 2011] Alcalá-Fernández, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Multiple-Valued Logic and Soft Computing*, 17:255−287.

[Ali-Gombe and Elyan, 2019] Ali-Gombe, A., and Elyan, E. (2019). MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing*, 361:212−221.

[Alvear-Sandoval and Figueiras-Vidal, 2018] Alvear-Sandoval, R., and Figueiras-Vidal, A. R. (2018). On building ensembles of stacked denoising auto-encoding classifiers and their further improvement. *Information Fusion*, 39:41−52.

[Anderson, 1984] Anderson, J. A. (1984). Regression and ordered categorical variables. *J. Royal Statistical Soc.: Series B (Methodological)*, 46:1−22.

[Anne et al., 2018] Anne, C., Mishra, A., Hoque, M. T., and Tu, S. (2018). Multiclass patent document classification. *Artificial Intelligence Research*, 7:1−14.

[Ballard, 1987] Ballard, D. H. (1987). Modular learning in neural networks. In *Proc. Nat. Conf. Artificial Intelligence*, pages 279−284. Seattle, WA: AAAI.

[Bahnsen et al., 2015a] Bahnsen, A. C., Aouada, D., and Ottersten, B. (2015a). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*, 42:6609−6619.

[Bahnsen et al., 2015b] Bahnsen, A. C., Aouada, D., and Ottersten, B. (2015b). A novel cost-sensitive framework for customer churn predictive modeling. *Decision Analytics*, 2:1−15.

[Barua et al., 2014] Barua, S., Islam, M. M., Yao, X., and Murase, K. (2014). MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans. Knowledge and Data Engineering*, 26:405−425.

[Basak et al., 2019] Basak, R., Sural, S., Ganguly, N., and Ghosh, S. K. (2019). Online public shaming on twitter: Detection, analysis, and mitigation. *IEEE Trans. Computational Social Systems*, 6:208−220.

[Batista et al., 2004] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6:20−29.

[Batuwita and Palade, 2009] Batuwita, R., and Palade, V. (2009). MicroPred: Effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, 25:989−995.

[Batuwita, and Palade, 2010] Batuwita, R., and Palade, V. (2010). FSVM-CIL: Fuzzy support vector machines for class imbalance learning. *IEEE Trans. Fuzzy Systems*, 18:558−571.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *J. Machine Learning Research*, 3:1137−1155.

[Bengio et al., 2007] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153−160. Vancouver (Canada): Curran Associates.

[Bengio, 2009] Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2:1–127.

[Benítez-Buenache et al., 2019] Benítez-Buenache, A., Álvarez-Pérez, L., Mathews, V. J., and Figueiras-Vidal, A. R. (2019). Likelihood ratio equivalence and imbalanced binary classification. *Expert Systems with Applications*, 130:84−96.

[Benítez-Buenache et al., 2020] Benítez-Buenache, A., Álvarez-Pérez, L., Mathews, V. J., and Figueiras-Vidal, A. R. (2020). Corrigendum to "Likelihood ratio equivalence and imbalanced binary classification" [Expert Systems with Applications, Volume 130 (2019), Pages 84–96]. *Expert Systems with Applications*, 146:113299.

[Benítez-Buenache et al., 2021] Benítez-Buenache, A., Álvarez-Pérez, L., and Figueiras-Vidal, A. R. (2021). On the design of Bayesian principled algorithms for imbalanced classification. Accepted for publication in *Knowledge-Based Systems*. Available at: https://doi.org/10.1016/j.knosys.2021.106969

[Blagus and Lusa, 2012] Blagus, R., and Lusa, L. (2012). Evaluation of SMOTE for high-dimensional class-imbalanced microarray data. In *Proc. 11th Intl. Conf. Machine Learning and Applications*, pages 89−94. Boca Raton, FL: IEEE Press.

[Bordes et al., 2005] Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *J. Machine Learning Research*, 6:1579−1619.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proc. 5th Annual Workshop on Computational Learning Theory*, pages 144−152. Pittsburgh, PA: ACM Press.

[Branco et al., 2016] Branco, P., Torgo, L., and Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys*, 49:1−50.

[Bregman, 1967] Bregman, L. M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Mathematics and Mathematical Physics*, 7:200−217.

[Breiman et al., 1984] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.

[Breiman, 1996] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123−140.

[Breiman, 1998] Breiman, L. (1998). Arcing classifier. *Annals of Statistics*, 26:801−849.

[Breiman, 1999a] Breiman, L. (1999a). Combining predictors. In Sharkey A. J. C., editor, *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, pages 31−50. London (UK): Springer.

[Breiman, 1999b] Breiman, L. (1999b). Prediction games and arcing algorithms. *Neural Computation*, 11:1493−1517.

[Breiman, 2000] Breiman, L. (2000). Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40:229−242.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45:5−32.

[Broomhead and Lowe, 1988] Broomhead, D. S., and Lowe, D. (1988). Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks. *Royal Signals and Radar Establishment, Memorandum No. 4148*, Malvern (UK).

[Bryson, 1961] Bryson, A. E. (1961). A gradient method for optimizing multi-stage allocation processes. In *Proc. Harvard Univ. Symposium on Digital Computers and Their Applications*, pages 125−135. Cambridge, MA: Harvard University Press.

[Bunkhumpornpat et al., 2009] Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2009). Safe-level-SMOTE: Safe-level-synthetic minority oversampling technique for handling the class imbalanced problem. In *Proc. Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining*, pages 475−482. Bangkok (Thailand): Springer.

[Bunkhumpornpat et al., 2012] Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2012). DBSMOTE: Density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36:664−684.

[Caruana and Joachims, 2004] Caruana, R., and Joachims, T. (2004). Protein Homology data-set. *KDD Cup 2004: Particle Physics; Plus Protein Homology Prediction*. Department of Computer Science, Cornell University.

[Castro and Braga, 2013] Castro, C. L., and Braga, A. P. (2013). Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE Trans. Neural Networks and Learning Systems*, 24:888−899.

[Chan and Stolfo, 1998] Chan, P. K., and Stolfo, S. J. (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In *Proc. 4th Intl. Conf. Knowledge Discovery and Data Mining*, pages 164−168. New York, NY: AAAI Press.

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *J. Artificial Intelligence Research*, 16:321−357.

[Chen et al., 2011] Chen, X., Fang, T., Huo, H., and Li, D. (2011). Graph-based feature selection for object-oriented classification in VHR airborne imagery. *IEEE Trans. Geoscience and Remote Sensing*, 49:353−365.

[Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv*, 1406.1078v3.

[Cid-Sueiro et al., 1999] Cid-Sueiro, J., Arribas, J. I., Urbán-Muñoz, S., and Figueiras-Vidal, A. R. (1999). Cost functions to estimate a posteriori probabilities in multiclass problems. *IEEE Trans. Neural Networks*, 10:645−656.

[Cid-Sueiro and Figueiras-Vidal, 2001] Cid-Sueiro, J., and Figueiras-Vidal, A. R. (2001). On the structure of strict sense Bayesian cost functions and its applications. *IEEE Trans. Neural Networks*, 12:445−455.

[Cortes and Vapnik, 1995] Cortes, C., and Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20:273−297.

[Cortez et al., 2009] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47:547−553.

[Cost and Salzberg, 1993] Cost, S., and Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57−78.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303−314.

[Dal Pozzolo et al., 2015] Dal Pozzolo, A., Caelen, O., and Bontempi, G. (2015). When is undersampling effective in unbalanced classification tasks?. In *Proc. Machine Learning and Knowledge Discovery in Databases*, pages 200−215. Porto (Portugal): Springer.

[De la Torre et al., 2015] De la Torre, M., Granger, E., Sabourin, R., and Gorodnichy, D. O. (2015). Adaptive skew-sensitive ensembles for face recognition in video surveillance. *Pattern Recognition*, 48:3385−3406.

[Devlin et al., 2018] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, 1810.04805v2.

[Domingos, 1999] Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. In *Proc. 5th ACM Intl. Conf. Knowledge Discovery and Data Mining*, pages 155−164. San Diego, CA: ACM Press.

[Dos Santos and Zadrozny, 2014] Dos Santos, C., and Zadrozny, B. (2014). Learning character-level representations for part-of-speech tagging. In *Proc. Intl. Conf. Machine Learning*, pages 1818−1826. Beijing (China): PMLR.

[Dua and Graff, 2019] Dua, D., and Graff, C. (2019). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science.

[Elkan, 2001] Elkan, C. (2001). The Foundations of cost-sensitive learning. In *Proc. 7th International Conference on Machine Learning*, pages 973−978. Stanford, CA: Morgan Kaufmann.

[Elman, 1990] Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179−211.

[Ertekin et al., 2007] Ertekin, S., Huang, J., Bottou, L., and Giles, C. L. (2007). Learning on the border: Active learning in imbalanced data classification. In *Proc. 16th ACM Conf. Information and Knowledge Management*, pages 127−136, Lisbon (Portugal): ACM Press.

[Estabrooks et al., 2004] Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20:18−36.

[Fan et al., 1999] Fan, W., Stolfo, S. J., Zhang, J., and Chan, P. K. (1999). AdaCost: Misclassification cost-sensitive boosting. In *Proc. 16th Intl. Conf. Machine Learning*, pages 97−105. Bled (Slovenia): Morgan Kaufmann.

[Fard et al., 2019] Fard, A. E., Mohammadi, M., Chen, Y., and Van de Walle, B. (2019). Computational rumor detection without non-rumor: A one-class classification approach. *IEEE Trans. Computational Social Systems*, 6:830−846.

[Fernández et al., 2013] Fernández, A., López, V., Galar, M., Del Jesús, M. J., and Herrera, F. (2013). Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97−110.

[Feurer and Hutter, 2013] Feurer, M., and Hutter, F. (2019). Hyperparameter optimization. In *Automated Machine Learning: Methods, Systems, Challenges*, pages 3−33. Cham (Germany): Springer.

[Freitas, 2011] Freitas, A. (2011). Building cost-sensitive decision trees for medical applications. *AI Communications*, 24:285−287.

[Freund and Schapire, 1995] Freund, Y., and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Proc. 2nd European Conf. Computational Learning Theory*, pages 23−37. Berlin (Germany): Springer.

[Freund and Schapire, 1996a] Freund, Y., and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. 13th Intl. Conf. Machine Learning*, pages 148−156. San Francisco, CA: Morgan Kaufmann.

[Freund and Schapire, 1996b] Freund, Y., and Schapire, R. E. (1996). Game theory, on-line prediction and boosting. In *Proc. 9th Annual Conf. Computational Learning Theory*, pages 325−332. Desenzano di Garda (Italy): ACM Press.

[Freund and Schapire, 1997] Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, 55:119−139.

[Freund and Schapire, 2012] Freund, Y., and Schapire, R. E. (2012). *Boosting: Foundations and Algorithms. Adaptive Computation and Machine Learning*. Cambridge, MA: MIT Press.

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193−202.

[Fung and Mangasarian, 2005] Fung, G. M., and Mangasarian, O. L. (2005). Multicategory proximal support vector machine classifiers. *Machine Learning*, 59:77−97.

[Galar et al., 2012] Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., and Herrera, F. (2012). A review on ensembles for the class imbalance problem:

Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Systems, Man, and Cybernetics, Pt. C*, 42:463−484.

[García-Laencina et al., 2013] García-Laencina, P. J., Sancho-Gómez, J. L., and Figueiras-Vidal, A. R. (2013). Classifying patterns with missing values using multi-task learning perceptrons. *Expert Systems with Applications*, 40:1333−1341.

[Glorot and Bengio, 2010] Glorot, X., and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proc. 13th Intl. Conf. Artificial Intelligence and Statistics*, pages 249−256. Sardinia (Italy): JMLR.

[Gómez-Verdejo et al.(2006)] Gómez-Verdejo, V., Ortega-Moral, M., Arenas-García, J., and Figueiras-Vidal, A. R. (2006). Boosting by weighting critical and erroneous samples. *Neurocomputing*, 69:679−685.

[Gómez-Verdejo et al., 2008] Gómez-Verdejo, V., Arenas-García, J., and Figueiras-Vidal, A. R. (2008). A dynamically adjusted mixed emphasis method for building boosting ensembles. *IEEE Trans. Neural Networks*, 19:3−17.

[González et al., 2013] González P., Álvarez, E., Barranquero, J., Díez, J., González-Quirós, R., Nogueira, E., López-Urrutia, Á., and del Coz, J. J. (2013). Multiclass support vector machines with example-dependent costs applied to plankton biomass estimation. *IEEE Trans. Neural Networks and Learning Systems*, 24:1901−1905.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672−2680. Montreal (Canada): Curran Associates.

[Gutiérrez et al., 2015] Gutiérrez, P. A., Pérez-Ortiz, M., Sánchez-Monedero, J., Fernández-Navarro, F., and Hervás-Martínez, C. (2015). Ordinal regression

methods: Survey and experimental study. *IEEE Trans. Knowledge and Data Engineering*, 28:127−146.

[Habibi et al., 2017] Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33:i37−i48.

[Haixiang et al., 2017] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220−239.

[Han et al., 2005] Han, H., Wang, W. Y., and Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proc. Intl. Conf. Intelligent Computing*, pages 878−887. Hefei (China): Springer.

[Hansen and Salamon, 1990] Hansen, L. K., and Salamon, P. (1990). Neural network ensembles. I*EEE Trans. Pattern Analysis and Machine Intelligence*, 12:993−1001.

[Harries et al., 2009] Harries, M., Gama, J., and Bifet, A. (2009). Electricity dataset. *OpenML Data Set Repository.*

[Hart, 1968] Hart, P. (1968). The condensed nearest neighbor rule. *IEEE Trans. Information Theory*, 14:515−516.

[He et al., 2008] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proc. Intl. Joint Conf. Neural Networks*, pages 1322−1328. Hong Kong (China): IEEE Press.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 770−778. Las Vegas, NV: IEEE Press.

[He and García, 2009] He, H., and García, E. A. (2009). Learning from imbalanced data. *IEEE Trans. Knowledge and Data Eng.*, 21:1263−1284.

[He and Ma, 2013] He, H., and Ma, Y. (2013). (Eds.), *Imbalanced Learning: Foundations, Algorithms, and Applications*, Hoboken, NJ: IEEE-Wiley.

[Hebb, 1949] Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New Tork, NY: Wiley.

[Herrera et al., 2016] Herrera, F., Charte, F., Rivera, A. J., and Del Jesús, M. J. (2016). *Multilabel Classification. Problem Analysis, Metrics and Techniques*. Cham (Germany): Springer.

[Hido et al., 2009] Hido, S., Kashima, H., and Takahashi, Y. (2009). Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining*, 2:412−426.

[Hinton and Sejnowski, 1986] Hinton, G. E., and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In Rumelhart, D. E., McClelland, J. L., and The PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations*, pages 282−317. Cambridge, MA: MIT Press.

[Hinton and Salakhutdinov, 2006] Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313:504−507.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20:832−844.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735−1780.

[Holmes et al., 2014] Holmes, G., Pfahringer, B., van Rijn, J., and Vanschoren, J. (2014). BNG (Page-Blocks) data-set. *OpenML Data Set Repository*.

[Hong et al., 2007] Hong, X., Chen, S., and Harris, C. J. (2007). A kernel-based two-class classifier for imbalanced data sets. *IEEE Trans. Neural Networks*, 18:28−41.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. National Academy of Sciences*, 79:2554−2558.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359−366.

[Hu et al., 2009] Hu, S., Liang, Y., Ma, L., and He, Y. (2009). MSMOTE: Improving classification performance when training data is imbalanced. In *Proc. 2nd Intl. Workshop Computer Science and Engineering*, pages 13−17. Quingdao (China): IEEE Press.

[Huang et al., 2011] Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I., and Tygar, J. D. (2011). Adversarial machine learning. In *Proc. 4th ACM Workshop on Security and Artificial Intelligencem*, pages 43−58. Chicago, IL: ACM Press.

[Husmeier, 2012] Husmeier, D. (2012). *Neural Networks for Conditional Probability Estimation: Forecasting Beyond Point Predictions*. London (UK): Springer.

[Imam et al., 2006] Imam, T., Ting, K., and Kamruzzaman, J. (2006). z-SVM: An SVM for improved classification of imbalanced data. In *Proc. 19th Australian Joint Conf. Artificial Intelligence*, pages 264−273, Hobart (Australia): Springer.

[Jacobs et al., 1991] Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3:79−87.

[Japkowicz, 2000] Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. In *Proc. AAAI Workshop in Learning from Imbalanced Data Sets*, pages 10−15. Austin, TX: AAAI Press.

[Japkowicz and Stephen, 2002] Japkowicz, N., and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429−449.

[Jo and Japkowicz, 2004] Jo, T., and Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter*, 6:40−49.

[Jordan, 1986] Jordan, M. I. (1986). *Serial order: A parallel distributed processing approach*. ICS Report 8604. Institute for Cognitive Science. La Jolla, CA.

[Jordan and Jacobs, 1994] Jordan, M. I., and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181−214.

[Jordan and Xu, 1995] Jordan, M. I., and Xu, L. (1995). Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks*, 8:1409−1431.

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 4401−4410. Long Beach, CA: IEEE Press.

[Kelley, 1960] Kelley, H. J. (1960). Gradient theory of optimal flight paths. *American Rocket Society*, 30:947−954.

[Khoshgoftaar et al., 2007] Khoshgoftaar, T. M., Seiffert, C., Van Hulse, J., Napolitano, A., and Folleco, A. (2007). Learning with limited minority class data. In *Proc. 6th Intl. Conf. Machine Learning and Applications*, pages 348−353. Cincinnati, OH: IEEE Press.

[Kingma and Ba, 2015] Kingma, D., and Ba, J. (2015). Adam: A method for stochastic optimization. *arXiv*, 1412.6980v9.

[Kowalczyk and Raskutti, 2002] Kowalczyk, A., and Raskutti, B. (2002). One class SVM for yeast regulation prediction. *SIGKDD Explorations Newsletter*, 4:99−100.

[Krawczyk, 2016] Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5:221−232.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097−1105. Lake Tahoe, NV: MIT Press.

[Kubat and Matwin, 1997] Kubat, M., and Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *Proc. 14th Intl. Conf. Machine Learning*, pages 179−186. Nashville, TN: Morgan Kaufmann.

[Kukar and Kononenko, 1998] Kukar, M., and Kononenko, I. (1998). Cost-Sensitive Learning with Neural Networks. In *Proc. European Conf. 13th Artificial Intelligence*, pages 445−449. Brighton (UK): Wiley.

[Kuncheva, 2004] Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley.

[Kwak, 2008] Kwak, N. (2008). Feature extraction for classification problems and its application to face recognition. *Pattern Recognition*, 41:1701−1717.

[Laurikkala, 2001] Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Proc. Conf. Artificial Intelligence in Medicine*, pages 63−66. Cascais (Portugal): Springer.

[Lázaro et al., 2018] Lázaro, M., Hayes, M. H., and Figueiras-Vidal, A. R. (2018). Training neural network classifiers through Bayes risk minimization applying unidimensional Parzen windows. *Pattern Recognition*, 77:204−215.

[Lázaro and Figueiras-Vidal, 2019] Lázaro, M., and Figueiras-Vidal, A. R. (2019). A Bayes risk minimization machine for example-dependent cost classification. Accepted for publication in *IEEE Trans. Cybernetics*. Available at: https://doi.org/10.1109/TCYB.2019.2913572

[LeCun, 1985] LeCun, Y. (1985). Une procedure d´ apprentissage pour réseau à seuil asymétrique. In *Proc. Cognitiva '85*, pages 599−604. Paris (France).

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541−551.

[LeCun and Bengio, 1995] LeCun, Y., and Bengio, Y. (1995). *Convolutional networks for images, speech, and time series*. Cambridge, MA: MIT Press.

[Lee, 1999] Lee, S. S. (1999). Regularization in skewed binary classification. *Computational Statistics*, 14:277−292.

[Lee, 2000] Lee, S. S. (2000). Noisy replication in skewed binary classification. *Computational Statistics and Data Analysis*, 34:165−191.

[Li et al., 2018] Li, J., Du, Q., Li, Y., and Li, W. (2018). Hyperspectral image classification with imbalanced data based on orthogonal complement subspace projection. *IEEE Trans. Geoscience and Remote Sensing*, 56:3838−3851.

[Liao, 2008] Liao, T. W. (2008). Classification of weld flaws with imbalanced class data. *Expert Systems with Applications*, 35:1041−1052.

[Ling and Li, 1998] Ling, C. X., and Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining*, pages 73−79. New York, NY: AAAI Press.

[Linnainmaa, 1970] Linnainmaa, S. (1970). *The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors*. Master's Thesis (in Finnish), Univ. Helsinki.

[Liu et al., 1999] Liu, B., Hsu, W., and Ma, Y. (1999). Mining association rules with multiple minimum supports. In *Proc. 5th ACM Intl. Conf. Knowledge Discovery and Data Mining*, pages 337−341. San Diego, CA: ACM Press.

[Liu and Yao, 1999a] Liu, Y., and Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, 12:1399−1404.

[Liu and Yao, 1999b] Liu, Y., and Yao, X. (1999). Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29:716−725.

[López et al., 2013] López, V., Fernández, A., García, S., Palade, V., and Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113−141.

[Manevitz and Yousef, 2001] Manevitz, L. M., and Yousef, M. (2001). One-class SVMs for document classification. *J. Machine Learning Research*, 2:139−154.

[Martens and Sutskever, 2011] Martens, J., and Sutskever, I. (2011). Learning recurrent neural networks with Hessian-free optimization. In *Proc. 28th Intl. Conf. Machine Learning*, pages 1033−1040. Bellevue, WA.

[Masnadi-Shirazi and Vasconcelos, 2010] Masnadi-Shirazi, H., and Vasconcelos, N. (2010). Risk minimization, probability elicitation, and cost-sensitive SVMs. In *Proc. 27th Intl. Conf. Machine Learning*, pages 759−766, Haifa (Israel).

[Masnadi-Shirazi and Vasconcelos, 2011] Masnadi-Shirazi, H., and Vasconcelos, N. (2011). Cost-sensitive boosting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33:294−309.

[Mazurowski et al., 2008] Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21:427−436.

[McCulloch and Pitts, 1943] McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115−133.

[Mediavilla-Relaño et al., 2020] Mediavilla-Relaño, J., Lázaro, M., and Figueiras-Vidal, A. R. (2020). Designing example-dependent cost principled classification algorithms. Submitted to *J. Machine Learning Research*.

[Mehrotra et al., 2016] Mehrotra, H., Singh, R., Vatsa, M., and Majhi, B. (2016). Incremental granular relevance vector machine: A case study in multimodal biometrics. *Pattern Recognition*, 56:63−76.

[Mena and González, 2009] Mena, L., and González, J. A. (2009). Symbolic one-class learning from imbalanced datasets: Application in medical diagnosis. *Intl. J. Artificial Intelligence Tools*, 18:273−309.

[Milletari et al., 2016] Milletari, F., Navab, N., and Ahmadi, S. A. (2016). V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. 4th Intl. Conf. 3D Vision*, pages 565−571. Stanford, CA: IEEE Press.

[Minsky and Papert, 1969] Minsky, M., and Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.

[Nahar et al., 2013] Nahar, J., Imam, T., Tickle, K. S., and Chen, Y. P. P. (2013). Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. *Expert Systems with Applications*, 40:96−104.

[Nallapati et al., 2017] Nallapati, R., Zhai, F., and Zhou, B. (2017). SummaRuNNer: A recurrent neural network based sequence model for extractive summarization

of documents. In *Proc. 31st AAAI Conference on Artificial Intelligence*, pages 3075−3081. San Francisco, CA: AAAI Press.

[Nami and Shajari, 2018] Nami, S., and Shajari, M. (2018). Cost-sensitive payment card fraud detection based on dynamic random forest and k-nearest neighbors. *Expert Systems with Applications*, 110:381−392.

[Ngai et al., 2009] Ngai, E. W., Xiu, L., and Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36:2592−2602.

[O'Mahony et al., 2019] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., and Walsh, J. (2019). Deep learning vs. traditional computer vision. In *Proc. Science and Information Conference. Advances in Computer Vision*, pages 128−144. Las Vegas, NV: Springer.

[Olteanu and Rynkiewicz, 2008] Olteanu, M., and Rynkiewicz, J. (2008). Estimating the number of components in a mixture of multilayer perceptrons. *Neurocomputing*, 71:1321−1329.

[Omari and Figueiras-Vidal, 2013] Omari, A., and Figueiras-Vidal, A. R. (2013). Feature combiners with gate-generated weights for classification. *IEEE Trans. Neural Networks and Learning Systems*, 24:158−163.

[Panigrahi et al., 2009] Panigrahi, S., Kundu, A., Sural, S., and Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. *Information Fusion*, 10:354−363.

[Park et al., 2013] Park, B. J., Oh, S. K., and Pedrycz, W. (2013). The design of polynomial function-based neural network predictors for detection of software defects. *Information Sciences*, 229:40−57.

[Parker, 1982] Parker, D. B. (1982). *Learning logic*. Technical Report 581-64, F1, Stanford Univ. Office of Technology Licesing, Stanford, CA.

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065−1076.

[Phua et al., 2004] Phua, C., Alahakoon, D., and Lee, V. (2004). Minority report in fraud detection: Classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6:50−59.

[Pratt, 1993] Pratt, L. Y. (1993). Discriminability-based transfer between neural networks. In *Advances in Neural Information Processing Systems*, pages 204−211. Denver, CO: Morgan Kaufmann.

[Provost and Fawcett, 2001] Provost, F., and Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, 42:203−231.

[Radivojac et al., 2004] Radivojac, P., Chawla, N. V., Dunker, A. K., and Obradovic, Z. (2004). Classification and knowledge discovery in protein databases. *J. Biomedical Informatics*, 37:224−239.

[Ramentol et al., 2012] Ramentol, E., Caballero, Y., Bello, R., and Herrera, F. (2012). SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. *Knowledge and Information Systems*, 33:245−265.

[Rao et al., 2006] Rao, R. B., Krishnan, S., and Niculescu, R. S. (2006). Data mining for improved cardiac care. *ACM SIGKDD Explorations Newsletter*, 8:3−10.

[Rao and Pais, 2019] Rao, R. S., and Pais, A. R. (2019). Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and Applications*, 31:3851−3873.

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 779−788. Las Vegas, NV: IEEE Press.

[Ríos-Insua et al., 2009] Ríos-Insua, D., Ríos, J., and Banks, D. (2009). Adversarial Risk Analysis. *J. American Statistical Association*, 104:841−854.

[Rokach, 2010] Rokach, L. (2010). *Pattern Classification Using Ensemble Methods*. Singapore: World Scientific.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Proc. Intl. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 234−241. Munich (Germany): Springer.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386−408.

[Rumelhart et al., 1986a] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Vol. I. Foundations*, pages 318−362. Cambridge, MA: MIT Press.

[Rumelhart et al., 1986b] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533−536.

[Rush et al., 2015] Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv*, 1509.00685v2.

[Salakhutdinov and Hinton, 2009] Salakhutdinov, R., and Hinton, G. (2009). Deep Boltzmann machines. In *Proc. 12th Intl. Conf. Artificial Intelligence and Statistics*, pages 448−455. Clearwater Beach, FL: JMLR.

[Samant and Agarwal, 2019] Samant, P., and Agarwal, R. (2019). Analysis of computational techniques for diabetes diagnosis using the combination of iris-

based features and physiological parameters. *Neural Computing and Applications*, 31:8441−8453.

[Sánchez-Morales, 2019] Sánchez-Morales, A., Sancho-Gómez, J. L., and Figueiras-Vidal, A. R. (2019). Exploiting label information to improve auto-encoding based classifiers. *Neurocomputing*, 370:104−108.

[Schapire, 1990] Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5:197−227.

[Schapire and Singer, 1999] Schapire, R. E., and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297−336.

[Schmidhuber, 2015] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85−117.

[Schuster and Paliwal, 1997] Schuster, M., and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45:2673−2681.

[Scott, 1992] Scott, D. W. (1992) *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York, NY: Wiley.

[Seiffert et al., 2014] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., and Folleco, A. (2014). An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Information Sciences*, 259:571−595.

[Settles, 2010] Settles, B. (2010). *Active Learning Literature Survey*, Tech. Report 1648, Computer Sci. Dept., Univ. Wisconsin-Madison.

[Shawe-Taylor and Cristianini, 2004] Shawe-Taylor, J., and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. New York, NY: Cambridge Univ. Press.

[Simonyan and Zisserman, 2014] Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv*, 1409.1556v6.

[Smith et al., 2017] Smith, S. L., Kindermans, P. J., Ying, C., and Le, Q. V. (2017). Don't decay the learning rate, increase the batch size. *arXiv*, 1711.00489v2.

[Smolensky, 1986] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition. Vol. I. Foundations*, pages 194−281. Cambridge, MA: MIT Press.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Machine Learning Research*, 15:1929−1958.

[Stallkamp et al., 2011] Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2011). The German traffic sign recognition benchmark: A multi-class classification competition. In *Proc. Intl. Joint Conf. Neural Networks*, pages 1453−1460. San Jose, CA: IEEE Press.

[Stefanowski and Wilk, 2008] Stefanowski, J., and Wilk, S. (2008). Selective preprocessing of imbalanced data for improving classification performance. In *Proc. Intl. Conf. Data Warehousing and Knowledge Discovery*, pages 283−292. Berlin (Germany): Springer.

[Stefanowski, 2016] Stefanowski, J. (2016). Dealing with data difficulty factors while learning from imbalanced data. In Matwin, S., and Mielniczuk, J., editors, *Challenges in Computational Statistics and Data Mining*, pages 333−363. Cham (Germany): Springer.

[Sun et al., 2007] Sun, Y., Kamel, M. S., Wong, A. K. C., and Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40:3358−3378.

[Sun et al., 2009] Sun, Y., Wong, A. K. C., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *Intl. J. Pattern Recognition and Artificial Intelligence*, 23:687−719.

[Tao et al., 2006] Tao, D., Tang, X., Li, X., and Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28:1088−1099.

[Tavallaee et al., 2010] Tavallaee, M., Stakhanova, N., and Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Trans. Systems, Man, and Cybernetics, Pt. C*, 40:516−524.

[Tieleman and Hinton, 2012] Tieleman, T., and Hinton, G. (2012). Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural Networks for Machine Learning*, 4:26−31.

[Ting, 2000] Ting, K. M. (2000). A comparative study of cost-sensitive boosting algorithms. In *Proc. 17th Intl. Conf. Machine Learning*, pages 983−990. Stanford, CA.

[Tomek, 1976] Tomek, I. (1976). Two modifications of CNN. *IEEE Trans. Systems, Man, and Cybernetics*, 6:769−772.

[Tong and Koller, 2001] Tong, S., and Koller, D. (2001). Support vector machine active learning with applications to text classification. *J. Machine Learning Resarch*, 2:45−66.

[Triguero et al., 2015] Triguero, I., del Río, S., López, V., Bacardit, J., Benítez, J. M., and Herrera, F. (2015). ROSEFW-RF: The winner algorithm for the ECBDL'14 big data competition: An extremely imbalanced big data bioinformatics problem. *Knowledge-Based Systems*, 87:69−79.

[Tsai et al., 2009] Tsai, C. H., Chang, L. C., and Chiang, H. C. (2009). Forecasting of ozone episode days by cost-sensitive neural network methods. *Science of the Total Environment*, 407:2124−2135.

[Valiant, 1984] Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, 27:1134−1142.

[Van Trees, 1968] Van Trees, H. L. (1968). *Detection, Estimation, and Modulation Theory, Part I*. New York, NY: Wiley.

[Vapnik, 1982] Vapnik, V. N. (1982). *Estimation of Dependences Based on Empirical Data*. Secaucus, NJ: Springer.

[Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York, NY: Springer.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. New York, NY: Wiley.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998−6008. Long Beach, CA.

[Ver Hoef and Cressie, 1993] Ver Hoef, J. M., and Cressie, N. (1993). Multivariable spatial prediction. *Mathematical Geology*, 25:219−240.

[Verbraken et al., 2014] Verbraken, T., Bravo, C., Weber, R., and Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European J. Operational Research*, 238:505−513.

[Veropoulos, 1999] Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proc. 20th Intl. Joint Conf. Artificial Intelligence*, pages 55−60, Stockholm (Sweden).

[Vincent et al., 2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Machine Learning Research*, 11:3371−3408.

[Viola and Jones, 2004] Viola, P., and Jones, M. J. (2004). Robust real-time face detection. *Intl. J. Computer Vision*, 57:137−154.

[Vuttipittayamongkol and Elyan, 2020] Vuttipittayamongkol, P., and Elyan, E. (2020). Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Information Sciences*, 509:47−70.

[Waibel et al., 1989] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 37:328−339.

[Wallace et al., 2011] Wallace, B. C., Small, K., Brodley, C. E., and Trikalinos, T. A. (2011). Class imbalance, redux. In *Proc. 11th Intl. Conf. Data Mining*, pages 754−763. Las Vegas, NV: IEEE Press.

[Wang and Yao, 2012] Wang, S., and Yao, X. (2012). Multiclass imbalance problems: Analysis and potential solutions. *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42:1119−1130.

[Werbos, 1974] Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph. D. Thesis, Harvard Univ., Cambridge, MA.

[Widrow and Hoff, 1960] Widrow, B., and Hoff, M. E. (1960). Adaptive switching circuits. In *1960 IRE WESCON Conv. Record*, pages 96−104. Los Angeles, CA.

[Wiener, 1949] Wiener, N. (1949). *Extrapolation, Interpolation and Smoothing of Stationary. Time Series, with Engineering Applications*. Cambridge, MA: MIT Press.

[Williams and Rasmussen, 2006] Williams, C. K., and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT press.

[Wilson, 1972] Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Systems, Man, and Cybernetics*, 2:408−421.

[Wolpert, 1992] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5:241−259.

[Woods et al., 1993] Woods, K., Doss, C., Bowyer, K., Solka, J., and Priebe, C. (1993). Mammography data-set. *OpenML Data Set Repository*.

[Wu and Chang, 2005] Wu, G., and Chang, E. (2005). KBA: Kernel boundary alignment considering imbalanced data distribution. *IEEE Trans. Knowledge and Data Engineering*, 17:786−795.

[Xu et al., 2019] Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*, pages 7335−7345. Vancouver (Canada): PMLR.

[Yang et al., 2009] Yang, C. Y., Yang, J. S., and Wang, J. J. (2009). Margin calibration in SVM class-imbalanced learning. *Neurocomputing*, 73:397−411.

[Yang et al., 2019] Yang, Z., Yu, W., Liang, P., Guo, H., Xia, L., Zhang, F., Ma, Y., and Ma, J. (2019). Deep transfer learning for military object recognition under small training set condition. *Neural Computing and Applications*, 31:6469−6478.

[Yu et al., 2013] Yu, H., Ni, J., and Zhao, J. (2013). ACOSampling: An ant colony optimization-based undersampling method for classifying imbalanced DNA microarray data. *Neurocomputing*, 101:309−318.

[Zadrozny et al., 2003] Zadrozny, B., Langford, J., and Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. In *Proc. Intl. Conf. Data Mining*, pages 435−442. Melbourne, FL: IEEE Comp. Soc.

[Zhang et al., 2018] Zhang, B. H., Lemoine, B., and Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proc. AAAI/ACM Conf. AI, Ethics, and Society*, pages 335−340. New Orleans, LA: ACM Press.

[Zhou, 2013] Zhou, L. (2013). Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems*, 41:16−25.