Programa de doctorado "Matemáticas"

PhD Dissertation

NEW MODELS AND METHODS FOR CLASSIFICATION AND FEATURE SELECTION. A MATHEMATICAL OPTIMIZATION PERSPECTIVE.

Author

Sandra Benítez Peña

Supervisors

Prof. Dr. *Rafael Blanquero Bravo*

Prof. Dr. *Emilio Carrizosa Priego*

Prof. Dr. *Pepa Ramírez Cobo*

A mis padres y mi hermana,
por todo su apoyo incondicional.

A todos aquellos que desde mi niñez
me inspiraron para llegar hasta aquí.

Agradecimientos

Y aquí llegué, por fin. Y vuelvo a mi yo de 10 años, que mira aquel tablón del cole en el que se anuncian unas olimpiadas de matemáticas, en las que sueña participar en unos años, así como crecer para poder conocer cada vez más y más de aquella que es su pasión, quien cierra los ojos y dice para sí: "*Lo logré. Sabía que podía.*".

Pero esto no es algo que haya hecho sola, no, ni mucho menos. Por eso, quiero expresar mi gratitud a todos los que habéis formado parte de esto. Quizás no vaya uno por uno, y quizás se me escape alguien por medio, pero intentaré expresarlo de la mejor manera que pueda.

En primer lugar a mis directores: Emilio, Rafa, Pepa. Así, en ese orden, os fui conociendo uno a uno. Y qué alegría hacerlo. Emilio: Gracias por ver en mí el potencial y las ganas, las suficientes como para acogerme en tu grupo de investigación, e intentar sacarle el máximo partido a lo que veías en mí. Gracias por todo el aprendizaje, desde aquellos días en que era estudiante de grado hasta ahora mismo, y todo lo que me queda por aprender de ti. Gracias por brindarme esta maravillosa oportunidad. Rafa: Gracias por, a pesar de no conocerme de nada, aceptar dirigirme aquel TFG junto con Emilio. Por, también desde ese entonces, aguantarme todos esos ratos echados en tu despacho (o por skype) intentando descifrar mis interminables scripts, y por lo muchísimo que me has enseñado de todo. Pepa: A ti te conocí la última, pero qué bien nos sentó conocerte a Reme y a mi, en todos los sentidos. Nos llenas de tranquilidad en aquellos momentos de nerviosismo, siempre con tu profesionalidad por delante, sentándote y trabajando mano a mano con nosotras, y echándonos horas y horas con tal de ayudarnos en todo lo posible y más. Gracias.

Although you are not my supervisors, I carry you both as if you were. Thank you so much Dolores and Peter. I met you practically at the beginning of my PhD studies, so I have grown professionally (and personally) also with you both. Thank you for giving me the opportunity of learning from you both, and I hope I can continue doing it for a long time. Also, thanks for made from Copenhagen my second home, and from the CBS like my home institution. Gracias, Loli, no solo desde el lado profesional, sino también por hacerme un hueco en tu vida y de la de tu familia, por permitirme disfrutar de Copenhague cual ciudadana nativa con bicicleta incluída. Por esas cenas,

cervezas, vinos y cafés, y por enseñarme tu maravilloso país de residencia, Dinamarca.

Por otro lado, estas estancias en Copenhague, el poder asistir a congresos conociendo además las ciudades en las que se celebraban (ay, bendita época sin COVID) así como esta misma tesis en sí, no sería posible sin los diferentes proyectos y sus IP: MTM2015-65915-R (Ministerio de Economía y Competitividad), PID2019-110886RB-I00 (Ministerio de Ciencia, Innovación y Universidades), FQM-329 y P18-FR-2369 (Junta de Andalucía), PR2019-029 (Universidad de Cádiz), Fundación BBVA y EC H2020 MSCA RISE NeEDS Project (Grant agreement ID: 822214). Y gracias, por supuesto a quienes en menor o mayor medida han gestionado todo esto desde su lado administrativo y me han ayudado en miles de cosas más: El IMUS y sus equipos de Administración, así como al Departamento de Estadística e Investigación Operativa de la Universidad de Sevilla.

Y como no podía faltar, ya que hablo del IMUS, una mención y agradecimiento a mis compañeros (ya, más que compañeros o amigos, mi familia IMUS). Algunos me habéis aguantado los 4 añitos al completo. Con otros he pasado menos tiempo, bien porque os fuísteis antes o bien porque hayáis llegado después, pero si algo es cierto, es que aún nos queda mucho por vivir juntos. Muchas gracias a todos, por vuestro lado profesional (sois unos máquinas), así como por vuestro lado personal. Es maravilloso teneros en mi vida, y saber que siempre os tendré. No quería hacer aquí menciones en concreto, pero creo que hay dos personas en particular que se lo merecen. Cristina: Sin ti y tu alegría, cantando y riendo a todas horas, estar trabajando en un despacho no será lo mismo. Reme, hemos sido prácticamente gemelas de tesis, juntas en todo momento, avanzando a la vez, y "sufriendo" y disfrutando casi al unísono cada uno de los momentos que hemos vivido. ¡Gracias, chicas!

Siguiendo con menciones particulares, de sobra sabíais que estaríais aquí: Caro, Cris, Andrés, Sara, Vero, Ro, Javi, Pablo. Gracias a todos, una necesita tener a amigos al lado en todos los momentos, buenos y malos, y vosotros habéis demostrado ser los mejores estando ahí siempre que lo he necesitado. Gracias por apoyarme, creer en mi, y querer sacarme una sonrisa.

Kevin, gracias por todo. Por escucharme hablar durante horas de mis cosas, a pesar de decir que no tienes idea de ellas (¡la tienes!) y por ayudarme siempre que has podido, de todas las formas habidas y por haber. Por aguantar interminables ensayos pre-charla y darme ánimos y consejos cuando yo ya no sabía de dónde sacarlos. Me ayudas a ser imparable.

Mamá, papá, gracias por hacer todo lo posible y más por hacerme llegar hasta aquí. Por todos los sacrificios que hicísteis para que, desde que era una cría, pudiese disfrutar de mis queridas matemáticas. Sin duda sois, junto con mis directores, los mayores responsables de esto. Raquel, tener a una casi ingeniera en casa ayuda. Gracias por entenderme y ayudarme en mis cosas, y no hacerme sentir la rarita de la casa, con tanta

ciencia, razonamientos y fórmulas siempre rodeándome.

A todos, muchísimas gracias. De una forma u otra me habéis ayudado a alcanzar esto, y sin vosotros no hubiera sido posible. Gracias por acompañarme en este viaje que solo acaba de comenzar.

Resumen

Esta tesis tiene como objetivo el desarrollo de nuevos modelos para Clasificación Supervisada y Benchmarking, haciendo uso de herramientas de Optimización Matemática y Estadística. Más concretamente, abordamos la fusión de técnicas de estas dos disciplinas, con el objetivo de extraer conocimiento de los datos. De esta forma, obtenemos métodos innovadores que mejoran a los ya existentes, unificando además las Matemáticas teóricas con problemas de la vida real.

El trabajo desarrollado en esta tesis se ha centrado en dos metodologias fundamentales en la Ciencia de los Datos: las máquinas de vectores soporte (SVM) y el Benchmarking. Con respecto a la primera, el classificador SVM se fundamenta en encontrar el hiperplano de máximo margen y se escribe como un problema cuadrático convexo. En el contexto de Benchmarking, el objetivo es calcular las diferentes eficiencias a través de un enfoque no paramétrico determinista. En esta tesis nos centraremos en el análisis envolvente de datos (DEA), consistente en un problema lineal.

Esta tesis está estructurada de la siguiente manera. En el Capítulo 1 se presentan de forma breve los retos considerados en esta tesis, junto con su estado del arte. De la misma forma, se exponen los diferentes modelos que se utilizarán como base, así como la notación usada en los siguientes capítulos.

En el Capítulo 2, se aborda el problema de la construcción de una versión del SVM que tiene en cuenta costes de error en la clasificación. Para ello, incorporamos nuevas restricciones de rendimiento de la classificación en el problema de optimización que define al SVM. En dichas restricciones imponemos cotas superiores sobre los errores de classificación. La formulación resultante consiste en un problema cuadrático convexo mixto con restricciones lineales.

El Capítulo 3 continúa con base en el SVM, planteando el problema de dar no solo un etiquetado fijo (clasificación binaria) a cada uno de los individuos que conforman la muestra, sino una probabilidad de pertenencia a la misma. Más aún, proporcionamos intervalos de confianza para los valores de score así como para las probabilidades de pertenencia a las diferentes clases. Además, al igual que en el capítulo anterior, llevaremos los resultados obtenidos al terreno en el que se tienen en cuenta los costes de error en la clasificación. Para este propósito, resolveremos bien un problema cuadrático

o un cuadrático convexo mixto con restricciones lineales, y siempre aprovechando la información obtenida en el ajuste de los parámetros del SVM, que generalmente es desaprovechada.

Basado en los resultados del Capítulo 2, en el Capítulo 4 tratamos el problema de la selección de atributos en SVM, teniendo en cuenta, como en los otros capítulos, los costes de error en la clasificación. Para esta técnica, integramos el modelado de la selección de atributos en la construcción del propio clasificador. El proceso se divide en dos etapas. En la primera de ellas, se hace la selección de atributos separando a su misma vez los datos mediante un clasificador lineal, teniendo en cuenta las restricciones de rendimiento. En la segunda, construimos el clasificador de máximo margen (SMV) usando los atributos seleccionados en el primer paso y considerando las mismas restricciones de antes.

En el Capítulo 5, nos movemos al problema del Benchmarking, donde comparamos las prácticas de diferentes entidades a través de sus productos o servicios, con la finalidad de realizar cambios o mejoras en cada una de ellas. En concreto, en este capítulo proponemos una formulación de programación lineal entero-mixta basada en el análisis envolvente de datos (DEA), con el propósito de hacer una selección de los inputs y outputs más relevantes, ayudando a la interpretabilidad y comprensión del modelo y las eficiencias obtenidas.

Finalmente, en el Capítulo 6 se recogen las conclusiones de esta tesis, así como futuras líneas de investigación.

Abstract

The objective of this PhD dissertation is the development of new models for Supervised Classification and Benchmarking, making use of Mathematical Optimization and Statistical tools. Particularly, we address the fusion of instruments from both disciplines, with the aim of extracting knowledge from data. In such a way, we obtain innovative methodologies that overcome to those existing ones, bridging theoretical Mathematics with real-life problems.

The developed works along this thesis have focused on two fundamental methodologies in Data Science: support vector machines (SVM) and Benchmarking. Regarding the first one, the SVM classifier is based on the search for the separating hyperplane of maximum margin and it is written as a quadratic convex problem. In the Benchmarking context, the goal is to calculate the different efficiencies through a non-parametric deterministic approach. In this thesis we will focus on Data Envelopment Analysis (DEA), which consists on a Linear Programming formulation.

This dissertation is structured as follows. In Chapter 1 we briefly present the different challenges this thesis faces on, as well as their state-of-the-art. In the same vein, the different formulations used as base models are exposed, together with the notation used along the chapters in this thesis.

In Chapter 2, we tackle the problem of the construction of a version of the SVM that considers misclassification errors. To do this, we incorporate new performance constraints in the SVM formulation, imposing upper bounds on the misclassification errors. The resulting formulation is a quadratic convex problem with linear constraints.

Chapter 3 continues with the SVM as the basis, and sets out the problem of providing not only a hard-labeling for each of the individuals belonging to the dataset, but a class probability estimation. Furthermore, confidence intervals for both the score values and the posterior class probabilities will be provided. In addition, as in the previous chapter, we will carry the obtained results to the field in which misclassified errors are considered. With such a purpose, we have to solve either a quadratic convex problem or a quadratic convex problem with linear constraints and integer variables, and always taking advantage of the parameter tuning of the SVM, that is usually wasted.

Based on the results in Chapter 2, in Chapter 4 we handle the problem of feature

selection, taking again into account the misclassification errors. In order to build this technique, the feature selection is embedded in the classifier model. Such a process is divided in two different steps. In the first step, feature selection is performed while at the same time data is separated via an hyperplane or linear classifier, considering the performance constraints. In the second step, we build the maximum margin classifier (SVM) using the selected features from the first step, and again taking into account the same performance constraints.

In Chapter 5, we move to the problem of Benchmarking, where the practices of different entities are compared through the products or services they provide. This is done with the aim of make some changes or improvements in each of them. Concretely, in this chapter we propose a Mixed Integer Linear Programming formulation based in Data Envelopment Analysis (DEA), with the aim of perform feature selection, improving the interpretability and comprehension of the obtained model and efficiencies.

Finally, in Chapter 6 we collect the conclusions of this thesis as well as future lines of research.

Contents

Chapter 1

Introduction

We are living in the era of Big Data, Data Science and Advanced Analytics. In Data Science, whose main purpose is to extract knowledge from datasets, some of the most important considered problems are Supervised Classification (Carrizosa and Romero Morales [2013]; Bertsimas and Shioda [2007]; Friedman et al. [2001]), including Class Probability Estimation (Kruppa et al. [2014]; Margineantu [2002]; Walker and Olshen [1992]), as well as Dimensionality Reduction (Huang et al. [2019]; Xu et al. [2019]; Sarveniazi [2014]) or Benchmarking (Bogetoft [2013], Moffett et al. [2008], Bogetoft and Otto [2010], Cook and Zhu [2006], Parsons [2002]). All previous topics are full of challenges, in particular from a Mathematical Optimization perspective, such as the construction of a mathematical model describing the different properties of the data, the study of relationships between different loss functions and the statistical performance of the model, the development of robust models, data scalability, among others as can be seen in Gambella et al. [2021]; Bartlett et al. [2006]; Ben-Tal et al. [2011]; Boţ and Lorenz [2011]; Bradley et al. [1999]; Carrizosa et al. [2016, 2017a]; Carrizosa and Romero Morales [2013]; Corne et al. [2012]; Marron et al. [2007]; Meisel and Mattfeld [2010]; Panagopoulos et al. [2016]; Bogetoft and Otto [2010]; Plastria and Carrizosa [2012]; Richtárik and Takáč [2016]; Sánchez et al. [2016]; Shen et al. [2003].

In general, in Data Science, we are given a set Ω_0 of individuals coming from a population Ω, each individual represented by a vector (x_i, y_i), where $x_i \in \mathbb{R}^N$ and $y_j \in \mathbb{R}^M$. Depending on the nature of the data set, some techniques of Data Science may be useful or not. Furthermore, the substantial meaning of x_i and y_i may vary. For instance, in Supervised (binary) Classification, $x_i \in \mathbb{R}^N$ is the attribute vector and $y_i \in \mathcal{C} = \{-1, +1\} \subset \mathbb{R}$ may be the membership of individual i. On the other hand, in Benchmarking, $x_i \in \mathbb{R}^N_+$ are the values of the input attribute vector and $y_i \in \mathbb{R}^M_+$ is the sampled output attribute vector.

The aim of this thesis is to design or improve existing state-of-the-art approaches in the commented fields. Therefore, in the following sections we present a detailed review of both SVM and Benchmarking, with special interest in feature selection problems.

1.1 Supervised Classification. Support Vector Machines

The usefulness of Classification lies in its wide variety of applicability, such as biology and medicine (Deo [2015]; Swan et al. [2013]; Tarca et al. [2007]; Kononenko [2001]), or even economics (Chaudhuri [2014]; Agarwal [2011]) and digital communication (Wei et al. [2019]; Chavda et al. [2018]). Some key examples of the use of classification in those and other fields are credit scoring (Baesens et al. [2003b,a]), in which the algorithm is used to detect good and bad customers; spam filtering (Zhang et al. [2004]), whose purpose is to distinguish between spam and desirable mails; or cancer screening (Kudva and Prasad [2018]; Guyon et al. [2002]; Mangasarian et al. [1995]), where we want to

differentiate between cancer and control patients.

For the task of classification, we need to build a classifier. Formally speaking, a classifier Ψ, i.e., a function $\Psi : \mathbb{R}^N \longrightarrow \mathcal{C}$, is sought to assign labels $c \in \mathcal{C}$ to incoming individuals for which the feature vector x is known but the label y is unknown and estimated through $\Psi(x)$. The different classification procedures differ in the way the classifier Ψ is obtained from the data set Ω_0. Nearest-neighbor methods represent one of the most well-known family of classifiers (Zhang [2020]; Biau and Devroye [2015]; Guo et al. [2003]). In its basic version, the k-nearest neighbors searches for the k objects in the so-called training sample $I \subseteq \Omega_0$ closest to each of the incoming instances, where closeness is measured via a similarity measure. Then, those new individuals will be labeled with the most frequent class in those k instances. Another well-known method to mention is the classification tree (Bertsimas and Dunn [2017]; Steinberg and Colla [2009]; Timofeev [2004]). This classifier is defined as a sequence of *if-then* statements, resulting in a hierarchy, encoded as a tree. Each element is sorted into a class by following the path of the tree, starting from the upmost statement. Another frequent approach consists of reducing the search of the classifier to the resolution of an optimization problem, namely, based on score functions. This is the case, among many others, of the state-of-the-art classifier for binary classification, known as Support Vector Machines (SVM), Carrizosa and Romero Morales [2013]; Cristianini and Shawe-Taylor [2000]; Vapnik [1995]; Vapnik and Vapnik [1998], on which we will focus throughout this thesis.

In its simplest version, SVM addresses two-class problems, i.e., $\mathcal{C}$ has two elements, say, $\mathcal{C} = \{-1, +1\}$. The SVM aims at separating both classes by means of a linear classifier, $\omega^\top x + \beta = 0$, where ω is the so called *score vector*. We will assume throughout this thesis that $\mathcal{C} = \{-1, +1\}$ and refer the reader to e.g. Allwein et al. [2001] for the reduction of multiclass problems to this case.
The linear SVM classifier is obtained by solving the following convex quadratic programming (QP) formulation with linear constraints:

$$\begin{array}{lll} \min\limits_{\omega,\beta,\xi} & \omega^\top\omega + C_+ \sum\limits_{i\in I:y_i=+1} \xi_i + C_- \sum\limits_{i\in I:y_i=-1} \xi_i & \\ \text{s.t.} & y_i(\omega^\top x_i + \beta) \geq 1 - \xi_i, & i \in I \\ & \xi_i \geq 0 & i \in I, \end{array} \qquad (\text{SVM}(C_+, C_-))$$

where I represents the set of training data, $\xi_i \geq 0$ are artificial variables which allow data points to be misclassified, and $C_+, C_- > 0$ are *regularization parameters* to be tuned that control the trade-off between margin minimization and misclassification

errors. The case $C_+ = C_-$ is frequently considered in the literature, but the use of different regularization parameters for the different classes may allow for a better control of misclassification costs or imbalancedness. See e.g. Lin et al. [2002].

Given an object i, it is classified in the positive or the negative class according to the sign of the so-called score function, $f(x) = sign(\omega^\top x_i + \beta)$, while for the case $\omega^\top x_i + \beta = 0$, the object is classified randomly.

A mapping into a high-dimensional feature space may be considered (Cortes and Vapnik [1995]), which allows us to transform this linear classification technique in a non-linear one using Mercer Theorem, Mercer [1909] and the so-called kernel trick, see e.g. Cristianini and Shawe-Taylor [2000]. In this way we can address problems with a very large number of features, such as those encountered in personalized medicine (Sánchez et al. 2016).

Hence, the general formulation of SVM is

$$\begin{aligned} \max_{\lambda} \quad & -\frac{1}{2}\sum_{j\in I}\sum_{k\in I}\lambda_j\lambda_k y_j y_k K(x_j,x_k) + \sum_{l\in I}\lambda_l \\ \text{s.t.} \quad & \sum_{i\in I}\lambda_i y_i = 0 \qquad\qquad (\text{GSVM}(C_+,C_-)) \\ & 0 \le \lambda_i \le C_+, \qquad i\in I: y_i = +1 \\ & 0 \le \lambda_i \le C_-, \qquad i\in I: y_i = -1, \end{aligned}$$

where $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a kernel function and λ are the usual variables of the dual formulation of the SVM.

As already mentioned, the goal in supervised classification is to classify objects in the correct class. However, ignoring imbalancedness (either in the classes size, either in the misclassification cost structure) may have dramatic consequences in the classification task, see Carrizosa et al. [2008]; He and Ma [2013]; Prati et al. [2015]; Maldonado et al. [2017]. For instance, for clinical databases, there are usually more observations of healthy populations than of the disease cases, and therefore smaller classification errors may be obtained for the first case. To illustrate this phenomenon, consider the well-known *Breast Cancer Wisconsin (Diagnostic) Data Set* from the UCI repository (Lichman 2013), for which the number of sick cases (212) is smaller than the size of control cases (357). Table 1.1 depicts the estimated rates (through a 10-fold cross validation approach) using a standard SVM. Even though both rates are high, it might be of interest to increase the accuracy of predicting cancer, perhaps at the expense of deteriorating the classification rates in the other class.

In addition to imbalancedness, another interesting problem arising in the context of the SVM is the extraction of probabilistic outputs. Given an object i with attribute vector x_i, then, as previously commented, the SVM produces a hard labeling in such a

	Mean	Std
`% benign instances well classified`	99%	1.7
`% malign instances well classified`	94.8%	4.9

Table 1.1: Performance of standard SVM with Radial Function Basis kernel for `wisconsin` dataset. Average values and standard deviations computed from 10 realizations.

way that i is classified in the positive or the negative class according to $sign(\omega^\top x_i + \beta)$. When an attribute vector x_0 is given, the value $f(x_0) = \omega^\top x_0 + \beta$ is called the *score value* of x_0, where $f(x)$ is the score function. However, the SVM method does not result in probabilistic outputs as posterior probabilities $P(y = +1 \mid x)$, which are of interest if a measure of confidence in the predictions is sought, see Murphy [2012]. This is of particular importance in several real-world applications such as the previously mentioned cancer screening or credit scoring, where the risk of a false-negative and a false-positive is significantly different.

1.2 Benchmarking and Performance Measurement

As already said, another challenging problem in Data Science is Benchmarking. Organisations need to know whether they are using the best practices to produce their products and services, and to do so they benchmark their performance with that of others. There are many documented examples of the use of benchmarking in the literature from both the private and the public sector such as airlines, banks, hospitals, universities, manufacturers, schools, and municipalities, see Bogetoft [2013] and references therein.

Benchmarking can be used both to make *intra* and *inter* organizational comparisons. The first one gains importance, for instance, when headquarters wants to promote cost efficiency in its different subunits. For the second one, a main example could be the one that involves a regulator seeking to induce cost-efficiency. Its use is also extended to make longitudinal, panel or dynamic comparisons.

We can define Benchmarking as the relative performance evaluation of entities (or firms) that converts the same types of inputs (resources) x into the same type of outputs (services) y. To calculate the relative efficiency of the firms, one can find in the literature numerous methods. Particularly, one can differentiate between parametric and non-parametric methods, as well as stochastic and non-stochastic methods. Two main approaches are Stochastic Frontier Analysis (SFA) (Coelli et al. [2005]), which is a parametric, stochastic approach, and the non-parametric, deterministic approach Data Envelopment Analysis (DEA) (Coelli et al. [2005]; Cooper et al. [2001]). Other methods that can be found in the literature are, for instance, Corrected Ordinary Least Squares (COLS) (Lovell et al. [1993]; Greene [2003, 1990]; Aigner and Chu [1968]) or

Stochastic Data Envelopment Analysis (SDEA) (Olesen and Petersen [1995]; Land et al. [1993]; Fethi et al. [2001]). In both SFA and DEA approaches, we can assume a firm (a given instance i) uses N inputs $x_i = (x_i^{(1)}, \ldots, x_i^{(N)}) \in \mathbb{R}_+^N$ to produce M outputs $y_i = (y_i^{(1)}, \ldots, y_i^{(M)}) \in \mathbb{R}_+^M$. The set of feasible input-output combinations for such firm is $T = \left\{(x, y) \in \mathbb{R}_+^N \times \mathbb{R}_+^M : x \text{ can produce } y\right\}$, which is denoted as the technology. SFA and DEA involve the different ways of building technologies T based on observations. Nevertheless, given any T, $(x, y) \in T$ is said to be efficient if and only if it cannot be dominated by some $(x, y) \in T$ and hence the efficient subset of T, T^E is $T^E = \{(x, y) \in T : (x, y) \text{ is efficient in } T\}$. In order to construct T via SFA, we make some a priori assumptions about the structure, providing a parametric functional based on the idea of estimating a stochastic cost or production frontier. On the other hand, DEA is based on Mathematical Programming. In what follows, we will focus on this particular approach for Benchmarking.

In DEA, the general setting involves a set F of firms that use N inputs to produce M outputs as specified before. In DEA, the estimate of T, T^*, is built following the so-called minimal extrapolation principle (Bogetoft [1996]; Bogetoft and Otto [2010]). Basic DEA models may differ in the assumptions they make about T, like free disposability or convexity. In what follows, we will assume a DEA model with constant returns to scale (CRS), so that the input-oriented efficiency of firm i, $E^{(i)}$ is equal to the optimal solution value of the following Linear Programming formulation,

$$E^{(i)} = \text{maximize}_{(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)} \sum_{m=1}^{M} \beta_i^{(m)} y_i^{(m)} \tag{1.1}$$

s.t. (DEA$^{(i)}$)

$$\sum_{m=1}^{M} \beta_i^{(m)} y_j^{(m)} - \sum_{n=1}^{N} \alpha_i^{(n)} x_j^{(n)} \leq 0 \qquad \forall j \in I \tag{1.2}$$

$$\sum_{n=1}^{N} \alpha_i^{(n)} x_i^{(n)} = 1 \tag{1.3}$$

$$\alpha_i \in \mathbb{R}_+^N \tag{1.4}$$

$$\beta_i \in \mathbb{R}_+^M, \tag{1.5}$$

where $\alpha_i^{(n)}$ is the weight for input n and $\beta_i^{(m)}$ the weight for output m. Problem (DEA$^{(i)}$) has $N + 1$ linear constraints and $N + M$ continuous variables, and thus can be solved efficiently even for large problem instances. In DEA literature, the evaluated entities are typically called Decision-Making Units (DMUs), so we will use equally the word DMU or firm.

1.3 Dimensionality reduction

The size of databases has suffered from a significant increase recently. This has resulted in huge databases with a high proportion of features that may turn out to be redundant, irrelevant or noisy. Dimensionality reduction has hence become a crucial procedure in Data Science tasks, since it identifies the relevant features, making thus the different procedures more interpretable, cheaper in terms of measurements and computational effort, and more effective since the noise is reduced. To perform dimensionality reduction, the original feature space is mapped onto a new space, with lower dimensionality. Such a mapping, can be implemented either by selecting a subset of the original feature space or building new features from the existing ones. There exist an assortment of techniques encompassed in what is known as Feature Selection, under the first choice. Inside this method, rather than using existing approaches like Genetic algorithms (Babatunde et al. [2014]; Tan et al. [2008]), Greedy selection/elimination (Caruana and Freitag [1994]), Lasso (Tibshirani [1996]), as well as filter or wrapper (Blum and Langley [1997]) methods, we will focus on embedded methods (Chandrashekar and Sahin [2014]; Saeys et al. [2007]), in which the Feature Selection procedure is integrated as part of the classifier. On the other hand, if we do not mind to reduce the size of the feature space at the expense of loose the original attributes, different methods can be applied such as Principal Component Analysis (Wold et al. [1987]; Abdi and Williams [2010]), Factor Analysis (Harman [1976]; Fodor [2002]) or Linear Discriminant Analysis (Balakrishnama and Ganapathiraju [1998]).

1.3.1 Feature selection for support vector machines

The SVM classifier uses all the features in the data, which may be rather problematic if measuring the features involve some non-negligible costs. This is particularly relevant when the dimension of the data set is large. It is then advisable to perform any type of dimensionality reduction, as it has been undertaken by Aytug [2015]; Bertolazzi et al. [2016]; Bradley et al. [1998]; Carrizosa et al. [2011]; Fung and Mangasarian [2004]; Guyon and Elisseeff [2003]; Le Thi et al. [2015]; Maldonado and Weber [2009]; Maldonado et al. [2011]; Weston et al. [2001]. In the previous works, the set of features is reduced so that an appropriate trade-off between classification accuracy and sparsity is obtained.

An amount of different FS procedures are found in the literature, some independent of the classification procedure (FS is performed in advance, based e.g. on the correlation between each feature and the label) and others embedded in the classification procedure, like the Holdout SVM (HOSVM), Maldonado and Weber [2009], Kernel-Penalized SVM (KP-SVM), Maldonado et al. [2011], or the methods presented in Chan et al. [2007], Maldonado et al. [2017] or Ghaddar and Naoum-Sawaya [2018]. Embedded

methods (Chandrashekar and Sahin [2014],Lal et al. [2006]) and other feature selection techniques main difference is the way feature selection and learning interact. In embedded methods, the learning and the feature selection parts cannot be separated, opposite to what happens for non-embedded procedures, which do not incorporate learning or use a learning machine to rank the quality of the different subsets of features without adding knowledge about the specific structure of the classifier.

1.3.2 Feature selection in Benchmarking

Another Data Science research field where Fature Selection becomes relevant is DEA, Cook et al. [2019]; Emrouznejad and Yang [2018]; Jiang and Lin [2015]; Landete et al. [2017]; Li et al. [2017b]; Petersen [2018]; Ruiz and Sirvent [2016, 2019]. DEA model specification, in the form of feature (where the term feature is used to refer to either outputs, inputs or environmental variables) selection, has a significant impact on the shape of the so-called efficient frontier as well as the insights given to the inefficient DMUs Golany and Roll [1989]. Moreover, feature selection is known to improve the discriminatory power of DEA models Bogetoft and Otto [2010].

In benchmarking projects, a good model should make conceptual sense not only from the theoretical but also from a practical point of view. The interpretation must be easy to understand and the properties of the model must be natural. We typically seek models that have significant features with the right signs and that do not leave a large unexplained variation. The complexity of this model specification phase partially explains the lack of enough guidance in the literature at this respect, Cook et al. [2014]; Luo et al. [2012]; Soleimani-Damaneh and Zarepisheh [2009], and most of the effort goes into the analysis and interpretation of a given DEA model. With the strand of literature on feature selection, the most common approach is to use a priori rules based on Statistical Analysis (such as correlations, dimensionality reduction techniques, and regression), and Information Theory (such as AIC or Shannon entropy). Alternatively, an ex-post analysis of the sensitivity of the efficient frontier to additional features can be run to detect whether relevant features have been left out. See Adler and Yazhemsky [2010]; Fernandez-Palacin et al. [2018]; Li et al. [2017a]; Nataraja and Johnson [2011]; Pastor et al. [2002]; Sirvent et al. [2005]; Soleimani-Damaneh and Zarepisheh [2009]; Wagner and Shimshak [2007], and references therein. Recently, there have been attempts to use LASSO techniques from Statistical Learning to build sparse benchmarking models, i.e., models using just a few features, J.-Y. Cai [2016]; Lee and Cai [2020]; Qin and Song [2014].

1.4 Contributions of this thesis

The goal of this thesis is to provide new methodological tools to improve aspects concerning classification via SVM, and Benchmarking. The considered framework is Mathematical Optimization that, as already commented, has proven very suitable in Data Science problems. In this section, we briefly describe the different problems we have dealt with.

In Chapter 2 we formulate a novel SVM model in which misclassification costs are considered by incorporating performance constraints in the problem formulation. This is of relevant interest in datasets that present imbalandecness, for example, in the class size. Specifically, our aim is to seek the hyperplane with maximal margin, yielding misclassification rates below given threshold values. Such maximal margin hyperplane is obtained by solving a quadratic convex problem with linear constraints and integer variables. This chapter is based on Benítez-Peña et al. [2019b] which also has a crucial role in the findings of Chapter 4. The reported numerical experience shows that our model gives the user control on the misclassification rates in one class (possibly at the expense of an increase in misclassification rates for the other class) and is feasible in terms of running times.

Chapter 3, which is based on Benítez-Peña et al. [2021], addresses the problem of obtaining class probabilities estimates for SVM. Such a classifier is based on a score procedure, yielding a deterministic classification rule, that can be transformed into a probabilistic rule, but does not provide this information in a natural way. Our probabilistic classification approach exploits the different results obtained during the process of the tunning of the regularization parameters, which is known to imply a high computational effort and generates pieces of information that is usually disregarded. The numerical results show a comparable or even better performance than benchmark approaches.

In Chapter 4, based on Benítez-Peña et al. [2019a], we propose a mathematical-optimization-based Feature Selection procedure embedded in Support Vector Machines, accommodating asymmetric misclassification costs by making use of the results from Chapter 2. The key idea in this work is to replace the traditional magin maximization by minimizing the number of features selected, but imposing upper bounds on the false positive and negative rates. The problem is written as an integer linear problem plus a quadratic convex problem. Our numerical results show how we can drastically reduce the number of variables in some benchmark data sets without losing too much performance in the classification task or even improving the results.

Chapter 5 is based on Benítez-Peña et al. [2020]. Here, we propose an integrative approach to feature (input and output) selection in Data Envelopment Analysis (DEA). DEA model is enriched with binary decision variables modelling the selection of fea-

tures, yielding a Mixed Integer Linear Programming formulation. Such a single-model approach can handle different objective functions as well as constraints to incorporate desirable properties from the real-world application. The numerical results highlight the advantages of our single-model approach provide to the user, in terms of making the choice of the number of features, as well as modeling their costs and their nature.

Finally, in Chapter 6, some final remarks conclusions as well as further research are presented.

Chapter 2

On support vector machines under a multiple-cost scenario

Many real-world classification problems, such as those found in medical diagnosis, churn or fraud prediction, involve misclassification costs which may be different in the different classes. In this chapter we propose a novel SVM model in which misclassification costs are considered by incorporating performance constraints in the problem formulation. Specifically, our aim is to seek the hyperplane with maximal margin yielding misclassification rates below given threshold values. Such maximal margin hyperplane is obtained by solving a quadratic convex problem with linear constraints and integer variables. The reported numerical experience shows that our model gives the user control on the misclassification rates in one class (possibly at the expense of an increase in misclassification rates for the other class) and is feasible in terms of running times.

2.1 Introduction

As mentioned in Section 1.1, the objective in supervised classification consists on classifying objects in the correct class. However, there may be imbalancedness related to the data, which can affect to the classification job. In order to cope with imbalancedness, either in class size or structure of misclassification costs, different methods have been suggested, see Bradford et al. [1998]; Freitas et al. [2007]; Carrizosa et al. [2008]; Datta and Das [2015]. Those methods are based on adding parameters or adapting the classifier construction, among others. For example, in Carrizosa et al. [2008] a biobjective problem of simultaneous minimization of misclassification rate, via the maximization of the margin, and measurement costs, is formulated.

In this chapter, a new formulation of the SVM is presented, in such a way that the focus is not only on the minimization of the overall misclassification rate but also on the performance of the classifier in the two classes (either jointly or separately). In order to do that, novel constraints are added to the SVM formulation. The keystone of the new model is its ability to achieve a deeper control over misclassification in contrast to previously existing models. The proposed methodology will be called Constrained Support Vector Machine (CSVM) and the resulting classification technique will be referred as CSVM classifier.

The remainder of this chapter is structured as follows. In Section 2.2, the CSVM is formulated as an optimization problem, and details concerning its feasibility are given. Section 2.3 aims to illustrate the performance of the new classifier. A description in depth about the experiments' design, real datasets to be tested as well as the obtained results will be given. The chapter ends with some concluding remarks in Section 2.4.

2.2 Constrained Support Vector Machines

In this section the Constrained Support Vector Machine (CSVM) model is formulated as a Mixed Integer Nonlinear Programming (MINLP) problem (Bonami et al. 2008; Burer and Letchford 2012), specifically in terms of a Mixed Integer Quadratic Programming (MIQP) problem.

This section is structured as follows. In Section 2.2.1 some theoretical foundations that motivate the novel constraints are given. Then, in Section 2.2.2 the formulation of the CSVM is presented. We will depart from the linear kernel case to later extend it to the general kernel case via the kernel trick. Finally, in Section 2.2.3, some issues about the CSVM formulation, as its feasibility, shall be discussed.

2.2.1 Theoretical Motivation

As commented before, the aim of the work in this chapter is to build a classifier so that the user may have control over the performance over the two classes. Specifically, given a set $\Omega_0 = \{(x_i, y_i)\}_i$ of data (a random sample of a vector (X, Y) with unknown distribution), the target is to obtain a classifier such that $p \geq p_0$, where p is the value of a performance measurement and p_0 is a threshold chosen by the user. The performance measure p is chosen by the user at her convenience and may be selected among the following rates: true positive rate (TPR) or sensitivity, true negative rate (TNR) or specificity and accuracy (ACC), which are defined as follows:

$$\begin{aligned} \text{TPR}: &\quad p = P(\omega^\top X + \beta > 0 | Y = +1) \\ \text{TNR}: &\quad p = P(\omega^\top X + \beta < 0 | Y = -1) \\ \text{ACC}: &\quad p = P(Y(\omega^\top X + \beta) > 0), \end{aligned} \tag{2.1}$$

see for example, Bewick et al. [2004].

In this work, for the sake of clarity, the positive class shall be identified with the class of interest to be controlled. For instance, in cancer screening studies, cancer is labelled as positive class whereas absence of cancer is labelled as negative. Also, in credit-scoring applications the positive class will be the defaulting clients. More examples will be discussed in Section 2.3.

If the random variable Z, defined as

$$Z = \begin{cases} 1, & \text{if an observation is well classified,} \\ 0, & \text{otherwise,} \end{cases}$$

is considered, then, the values of p as in (2.1) corresponding to the probability of correct

classification can be rewritten as

$$\begin{aligned} \text{TPR}: &\quad p = E[Z|Y=+1] \\ \text{TNR}: &\quad p = E[Z|Y=-1] \\ \text{ACC}: &\quad p = E[Z] \end{aligned}$$

and estimated from an independent and identically distributed (i.i.d.) sample $\{Z_i\}_{i\in S}$, by

$$\begin{aligned} \text{TPR}: &\quad \hat{p} = \bar{Z}_+ = \frac{\sum\limits_{i\in S_+} Z_i}{|S_+|} \\ \text{TNR}: &\quad \hat{p} = \bar{Z}_- = \frac{\sum\limits_{i\in S_-} Z_i}{|S_-|} \\ \text{ACC}: &\quad \hat{p} = \bar{Z} = \frac{\sum\limits_{i\in S} Z_i}{|S|}, \end{aligned}$$

where S_+ and S_- denote, respectively, the subsets $\{i \in S : y_i = +1\}$ and $\{i \in S : y_i = -1\}$, $S \subseteq \Omega_0$.

From a hypothesis testing viewpoint, our aim is to build a classifier such that, for a given sample, one can reject the null hypothesis in

$$\begin{cases} H_0: & p \leq p_0 \\ H_1: & p > p_0. \end{cases}$$

Under the classic decision rule, H_0 is rejected if $\hat{p} \geq p_0^*$ assuming that $\alpha = P(\text{type I error})$. From Hoeffding Inequality (Hoeffding 1963),

$$P(\hat{p} \geq p + c) \leq \exp(-2nc^2). \tag{2.2}$$

As $\alpha = P(\text{type I error}) = P(\hat{p} \geq p_0^* | p = p_0)$, substituting p by p_0 in (2.2) yields

$$P(\hat{p} < p_0 + c) \geq 1 - \exp(-2nc^2) = 1 - \alpha, \tag{2.3}$$

where $p_0 + c = p_0^*$. Therefore, we can take

$$p_0^* = p_0 + \sqrt{\frac{\log \alpha}{-2n}}. \tag{2.4}$$

Note that n equals $|S_+|$, $|S_-|$ or $|S|$, respectively, when considering the TPR, the TNR or the accuracy.

Here, the selection of the Hoeffding Inequality is motivated by its distribution-free character, but other options as the Binomial-Normal approximation could have been

chosen instead.

2.2.2 CSVM formulation

In this section, the CSVM formulation is presented. As it will be seen, the formulation includes novel performance constraints, which make the optimization problem a MIQP problem in terms of some integer variables.

We assume to be given a dataset Ω_0 with known labels. From such set we identify the training set $I \subseteq \Omega_0$, used to build the classifier, and the anchor set $J \subseteq \Omega_0$, used to impose a lower bound on the classifier performance. These sets will be considered disjoint.

With the purpose of building the CSVM, the performance constraints will be formulated in terms of binary variables $\{z_j\}_{j\in J}$, which are realizations of the variable Z in Section 2.2.1 and defined as:

$$z_j = \begin{cases} 1, & \text{if instance } j \text{ is counted as well classified} \\ 0, & \text{otherwise.} \end{cases}$$

In order to formulate the CSVM, novel constraints are added to the standard soft-margin SVM formulation as follows:

$$\begin{aligned}
\min_{\omega,\beta,\xi,z} \quad & \omega^\top\omega + C_+ \sum_{i\in I:y_i=+1} \xi_i + C_- \sum_{i\in I:y_i=-1} \xi_i & & & \\
\text{s.t.} \quad & y_i(\omega^\top x_i + \beta) \geq 1 - \xi_i, & i \in I & (2.5) & \\
& \xi_i \geq 0 & i \in I & (2.6) & \\
& y_j(\omega^\top x_j + \beta) \geq 1 - M_1(1 - z_j), & j \in J & (2.7) & (\text{CSVM}_0) \\
& z_j \in \{0,1\} & j \in J & (2.8) & \\
& \hat{p}_\ell \geq p^*_{0\ell} & \ell \in L. & (2.9) &
\end{aligned}$$

In the previous optimization problem, (2.5) and (2.6) are the usual constraints in the SVM formulation. Constraints (2.7) ensure that observations $j \in J$ with $z_j = 1$ will be correctly classified, without imposing any restriction when $z_j = 0$, provided that M_1 is big enough. A collection of requirements on the performance of the classifier over J can be specified by means of (2.9). Also, L is the set of indexes of the constraints that has the form of (2.9). These constraints can be modeled via the binary variables z_j,

for instance:

$$\begin{aligned}\text{TPR}: &\quad \sum_{j\in J_+} z_j \geq p_0^*|J_+|\\ \text{TNR}: &\quad \sum_{j\in J_-} z_j \geq p_0^*|J_-|\\ \text{ACC}: &\quad \sum_{j\in J} z_j \geq p_0^*|J|,\end{aligned}$$

where J_+ and J_- denote, respectively, the subsets $\{i \in J : y_i = +1\}$ and $\{i \in J : y_i = -1\}$.

As before, by considering the (partial) dual problem of (CSVM_0) and the *kernel trick*, the general formulation of the CSVM is obtained as follows (the intermediate steps can be found in Appendix A):

$$\begin{aligned}\min_{\lambda,\mu,\beta,\xi,z} \quad & \sum_{s\in I}\sum_{s'\in I}\lambda_s y_s \lambda_{s'} y_{s'} K(x_s,x_{s'}) + \sum_{t\in J}\sum_{t'\in J}\mu_t y_t \mu_{t'} y_{t'} K(x_t,x_{t'}) \\ & +2\sum_{s\in I}\sum_{t\in J}\lambda_s y_s \mu_t y_t K(x_s,x_t) + C_+ \sum_{i\in I:y_i=+1}\xi_i + C_- \sum_{i\in I:y_i=-1}\xi_i \\ \text{s.t.} \quad & z_j \in \{0,1\} && j\in J \\ & \hat{p}_\ell \geq p_{0\ell}^* && \ell \in L \\ & y_i\left(\sum_{s\in I}\lambda_s y_s K(x_s,x_i) + \sum_{t\in J}\mu_t y_t K(x_t,x_i) + \beta\right) \geq 1-\xi_i && i\in I \\ & y_j\left(\sum_{s\in I}\lambda_s y_s K(x_s,x_j) + \sum_{t\in J}\mu_t y_t K(x_t,x_j) + \beta\right) \geq 1 - M_1(1-z_j) && j\in J \qquad \text{(CSVM)} \\ & \xi_i \geq 0 && i\in I \\ & \sum_{i\in I}\lambda_i y_i + \sum_{j\in J}\mu_j y_j = 0 \\ & 0 \leq \lambda_i \leq C_+/2 && i\in I\text{: } y_i = +1 \\ & 0 \leq \lambda_i \leq C_-/2 && i\in I\text{: } y_i = -1 \\ & 0 \leq \mu_j \leq M_2 z_j && j\in J.\end{aligned}$$

Here $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ is again a kernel function, M_1 and M_2 are big enough numbers, and (λ, μ) are the usual variables in the dual formulation of the SVM.

2.2.3 Solving the CSVM

In this section we give details about the complexity of our problem as formulated in (CSVM). The problem belongs to the class of MIQP problems, and thus it can be addressed by standard mixed integer quadratic optimization solvers. In particular, the solver Gurobi (Gurobi Optimization 2016) and its Python language interface (Van Rossum and Drake 2011) have been used in our numerical experiments. In contrast to the standard SVM formulation, which is a continuous quadratic problem, the

CSVM is harder to solve due to the presence of binary variables. Hence, the optimal solution may not be found in a short period of time; however, as discussed in our numerical experience, good results are obtained when the problems are solved heuristically by imposing a short time limit to the solver.

Performance constraints (2.9) may define an infeasible problem since the values of the $p^*_{0\ell}$ may be unattainable in practice. Hence, the study of the feasibility of Problem (CSVM) is an important issue. As an example, consider data composed by two different classes, each one represented respectively by black and white dots in the top picture in Figure 2.1. If the optimization problem for the linear kernel SVM is solved, the resulting classifier is a hyperplane that aims at separating both classes and maximizes the margin. An approximate representation of the data and the classifier is shown in the middle panel in Figure 2.1. If the aim is to correctly classify all the data corresponding to a given class, it is intuitively easy to see that this objective can be reached by moving the SVM hyperplane. In fact, it can be seen in the bottom picture in Figure 2.1 how hyperplanes 1 and 2 classify correctly all white points, and hyperplane 3 classifies all the black dots in the correct class. Among all those hyperplanes, the SVM selects the one which maximizes the margin. So, intuitively, it is evident that if just one constraint of performance is imposed in only one of the classes, the problem is always feasible. However, and using the data in Figure 2.1 again, as well as the linear kernel SVM, it is clear that it is impossible to classify correctly all the instances at the same time; thus, the problem is then infeasible. However, there exist results, as Theorem 5 in Burges [1998], that show that the class of Mercer kernels for which $K(x, x') \to 0$ as $\|x - x'\| \to \infty$, and for which $K(x, x)$ is $O(1)$, builds classifiers that get a total correct classification in all the classes in the training sample, without regard how arbitrarily the data have been chosen. Thus, if a kernel satisfies the previous conditions, then feasibility is guaranteed. In particular, Radial Function Basis (RBF) kernel meets these conditions. Therefore, to be on the safe side, if the performance thresholds imposed are not too low, they should refer only to one class misclassification rates (so that we can shift the variable β to make the problem feasible) or to use a kernel, such as the RBF, known to have large VC dimension (Burges 1998; Cristianini and Shawe-Taylor 2000), defined as the maximal training sample size for which perfect separation can always be enforced.

2.3 Computational results

This section illustrates the performance of the novel method, the CSVM, in comparison with benchmark approaches. To do that, an assortment of datasets with different properties concerning size and imbalanceness shall be analyzed. Section 2.3.1 describes the experiments to be carried out, while Section 2.3.2 details the choice of parameters.

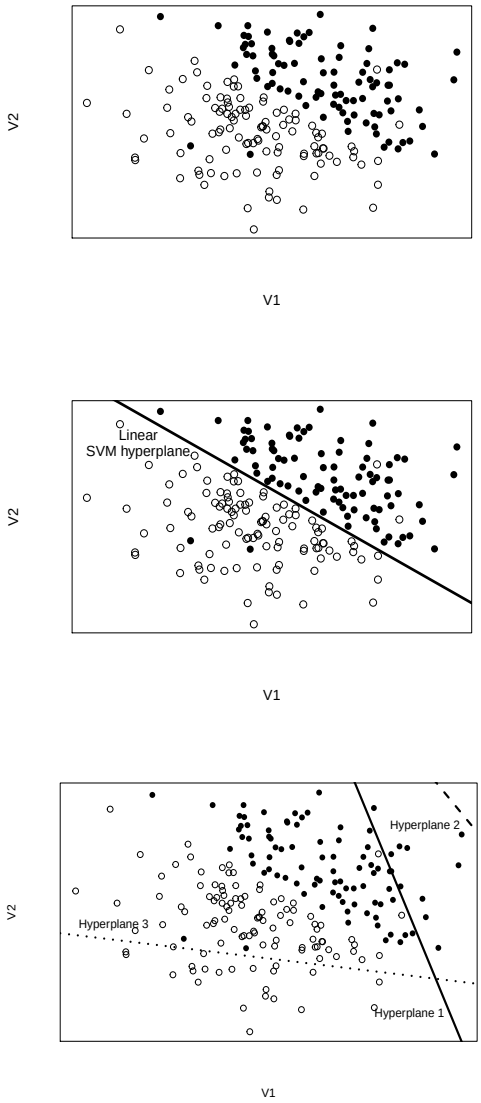


Figure 2.1: Study of feasiblity and unfeasibility of the CSVM.

Section 2.3.3 is devoted to clarify different aspects of the cross-validation procedure for estimating the performance of the approach, and Section 2.3.4 presents the datasets to be analyzed. Finally, Section 2.3.5 contains the obtained results and a deep discussion about them.

2.3.1 Description of the experiments

The objective of this chapter, as has been stated before, is to build a classifier whose performance can be controlled by means of some constraints, as in Problem (CSVM). As explained in Section 2.2.1, if we want a performance measurement p to be greater than a value p_0 with a specified confidence $100(1-\alpha)\%$, we should use an estimator of p, $\hat{p}$, and impose it to be greater than $p_0^* = p_0 + \sqrt{\frac{\log \alpha}{-2n}}$, according to (2.4).
Experiments whose aim will be to increase the performance rate of interest in one class will be performed. However, as it will be shown, a damage may be produced in the other class. In particular, since the interest is to improve the classification in the positive class, the TPR will be the rate to be included in the novel constraints. Assume that an estimator of the TPR, TPR_0 is given. The aim will be to impose $\mathrm{TPR} \geq \mathrm{TPR}_0 + \delta$, were $\delta = 0.025$, although other values can also be tested. Therefore, our experiments will consist of:

$$\text{Impose } \mathrm{TPR} \geq \min\{1, \mathrm{TPR}_0 + 0.025\} = p_0,$$

which implies that, for $\alpha = 0.05$, the performance constraints in the optimization problem defining the novel CSVM are:

$$\widehat{\mathrm{TPR}} \geq \min\left\{1, \mathrm{TPR}_0 + \sqrt{\frac{\log 0.05}{-2n}} + 0.025\right\} = p_0^*.$$

The novel CSVM will be compared with benchmark approaches. The first method to be compared with is the classic SVM where two different values of C (C_+ and C_-) are used for each class. This approach shall be noted as SVM(C_+, C_-) (see Section 1.1). The second benchmark method consists of moving the original hyperplane resulting from performing the standard SVM until the value p_0^* obtained by Hoeffding Inequality is achieved. This approach will be called from now on *Sliding β strategy*.

2.3.2 Parameters Setting

One of the most popular kernels $K(x, x')$ in literature is the well-known RBF kernel (Cristianini and Shawe-Taylor 2000; Hastie et al. 2001; Hsu et al. 2003; Smola and Schölkopf 2004; Horn et al. 2016), given by

$$K(x, x') = \exp\left(-\gamma \|x - x'\|^2\right),$$

where $\gamma > 0$ is a parameter to be tuned. This will be the kernel chosen for implementing the CSVM, although the method is valid for an arbitrary kernel.

The time limit for the solver was set equal to 300 seconds. In addition, the M_1 and M_2 values in Problem (CSVM) were set both equal to 100. The choice of these values is motivated as follows. First, for the sake of computational tractability, the time limit

should not be too high, but high enough so that the optimizer is able to solve the problem or at least to provide good feasible solutions. In our experiments, the choice of a time limit equal to 300s gave a good balance between the computational cost and the quality of the solutions. In the case of the values of M_1 and M_2, if small values are chosen, there may be many discarded hyperplanes, including the optimal one. However, if M_1 and M_2 are too big, it might cause computational difficulties (Camm et al. 1990) because of numerical instabilities and large gaps in the continuous relaxation, making the branch and bound too slow. A compromise solution is obtained by considering $M_1 = M_2 = 100$ in our problems. Setting M_1 and M_2 equal to 100, not a huge number, may indeed exclude the optimal solution of the original problem. This is not a big issue, since the original problem is nothing but a surrogate of our real aim, namely, classifying correctly forthcoming individuals. On the other hand, this constraint may also be seen as a regularization constraint, since it forces the variables involved to take relatively small values, as already happens with the variables λ_i, already force to be below $C/2$. In other words, though at the expense of excluding the optimal solution of the proxy optimization problem, setting a not too large value for M_1 and M_2 can be seen as an extra regularization, thus preventing overfit.

Note that an alternative formulation, avoiding big M constraints is obtained by using the Specially Ordered Sets of Type 1 (SOS1) (Bertsimas and Weismantel [2005], see also Silva [2017] and Bertsimas et al. [2016] for some examples of SOS1). However, we prefer to maintain the big M_1 and M_2 for two reasons. On the one hand, the use of SOS1 would involve quadratic constraints, which would make the problem even more difficult to solve. For example, constraint $0 \leq \mu_t \leq M_2 z_t$ would become $(\mu_t, 1-z_t)$:SOS1 and $0 \leq \mu_t$. This is equivalent to $\mu_t(1 - z_t) = 0$ and $0 \leq \mu_t$, which includes, as we can see, a non-convex quadratic constraint. On the other hand, not every solver has implemented the SOS1 method or is capable to solve quadratic mixed integer problems with non-convex quadratic constraints. In addition, even if it can manage SOS1-type constraints, it might perform the conversion to the problem with a big M automatically, and thus we would be again with big M constraints, now controlled by the solver and not by ourselves.

2.3.3 Performance estimation

The estimation of the performance of the novel CSVM is based on a K-fold cross validation (CV) as follows, see Kohavi et al. [1995]. Generally, K=10, but for those datasets with more than 1000 samples, $K = 5$ so that the running times are lower. Note that, apart from tuning γ, the regularization parameters C_+ and C_- introduced in Section 2.1 also need to be tuned. In order to make the CSVM procedure quicker, our experiments are based on choosing $C_+ = C/|I_+|$ and $C_- = C/|I_-|$, so only one parameter C shall be tuned for the CSVM, but not for the SVM(C_+, C_-), in which

both C_+ and C_- are tuned independently. As it will be seen later, this is not a crucial issue. Hence, for a given pair of parameters (C, γ), the process consists mainly on solving a standard SVM using all the instances $(I \cup J)$, and collect the values of λ (from the dual formulation of the SVM) as well as the value of β. Once the SVM is solved, and with the purpose of providing an initial solution for the CSVM, the value of β is slightly changed (maintaining the values of λ's fixed) until the desired number of instances well classified is reached. Then, the values of β and λ's obtained are set as initial solutions for CSVM. In addition, depending on whether each instance in J is well classified or not, we set their values of z as 0 or 1 as initial values for the CSVM.

We should make the selection of the best pair (C, γ) in each of the previous folds. In order to do that, a 10-fold CV (5-fold CV for datasets with more of 1000 samples, in order to reduce the running times) as before is made for each pair in a grid given by the 121 different combinations of $C = 2^{(-5:5)}$ and $\gamma = 2^{(-5:5)}$ ($C_+ = 2^{(-5:5)}$, $C_- = 2^{(-5:5)}$ and $\gamma = 2^{(-5:5)}$ for GSVM(C_+, C_-)). The general criterion used to select the best pair of parameters is the accuracy. However, in cases where the datasets are severely imbalanced in the classes size (when one of the classes has a weight less than a 30% of the total size), the G-mean (Tang et al. 2009), which is defined as $\sqrt{TPR \times TNR}$, is used to perform the parameter tuning instead. Finally, the average values of TPR and TNR obtained in the first CV, in addition to their standard deviations, are calculated.

For a better understanding, the previous algorithm is summarized in Algorithm 3.

Finally, we want to clarify that for our experiments we have selected I as the first half of $I \cup J$ and J as the second one.

2.3.4 Data description

The performance, in terms of correct classification probabilities and accuracy, is illustrated using 6 real-life datasets from the UCI and Keel repositories (Lichman 2013 and Alcalá-Fdez et al. 2009). In particular, the datasets are `australian` (Statlog (Australian Credit Approval) Data Set), `votes` (Congressional Voting Records Data Set), `wisconsin` (Breast Cancer Wisconsin (Diagnostic) Data Set), `german` (Statlog (German Credit Data) Data Set), `pageBlocks` (Page Blocks Classification (Imbalanced: 0) data set) and `biodeg` (QSAR biodegradation Data Set).

Details concerning the distribution of the classes in the considered datasets are provided by Table 4.1.

The first two columns give the name and number of attributes for each set. The values $|\Omega_0|$, $|\Omega_{0V}|$ and $|\Omega_{0+}|$ represent, respectively, the size for each dataset, the size of the validation sample in each step of the 10(or 5)-fold CV, and the number of positive instances in Ω_0. Finally, the percentage of positive instances is compiled in the last column.

Algorithm 1: Pseudocode for CSVM

Split data (D) into "*folds*" subsets, $D = \{D_1, \ldots, D_{folds}\}$.
for *kf = 1,...,folds* **do**
  Set $Validation = D_{kf}$ and $I \cup J = D - \{D_{kf}\}$.
  for *each pair* (C, γ) *in grid* $(\{2^{(-5:5)}\}, \{2^{(-5:5)}\})$ **do**
    Split $D - \{D_{kf}\} = D^*$ into "*folds2*" subsets, $D^* = \{D_1^*, \ldots, D_{folds2}^*\}$.
    for *kf2 = 1,..., folds2* **do**
      Set $Validation^* = D_{kf2}^*$ and $I^* \cup J^* = D^* - \{D_{kf2}\}$.
      Run standard SVM over $I^* \cup J^*$.
      Move β of SVM until the instances are correctly classified.
      Run problem CSVM over I^*, J^* with initial solutions from before.
      Validate over $Validation^*$, getting the accuracy ($ACC[kf2]$).
    end
    Calculate the average accuracy $(\sum_{kf2} ACC[kf2])/folds2 = \overline{ACC}$.
    if $\overline{ACC} \geq bestACC$ **then**
      Set $bestACC = \overline{ACC}$, $best\gamma = \gamma$ and $bestC = C$.
    end
  end
  Run standard SVM over $I \cup J$ with the parameters $best\gamma$ and $bestC$.
  Move β of SVM until the instances are correctly classified.
  Run problem CSVM over I, J with initial solutions from the previous step.
  Validate over $Validation$, getting the correct classification probabilities ($TPR[kf]$, $TNR[kf]$).
end
Calculate the average values for TPR and TNR.

Name	$\lvert A\rvert$	$\lvert\Omega_0\rvert$	$\lvert\Omega_{0V}\rvert$	$\lvert\Omega_{0+}\rvert$ (%)
`australian`	14	690	69	307 (44.5%)
`votes`	16	435	44	267 (61.4 %)
`wisconsin`	30	569	57	212 (37.3 %)
`german`	45	1000	200	300 (30%)
`pageBlocks`	10	5472	1094	558 (10.2%)
`biodeg`	41	1055	211	356 (33.7%)

Table 2.1: Details concerning the implementation of the CSVM for the considered datasets.

Note that prior to running the different experiments, data have been standardized, that is to say, each attribute has zero mean and unit variance.

As a remark, we want to express that for the two biggest datasets (those that have more than 1000 samples), an alternative is proposed in order to reduce the computational times. First, to train the classifier, instead of using the training samples, we have built clusters of training points of the same class via the k-means method. The number of clusters was selected so that the proportion of original positive and negative instances was maintained. Also, we took into consideration the number of instances per cluster to train the SVM. In the validation sample, we kept the instances as they were originally.

2.3.5 Results

In this section we illustrate the performance of the CSVM in comparison with the classic SVM, the SVM(C_+, C_-) and the *Sliding β strategy*. As previously commented, the purpose will be to increase the TPR. Note that, even though from Section 2.2.3 the CSVM problem is always feasible using the training sample, it may happen that the desired performance is not achieved in the validation sample.

Table 2.2 reports the average rates (and under them, and in parenthesis, their standard deviations) obtained under the SVM, SVM(C_+, C_-), *Sliding β strategy* and CSVM, for the experiment described in Section 2.3.1, that is, when $\widehat{\text{TPR}} \geq \min\left\{1, \text{TPR}_0 + \sqrt{\frac{\log 0.05}{-2n}} + 0.025\right\}$ is imposed. Also, the target values (in parenthesis in the third and forth columns) to be achieved for the TPR are shown.

Some comments arise from the table. In the case of `australian`, we trivially considered "+" as the positive class and "−" as the negative. The SVM(C_+, C_-) slightly improves the TNR when it is compared with the standard SVM, but yields a worse value in the TPR, which is the rate to be improved. When the *Sliding β strategy* is used, although a target value of 0.855 is imposed, even a lower value that the one got with the SVM is obtained, with a lower TNR value also. On the other hand, when the

Name		SVM	SVM(C_+, C_-)	Sliding β	CSVM
		Mean (Std)	Mean (Std)	Mean (Target) (Std)	Mean (Target) (Std)
`australian`	TPR	**0.83** (0.071)	**0.806** (0.093)	**0.821** (0.855) (0.073)	**0.903** (0.855) (0.05)
	TNR	0.863 (0.079)	0.878 (0.088)	0.855 (0.068)	0.772 (0.081)
`votes`	TPR	**0.963** (0.04)	**0.945** (0.042)	**0.971** (0.988) (0.037)	**0.978** (0.988) (0.026)
	TNR	0.951 (0.031)	0.941 (0.037)	0.91 (0.063)	0.922 (0.04)
`wisconsin`	TPR	**0.948** (0.049)	**0.962** (0.027)	**0.989** (0.973) (0.017)	**0.965** (0.973) (0.037)
	TNR	0.99 (0.017)	0.931 (0.07)	0.953 (0.045)	0.945 (0.045)
`german`	TPR	**0.464** (0.103)	**0.89** (0.08)	**0.043** (0.65) (0.023)	**0.671** (0.65) (0.164)
	TNR	0.847 (0.031)	0.407 (0.069)	0.996 (0.009)	0.668 (0.111)
`pageBlocks`	TPR	**0.807** (0.03)	**0.557** (0.361)	**0.819** (0.832) (0.981)	**0.859** (0.832) (0.045)
	TNR	0.988 (0.004)	**0.901** (0.088)	0.981 (0.006)	0.965 (0.012)
`biodeg`	TPR	**0.783** (0.084)	**0.793** (0.083)	**0.797** (0.808) (0.095)	**0.852** (0.808) (0.057)
	TNR	0.909 (0.032)	0.839 (0.037)	0.891 (0.037)	0.833 (0.05)

Table 2.2: Results under the SVM, SVM(C_+, C_-), the *Sliding β strategy* and the novel CSVM. Target rate: TPR

CSVM is used instead, the increase is not only of 0.025 points but of 0.073, obviously at the expense of the other class. Hence, the best TPR is obtained for the CSVM.

We shall analyze next the results for `votes`, which has two classes: "democrat" and "republican". Since in principle there is no interest in a better classification of one of the classes, the majority class ("democrat") will be identified as the positive class. From the table it can be seen how the results under the SVM(C_+, C_-) are poorer than under the classic SVM. If the *Sliding β strategy* is used instead, an increase in the TPR is obtained but the rate does not achieve the target value. Even though the CSVM does not achieve the target value in the validation set, here again, this novel approach achieves the best TPR.

Concerning `wisconsin` dataset, it has two classes: "malignant" and "benign". Here, we consider as positive the "malignant" class, which is clearly the class of interest. The results for the SVM(C_+, C_-) are better than those obtained under the SVM, but it does not achieve the target value. When the *Sliding β strategy* is used, the target value for the TPR is achieved, while reducing the value for the TNR with respect to the SVM. Then, when we use the CSVM, the TPR is a bit higher than when the SVM(C_+, C_-)

is used, but lower than the one obtained for the *Sliding β strategy*. The same happens for the TNR. In this case, the method that performs the best is the *Sliding β strategy*. Next, we shall analyze `german` dataset, which is composed by two classes: good and bad credit risk. The class of interest and hence the positive one, is "bad credit risk". Here, the SVM(C_+, C_-) improves in a significant way the estimation of the TPR in comparison to the classic SVM; however, this is achieved at the expense of worsening the TNR. The *Sliding β strategy* performs very poorly in the case of the TPR but provides in contrast a very high TNR. The CSVM gets the most balanced result: the TPR exceeds the target values, and at the same time, the TNR is not notably affected. We next describe the results obtained for the `pageBlocks` dataset which, as it has been previously commented, is a strongly imbalanced dataset with a dimension higher than in the previous cases. The two classes for this dataset are "text" and "graphic" areas. In addition, the "graphic" areas instances are less frequent (10.2 %). Assume that for this problem the interest is in distinguishing the "graphic" areas from the "text" areas, therefore, the class to be controlled will be the "graphic" one. The results show how the SVM(C_+, C_-) obtains the opposite effect than the pursued. Both the TPR and TNR are lower than when the classic SVM is used. In the case of using the *Sliding β strategy*, the TPR is increased but it does not reach the imposed target. On the other hand, the TNR is slightly reduced. For the CSVM, the target value in the TPR is reached, resulting in a small decrease in the TNR.
Finally, we present the results for `biodeg`, with two classes: "ready biodegradable" and "not ready biodegradable". Originally, Mansouri et al. [2013], classification models were used to discriminate "ready biodegradable" from "not ready biodegradable", being "ready biodegradable" considered as the positive class. Here again, SVM(C_+, C_-) improves the TPR with respect to the classic SVM. The *Sliding β strategy* outperforms the SVM(C_+, C_-) but only the CSVM obtains an estimated TPR larger than the imposed lower bound. Note that, in contrast, the TNR under the CSVM is slightly lower than the values under the benchmark approaches.

Overall, the target value is almost always achieved when the CSVM is used. In the cases this does not occur, we obtain a close value. However, although initially one may think that good results will be obtained for the *Sliding β strategy*, such naive procedure does not achieve the target value so frequently. The same occurs with the SVM(C_+, C_-). Hence, we can conclude that the method that provides more control on the performance measures is the CSVM, which highlights the novelty of our proposal.

2.4 Chapter Summary

In this chapter, we propose a new supervised learning SVM-based method, the CSVM, with the purpose of controlling a specific performance measure. Such classifier is built

via a reformulation of the classic SVM, where novel constraints including integer variables are added. The final optimization problem is a MIQP problem, which can be solved using standard solvers as Gurobi or CPLEX. In order to guarantee that the performance rate is lower bounded by a fixed constant with a high confidence, some theoretical foundations are provided. The applicability of this cost-sensitive SVM has been demonstrated by numerical experiments on benchmark data sets.

We conclude that it is possible to control the classification rates in one class, possibly, but not necessarily, at the expense of the performance on the other class. This highly contrasts with the naive approach in which, once the SVM is solved, its intercept is moved to enhance the positive rates in one class, necessarily deteriorating the performance in the other class. The results presented confirm the power of our approach.

Although, for simplicity, all numerical results are presented just adding one performance constraint, one constraint per class, as well as an overall accuracy, may be added in our approach.

Chapter 3

Cost-sensitive class probability estimation in support vector machines

As formerly expounded, classification in SVM is based on a score procedure, yielding a deterministic classification rule, which can be transformed into a probabilistic one (as implemented in off-the-shelf SVM libraries), but is not probabilistic in nature. On the other hand, the tuning of the regularization parameters in SVM is known to imply a high computational effort and generates pieces of information that are not fully exploited, not being used to build a probabilistic classification rule.

In this chapter we propose a novel approach to generate probabilistic outputs for the SVM. The new method has the following three properties. First, it is designed to be cost-sensitive, and thus the different importance of sensitivity (or true positive rate, TPR) and specificity (true negative rate, TNR) is readily accommodated in the model. As a result, the model can deal with imbalanced data which are common in operational business problems as churn prediction or credit scoring. Second, the SVM is embedded in an ensemble method to improve its performance, making use of the valuable information generated in the parameters tuning process. Finally, the probabilities estimation is done via bootstrap estimates, avoiding the use of parametric models as competing approaches. This allows us to compute confidence intervals for the score and class probabilities for a given individual. Numerical tests on a wide range of datasets show the advantages of our approach over benchmark procedures.

3.1 Introduction

There exist many real-world contexts where it is relevant to get probabilistic outputs as posterior probabilities $P(y = +1 \mid x)$, which are of special interest if a measure of confidence in the predictions is sought, see Murphy [2012]. This is of particular importance in several business problems such as churn prediction, e.g., Huang et al. [2012], to calibrate the probability of churning of a customer, or credit scoring, e.g., Thomas et al. [2017], where the probability of defaulting is to be estimated. In these situations, it is important not only to get a hard label for the individual but also, an estimation of the degree of confidence in the assignation.

Several attempts to obtain the posterior probabilities $P(y = +1 \mid x)$ for SVM have been already carried out previously. One of them is based on assigning posterior class probabilities assuming a specific parametric family for the posterior probability. For example, Wahba [1992], Wahba et al. [1999] proposed a logistic link function,

$$P(y = +1 \mid x) = \frac{1}{1 + exp(-f(x))}. \tag{3.1}$$

Also, Vapnik and Vapnik [1998] suggested to estimate $P(y = +1 \mid x)$ in terms of a series of the trigonometric functions, where the coefficients of the trigonometric expansion minimizes a regularized function. Another considered option has been to fit

Gaussians to the class-conditional densities $P(f(x) \mid y = +1)$ and $P(f(x) \mid y = -1)$, as proposed in Hastie and Tibshirani [1998]. From such a choice, the posterior probability $P(y = +1 \mid f(x))$ is assumed to be a sigmoid, whose slope is determined by the tied variance. One of the best-known heuristics to obtain probabilities is due to Platt [2000], which considers $f(x)$ as the log-odds ratio $\log \dfrac{P(y = +1 \mid x)}{P(y = -1 \mid x)}$. This implies that

$$P(y = +1 \mid x) = \frac{1}{1 + \exp(Af(x) + B)}, \tag{3.2}$$

and A and B can be estimated by maximum likelihood on a validation set. This technique is implemented by well-known statistical packages such as the `ksvm()` function in R, see Karatzoglou et al. [2006], `predict_proba` in scikit-learn in Python (Pedregosa et al. [2011]) or in the software LIBSVM (see Chang and Lin [2011]), which uses a better implementation of the method, as presented in Lin et al. [2007]. Although SVM is designed for binary classification, there are several extensions for multiclass problems, e.g. Carrizosa et al. [2008]; Lorena and de Carvalho [2008]; Wang and Shen [2007], and also some attempts to construct class probabilities are found in the literature. In particular, multiclass versions of Platt's approach can be found in Milgram et al. [2005] and have been implemented in software packages like LIBSVM (Chang and Lin [2011]).

Platt's approach has been criticized for failing to provide insight and for interpreting $f(x)$ as a log-odds ratio, which may be not accurate for some datasets, see Murphy [2012]; Tipping [2001]; Franc et al. [2011]. To illustrate such a phenomenon, consider Figure 3.1, which shows the fit of the sigmoid function (3.2) to the empirical class probabilities of two different, well-referenced datasets: `adult` and `wisconsin`, respectively (see Section 3.3.1). It can be seen that, while for `adult` dataset the fit provided by the method given in (3.2) performs reasonably well, the performance is poor for `wisconsin`.

Sollich [2002] considers a different probabilistic framework for SVM classification, based on Bayesian theory. In particular, it relates the SVM kernel to the covariance function for a Gaussian process prior and as a result, optimal values of the tuning parameter C and class probabilities are obtained in a natural way. Again, this method as the previously commented approaches make modeling assumptions that might not be satisfied by the data. Finally, other procedures seeking probabilistic outputs are found in the literature, as Seeger [2000], Kwok [1999, 1998], Herbrich et al. [1999].

None of the previously mentioned works produce cost-sensitive models, which are of crucial importance in many managerial decision-making problems. For instance, in a churn prediction context, classifying a churner customer as non-churner may have important negative consequences. In a similar way and in order to avoid high costs, it is more important for a financial institution to correctly classify a defaulting customer than a non-defaulting one. Comparable situations arise in other settings different from business and management domains as medical diagnosis, in which failing to detect a

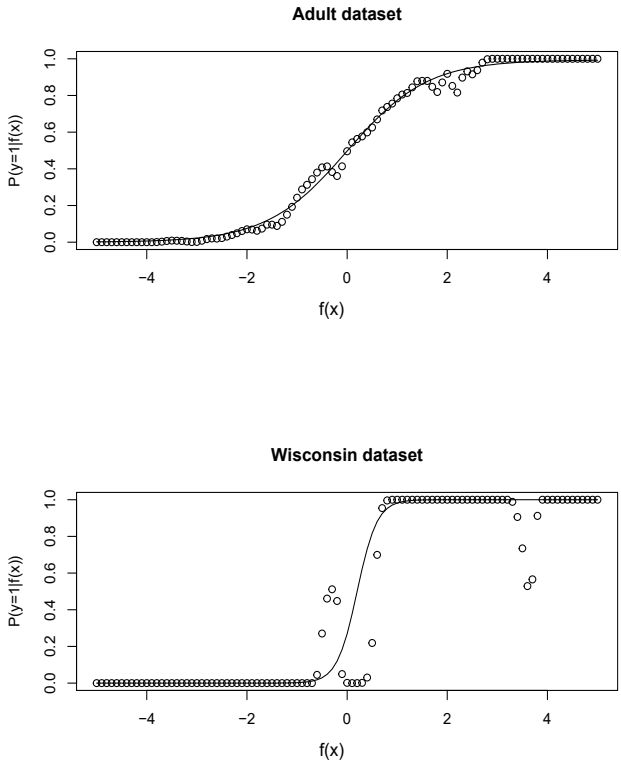


Figure 3.1: Fit (in solid line) of the sigmoid function to the empirical class probabilities (dots) of `adult` and `wisconsin` datasets.

disease may have fatal aftermaths. Because of that, cost-sensitive classification has become a trending issue lately and a number of references can be found regarding this, see Maldonado et al. [2021]; Coussement [2014]; Bradford et al. [1998]; Freitas et al. [2007]; Carrizosa et al. [2008]; Datta and Das [2015]; Benítez-Peña et al. [2019a,b].

Cost-sensitivity is closely related to the problem of imbalancedness in datasets. Imbalancedness may produce unnacurate classification rates for the minority class that is often the most critical one, Ghatasheh et al. [2020]. Several attempts in the literature have considered probabilistic outputs for the SVM in a context of imbalancedness. For example, Tao et al. [2005] propose robust SVM that turn out insensitive to the class imbalancedness. Their approach, the *posterior probability support vector machine* (PPSVM), is distribution-free and weighs imbalanced training samples. A multiclass approach based on the method in Tao et al. [2005] is proposed by Gonen et al. [2008]. Also, a more sophisticated and computationally expensive alternative is proposed by Kim et al. [2015], which combines layers of SVM with class probability output networks (CPONs), in which strong statistical assumptions are imposed.

In this work not only we provide a method for obtaining point estimates for the class probabilities for the SVM (as other works have done before), but also we construct confidence intervals for the scores and the posterior class probabilities. In addition, our methodology addresses properly cost sensitivity, since the rates of main interest (either TPR or TNR, which are the probabilities of an individual with label $y = +1$ and $y = -1$ respectively, being classified in class $+1$ and -1, respectively) are explicitly controlled. As an example, and continuing with the credit scoring problem we consider the dataset `german`, available at the UCI Repository [Dheeru and Karra Taniskidou, 2017] and described in detail in Section 3.3. This dataset is slightly imbalanced (the class of defaulting customers represents the 30% of the total). The mean squared error (MSE) of the probability prediction for the defaulting class under the non cost-sensitive version of the novel method is equal to 0.51 (average value over a testing sample). If we use instead the cost-sensitive version, this error decreases down to 0.134. In the setting of churn prediction we obtain similar results. Again, the considered database `churn` is imbalanced, being the percentage of churners equal to 15.71% (see Section 3.3). From an initial mean squared error equal to 0.8 the new model is able to decrease it down to 0.149. As it will be commented, such reductions may be at the expense of damaging the prediction of the posterior negative class probabilities, which are assumed to be less relevant.

Another distinctive feature of our approach is that the SVM is embedded in an ensemble method which, as will be shown, means an improvement in performance. It is known that, in order to solve the SVM problem (SVM(C_+, C_-)), a tuning process concerning the regularization parameter C in the grid $\mathcal{C}$ needs to be performed. Traditionally, all the information resulting from this tuning procedure is discarded and only

the *best* C value is used to build the classifier. Instead, in this work, the final posterior class probability estimate is a weighted mean of different posterior probabilities, each one related to specific values of C. In addition, here we propose a novel methodology that does not make use of parametric models based on the score function $f(x)$ obtained after tuning the SVM parameters. Instead, we consider a bootstrap framework (Efron and Tibshirani [1986]; Efron [2000]), which, to the best of our knowledge, has not been addressed before for this type of problems. The use of a bootstrap sampling allows us to obtain accurate values for the density of the score values, which translates into a better prediction of the posterior class probability $P(y = +1 \mid x)$, and, on top of the point estimates, confidence intervals for the score values and estimated class probabilities for a given individual. This is another novelty of our approach with respect to off-the-shelf procedures in the literature.

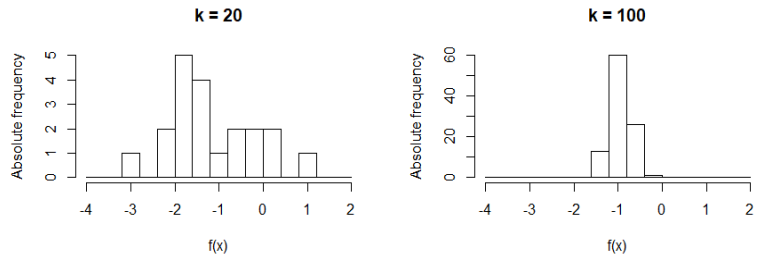
The chapter is structured as follows. First, in Section 3.2, our methodology is introduced. Section 3.2.1 describes how to integrate a bootstrap sampling into an SVM to enhance accuracy and to produce posterior class probabilities estimates as well as their corresponding confidence intervals of the score values. Section 3.2.2 explains two different ways to obtain cost-sensitive probabilistic predictions. In Section 3.3 some experimental results are presented. In particular, several well-referenced datasets from business, social sciences and other contexts are analyzed. Estimates of the posterior class probabilities under our methodology are compared to those obtained under benchmark approaches. Finally, the posterior probabilities of the classes of interest are controlled via the two different approaches described in Section 3.2.2. Conclusions can be found in Section 3.4.

3.2 Cost-sensitive predictive probabilities for SVM

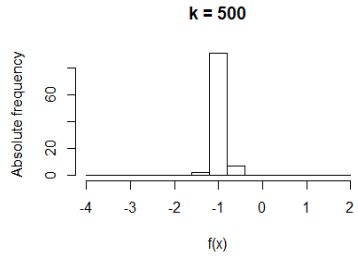
In this section we present our methodology to obtain point estimates for the posterior class probabilities as well as confidence intervals for them, using the SVM classifier. First, in Section 3.2.1 we explain how to integrate a bootstrap sampling into the SVM to produce posterior class probabilities estimates $P(y = +1 \mid x)$. Second, in Section 3.2.2 we describe two different approaches that allow us to control the posterior probability estimates in case we have a binary classification problem with one of the classes of most interest.

3.2.1 SVM posterior class probabilities based on Bootstrap

Assume that we want to solve SVM (SVM(C_+, C_-)) to classify the observations in a dataset, such as e.g. `churn` dataset (see Section 3.3.1 for details concerning the database). In order to estimate the classification error of the SVM classifier, it is standard to consider a k-fold cross-validation (CV), see Kohavi et al. [1995]. Figure 3.2



(a) (b)

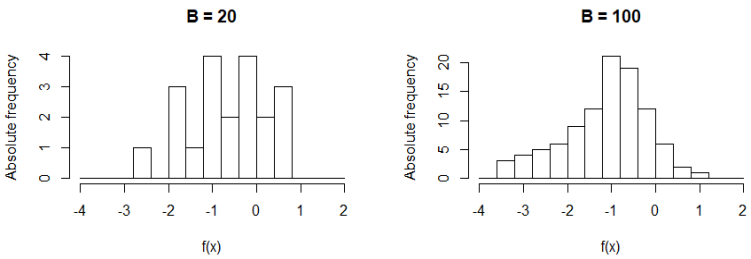


(c)

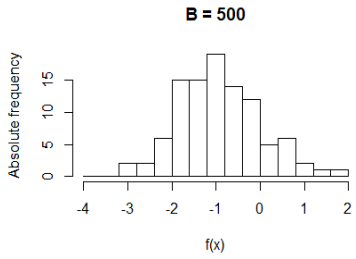
Figure 3.2: Histogram of scores for a single instance when a k-fold CV is used. Here, we set $k = 20$, 100 and 500, obtaining as many score values as the value of k.

shows the histogram (in absolute frequencies) of the score values under three different choices of k ($k = 20, 100, 500$) for a given individual (randomly chosen). It can be observed that as k increases, the score values are less disperse, a consequence of the fact that the different samples share more elements, and thus they yield more similar scores. Our procedure to calculate $P(y = +1 \mid x)$ for a given instance, as it will be explained later, will be based on considering the proportion of positive scores using different classifiers, taking advantage of the fact that different classifiers have to be build in order to measure the performance. However, in the k-fold CV situation, it might not be possible to obtain accurate posterior class probabilities $P(y = +1 \mid x)$, especially when the observations of the two groups strongly overlap. What it is proposed in this paper is to replace the k-fold CV approach by a bootstrap sampling that allows us to avoid the degenerate behaviour observed in Figure 3.2. The results are those illustrated in Figure 3.3, where the analogous histograms to Figure 3.2 are shown, but where a bootstrap sampling with B replications ($B = 20, 100, 500$) has been considered instead.

The idea of using those values is just to illustrate the behavior of this method when increasing the sample size in contrast with the one shown in Figure 3.2. Finally, as already exposed, the estimates for the posterior class probabilities $P(y = +1 \mid x)$ will be



(a) (b)



(c)

Figure 3.3: Histogram of scores for a single instance when the Bootstrap with B replications is used. As in the k-fold CV, $B = 20$, 100 and 500.

obtained as the relative frequency of the positive (negative in the case of $P(y = -1 \mid x)$) score values.

Now we explain how our procedure works. The pseudocode can be found as Algorithm 2 below, and a flowchart is given in Figure 3.4.

The goal here will be to estimate, for a generic individual with predictor variables x, the probability $P(y = +1 \mid x)$. For that, roughly speaking, we will make use of the bootstrap resampling, obtaining for each sample and each choice of the parameter C a different classifier that will be used in order to obtain different scores for a given individual. The probability $P(y = +1 \mid x)$ will be calculated as (a correction of) the proportion of positive scores obtained. The procedure is carried out as follows. First, to carry out our procedure of obtaining probabilities, consider a complete dataset Ω_0 composed of m instances, n variables, and 2 classes ($+1$ or -1). The dataset is divided in two samples: the training sample T (of sample size mtr) in which we have the class label information, in order to train and estimate the accuracy of the classifier (with the aim of obtaining the weights w_j in (3.3) and therefore being able to calculate the probabilities as in (3.4)) and an outer sample V (of size $m - mtr$) which consist of the instances for which the probabilities $P(y = +1 \mid x)$ have to be calculated, see Step 1 of Algorithm 2. As it is usual in the SVM implementation, a grid $\mathcal{C}$ for the regularization parameter needs to be set. Then, a matrix PX with as many rows as the number of instances in the outer sample ($m - mtr$) and as many columns as the number of values $\mathcal{C}$ to be used is built. This matrix contains the proportions of negative scores and is generated through the algorithm as follows. Given a fixed value $C \in \mathcal{C}$, and B bootstrap samples from the training sample T (that will be denoted as $T_b^*,\ b = 1, \ldots, B$), SVM is run over each bootstrap sample and validated over the out of bag sample, i.e., the set of instances that are in T but not in the considered bootstrap sample (we will denote them as $V_b^*,\ b = 1, \ldots, B$). In case we have class information in V, we can validate twice: the first time (over the validation in the bootstrap procedure V_b^*), to measure the performance led by the chosen value C and then over a outer set V, to estimate the posterior negative class probabilities conditioned to C, $P(y_{V_i} = -1 \mid x, C)$, where V_i denotes the i-th instance in V. In this way, we shall obtain B score values for each instance in V, which will allow us to calculate $P(y = +1 \mid x)$ in a frequentist way. Such score values shall be recorded in a matrix with as many rows as number of instances in the validation sample ($m - mtr$) and B columns, $PredictionV(f(x_i)_j^*)$. Finally, we propose to estimate its posterior negative class probability $P(y_{V_i} = -1 \mid x)$ as a weighted average (using the weights in (3.3) and calculated as in (3.4)) of the estimates for $P(y_{V_i} = -1 \mid x, C)$ when using different values of C, but taking only into account those values of C that lead to accuracies close to the best one, since we want to give more weight to those most promising values of C. Let $acc_{r,b}$ denote the standard accuracy in V_b^* (fraction of instances in V_b^* correctly classified), given the SVM built

from using the r-th value of $\mathcal{C}$, and the b-th bootstrap sample; let $\overline{acc}_r$ denote the average value of the coefficients $acc_{r,b}$, namely

$$\overline{acc}_r = \frac{\sum_{b=1}^{B} acc_{r,b}}{B}.$$

Only values yielding high estimates of $\overline{acc}_r$ are taken into consideration, the remaining ones being discarded. Those considered are stored in the set J, defined as $J = \{j : \overline{acc}_j \geq \max_l \overline{acc}_l - \varepsilon\}$, where $\varepsilon > 0$ is a fixed parameter.

Finally, if weights $w_j,\ j \in J$, are defined according to

$$w_j = \frac{\overline{acc}_j^2}{\sum_{l \in J} \overline{acc}_l^2}, \tag{3.3}$$

then, the estimation $P(y_{V_i} = -1 \mid x)$ is:

$$P(y_{V_i} = -1 \mid x) = \sum_{j \in J} w_j P(y_{V_i} = -1 \mid x, C_j). \tag{3.4}$$

As commented before, the bootstrap sample allows us to obtain point estimates for the posterior class probabilities $P(y = +1 \mid x)$. From the classic confidence interval for a proportion, it is straightforward to construct a $(1-\alpha)\%$ confidence interval for $\hat{p} = P(y = +1 \mid x)$ as

$$\left(\hat{p} - z_{1-\alpha/2}\sqrt{\frac{1/4}{B}}, \hat{p} + z_{1-\alpha/2}\sqrt{\frac{1/4}{B}}\right), \tag{3.5}$$

where $z_{1-\alpha/2}$ denotes the $1-\alpha/2$-th percentile of a standard normal distribution. Note that (3.5) is not a bootstrap confidence interval per se but a confidence interval constructed from a bootstrap estimate $\hat{p}$. However, it is easy to obtain a bootstrap confidence interval for the score value $\widehat{f(x_i)}$ of a given individual i. Specifically, given a confidence level $1-\alpha$, the *basic bootstrap* (Wehrens et al. [2000]) procedure leads to

$$\left(2\widehat{f(x_i)} - f(x_i)^*_{(1-\alpha/2)}, 2\widehat{f(x_i)} - f(x_i)^*_{(\alpha/2)}\right), \tag{3.6}$$

where

$$\widehat{f(x_i)} = \overline{f(x_i)}^* = \sum_j \frac{f(x_i)^*_j}{B}$$

denotes a point estimate of the score value for individual i and $f(x_i)^*_{(1-\alpha/2)}$ is the $(1-\alpha/2)$-th percentile of the bootstrapped coefficients $f(x_i)^*$ for the same individual. Note that from the confidence interval as in (3.6) it is straightforward to decide if there is evidence to reject the null hypothesis of a positive score (which would imply that the

individual is classified in the negative class).

Algorithm 2: Pseudocode for the Bootstrap SVM

```
Split Ω_0 into T and V: T ∪ V = Ω_0, T ∩ V = ∅.
PX = Initialize as an empty matrix
for each C in grid of C_r, r = 1,...,R do
    PredictionV(f(x_i)*_j) = Initialize as an empty matrix
    for b in 1,2,...,B do
        Create a bootstrap sample T*_b from T.
        Run SVM over T*_b. Validate over V*_b = T \ T*_b.
        Obtain (for each instance in V*_b) the accuracy acc_{r,b}.
        Obtain (for each instance in V) scores values for V, denoted as scoresV.
        Insert scoresV as a new column in PredictionV(f(x_i)*_j)
    end
    Estimate P(y = −1 | x, C) for each of the (m − mtr) instances in V, as the
     proportion of negative scores in each of the (m − mtr) rows in
     PredictionV(f(x_i)*_j). These estimates are inserted as a new column in
     PX.
end
Using the values acc_{r,b}, calculate the weighted average given by (3.4).
```

The novel methodology will be illustrated in Section 3.3.2 where, in addition, some comparisons with respect to benchmark approaches will also be presented.

3.2.2 Control over the sensitivity measure

In the previous section, an approach for estimating posterior class probabilities $P(y = +1 \mid x)$ and $P(y = -1 \mid x)$ has been described, and confidence intervals are obtained as well. In this section, we deal with the issue of improving the sensitivity of the classifier (or TPR) which, as commented in Section 3.1, may be a problem of interest, among others, in business, social sciences or biomedical contexts. To do this, we propose two different approaches, *Ctrl1* and *Ctrl2*, which are discussed in what follows and are empirically analyzed in Section 3.3.3.

Method *Ctrl1* is based on the fact that the sensitivity measure can be controlled by the posterior class probabilities, as is explained next. In Algorithm 2, the posterior negative class probabilities have been estimated taking into account the proportion of negative scores. However, if instead of 0, we consider a different threshold (say a value $-a$, with a positive), then the estimates for the values of $P(y_{V_i} = -1 \mid x, C)$ decrease (that is, the posterior positive class probabilities $P(y_{V_i} = +1 \mid x, C)$ increase). This is illustrated by two examples in Figure 3.5, Figures 3.6(a) and 3.5(b), which represent the histograms of the scores for two different individuals of the `churn` dataset. Figure 3.6(a)

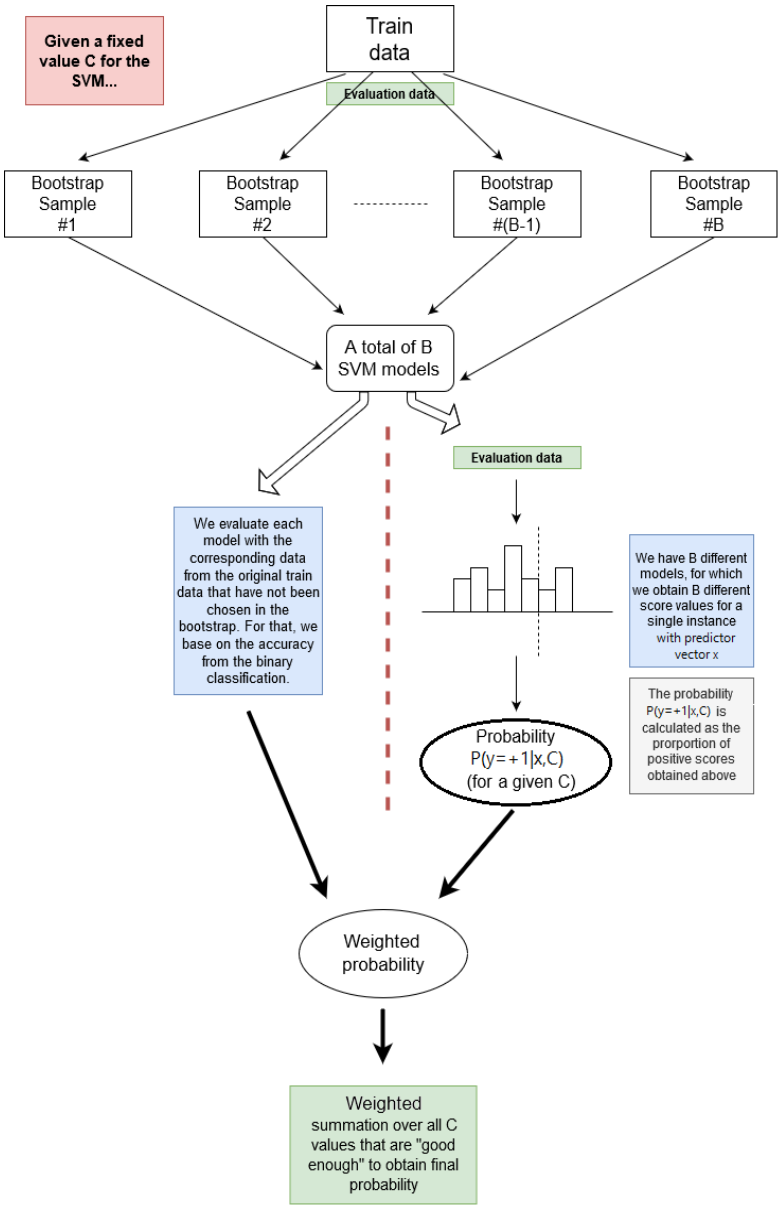
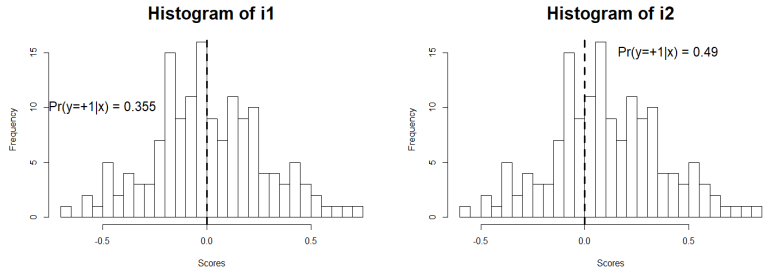
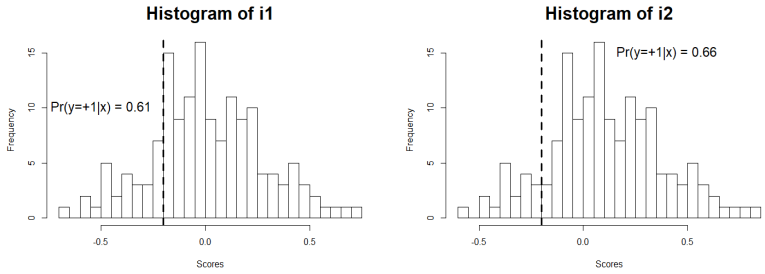


Figure 3.4: Flowchart of the Bootstrap-based methodology for obtaining $P(y = +1 \mid x)$.



(a)



(b)

Figure 3.5: Control over the probabilities estimation. In Subfigures 3.6(a) we can observe the original estimated probabilities, whereas in Subfigures 3.5(b) the new cost-sensitive probabilities for 3.6(a), obtained by moving the threshold, are depicted.

shows the posterior positive class probability estimates using Algorithm 2, that is, where the value 0 is used as a threshold to classify in the positive or the negative class. In Figure 3.5(b), the threshold value has been moved to the left and, as a consequence, the resulting estimates have increased. Note from Figure 3.5(b) that, with this approach, the probability of an instance to belong to the positive class may change from below to above 0.5. In practice, in order to obtain a desired posterior positive class probability estimate, the threshold is moved until a certain proportion of the instances of the positive class are correctly classified.

Method *Ctrl2* also results from Algorithm 2, but, instead of changing the threshold for the scores, we consider a different classifier. Specifically, we propose to use a novel version of the SVM, the so-called Constrained SVM (CSVM), which has been particularly designed to obtain cost-sensitive results, see Benítez-Peña et al. [2019b]. Without going into much detail, the CSVM formulation is obtained by solving a convex quadratic optimization problem with linear constraints and some integer variables:

$$
\begin{array}{lll}
\min_{\omega,\beta,\xi,z} & \omega^\top\omega + C\sum_{i\in I}\xi_i & \\
s.t. & y_i(\omega^\top x_i+\beta) \geq 1-\xi_i, & i\in I \\
& 0\leq \xi_i \leq M(1-z_i) & i\in I \\
& \hat{p}_\ell \geq p^*_{0\ell} & \ell \in L \\
& z_i \in \{0,1\} & i\in I.
\end{array} \tag{3.7}
$$

Problem (4.1) is simply the formulation for the standard SVM with linear kernel, to which performance constraints have been added: $\hat{p}_\ell \geq p^*_{0\ell}$, where $\hat{p}_\ell$ are different performance measures, forced to take values above thresholds $p^*_{0\ell}$, and z_i are new binary variables that check whether record i is counted as correctly classified, and M is a large number. We refer the reader to the original reference (Benítez-Peña et al. [2019b]) for a more detailed description of the cost-sensitive classifier. The important message to be kept here is that solving (4.1) with a standard software package for different values of the parameters $p^*_{0\ell}$ yields classifiers with different trade-off between sensitivity and specificity.

Both methods (*Ctrl1* and *Ctrl2*) will be illustrated through numerical examples in Section 3.3.3.

3.3 Experimental results

In this section we illustrate the performance of our method for computing posterior class probability estimates as described in Section 3.2.1. The results will be compared to those of benchmark approaches by Platt [2000], Sollich [2002] and Tao et al. [2005]. For that, a variety of datasets with different properties concerning size (in the number of instances and/or variables) and imbalancedness shall be analyzed. Moreover, we test the methods described in Section 3.2.2 to control the posterior positive class probability. Specifically, this section is organized as follows. In Section 3.3.1 we present a brief description of the different datasets we have used and describe how the different experiments have been implemented. Section 3.3.2 shows the performance of the novel approach in comparison to benchmark methodologies to build point estimates of the probabilities. Also, confidence intervals are computed for the score values and class probabilities for `german` dataset. Finally, in Section 3.3.3 we apply both *Ctrl1* and *Ctrl2* to improve the posterior probability of the class of interest.

3.3.1 Datasets and description of the experiments

The performance of the different methodologies presented in this paper is illustrated using fourteen real-life datasets: `absenteeism` (Absenteeism at work Data Set), `adult` (Adult), `australian` (Statlog (Australian Credit Approval) Data Set), `banknote` (banknote authentication), `careval` (Car Evaluation Data Set), `cervical-cancer` (Cervi-

cal cancer (Risk Factors)), `churn` (Customer churn), `german` (German Credit Data), `heart` (Heart Disease), `housing` (The Boston Housing Dataset), `leukemia` (Leukemia), `productivity` (Productivity Prediction of Garment Employees Data Set), `SRBCT` (Small Round Blue Cell Tumor) and `wisconsin` (Breast Cancer Wisconsin (Diagnostic)), `SRBCT` dataset can be obtained from the R package `plsgenomics` (Boulesteix et al. [2011]) and `leukemia` from Golub et al. [1999b]. On the other hand, `housing` is taken from Harrison and Rubinfeld [1978] and `churn` from Keramati and Ardabili [2011]. The other eleven datasets are obtained from the UCI Repository, (Dheeru and Karra Taniskidou [2017]). Dataset `cervical-cancer` has been split into two different datasets since it contains 4 different variables of class. We show 2 of them as an illustration. Table 4.1 contains relevant information of the previous datasets. In the second and third columns, the sample sizes of the validation ($|\Omega_V|$) and the complete datasets ($|\Omega|$) are shown, respectively. The fourth column contains the number of original variables or attributes ($|A|$) in the dataset. Finally, the last column collects the number ($|\Omega_+|$) and percentage (%) of positive instances in the complete dataset.

Name	$\lvert\Omega_{0V}\rvert$	$\lvert\Omega_0\rvert$	$\lvert A\rvert$	$\lvert\Omega_{0+}\rvert$ *(%)*
`absenteeism`	74	739	20	272 (36.81%)
`adult`	3256	32561	14	7841 (24.08%)
`australian`	69	690	14	307 (44.49%)
`banknote`	137	1372	5	610 (44.46%)
`careval`	173	1728	6	518 (29.98%)
`cervical-cancer-1`	86	858	36	35 (4.08%)
`cervical-cancer-2`	86	858	36	74 (8.62%)
`churn`	315	3150	11	495 (15.71%)
`german`	100	1000	20	300 (30%)
`heart`	72	720	75	362 (50.28%)
`housing`	51	506	13	256 (50.59%)
`leukemia`	7	72	7128	25 (34.72%)
`productivity`	120	1196	14	474 (39.63%)
`SRBCT`	8	83	1022	29 (34.94%)
`wisconsin`	57	569	30	212 (37.26%)

Table 3.1: Datasets

In a pre-processing step, the categorical variables were transformed into a set of dummy variables. In addition, those datasets with three classes or more were converted into two-class datasets by giving negative label to the largest class and positive labels to the remaining records. In the case of missing values, they were replaced by the median in the case of numerical variables and by the mode in the case of categorical ones. Standardization of the data to have each numerical variable coming from a distribution with mean 0 and unit variance has been consider in each fold (for both the k-fold CV and in the bootstrap), performing it first over the training data and then using the obtained average and standard deviation to standardize the validation one. Finally, when running the SVM and the constrained SVM in (SVM(C_+, C_-)) or (4.1), the linear

kernel versions were considered. All the experiments have been carried out using the solver Gurobi (Gurobi Optimization [2016]) and its Python language interface (Python Core Team [2015]). No timelimit was imposed when solving Problem (SVM(C_+, C_-)), whereas 300 seconds was set when solving (4.1). Also, for the latter problem, M was equal to 1000 (see, Benítez-Peña et al. [2019b] for more details).

In our experiments, the number of folds selected for the k-fold CV is 10 external folds (and we estimate the performance measure by the average over the 10 folds) and 10 internal folds (in order to obtain the best parameter C). The number of bootstrap samples B has been set equal to 500 and each bootstrap training sample has the same size as the original training sample. Note that we cope with the imbalancedness, if present, though one could have performed under or oversampling in the majority or the minority class, respectively, in a preprocessing phase. The grid $\mathcal{C}$ of values selected in our experiments is $\{2^{-5}, 2^{-4}, ..., 2^4, 2^5\}$.

3.3.2 Performance of the bootstrap-based approach

In this section we obtain point estimates of the posterior class probabilities according to the bootstrap-based novel method described in Section 3.2.1 and compare the results with those obtained by the benchmark approaches by Platt [2000], Sollich [2002] and Tao et al. [2005] commented in Section 1. The obtained results are summarized in Table 3.2, whose second, third and fourth columns contain the mean squared errors (MSE) values obtained when the deterministic class membership is approximated to its probabilistic counterpart. Note that, according to Tao et al. [2005], a value for the parameter r needs to be selected. In this case, we tested the results for four different choices of r (0, $\sqrt{10}$, $\sqrt{20}$, $\sqrt{30}$). The best results have been highlighted in bold style.

It can be seen from Table 3.2, how our methodology is the one performing best for `absenteeism`, `cervical-cancer-1`, `cervical-cancer-2`, `housing`, `leukemia` and `wisconsin`, obtaining the lowest values of MSE. Additionally, the method proposed by Platt [2000] obtains the lowest MSE in `absenteeism`, `adult`, `australian`, `careval`, `churn`, `german` and `productivity`. Finally, with the method of Tao et al. [2005], the lowest MSE is obtained in `banknote` and `heart`, and also a zero MSE for `SRBCT` is achieved. On the other hand, the method proposed by Sollich [2002] performs poorly in all cases except in `banknote`. In conclusion, we have built a method that is comparable in terms of performance to benchmark approaches, outperforming them in some datasets. As described in Section 3.2.1, the final estimate of the posterior class probabilities is set in terms of the results obtained for a range of the regularization parameter C (see expression 3.4). It is of interest to compare the results with those computed using only the value of C that provides the best accuracy measure. The results are shown in Table 3.3, from which it can be concluded that embedding the SVM in an ensemble method actually improves its performance in most of the cases.

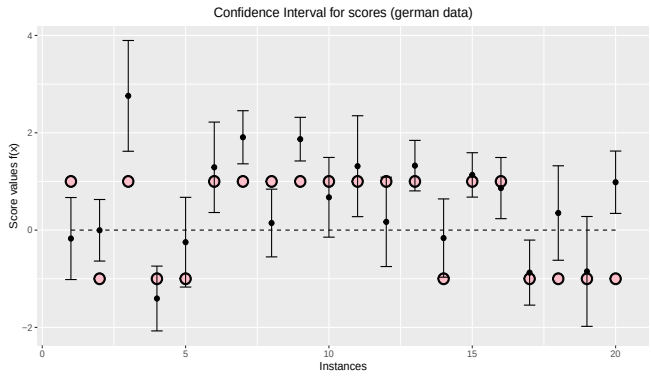
Dataset	*Bootstrap-based approach*	*Sollich*	*Platt*	*Tao et al.* ($r = 0, \sqrt{10}, \sqrt{20}, \sqrt{30}$)
absenteeism	**0.133**	0.16	**0.133**	0.176, 0.176, 0.176, 0.176
adult	0.158	0.232	**0.105**	0.151, 0.146, 0.128, 0.13
australian	0.173	0.223	**0.121**	0.151, 0.149, 0.135, 0.133
banknote	0.011	**0.008**	0.09	**0.008**, 0.145, 0.221, 0.238
careval	0.052	0.074	**0.04**	0.047, 0.047, 0.043, 0.093
cervical-cancer-1	**0.013**	0.234	0.04	0.045, 0.045, 0.045, 0.045
cervical-cancer-2	**0.075**	0.199	0.08	0.103, 0.103, 0.103, 0.103
churn	0.108	0.227	**0.093**	0.104, 0.104, 0.101, 0.128
german	0.203	0.203	**0.163**	0.235, 0.224, 0.187, 0.182
heart	0.171	0.217	0.124	0.157, 0.119, **0.113**, 0.168
housing	**0.078**	0.142	0.097	0.152, 0.118, 0.141, 0.163
leukemia	**0**	0.238	0.019	0.014, 0.014, 0.014, 0.014
productivity	0.212	0.238	**0.190**	0.242, 0.239, 0.262, 0.259
SRBCT	0.039	0.237	0.006	**0**, **0**, **0**, **0**
wisconsin	**0.018**	0.094	0.034	0.028, 0.019, 0.037, 0.055

Table 3.2: Mean squared errors (MSE) obtained when predicting the posterior class probabilities in a linear SVM.

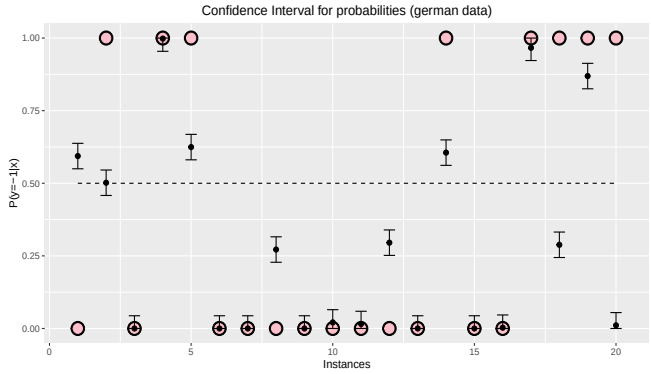
Dataset	*Best C*	*Bootstrap-based approach*
absenteeism	0.176	0.133
australian	0.217	0.173
banknote	0.007	0.011
careval	0.055	0.052
cervical-cancer-1	0.012	0.013
cervical-cancer-2	0.105	0.075
churn	0.108	0.108
divorce	0.059	0.059
german	0.24	0.203
heart	0.194	0.171
housing	0.118	0.078
leukemia	0	0
productivity	0.286	0.212
SRBCT	0.125	0.039
wisconsin	0.035	0.018

Table 3.3: Mean squared errors (MSE) using only the best C and under the bootstrap-based approach.

As already mentioned, not only point estimates for the class probabilities can be calculated, but also confidence intervals for both the scores and the probabilities. In Figure 3.6 we depict the confidence intervals for the scores and the probabilities $P(Y = -1 \mid x)$ of 20 different instances in the validation sample, for german dataset. In Figure 3.6(a) we present 95% confidence interval for the scores according to (3.6). For those intervals that do not contain the score 0, we can guarantee (with a confidence of a 95%) that the instance belongs to the positive class if the interval is above 0 or to the negative class, otherwise. Together with the intervals, we have represented with



(a)



(b)

Figure 3.6: Confidence intervals for the scores values (Top panel) and the probabilities $P(y = -1 \mid x)$ (Bottom panel), as well as actual values $P(y = -1)$ (light colour) for some instances of `german` dataset.

the clearer point the true class of the instance. Finally, Figure 3.6(b) represents the confidence intervals for the estimated probabilities $P(y = -1 \mid x)$, calculated from (3.5). Also, the true probabilities (known since we know the true classes) are represented using the clearer points.

3.3.3 Results when the posterior class probabilities are controlled

In this section we apply the methodologies described in Section 3.2.2 in order to control $P(y = +1 \mid x)$ or $P(y = -1 \mid x)$. In particular, Tables 3.4 and 3.6 are obtained under *Ctrl1* and Tables 3.5 and 3.7 show the results when the method based on the CSVM (*Ctrl2*) is implemented. For all the tables, the class of interest to be controlled is assumed to be the positive one, that is, we aim to control the true positive rate (TPR). Tables 3.4 and 3.5 show the MSE when considering only the positive instances, while Tables 3.6 and 3.7 depict the MSE for the negative instances. The values on the

"imposed TPR" depends on the original value obtained in the TPR when no control on the misclassification rates is imposed. Therefore, for values of "imposed TPR" lower than the one obtained, a "−" is placed instead of the resulting MSE. Also, depending on the obtained TPR, and in order to obtain more representation than a single value, a imposed TPR of 0.95 is imposed for some datasets. Finally, if the MSE was originally 0 (TPR = 1), no TPR has been imposed, as occurs for `banknote` and `leukemia`.

From Tables 3.4 and 3.5, we can see how as the threshold for obtaining a given proportion of the instances in the correct class is increased, the MSE becomes lower, as expected. In fact, there are some datasets (`banknote`, `careval`, `heart`, `housing`, `SRBCT` and `wisconsin`), for which the obtained MSEs are very close to 0, for both Tables 3.4 and 3.5. However, Tables 3.6 and 3.7 present different patterns. While Table 3.7 behaves as expected (as the MSEs for the sensitivity become smaller, the MSEs for the specificity become constant or higher), the specificity depicted by Table 3.6 remains almost unaltered or have even descending MSEs. Here again, some datasets result in almost null MSEs (`cervical-cancer`, `divorce`, `leukemia`, `SRBCT` and `wisconsin`).

Dataset \ TPR imposed	0	0.5	0.6	0.7	0.8	0.9	0.95	1
absentism	0.266	-	-	-	0.266	0.266	-	0.266
adult	0.476	-	0.476	0.476	0.476	0.476	-	0.476
australian	0.071	-	-	-	-	0.071	-	0.070
banknote	0	-	-	-	-	-	-	-
careval	0.044	-	-	-	-	-	0.040	0.038
cervical-cancer-1	1	1	1	1	1	1	-	1
cervical-cancer-2	1	0.964	0.963	0.963	0.963	0.963	-	0.963
churn	0.8	0.8	0.8	0.8	0.8	0.8	-	0.8
german	0.510	0.258	0.255	0.255	0.255	0.255	-	0.255
heart	0.141	-	-	-	-	-	0.071	0.071
housing	0.114	-	-	-	-	0.057	-	0.057
leukemia	0	-	-	-	-	-	-	-
productivity	0.354	-	0.352	0.180	0.180	0.180	-	0.180
SRBCT	0.063	-	-	-	-	0.063	-	0.063
wisconsin	0.036	-	-	-	-	-	0.018	0.018

Table 3.4: MSE for the positive class probability predictions of each dataset. *Ctrl1*

An important remark to be made concerning the performance of *Ctrl1* and *Ctrl2* is as follows. The first one seems to be able to improve the sensitivity without damaging too much the specificity or even improving it at the same time, while the second method damages in a more significant way the specificity, but at the same time it leads to better sensitivity values.

Dataset \ TPR imposed	0	0.5	0.6	0.7	0.8	0.9	0.95	1
absenteeism	0.266	-	-	-	0.254	0.237	-	0.221
adult	0.476	-	0.112	0.003	0.000	0.000	-	0.219
australian	0.071	-	-	-	-	0.056	-	0.091
banknote	0	-	-	-	-	-	-	-
careval	0.044	-	-	-	-	-	0.019	0.014
cervical-cancer-1	1	1	1	1	1	0.999	-	0.999
cervical-cancer-2	1	0.765	0.598	0.539	0.486	0.494	-	0.493
churn	0.8	0.437	0.416	0.271	0.228	0.175	-	0.149
german	0.510	0.427	0.192	0.175	0.157	0.155	-	0.134
heart	0.141	-	-	-	-	-	0.023	0.028
housing	0.114	-	-	-	-	-	0.066	0.035
leukemia	0.000	-	-	-	-	-	-	-
productivity	0.354	-	0.156	0.137	0.123	0.103	-	0.081
SRBCT	0.063	-	-	-	-	0	-	0
wisconsin	0.036	-	-	-	-	-	0.014	0.003

Table 3.5: MSE for the positive class probability predictions of each dataset. *Ctrl2*

3.4 Chapter Summary

In this chapter we have proposed a procedure to obtain probabilistic outputs for the Support Vector Machines, through both point estimates and confidence interval estimates. Contrary to existing proposals, we present a method that is distribution-free and cost-sensitive. Also, it makes use of not only a single classifier but a weighted average of them, obtaining more accurate results. The method turns out advantageous for operational business processes as credit scoring or churn prediction, where the class of interest may suffer from imbalancedness.

Our proposal is compared to some benchmark methodologies. The results show that our approach is comparable or better than such approaches if the focus is on point estimates. Up to our knowledge, no prior work has undertaken posterior class probabilities estimation by confidence intervals. We have shown how the bootstrap approach naturally leads to confidence intervals for the scores values. Two cost-sensitive alternatives are proposed here. The first one is based on changing the way the probabilities are estimated and the second one proposes to modify the original classifier by a cost-sensitive version. Results for real datasets have been shown, proving the usefulness of the novel approach.

For simplicity, the baseline SVM classifiers are taken with a linear kernel; more powerful classifiers will be obtained if nonlinear kernels (as RBF) are used, though at the expense of making the computational effort higher.

Dataset \ TPR imposed	0	0.5	0.6	0.7	0.8	0.9	0.95	1
absentism	0.088	-	-	-	0.088	0.044	-	0.044
adult	0.044	-	0.022	0.022	0.022	0.022	-	0.022
australian	0.271	-	-	-	-	0.135	-	0.135
banknote	0.019	-	-	-	-	-	-	-
careval	0.061	-	-	-	-	-	0.030	0.030
cervical-cancer-1	0.001	0.001	0.001	0.001	0.001	0.001	-	0.001
cervical-cancer-2	0.008	0.008	0.008	0.008	0.008	0.008	-	
churn	0.008	0.008	0.004	0.004	0.004	0.004	-	0.004
german	0.051	0.051	0.051	0.051	0.051	0.051	-	0.051
heart	0.207	-	-	-	-	-	0.207	0.071
housing	0.046	-	-	-	-	-	0.046	0.046
leukemia	0	-	-	-	-	-	-	-
productivity	0.119	-	0.136	0.140	0.140	0.140	-	0.140
SRBCT	0	-	-	-	-	-	0	0
wisconsin	0.006	-	-	-	-	-	0.006	0.006

Table 3.6: MSE for the negative class probability predictions of each dataset. *Ctrl1*

Dataset \ TPR imposed	0	0.5	0.6	0.7	0.8	0.9	0.95	1
absenteeism	0.088	-	-	-	0.043	0.045	-	0.051
adult	0.044	-	0.202	0.825	0.969	0.976	-	0.095
australian	0.271	-	-	-	-	0.304	-	0.171
banknote	0.019	-	-	-	-	-	-	-
careval	0.061	-	-	-	-	-	0.066	0.119
cervical-cancer-1	0.001	0.024	0.024	0.074	0.074	0.113	-	0.113
cervical-cancer-2	0.008	0.065	0.044	0.073	0.107	0.126	-	0.171
churn	0.008	0.049	0.060	0.076	0.044	0.065	-	0.097
german	0.051	0.093	0.125	0.145	0.207	0.252	-	0.421
heart	0.207	-	-	-	-	-	0.244	0.44
housing	0.046	-	-	-	-	-	0.236	0.296
leukemia	0	-	-	-	-	-	-	-
productivity	0.119	-	0.142	0.159	0.201	0.234	-	0.294
SRBCT	0	-	-	-	-	0.137	-	0.090
wisconsin	0.006	-	-	-	-	-	0.007	0.017

Table 3.7: MSE for the negative class probability predictions of each dataset. *Ctrl2*

Chapter 4

Cost-sensitive feature selection in support vector machines

The relevance of features in a classification procedure is linked to the fact that misclassifications costs are frequently asymmetric, since false positive and false negative cases may have very different consequences. However, off-the-shelf Feature Selection procedures seldom take into account such cost-sensitivity of errors.

In this chapter we propose a mathematical-optimization-based Feature Selection procedure embedded in one of the most popular classification procedures, namely, Support Vector Machines, accommodating asymmetric misclassification costs. The key idea is to replace the traditional margin maximization by minimizing the number of features selected, but imposing upper bounds on the false positive and negative rates. The problem is written as an integer linear problem plus a quadratic convex problem for Support Vector Machines with both linear and radial kernels.

The reported numerical experience demonstrates the usefulness of the proposed Feature Selection procedure. Indeed, our results on benchmark data sets show that a substantial decrease of the number of features is obtained, whilst the desired trade-off between false positive and false negative rates is achieved.

4.1 Introduction

An amount of different FS procedures are found in the literature, some independent of the classification procedure and others embedded in the classification procedure, like the Holdout SVM (HOSVM), Maldonado and Weber [2009], Kernel-Penalized SVM (KP-SVM), Maldonado et al. [2011], or the methods presented in Chan et al. [2007] or Ghaddar and Naoum-Sawaya [2018]. Also, one can minimize the number of relevant features or even their cost, as in Maldonado et al. [2017]. The embedded method together with whichever of the previous optimization schemes is the approach considered in this chapter, since we aim to obtain a SVM-based classifier, and, at the same time, perform the selection of the features. The core idea is the optimization problem to be solved: instead of maximizing the margin, as in the traditional SVM, we seek the classifier with lowest number of features (or cost of the features), but without damaging too much the original performance. In order to be able to control the classifier's performance, we will make use of the model presented in Chapter 2, but choosing the anchor sample J to be the same as the training sample I. Specifically, the formulation results

$$\begin{array}{lll} \min_{\omega,\beta,\xi,z} & \omega^\top \omega + C \sum_{i\in I} \xi_i & \\ s.t. & y_i(\omega^\top x_i + \beta) \geq 1 - \xi_i, & i \in I \\ & 0 \leq \xi_i \leq M_1(1 - z_i) & i \in I \\ & \hat{p}_\ell \geq \hat{p}^*_{0\ell} & \ell \in L \\ & z_i \in \{0,1\} & i \in I, \end{array} \tag{4.1}$$

where M_1 is a large number.

In essence, this is simply the formulation for the SVM with linear kernel, to which performance constraints have been added: $\hat{p}_\ell \geq \hat{p}^*_{0\ell}$, where $\hat{p}_\ell$ are different performance measures, forced to take values above thresholds $\hat{p}^*_{0\ell}$, and z_i are new binary variables that check whether sample i is counted as correctly classified. Its (partial) dual formulation is

$$\begin{array}{lll} \min_{\lambda,\beta,\xi,z} & \sum\limits_{i,j\in I} \lambda_i y_i \lambda_j y_j K(x_i,x_j) + C\sum_{i\in I}\xi_i & \\ s.t. & y_i(\sum\limits_{j\in I} \lambda_j y_j K(x_j,x_i) + \beta) \geq 1-\xi_i, & i\in I \\ & \sum\limits_{i\in I} \lambda_i y_i = 0 & \\ & 0 \leq \lambda_i \leq C/2 & i\in I \\ & 0 \leq \xi_i \leq M_1(1-z_i) & i\in I \\ & \hat{p}_\ell \geq \hat{p}^*_{0\ell} & \ell \in L \\ & z_i \in \{0,1\} & i\in I. \end{array} \tag{4.2}$$

As before, this is similar to the standard partial dual formulation of the SVM with general kernel and constraints in the performance measures, as in (4.1). For more information about how formulation (4.2) is obtained, the reader is referred to Chapter 2 and the Appendix. Note that, while mathematical optimization problems addressed in the statistical literature are, traditionally, as (SVM(C_+, C_-)) or (GSVM(C_+, C_-)), nonlinear programs in continuous variables, our approach involves integer variables, which define harder optimization problems. However, Integer Programming has shown to be rather competitive thanks to the impressive advances in (nonlinear) integer programming solvers, as demonstrated in recent papers addressing different topics in data analysis, Bertsimas et al. [2016, 2014]; Carrizosa et al. [2011, 2016, 2017a,b].

The remainder of the chapter is structured as follows. In Section 4.2 we present the new FS methodology for SVM. For either linear or nonlinear kernels, we reduce the optimization problem to solving a standard linear integer program plus, eventually, a quadratic convex problem. The performance of our FS approach is empirically tested under different experiments described in Section 4.3. The results of those experiments are shown in Section 4.4. Comparisons between the use of linear and radial kernels, and between the standard linear SVM with and without embedded FS are also provided. The chapter ends with conclusions in Section 4.5.

4.2 Cost-sensitive Feature Selection

In this section we present a novel linear formulation for SVM where classification costs are modeled via certain constraints, and where, in addition, a FS approach is embedded in such a way that only the relevant features are considered.

In order to cope with classification costs, first we recall some performance measures, namely, TPR, TNR and ACC as in (2.1).

The objective is to perform classification using a reduced set of features, in such a way that certain constraints over the performance, such as TPR $\geq p_{0(+1)}$ or TNR $\geq p_{0(-1)}$ (for threshold values $p_{0(+1)}$, $p_{0(-1)} \in [0,1]$), are fulfilled.

Note that the pair (X, Y) is a random vector with unknown distribution from which a sample $\{(x_i, y_i)\}_{i \in I}$ is generated. This implies that TPR and TNR should be estimated from sample data. This leads to the empirical constraints $\widehat{\text{TPR}} \geq p^*_{0(+1)}$ and $\widehat{\text{TNR}} \geq p^*_{0(-1)}$, for $p^*_{0(+1)} \geq p_{0(+1)}$ and $p^*_{0(-1)} \geq p_{0(-1)}$, where the performance measures are replaced by their sample estimates. Two possible choices, which shall be explored in this work, are

$$\begin{aligned} p^*_{0(+1)} &= p_{0(+1)} \\ &\text{and} \\ p^*_{0(-1)} &= p_{0(-1)}, \end{aligned} \tag{4.3}$$

or the more conservative approach based on Hoeffding inequality,

$$\begin{aligned} p^*_{0(+1)} &= p_{0(+1)} + \sqrt{\frac{-\log \alpha}{2|I_+|}} \\ &\text{and} \\ p^*_{0(-1)} &= p_{0(-1)} + \sqrt{\frac{-\log \alpha}{2|I_-|}}, \end{aligned} \tag{4.4}$$

where α is the significance level for the hypothesis test whose null hypothesis is either TPR $\leq p_{0(+1)}$ or TNR $\leq p_{0(-1)}$. See Chapter 2 for more details.

Note that it is straightforward to extend our results to the case in which measurement costs are associated with the features, as in e.g. Carrizosa et al. [2008], and then the minimum-cost feature set is sought instead.

4.2.1 The cost-sensitive FS procedure

Assume that we have a linear kernel, i.e., the kernel K in (GSVM(C_+, C_-)) is given by $K(x, x') = x^\top x'$, and thus the SVM with all features is obtained by solving (SVM(C_+, C_-)). We state the feature selection problem as a Mixed Integer Linear Program. Consider an auxiliary variable z_i that in case of being equal to 1, the instance i is counted as correctly classified. Hence, estimates of TPR and TNR from sample I have lower bounds $\widehat{\text{TPR}} \geq \sum_{i \in I} z_i(1+y_i)/\sum_{i \in I}(1+y_i)$ and $\widehat{\text{TNR}} \geq \sum_{i \in I} z_i(1-y_i)/\sum_{i \in I}(1-y_i)$, respectively. Associated with each feature k, $1 \leq k \leq N$, we define the variable δ_k taking the value 1 if feature k is selected for classifying, and 0 otherwise. Hence, the optimization problem that defines a linear classifier (hyperplane) taking into account the classification rates and in which a cost-based FS procedure is integrated is given by

$$
\begin{array}{lll}
\min_{\omega,\beta,\delta,z} & \sum_{k=1}^{N} c_k\delta_k & \\
s.t. & y_i(\omega^\top x_i+\beta)\geq 1-M_2(1-z_i), & \forall i\in I \\
& \sum_{i\in I} z_i(1-y_i)\geq p^*_{0(-1)}\sum_{i\in I}(1-y_i) & \\
& \sum_{i\in I} z_i(1+y_i)\geq p^*_{0(+1)}\sum_{i\in I}(1+y_i) & \\
& |\omega_k|\leq M_3\delta_k & \forall k\in 1,\ldots,N \\
& z_i\in\{0,1\} & \forall i\in I \\
& \delta_k\in\{0,1\} & \forall k\in 1,\ldots,N
\end{array}
\qquad (P1)
$$

where M_2 and M_3 are sufficiently large numbers. Also, c_k is the cost associated to the k-th feature, so we perform the FS by reducing the overall cost of the features. The case $c_k = 1 \quad \forall k$, is the standard FS in which the number of features selected is minimized.

Let us discuss the rationality of the formulation $(P1)$. The overall cost associated with the features used for classifying is to be minimized in the objective. The first constraint identifies which individuals are counted as correctly classified, since, as soon as $z_i = 1$, the score $f(x_i) = \omega^\top x_i + \beta$ is forced to be $f(x_i) \geq 1$ (if $y_i = +1$) or $f(x_i) \leq -1$ (if $y_i = -1$). Furthermore, the constant $\sum_{i\in I}(1 - y_i)$ is equal to two times the cardinality of the set $\{i \in I : y_i = -1\}$, whereas $\sum_{i\in I} z_i(1-y_i)$ yields two times the number of individuals counted as correctly classified in the class -1. Hence, the second and third constraints force respectively the fraction of individuals with label $y_i = -1$ (respectively, $y_i = +1$) counted as correctly classified to be at least $p^*_{0(-1)}$ (respectively, at least $p^*_{0(+1)}$). Finally, the fourth constraint forces to select those features k with $\delta_k = 1$. Note that, if very demanding classification rates are imposed, problem $(P1)$ may be infeasible. The solver will return this message, advising thus the user to lower the threshold values $p_{0(+1)}$, $p_{0(-1)}$.

Solving $(P1)$ identifies the features to be used in the classification. However, an SVM classifier has not been built yet, since the margin has not been maximized. The next section shall address such problem by using the SVM either with the linear kernel or with an arbitrary one.

We should stress that the feature selection is based on the linear kernel, yielding the tractable linear integer optimization problem $(P1)$. Extension of our FS approach to nonlinear kernels are formally straightforward, but the resulting nonconvex mixed integer nonlinear problems are not tractable, even for low dimensions. For this reason, we perform the FS by assuming a linear kernel, and then, once the features are selected, the classifier is built using an arbitrary kernel, as detailed in Section 4.2.2.

Of course more flexibility is gained if, in a preprocessing step, data x are embedded in a higher dimensional space through a nonlinear mapping ϕ, and thus the original x

is replaced by $\phi(x)$ in $(P1)$.

4.2.2 Cost-sensitive sparse SVMs: linear vs arbitrary kernels

Here we explain how the sparse SVM is built. Let us first consider the case of the classifier with linear kernel. Hence, the sparse SVM that controls the classification rates is formulated as

$$\begin{array}{lll} \min_{\omega,\beta,\xi} & \sum_{j=1}^{N} \omega_j^2 \delta_j + C \sum_{i\in I} \xi_i & \\ s.t. & y_i(\sum_{j=1}^{N} \omega_j \delta_j x_{ij} + \beta) \geq 1 - \xi_i, & \forall i \in I \\ & 0 \leq \xi_i \leq M_1(1 - z_i) & \forall i \in I \\ & z_i \in \{0,1\} & \forall i \in I \\ & \sum_{i\in I} z_i(1-y_i) \geq p^*_{0(-1)} \sum_{i\in I}(1-y_i) & \\ & \sum_{i\in I} z_i(1+y_i) \geq p^*_{0(+1)} \sum_{i\in I}(1+y_i). & \end{array} \qquad (P2)$$

Note that $(P2)$ is defined similarly as a standard linear SVM optimization problem. The slight difference is that in $(P2)$ only the variables selected by the FS approach described in Section 4.2.1. are considered. This means that the values of z in $(P2)$ are those obtained in problem $(P1)$. Note too that the constraints concerning the performance measures are also added here.

Now, assume the SVM classifier has the form

$$\Psi(x) = \begin{cases} +1, & \text{if } \omega^\top \phi(x) + \beta \geq 0 \\ -1, & \text{else,} \end{cases}$$

and an arbitrary kernel function $K(x, x') = \phi(x)^\top \phi(x')$ is used instead of the linear one. See e.g. Carrizosa and Romero Morales [2013]; Cristianini and Shawe-Taylor [2000]; Vapnik [1995]; Vapnik and Vapnik [1998] for details. Although formally similar, the case of an arbitrary kernel K implies that, if an FS procedure as $(P1)$ is desired, nonlinear constraints are involved and thus the optimization problem is much harder to solve. For this reason, instead of coping with such hard problem, we propose an alternative strategy: first, $(P1)$ is solved (as before), and then the SVM classifier (with the selected kernel) is built, using only the features selected in the problem described in Section 4.2.1. In what follows we focus on the radial kernel, even though one can consider any arbitrary kernel K. First, we define the binary variables δ identifying the features which are selected for classifying. The choice of the features, identified with the vector δ, leads to the kernel K_δ, defined as

$$K_\delta(x, x') = exp\left(-\gamma\left(\sum_{k=1}^{N} \delta_k (x^{(k)} - x'^{(k)})^2\right)\right),$$

where $x^{(k)}$ denotes the k-th component of vector x.

For δ (and thus K_δ) fixed, the aim is to solve (SVM(C_+, C_-)), but replacing the terms $\omega^\top\omega$ and $\omega^\top\phi(x_i)$, respectively, by the expressions $\sum_{i,j\in I}\lambda_i y_i \lambda_j y_i K(x_i, x_j)$ and $\sum_{i\in I}\lambda_i y_i K(x_i, x)$, apart from adding the constraints related to the performance measurements, as described in Benítez-Peña et al. [2019b]. Therefore, the cost-sensitive sparse SVM with an arbitrary kernel K is defined (once δ is fixed) as

$$
\begin{array}{lll}
\min_{\lambda,\xi,\beta,\mathbf{z}} & \sum_{i,j\in I}\lambda_i y_i \lambda_j y_j K_\delta(x_i, x_j) + C\sum_{i\in I}\xi_i & \\
s.t. & y_i(\sum_{j\in I}\lambda_j y_j K_z(x_i, x_j) + \beta) \geq 1 - \xi_i, & \forall i \in I \\
& 0 \leq \xi_i \leq M_1(1 - z_i) & \forall i \in I \\
& \sum_{i\in I}\lambda_i y_i = 0 & \\
& 0 \leq \lambda_i \leq C/2 & \forall i \in I \qquad (P3) \\
& \sum\limits_{i\in I} z_i(1 - y_i) \geq p^*_{0(-1)}\sum\limits_{i\in I}(1 - y_i) & \\
& \sum\limits_{i\in I} z_i(1 + y_i) \geq p^*_{0(+1)}\sum\limits_{i\in I}(1 + y_i) & \\
& z_i \in \{0, 1\} & \forall i \in I
\end{array}
$$

Let us discuss the formulation $(P3)$. The set of features is fixed through δ. The objective function, the first, third and fourth constraints are the usual ones in SVM. The second constraint together with the fifth, sixth and seventh constraints force some samples to be correctly classified, as in $(P1)$.

4.3 Experiment Description

In this section, a description of the experiments to be carried out in Section 4.4 of the cost-sensitive sparse SVM with linear kernel (problem $(P2)$) are compared to those under the radial kernel (problem $(P3)$), where, as described in the previous section, the variables z in both $(P2)$ and $(P3)$ are the solutions of the FS problem formulated by $(P1)$. Also, the solutions under the sparse methodology will be tested against the standard linear SVM. Although it would be natural to compare the solutions of $(P3)$ with the solutions of a standard radial SVM, this comparison is not straightforward since $(P1)$ may become infeasible when the performance measures obtained with the radial SVM are higher than those under the linear SVM. For simplicity we assume all measurement costs equal to 1, and then our aim is to minimize the number of features used.

Next, a description of how the experiments have been carried out is given. In order to solve problems $(P1)$, $(P2)$ and $(P3)$, the solver Gurobi, Gurobi Optimization [2016], and its Python language interface, Python Core Team [2015], are used. In order to estimate the performance of these FS procedures, a 10-fold cross-validation (CV),

Kohavi et al. [1995], is used, and out of samples accuracies are reported. However, for those datasets that have less than 100 instances, a Leave-One-Out procedure is carried out, in order to have a good size in the training sample. Also, depending on whether the linear or the radial kernel is considered, a parameter C or a pair of parameters (C, γ) must be tuned. Hence, in either the first or in the second case, $C \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}, \gamma \in \{2^{-5}, 2^{-4}, \ldots, 2^4, 2^5\}$ are considered. Problems in integer variables are hard to solve to optimality. However, excellent solutions are obtained in reasonable time. A time limit of 300 seconds is set, giving the solver enough time for finding (sub)optimal solutions. Parameters M_1, M_2 and M_3 are set as 100. Moreover, parameters tuning is done by another 10-fold CV (respectively, another Leave-One-Out), and the best set of parameters selected is the one with highest accuracy in average (or, in the case of imbalanced data, with the highest geometric mean between the TPR and the TNR).

For a better understanding, the whole procedure is summarized in Algorithm 3.

Algorithm 3: Pseudocode for general kernel approach.

for *kf = 1,...,folds* **do**
  Split data (D) into "*folds*" subsets, $D = \{D_1, \ldots, D_{folds}\}$
  Set $Validation = D_{kf}$ and set $I = D - \{D_{kf}\}$
  for *each pair* $(C, gamma)$ **do**
    for *kf2 = 1,..., folds2* **do**
      split $D' = D - \{D_{kf}\}$ into "*folds2*" subsets, $D' = \{D'_1, \ldots, D'_{folds2}\}$
      Set $Validation' = D'_{kf2}$ and set $I' = D' - \{D_{kf2}\}$
      Run ($P1$) over I, and select the relevant features.
      Run ($P2$) or ($P3$) over I with the corresponding modified kernel.
      Validate over $Validation'$, getting the accuracy ($acc[kf2]$)
    end
    Calculate the average accuracies $(\sum_{kf2} acc[kf2])/folds2$
    if $acc[kf2] \geq bestacc$ **then**
      Set $bestacc = acc[kf2]$, $bestgamma = gamma$ and $bestC = C$
    end
  end
  Run ($P1$) over I, and select the relevant features.
  Run ($P2$) or ($P3$) with the corresponding modified kernel and the parameters *bestgamma* and *bestC*, using I.
  Validate over $Validation$, getting the accuracy ($acc2[kf]$), and the correct classification probabilities ($TPR[kf]$, $TNR[kf]$) as well as the number of features selected $\Delta[kf] = \sum_{k=1}^{N} \delta[k]$.
end
Calculate and display the average performance measures: $(\sum_{kf} acc2[k2])/folds$, $(\sum_{kf} TPR[kf])/folds$, $(\sum_{kf} TNR[kf])/folds$ and $(\sum_{kf} \Delta[kf])/folds$

4.4 Numerical Results

Here, the experimental results are presented. We have chosen the datasets `wisconsin` (Breast Cancer Wisconsin (Diagnostic) Data Set), `votes` (Congressional Voting Records Data Set), `nursery` (Nursery Data Set), `Australian` (Statlog (Australian Credit Approval) Data Set), `careval` (Car Evaluation Data Set) and `gastrointestinal` (Gastrointestinal Lesions in Regular Colonoscopy Data Set), all well referenced and described with detail in Lichman [2013], and `leukemia` (Leukemia data), described in Golub et al. [1999a]. First, a brief data description is given in Section 4.4.1. Then, results under the linear kernel approach will be presented and discussed in Section 4.4.2. Finally, the case of the radial kernel will be analyzed in Section 4.4.3.

Note that the main idea of a FS approach is to reduce the number of features or, more generally, the overall associated costs, in such a way that the performance is not severely affected. As we can control the proportion of samples well classified, this is not a problematic issue. In fact, experiments are done so that new performance measurements will not be 0.025 points lower than the originals ones, i.e., those obtained under the standard version of the SVM with linear kernel. Using the notation as in Benítez-Peña et al. [2019b], TNR and TPR are the true negative and true positive rates, and TNR_0 and TPR_0 are their obtained values under the standard SVM with linear kernel on a validation sample, $\text{TNR} \geq p_{0(-1)} = \min\{1, \text{TNR}_0 - 0.025\}$ and $\text{TPR} \geq p_{0(+1)} = \min\{1, \text{TPR}_0 - 0.025\}$ are desired. For both linear and radial cases we have considered the two possible selection of the thresholds, defined by (4.3) and (4.4).

We stress that the purpose of this experimental section is to show how we can control TPR or TNR without a severe deterioration of overall classification rates, hopefully with a strong decrease in the number or cost of the features selected. This is the reason why we are comparing the performance of our approach with respect to the performance of the standard SVM. In a real application, the thresholds $p_{0(+1)}$, $p_{0(-1)}$ are to be given by the user, either based or not on SVM classification rates.

4.4.1 Data description

The performance of these novel approaches is illustrated using six real-life datasets from the UCI Repository, Lichman [2013], as well as the `leukemia` dataset, Golub et al. [1999a]. The positive label will be assigned to the majority class in 2-class datasets. In addition, multiclass datasets are transformed into 2-class ones, by giving positive label to the largest class and negative labels to the remaining samples. Categorical variables are transformed into dummy variables, i.e, if a categorical variable with ν levels is present, it will be replaced by $\nu - 1$ binary variables. Also, if there exist missing values, they are replaced by the median. A description of the datasets can be found in Table 4.1. Such table is split in 4 columns. The first shows the name of

the dataset (the actual names of the datasets are presented at the beginning of this section). The total number of samples of the dataset is given in the second column. The number of variables considered, and the number (and percentage) of positive samples in the dataset, are given in the last two columns.

Name	$\|\Omega_0\|$	$\|A\|$	$\|\Omega_{0+}\|$ (%)
`wisconsin`	569	30	357 (62.7 %)
`votes`	435	32	267 (61.4 %)
`nursery`	12960	19	4320 (33.3 %)
`Australian`	690	34	383 (55.5 %)
`careval`	1728	15	1210 (70.023 %)
`leukemia`	72	7128	47 (65.278 %)
`gastrointestinal`	76	698	55 (72.368 %)

Table 4.1: Details concerning the implementation of the CSVM for the considered datasets.

4.4.2 Results under the cost-sensitive sparse SVM with linear kernel

Two types of results will be shown, corresponding to the choices (4.3) and (4.4) of the thresholds. As a summary, it will turn out that (4.3) yields sparser classifiers, while (4.4), which is a more conservative choice, usually yields less sparse classifiers but with better accuracies.

The choice of threshold parameters in (4.3) leads to results summarized in Table 4.2. The first column of Table 4.2 gives the name of the dataset used (the abbreviation we have chosen for the dataset). Then, the second and third columns show, respectively, the performance measures for the standard SVM (using the linear kernel) and the proposed cost-sensitive sparse methodology. Such columns are split into two subcolumns: the first one shows the average values and the second one the standard deviations. The last column reports the feature reduction, by indicating the original and selected (average) number of variables.

From the table, it can be concluded that the approach with a linear kernel works well in general. In the case of `wisconsin`, the TPR has desirable values, since it only differentiates -0.019 points from the original. However, in the case of the accuracy and TNR, the loss is bigger than 0.025 points. This is due mainly to two reasons: first, the constraints are imposed on the training sample, while the performance is calculated using a test sample. Second, since the thresholds are considered as $p^*_{0(+1)} = p_{0(+1)}$, $p^*_{0(-1)} = p_{0(-1)}$, this implies we are not much restrictive as if $p^*_{0(+1)} > p_{0(+1)}$ ($p^*_{0(-1)} > p_{0(-1)}$) were required. Nevertheless, the new TNR value is only 0.038 points smaller than the original, and the reduction of features is significant since only two variables out of 30 are used. Also, in `votes` the features are significantly reduced and the most affected performance measure is the TPR, which decreases 0.027 points, making the

Name		SVM		FS		Feature reduction
		Mean	Std	Mean	Std	
wisconsin	Acc	0.975	0.021	0.947	0.025	30 → 2 (0 Std)
	TPR	0.992	0.013	0.973	0.031	
	TNR	0.943	0.051	0.905	0.063	
votes	Acc	0.954	0.033	0.949	0.036	32 → 2 (0 Std)
	TPR	0.955	0.038	0.928	0.059	
	TNR	0.947	0.059	0.979	0.036	
nursery	Acc	1	0	1	0	19 → 1 (0 Std)
	TPR	1	0	1	0	
	TNR	1	0	1	0	
Australian	Acc	0.848	0.051	0.855	0.057	34 → 1 (0 Std)
	TPR	0.798	0.083	0.801	0.087	
	TNR	0.912	0.05	0.926	0.041	
careval	Acc	0.956	0.017	0.946	0.019	15 → 9 (0 Std)
	TPR	0.96	0.022	0.963	0.017	
	TNR	0.948	0.024	0.907	0.04	
leukemia	Acc	0.972	0.164	0.875	0.331	7128 → 3.139 (1.205 Std)
	TPR	0.979	0.196	0.896	0.305	
	TNR	0.96	0.144	0.833	0.373	
gastrointestinal	Acc	0.895	0.307	0.829	0.379	698 → 1 (0 Std)
	TPR	0.929	0.258	0.839	0.367	
	TNR	0.8	0.4	0.8	0.4	

Table 4.2: Performance measures under the cost-sensitive sparse SVM with linear kernel and $p^*_{0(+1)} = p_{0(+1)}$, $p^*_{0(-1)} = p_{0(-1)}$.

Name		SVM		FS		Feature reduction
		Mean	Std	Mean	Std	
wisconsin	Acc	0.975	0.021	0.965	0.023	$30 \rightarrow 6.2$ (0.919 Std)
	TPR	0.992	0.013	0.975	0.023	
	TNR	0.943	0.051	0.947	0.048	
votes	Acc	0.954	0.033	0.954	0.033	$32 \rightarrow 9.3$ (1.16 Std)
	TPR	0.955	0.038	0.96	0.034	
	TNR	0.947	0.059	0.945	0.052	
nursery	Acc	1	0	1	0	$19 \rightarrow 1$ (0 Std)
	TPR	1	0	1	0	
	TNR	1	0	1	0	
Australian	Acc	0.848	0.051	0.837	0.057	$34 \rightarrow 5.5$ (1.78 Std)
	TPR	0.769	0.083	0.772	0.074	
	TNR	0.912	0.05	0.924	0.053	
careval	Acc	0.956	0.017	0.954	0.018	$15 \rightarrow 11$ (0 Std)
	TPR	0.96	0.022	0.962	0.018	
	TNR	0.948	0.024	0.935	0.039	
leukemia	Acc	0.972	0.164	0.944	0.229	$7128 \rightarrow 2$ (0 Std)
	TPR	0.979	0.196	0.957	0.202	
	TNR	0.96	0.144	0.92	0.272	
gastrointestinal	Acc	0.895	0.307	0.842	0.365	$698 \rightarrow 3.105$ (0.552 Std)
	TPR	0.929	0.258	0.927	0.26	
	TNR	0.8	0.4	0.619	0.486	

Table 4.3: Performance measures under the cost-sensitive sparse SVM with linear kernel and $p^*_{0(+1)} = p_{0(+1)} + \sqrt{-\log\alpha/(2|I_1|)}$, $p^*_{0(-1)} = p_{0(-1)} + \sqrt{-\log\alpha/(2|I_{-1}|)}$.

accuracy smaller. However, the value on the TNR is increased. As happened with `wisconsin`, the loss is due mainly to the two facts previously mentioned. For `nursery`, an amazing reduction to only one feature is achieved, in addition getting a perfect classification. This is explained as follows. As commented in Section 4.4.1, multiclass datasets are transformed into 2-class ones, and this is the case, obtaining the classes "`not_recom`" and "`others`", which are the positive and negative classes, respectively. In addition, one of the (categorical) features in the data (which is the one selected by our procedure) completely determines the class. In `Australian`, the total number of variables is also reduced to only one, having similar performance measures values as in the standard SVM. In fact, we obtain here even better results than under the original linear SVM. If the variable selected with the algorithm is studied, one can observe that it is a binary variable X, where the contingency table together with the class variable is given in Table 4.4. Hence this variable is by itself a good predictor, as the FS procedure pointed out. In the case of `careval`, we got the smallest reduction in the number of variables selected, maintaining the performance measures values above the imposed thresholds. On the other hand, in the case of `leukemia`, the number of variables is significantly reduced. However, since the number of instances is small, the performance measurements are affected by this reduction of features. Also, for `gastrointestinal`, the results are similar to what happened for `leukemia`, but the TNR has not been affected at all.

	X = 0	X = 1
Class +	306	77
Class −	23	284

Table 4.4: Contingency table of the feature selected in `Australian`.

Consider next the results shown by Table 4.3, for the case where we are restrictive regarding the performance values, that is, when $p^*_{0(+1)} = p_{0(+1)} + \sqrt{-\log\alpha/(2|I_1|)}$ and $p^*_{0(-1)} = p_{0(-1)} + \sqrt{-\log\alpha/(2|I_{-1}|)}$. From the table, it can be seen how this approach tends to work better concerning the performance measures, but achieves less sparse solutions. For example, if we focus on `wisconsin`, the TNR, the TPR and the accuracy as well, obtain the desired performance requirements. However, only a reduction of variables of one fifth is obtained. In the case of `votes`, an analogous result is obtained for the performance measures and only a reduction in one third of the variables is achieved. The same pattern as before is observed for `nursery`. For `Australian`, we obtain even an improvement in all the three performance measures considered, reducing the number of features to one fifth. In addition, we get again in `careval` the smallest reduction in the number of variables selected, maintaining the performance measures values above the thresholds imposed as before, but using a larger number of features. On the contrary, and surprisingly, we have obtained for `leukemia`

even a bigger reduction in the number of features and better results using Hoeffding inequality. However, `gastrointestinal` dataset goes with the flow and the number of features is increased when using the mentioned inequality. Nevertheless, the TPR has not been affected now whereas the TNR has significantly decreased.

4.4.3 Results under the cost-sensitive sparse SVM with radial kernel

The analogous results to those in Section 4.4.2 are presented here, for the case of the radial kernel. However, only `wisconsin`, `votes` and `Australian` datasets are used here. As shown by Tables 4.5 and 4.6 and similarly as occurred in Section 4.4.2, the use of the threshold values obtained by the Hoeffding inequality (as in (4.4)) tends to yield a lower level of sparsity, but also, a higher predictive power in general (particularly, when achieving the desired bounds). Concerning the performance measures, it can be deduced from Tables 4.5 and 4.6 that this approach works well in general, especially when using Hoeffding. Finally, it should be noted how the reduction in the number of features is quite notable for some datasets, as before.

Name		SVM		FS		Feature reduction
		Mean	Std	Mean	Std	
`wisconsin`	`Acc`	`0.975`	`0.021`	`0.956`	`0.012`	$30 \rightarrow 2$ (0 Std)
	`TPR`	`0.992`	`0.013`	`0.988`	`0.016`	
	`TNR`	`0.943`	`0.051`	`0.893`	`0.051`	
`votes`	`Acc`	`0.954`	`0.033`	`0.947`	`0.034`	$32 \rightarrow 2$ (0 Std)
	`TPR`	`0.955`	`0.038`	`0.928`	`0.059`	
	`TNR`	`0.947`	`0.059`	`0.974`	`0.036`	
`nursery`	`Acc`	`1`	`0`	`1`	`0`	$19 \rightarrow 1$ (0 Std)
	`TPR`	`1`	`0`	`1`	`0`	
	`TNR`	`1`	`0`	`1`	`0`	

Table 4.5: Performance measures under the cost-sensitive sparse SVM with radial kernel and $p^*_{0(+1)} = p_{0(+1)}$, $p^*_{0(-1)} = p_{0(-1)}$.

Name		SVM		FS		Feature reduction
		Mean	Std	Mean	Std	
wisconsin	Acc	0.975	0.021	0.947	0.03	$30 \to 6.2$ (0.919 Std)
	TPR	0.992	0.013	0.967	0.039	
	TNR	0.943	0.051	0.907	0.02	
votes	Acc	0.954	0.033	0.949	0.03	$32 \to 9.3$ (1.16 Std)
	TPR	0.955	0.038	0.959	0.034	
	TNR	0.947	0.059	0.939	0.043	
nursery	Acc	1	0	1	0	$19 \to 1$ (0 Std)
	TPR	1	0	1	0	
	TNR	1	0	1	0	

Table 4.6: Performance measures under the cost-sensitive sparse SVM with radial kernel and $p^*_{0(+1)} = p_{0(+1)} + \sqrt{-\log\alpha/(2|I_1|)}$, $p^*_{0(-1)} = p_{0(-1)} + \sqrt{-\log\alpha/(2|I_{-1}|)}$.

4.4.4 Comparison with other methodologies

The cost-sensitive FS procedure presented here can be compared in a certain way with some other benchmark methodologies. However, the authors are not aware of FS methods for SVM controlling, as we do, TPR or TNR. Among the different FS techniques that can be applied to SVM we can find, for example, the following ones: **Filter methods** (they are based on measures like Pearson's Correlation, Linear Discriminant Analysis or Chi-Square), **Wrapper methods** (Forward Selection, Backward Elimination, Recursive Feature elimination, ...) and **Embedded methods** (such as the presented in the Introduction Section).

In order to make a comparison with another state-of-the-art method, we have selected the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018]. The results can be seen in Tables 4.7 and 4.8. In Table 4.7 we can see the results for the standard SVM, the results of our FS approach when $p^*_{0(+1)} = p_{0(+1)}$ and $p^*_{0(-1)} = p_{0(-1)}$, and the results of the state-of-the-art method when the maximum number of features selected is the same as the obtained for our methodology. In Table 4.8, the results for the standard SVM are reported together with the results of our FS approach when $p^*_{0(+1)} = p_{0(+1)} + \sqrt{-\log\alpha/(2|I_1|)}$ and $p^*_{0(-1)} = p_{0(-1)} + \sqrt{-\log\alpha/(2|I_{-1}|)}$, as well as the results of the state-of-the-art method when the maximum number of features selected is the same as the obtained with our methodology.

We can observe that, except for gastrointestinal dataset (when using Hoeffding inequality), where we obtain better results than the comparative, similar results are

Name		SVM		FS		Compar.	
		Mean	Std	Mean	Std	Mean	Std
wisconsin	Acc	0.975	0.021	0.947	0.025	0.954	0.021
	TPR	0.992	0.013	0.973	0.031	0.977	0.025
	TNR	0.943	0.051	0.905	0.063	0.911	0.056
votes	Acc	0.954	0.033	0.949	0.036	0.956	0.026
	TPR	0.955	0.038	0.928	0.059	0.949	0.039
	TNR	0.947	0.059	0.979	0.036	0.969	0.034
nursery	Acc	1	0	1	0	1	0
	TPR	1	0	1	0	1	0
	TNR	1	0	1	0	1	0
Australian	Acc	0.848	0.051	0.855	0.057	0.855	0.054
	TPR	0.798	0.083	0.801	0.087	0.801	0.082
	TNR	0.912	0.05	0.926	0.041	0.925	0.039
careval	Acc	0.956	0.017	0.946	0.019	0.949	0.016
	TPR	0.96	0.022	0.963	0.017	0.967	0.012
	TNR	0.948	0.024	0.907	0.04	0.91	0.043
leukemia	Acc	0.972	0.164	0.875	0.331	0.653	0.471
	TPR	0.979	0.196	0.896	0.305	0.66	0.474
	TNR	0.96	0.144	0.833	0.373	0.68	0.466
gastrointestinal	Acc	0.895	0.307	0.829	0.379	0.857	0.35
	TPR	0.929	0.258	0.839	0.367	0.9	0.3
	TNR	0.8	0.4	0.8	0.4	0.75	0.433

Table 4.7: Performance measures under the cost-sensitive sparse SVM with linear kernel and $p^*_{0(+1)} = p_{0(+1)}$, $p^*_{0(-1)} = p_{0(-1)}$ and comparative with the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018].

Name		SVM		FS		Compar.	
		Mean	Std	Mean	Std	Mean	Std
wisconsin	Acc	0.975	0.021	0.965	0.023	0.967	0.018
	TPR	0.992	0.013	0.975	0.023	0.989	0.017
	TNR	0.943	0.051	0.947	0.048	0.926	0.033
votes	Acc	0.954	0.033	0.954	0.033	0.954	0.033
	TPR	0.955	0.038	0.96	0.034	0.948	0.036
	TNR	0.947	0.059	0.945	0.052	0.961	0.035
nursery	Acc	1	0	1	0	1	0
	TPR	1	0	1	0	1	0
	TNR	1	0	1	0	1	0
Australian	Acc	0.848	0.051	0.837	0.057	0.851	0.053
	TPR	0.798	0.083	0.772	0.074	0.798	0.081
	TNR	0.912	0.05	0.924	0.053	0.919	0.046
careval	Acc	0.956	0.017	0.954	0.018	0.954	0.017
	TPR	0.96	0.022	0.962	0.018	0.97	0.016
	TNR	0.948	0.024	0.935	0.039	0.917	0.027
leukemia	Acc	0.972	0.164	0.944	0.229	0.932	0.252
	TPR	0.979	0.196	0.957	0.202	0.938	0.242
	TNR	0.96	0.144	0.92	0.272	0.917	0.276
gastrointestinal	Acc	0.895	0.307	0.842	0.365	0.714	0.452
	TPR	0.929	0.258	0.927	0.26	0.75	0.433
	TNR	0.8	0.4	0.619	0.486	0.625	0.484

Table 4.8: Performance measures under the cost-sensitive sparse SVM with linear kernel and $p^*_{0(+1)} = p_{0(+1)} + \sqrt{-\log\alpha/(2|I_1|)}$, $p^*_{0(-1)} = p_{0(-1)} + \sqrt{-\log\alpha/(2|I_{-1}|)}$ and comparative with the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018].

obtained for our method and the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018] in terms of accuracy, while our methodology is cost-sensitive and we can control the performance measurements. As an illustration, in Table 4.9 we have collected all the results for dataset `australian` when applying the method in Chan et al. [2007], Ghaddar and Naoum-Sawaya [2018] and varying the number of features from 1 (minimum) to 34 (maximum). There, we can see how the maximum TPR obtained is 0.8007, so with our methodology, and maybe at the expense of increasing misclassification rates in the another class, we can improve the accuracy rates in the target class. The results obtained when either TPR or TNR are varied are summarized in Table 4.10.

Australian							
# Feat.	Acc	TPR	TNR	# Feat.	Acc	TPR	TNR
1	0.8551	0.8007	0.9248	18	0.8464	0.7954	0.9121
2	0.8551	0.8007	0.9248	19	0.8464	0.7954	0.9121
3	0.8551	0.8007	0.9248	20	0.8464	0.7954	0.9121
4	0.8551	0.8007	0.9248	21	0.8464	0.7954	0.9121
5	0.8507	0.798	0.9186	22	0.8578	0.7953	0.9154
6	0.8507	0.798	0.9186	23	0.8464	0.7954	0.9121
7	0.8507	0.798	0.9186	24	0.8464	0.7954	0.9121
8	0.8507	0.798	0.9186	25	0.8464	0.7954	0.9121
9	0.8507	0.798	0.9186	26	0.8449	0.7929	0.9121
10	0.8507	0.798	0.9186	27	0.8478	0.7981	0.9121
11	0.8507	0.798	0.9186	28	0.8478	0.7954	0.9153
12	0.8478	0.798	0.9121	29	0.8464	0.7927	0.9153
13	0.8478	0.798	0.9121	30	0.8493	0.798	0.9153
14	0.8478	0.798	0.9121	31	0.8478	0.7954	0.9153
15	0.8478	0.798	0.9121	32	0.8478	0.7981	0.9121
16	0.8464	0.7954	0.9121	33	0.8478	0.7981	0.9121
17	0.8478	0.798	0.9121	34	0.8478	0.7981	0.9121

Table 4.9: Performance measures using the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018], varying the maximum number of features from 1 (minimum) to 34 (maximum) in `Australian` dataset.

From these experimental results we can conclude that, indeed, our method is able not only to reduce the number of features but it also controls the performance measures. If we observe the cases where $(p^*_{0(+1)},p^*_{0(-1)})$ is (0.85,0.5) or (0.85,0.55), we see how we have strongly increased the value in the TPR, although the TNR has decreased a lot. A similar behavior is observed for the pair (0.9,0.5). However, when using (0.85,0.575)

Australian					
$p^*_{0(+1)}$	$p^*_{0(-1)}$	Acc	TPR	TNR	Aver. # Feat. Selected
0.85	0.5	0.738	0.94	0.484	1
0.85	0.55	0.738	0.94	0.484	1
0.85	0.575	0.812	0.854	0.754	1.7
0.85	0.6	0.855	0.801	0.92	2
0.9	0.5	0.757	0.896	0.582	1.333

Table 4.10: Performance measures using the method in Chan et al. [2007],Ghaddar and Naoum-Sawaya [2018], varying the maximum number of features from 1 (minimum) to 34 (maximum) in `Australian` dataset.

a different trade-off is found, whereas for (0.85,0.6) we recover the original results.

4.5 Chapter Summary

In this chapter we have proposed a Feature Selection procedure for Support Vector Machines that yields a novel, sparse, SVM. Contrary to existing Feature Selection approaches, we take explicitly into account that misclassification costs may be rather different in the two groups, and thus, instead of seeking the classifier maximizing the margin, we seek the most sparse classifier that attains certain true positive and true negative rates on the dataset. For both SVM with linear and radial kernel, the problem is written in a straightforward manner, solving first a mixed integer linear problem and then their standard SVM formulations, considering only the features obtained in the first problem as well as the performance constraints. The reported numerical results show that the novel approaches lead to comparable or better performance rates, in addition to an important reduction in the number of variables.

Chapter 5

Feature selection for benchmarking via DEA formulation

In this chapter we propose an integrative approach to feature (input and output) selection in Data Envelopment Analysis (DEA). Here, the DEA model is enriched with zero-one decision variables modelling the selection of features, yielding a Mixed Integer Linear Programming formulation. This single-model approach can handle different objective functions as well as constraints to incorporate desirable properties from the real-world application. Our approach is illustrated on the benchmarking of electricity Distribution System Operators (DSOs). The numerical results highlight the advantages of our single-model approach provide to the user, in terms of making the choice of the number of features, as well as modeling their costs and their nature.

5.1 Introduction

Within benchmarking, Data Envelopment Analysis (DEA) [Charnes et al., 1978] is one of the most widely used tools, Cook et al. [2019]; Emrouznejad and Yang [2018]; Jiang and Lin [2015]; Landete et al. [2017]; Li et al. [2017b]; Petersen [2018]; Ruiz and Sirvent [2016, 2019]. It aims at benchmarking the performance of decision marking units (DMUs), which use the same types of inputs and produce the same types of outputs, against each other. DEA calculates an efficiency score for each of the DMUs, so that DMUs with a score equal to one are in the so-called efficient frontier. DMUs outside the efficient frontier are deemed as underperforming, and a further analysis gives insights as to what they can do to improve their efficiency. The efficiency of DMUs in DEA is measured as the weighted summation of the outputs divided by the weighted summation of the inputs, and the weights are found solving a Linear Programming problem for each DMU. DEA model specification, in the form of feature (where the term feature is used to refer to either outputs, inputs or environmental variables) selection, has a significant impact on the shape of the efficient frontier in DEA as well as the insights given to the inefficient DMUs [Golany and Roll, 1989]. Moreover, it is known to improve the discriminatory power of DEA models [Bogetoft and Otto, 2010]. This chapter proposes and investigates a mathematical optimization approach for feature selection in DEA.

In benchmarking projects, as in most applied statistical analyses, one of the most challenging tasks is the choice of the DEA model specification. First, a good model should make conceptual sense not only from the theoretical but also from a practical point of view. The interpretation must be easy to understand and the properties of the model must be natural. This contributes to the acceptance of the model by stakeholders and provides a safeguard against spurious models developed without much understanding of the industry. More precisely, this has to do with the choice of outputs in DEA that are natural cost drivers and with functional forms that, for example, have reasonable returns to scale and curvature properties. Second, it is important to

guide the search for a good model with classical statistical tests. We typically seek models that have significant features with the right signs and that do not leave a large unexplained variation. Third, intuition and experience is a less stringent but important safeguard against false model specifications and the over- or underuse of data to draw false conclusions. It is important that the models produce results that are not that different from the results one would have found in other data sets, e.g., from other countries or related industries. The intuition and experience must be used with caution. We may screen away extraordinary but true results (Type 1 error) and we may go for a more common set of results based on false models (Type 2 error). One aspect of this is that one will tend to be more confident in a specification of inputs and outputs that leads to comparable results in alternative estimation approaches, e.g., in the DEA and Stochastic Frontier Analysis models. Finally, the choice of model specification has to be pragmatic. We need to take into account the availability of data as well as what the model is going to be used for. In benchmarking, it matters if the model is used to learn best practices, to reallocate resources between entities or to directly incentivize firms or managers by performance based payment schemes. Our approach gives a tool that can support the selection of features in benchmarking, allowing the user to navigate through a large amount of DEA models and a large amount of constraints modeling knowledge in the form of intuition and experience, in an efficient manner.

The complexity of the model specification phase partially explains the lack of enough guidance in the literature at this respect, Cook et al. [2014]; Luo et al. [2012]; Soleimani-Damaneh and Zarepisheh [2009], and most of the effort goes into the analysis and interpretation of a given DEA model. With the strand of literature on feature selection, the most common approach is to use a priori rules based on Statistical Analysis (such as correlations, dimensionality reduction techniques, and regression), and Information Theory (such as AIC or Shannon entropy). Alternatively, an ex-post analysis of the sensitivity of the efficient frontier to additional features can be run to detect whether relevant features have been left out. See Adler and Yazhemsky [2010]; Fernandez-Palacin et al. [2018]; Li et al. [2017a]; Nataraja and Johnson [2011]; Pastor et al. [2002]; Sirvent et al. [2005]; Soleimani-Damaneh and Zarepisheh [2009]; Wagner and Shimshak [2007], and references therein. Recently, there have been attempts to use LASSO techniques from Statistical Learning to build sparse benchmarking models, i.e., models using just a few features, J.-Y. Cai [2016]; Lee and Cai [2020]; Qin and Song [2014].

In this chapter, the DEA Linear Programming formulation is enriched with zero-one decision variables modelling the selection of features for different objective functions, such as the average efficiency or the squared distance to the ideal point where all DMUs are efficient, and for different set of constraints that incorporate knowledge from the industrial application, such as bounds on the weights as well as costs on the features. This yields either a Mixed Integer Linear Programming (MILP) problem, or a Mixed

Integer Quadratic Programming (MIQP) one. Thus, in contrast to the existing literature, that tends to combine statistical analysis with the mathematical programming based DEA, we propose an approach that is entirely driven by mathematical optimization. We illustrate our models in the benchmarking of electricity Distribution System Operators (DSOs), where there is a pool of 100 potential outputs.

The contributions of our approach are threefold. First, our single-model mathematical approach can guide better the selection of features: it controls directly the number of chosen features, as opposed to techniques based on seeking sparsity, being thus able to quantify the added value of additional features; works directly with the original features, as opposed to dimensionality reduction techniques, which create artificial features that are difficult to interpret; and can derive a collection of models by shaping in alternative ways the distribution of the efficiencies, using different objective functions that focus on different groups of DMUs, which can be combined through, for instance, Shannon entropy Soleimani-Damaneh and Zarepisheh [2009]. Second, while the previous literature has focused on the choice of variables from a small set of candidates, e.g., Lee and Cai [2020], in the era of Big Data, the set of alternatives to choose from is expanding at a fast pace, and the challenge is often not the lack of data, but the abundance of data, Zhu et al. [2018]. In the numerical section, we show how our MILP/MIQP approach is able to make the selection from a large pool of outputs. Third, we introduce an element of game theory when selecting features. In applied projects, the evaluated DMUs will typically try to influence the feature selection since this will affect how one firm is evaluated relative to others. It is therefore important to think about the conflict the DMUs (the players of the game) when choosing the set of features (the strategies of the game) used in the calculation of the efficiencies (outcome). We illustrate the results for a simpler game setting where the strategies are derived from the individual and the joint models.

The reminder of the chapter is structured as follows. In Section 5.2, we introduce the individual feature selection problem where the selection is tailored to a given DMU. In Section 5.3, we introduce the joint feature selection problem where the selection is imposed to be the same for all DMUs. Section 5.4 is devoted to the illustration of our models in the benchmarking of electricity Distribution System Operators (DSOs). We end the chapter in Section 5.5 with some conclusions.

5.2 The individual selection model

In this section, and for an individual DMU, we propose a Mixed Integer Linear Programming (MILP) formulation to select outputs in Data Envelopment Analysis (DEA).

Consider $|\Omega_0| = K$ DMUs (indexed by k), using N inputs (indexed by i), and producing M outputs (indexed by m). DMU k uses vector of inputs $\mathbf{x}^{(k)} \in \mathbb{R}^N_+$ to produce

vector of outputs $\mathbf{y}^{(k)} \in \mathbb{R}_+^M$. Let $E^{(k)}$ be the so-called Farrell input-oriented efficiency of DMU k, which is the optimal solution value to a DEA model. Our goal is to select the p outputs from the M potential ones that yield the maximal efficiency for DMU k. We first start with the classical formulation of the problem which solves a Linear Programming model, and subsequently include the output selection decision variables. The output selection model for DMU k is enriched with input selection decision variables, as well as constraints modeling desirable properties about the selected features. Please note that our approach can easily be extended to the use of other efficiency measures, including the output-based Farrell efficiency.

5.2.1 The selection model for a DMU

We start with the formulation of the classical DEA model, in which we can make use of the M outputs available. The input-oriented efficiency of DMU k, $E^{(k)}$, in a DEA model with constant returns to scale (CRS) is equal to the optimal solution value of problem (DEA$^{(k)}$) in Chapter 1.

We continue with the model in which p outputs are to be selected from the M available ones such that the efficiency of DMU k is maximized. Let $z_o^{(k)}$ be equal to 1 if output o can be used in the calculation of the efficiency of DMU k, and 0 otherwise. Let $E^{(k)}(\mathbf{z}^{(k)})$ denote the corresponding efficiency. The decision variables $\beta_o^{(k)}$ and $\alpha_i^{(k)}$ are defined as above. The Output Selection for DMU k, where p outputs must be selected such that $E^{(k)}(\mathbf{z}^{(k)})$ is maximized, can be written as the following MILP:

$$v^{(k)}(p) := \text{maximize}_{(\boldsymbol{\alpha}^{(k)},\boldsymbol{\beta}^{(k)},\mathbf{z}^{(k)})} \sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)} \tag{5.1}$$

s.t. (OSDEA$^{(k)}(p)$)

$$(1.2) - (1.5)$$

$$\beta_o^{(k)} \leq M_1 z_o^{(k)} \qquad \forall o = 1, \ldots, M \tag{5.2}$$

$$\sum_{o=1}^{M} z_o^{(k)} = p \tag{5.3}$$

$$z_o^{(k)} \in \{0, 1\} \qquad \forall o = 1, \ldots, M, \tag{5.4}$$

where M_1 is a big constant. Constraints (1.2)-(1.5) were already present in the classical DEA model. Constraints (5.2) make sure that the selection variables $z_o^{(k)}$ are well defined: if $z_o^{(k)}$ equals 0, then $\beta_o^{(k)}$ equals 0 too. Constraint (5.3) models the number of features to be selected. Finally, constraints (5.4) relate to the range of decision variables $z_o^{(k)}$. (OSDEA$^{(k)}(p)$) has $K + M + 2$ linear constraints and $N + 2\,M$ variables, where $N + M$ are continuous and M are binary ones. Our numerical experiments show

that this problem can be solved efficiently, although the solution time is affected by the value of the M_1 constant. The value of M_1, and thus the computational burden of the problem, can be reduced using an upper bound on the weight associated with output o, for each $o = 1, \ldots, M$. It is not difficult to see that, without loss of optimality, $\beta_o^{(k)} = 0$ if $y_o^{(k)} = 0$, and thus $z_o^{(k)} = 0$. Otherwise, $\beta_o^{(k)} \leq \frac{1}{y_o^{(k)}}$, by combining (1.2) and (1.3). Thus, constraints (5.2) can be tighten to

$$\begin{aligned} \beta_o^{(k)} &= 0 \qquad \forall o = 1, \ldots, M, \text{such that } y_o^{(k)} = 0 \\ \beta_o^{(k)} &\leq \frac{1}{y_o^{(k)}} \, z_o^{(k)} \qquad \forall o = 1, \ldots, M, \text{such that } y_o^{(k)} > 0. \end{aligned}$$

Let $z^{(k)}(p)$ denote the optimal selection variables to (OSDEA$^{(k)}(p)$), i.e., the p outputs that yield the maximum efficiency for DMU k. Thus, the optimal solution value to (OSDEA$^{(k)}(p)$), denoted above by $v^{(k)}(p)$, is equal to $E^{(k)}(z^{(k)}(p))$. A few things are known about the maximum efficiency $v^{(k)}(p)$ as function of p. The efficiency $v^{(k)}(p)$ is non decreasing in p, i.e., the more outputs we select the better the efficiency of DMU k can be. Moreover, in the extreme case when all outputs are considered, we have that $v^{(k)}(M) = E^{(k)}$. Thus, a plausible strategy to choose the value of p is to look at the marginal contribution of an additional feature, i.e., $v^{(k)}(p+1) - v^{(k)}(p)$, and stop when this is below a threshold.

5.2.2 Extensions

In this section we discuss several interesting extensions that can be carried out using the previous model as a basis. First, we present the formulation when both inputs and outputs are to be selected, all at once. Second, we model constraints on the weights attached to the outputs. Finally, we discuss how other attributes attached to the outputs, such as costs and correlations, may constrain the feature selection.

Selection of inputs and outputs

Note that up to now, and for the sake of clarity, we have focused on the selection of outputs. The selection of $\tilde{p}$ inputs from the N potential ones can be included in a similar fashion. Indeed, let us consider the new binary variables $\tilde{z}_i^{(k)}$, equal to 1 if input i can be used in the calculation of the efficiency for DMU k, and 0 otherwise. Hence, the Feature Selection for DMU k, (FSDEA$^{(k)}(p)$), where $\tilde{p}$ inputs and p outputs are selected, can be written also as an MILP

$$\text{maximize}_{(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{z}^{(k)}, \tilde{\mathbf{z}}^{(k)})} \sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)} \tag{5.5}$$

s.t. (FSDEA$^{(k)}(p)$)

$$(1.2)-(1.5),(5.2)-(5.4)$$

$$\alpha_i^{(k)} \leq \tilde{M}_1 \tilde{z}_i^{(k)} \qquad \forall i = 1,\dots,N \tag{5.6}$$

$$\sum_{i=1}^{N} \tilde{z}_i^{(k)} = \tilde{p} \tag{5.7}$$

$$\tilde{z}_i^{(k)} \in \{0,1\} \qquad \forall i = 1,\dots,N, \tag{5.8}$$

where $\tilde{M}_1$ is another big constant. Constraints (5.6)–(5.8) are the counterparts of (5.2)–(5.4) but modelling the selection of inputs instead of outputs. (FSDEA$^{(k)}(p)$) has $K+M+N+3$ linear constraints and $2\,N+2\,M$ variables, where half of them are continuous and the other half binary. Although running times are not an issue for this model, one can lower them even further by finding tighter values of M_1 and $\tilde{M}_1$. As above, this can be done using bounds on the inputs and the outputs.

Modeling constraints on weights

Our (OSDEA$^{(k)}(p)$) improves the discriminatory power of the DEA model by focusing on a few outputs, and eliminating the rest from the calculation of the efficiency of DMU k. There is a strand of literature that, using also as a basis the discriminatory power, argue the necessity of controlling the values of the weights [Allen et al., 1997; Green et al., 1996; Joro and Korhonen, 2015; Podinovski, 2016; Ramón et al., 2010; Sexton et al., 1986]. In these works, it is assumed that we have upper and lower bounds on the weight $\beta_o^{(k)}$, say, $L_o^{(k)}$ and $U_o^{(k)}$, for $o = 1,\dots,M$, i.e.,

$$L_o^{(k)} \leq \beta_o^{(k)} \leq U_o^{(k)} \quad \forall o = 1,\dots,M. \tag{5.9}$$

Gathering non trivial values for these bounds is not a straightforward task for the user in the presence of many outputs, as in dataset on benchmarking of electricity DSOs in our numerical section. In any case, we can enrich our (OSDEA$^{(k)}(p)$), to not only control whether an output can be used, but also the range of values for the corresponding weight. These bounds can be incorporated in constraints (5.2) in (OSDEA$^{(k)}(p)$) yielding

$$L_o^{(k)} z_o^{(k)} \leq \beta_o^{(k)} \leq U_o^{(k)} z_o^{(k)} \quad \forall o = 1,\dots,M. \tag{5.7'}$$

There are a few observations to be made. First, the knowledge of upper bounds on the weights naturally tightens the value of M_1. Second, if there are meaningful lower bounds on the weights, i.e., if $L_o^{(k)} > 0$, then $z_o^{(k)}$ must be equal to 1. Third, these positive lower bounds make the selection problem (OSDEA$^{(k)}(p)$) infeasible for small

values of p. Indeed, this is the case when there are more than p outputs with a positive lower bound.

Modeling attributes of the outputs

Outputs may have attributes attached to them, which may affect the selection. We will model two of those.

First, we will consider that outputs are different in nature and therefore we will attach a different cost to them. Let c_o denote the cost associated with output $y_o, o = 1, \ldots, M$, which can measure the collection and the verification of this output in a repeated setting. To select p outputs so that their total cost does not exceed a given amount C, we need to add to (OSDEA$^{(k)}(p)$) the following constraint

$$\sum_{o=1}^{M} c_o z_o^{(k)} \leq C. \tag{5.10}$$

Second, we can consider the outputs being partitioned into S clusters, with outputs within a cluster being similar in terms of what they measure. In the context of benchmarking electricity Distribution System Operators (network companies), clusters may related to the many different measurements of connections, transformers, lines, cables, etc. Let $\mathcal{H} = \{H_1, \ldots, H_S\}$ denote the partitioning of the outputs, namely $H_\ell \cap H_s = \emptyset$ and $\cup_{\ell=1}^{S} H_\ell = \{1, \ldots, M\}$. Given the similarity of outputs within a cluster, we will impose that at most (respectively, at least) $p_\ell^{(\max)}$ (respectively, $p_\ell^{(\min)}$) outputs can be selected from H_ℓ. In order to do so, we need to add to (OSDEA$^{(k)}(p)$) the following constraint

$$\sum_{o \in H_\ell} z_o^{(k)} \leq p_\ell^{(\max)} \qquad \forall \ell = 1, \ldots, S. \tag{5.11}$$

$$\sum_{o \in H_\ell} z_o^{(k)} \geq p_\ell^{(\min)} \qquad \forall \ell = 1, \ldots, S. \tag{5.12}$$

Finally, we have correlation $\rho_{oo'}$ between outputs o and o' as another attribute. If two outputs are highly correlated, we may be interested in using only one of them, since the information they provide is almost the same and can derive in the problem of multicollinearity Bertsimas and King [2016]. Hence, we want to impose that if $\rho_{oo'}$ is greater than a preselected threshold, then outputs o and o' cannot be chosen simultaneously. We can model this by first defining a 0–1 matrix R, in which $R_{oo'} = 0$ if $\rho_{oo'}$ is lower than the threshold, and 1 otherwise. Then, we have simply to add to (OSDEA$^{(k)}(p)$) the constraints

$$z_o^{(k)} + z_{o'}^{(k)} \leq 2 - R_{oo'}, \quad \forall o < o'. \tag{5.13}$$

The choice of the threshold have to be done with care, since some works like Nunamaker [1985] suggest that the addition of a highly correlated variable may increase the efficiency.

Throughout this section, we have made the selection of outputs individually for DMU k with the goal to maximize the efficiency of DMU k. Therefore, for two different DMUs, k and k', the selected outputs, $z^{(k)}(p)$ and $z^{(k')}(p)$, may differ. In model specification one is interested in finding the most discriminatory features in order to build a valid model for all DMUs. With this in mind, we propose in the next section a mathematical optimization model that selects the outputs jointly for all DMUs, ensuring they will be the same ones for all DMUs.

5.3 The joint selection model

In this section, we address the problem in which the selected outputs have to be the same for all DMUs. First, this joint selection is made maximizing the average efficiency of all DMUs, yielding an MILP formulation. The model can be enriched as in previous section with input selection decision variables, as well as constraints modeling desirable properties about the selected features. Second, we propose alternatives to the maximization of the average efficiency when making the joint selection of outputs, such as the maximization of the weighted average efficiency, the minimum efficiency, or a percentile of the efficiencies. The joint selection model can again be formulated as an MILP problem. We also consider the minimization of the square of the Euclidean distance from each DMU efficiency to the ideal value of 1, where the joint selection model can be rewritten as a Mixed Integer Quadratic Programming problem.

5.3.1 The selection model for all DMUs

To obtain all efficiencies $E^{(k)}$ simultaneously, one can solve the following single-objective Linear Programming formulation

$$\frac{1}{K}\sum_{k=1}^{K} E^{(k)} = \text{maximize}_{(\boldsymbol{\alpha},\boldsymbol{\beta})}\ \frac{1}{K}\sum_{k=1}^{K}\sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)} \tag{5.14}$$

s.t.

$$\sum_{o=1}^{M} \beta_o^{(k)} y_o^{(j)} - \sum_{i=1}^{N} \alpha_i^{(k)} x_i^{(j)} \leq 0 \qquad \forall j = 1, \ldots, K; \forall k = 1, \ldots, K \tag{5.15}$$

$$\sum_{i=1}^{N} \alpha_i^{(k)} x_i^{(k)} = 1 \qquad \forall k = 1, \ldots, K \tag{5.16}$$

$$\alpha \in \mathbb{R}_+^{N \cdot K} \tag{5.17}$$

$$\beta \in \mathbb{R}_+^{M \cdot K}. \tag{5.18}$$

It is easy to see that this problem decomposes by DMU, and that each of the subproblems are equivalent to (DEA$^{(k)}$), which optimal solution value is $E^{(k)}$.

We continue with the model in which p outputs are to be selected from the M available ones, the same ones for all DMUs. The goal in this section is to maximize the average efficiency across all DMUs. Let z_o be equal to 1 if output o can be used in the calculation of the efficiencies, and 0 otherwise. The decision variables $\beta_o^{(k)}$ and $\alpha_i^{(k)}$ are defined as above. The Output Selection for DEA problem, (OSDEA(p)), where p outputs must be selected such that the average efficiency across all DMUs is maximized, can be written as the following MILP:

$$v(p) := \text{maximize}_{(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z})} \ \frac{1}{K} \sum_{k=1}^{K} \sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)} \tag{5.19}$$

s.t. (OSDEA(p))

$$(5.15) - (5.18)$$

$$\beta_o^{(k)} \leq M_1 z_o \qquad \forall o = 1, \ldots, M; \forall k = 1, \ldots, K \tag{5.20}$$

$$\sum_{o=1}^{M} z_o = p \tag{5.21}$$

$$z_o \in \{0, 1\} \qquad \forall o = 1, \ldots, M, \tag{5.22}$$

where M_1 is a big constant. Constraints (5.15)-(5.18) are necessary to find the weights of the inputs and the outputs that the efficiency for each DMU. Constraints (5.20) make sure that the selection variables z_o are well defined with respect to $\beta_o^{(k)}$. Constraint (5.21) models the number of features to be selected. Finally, constraints (5.22) relate to the range of decision variables z_o. (OSDEA(p)) has $K(K+M+1)+1$ linear constraints and $K(M+N)+M$ variables, where $K(M+N)$ are continuous and M are binary ones. We have multiplied the size of the problem by K, except for the number of binary variables, which are still one per output. Our numerical experiments show that this problem can still be solved efficiently. Moreover, and as in previous section, the computational burden of the problem depends on the value M_1 and we can tighten it

using similar bounds. As before, we might extend the model as in Section 5.2.2, with input selection decision variables, as well as constraints to model desirable properties of the outputs.

Let $z(p)$ denote the optimal selection variables to (OSDEA(p)), i.e., the p outputs that yield the maximum average efficiency. Thus, the optimal solution value to (OSDEA(p)), denoted above by $v(p)$, is equal to $\frac{1}{K}\sum_{k=1}^{K} E^{(k)}(z(p))$. In general, we have that $E^{(k)}(z(p)) \leq E^{(k)}(z^{(k)}(p))$, since $z^{(k)}(p)$ is the best strategy for DMU k. As in previous section, the maximum average efficiency $v(p)$ is non decreasing in p, i.e., the more outputs we select the better the average efficiency can be. In the limit case, we have that $v(M) = \frac{1}{K}\sum_{k=1}^{K} E^{(k)}$. The number of selected outputs p is a parameter of our model. The user should make the choice of p after inspecting the curve $v(p)$. As before, a plausible strategy to choose the value of p is to look at the marginal contribution of an additional feature, i.e., $v(p+1) - v(p)$, and stop when this is below a threshold. The question is whether this marginal contribution is nonincreasing, i.e., $v(p+2) - v(p+1) \leq v(p+1) - v(p)$, for all $p = 1, \ldots, M-2$. Below, we show a toy example where this inequality is not satisfied, and thus $v(p)$ is not a concave function of p. In the numerical section, devoted to the benchmarking of electricity DSOs, the function $v(\cdot)$ that we obtain empirically is concave, and thus, not convex.

Counterexample 1. *Consider* 5 *DMUs, each one described by a single input and four different outputs, as can be seen in Table 1. When performing the feature selection procedure, the results that we obtain are the following. In the case of selecting just one output, say* $p = 1$*, the procedure chooses "Output 1" and the obtained efficiencies are then just the same as the values of "Output 1" for each DMU. Hence, the average efficiency is 0.8. When two outputs are selected, the procedure chooses "Output 1" and "Output 2". These outputs make the average efficiency to be 0.867. Furthermore, if three outputs are selected, the procedure chooses "Output 2", "Output 3" and "Output 4". These outputs make all the DMUs efficient (i.e., efficiency equal to* 1*) and thus the average efficiency is 1. Clearly,*

$$v(3) - v(2) > v(2) - v(1),$$

and therefore $v(\cdot)$ *is not concave.*

A greedy approach is provided in Pastor et al. [2002] to address the feature selection problem in a nested fashion. In short, this greedy nested procedure works as follows. For $p = 1$, (OSDEA(p)) is solved to optimality. Let $o(1)$ be its best output. For $p = 2$, (OSDEA(p)) is solved to optimality, with the additional constraint that $z_{o(1)} = 1$. Let $o(2)$ be its best output. In general, for p, (OSDEA(p)) is solved to optimality, with the additional constraints that $z_{o(1)} = z_{o(2)} = \ldots = z_{o(p-1)} = 1$. Let $o(p)$ be its best output. Clearly, this greedy approach returns a sequence of outputs that is nested, i.e.,

DMU	Input 1	Output 1	Output 2	Output 3	Output 4
1	1	0.6	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
2	1	0.7	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
3	1	0.8	1	0	0
4	1	0.9	0	1	0
5	1	1	0	0	1

Table 5.1: Toy example for which $v(\cdot)$ is not concave

DMU	Input 1	Output 1	Output 2	Output 3
1	1	0.85	0.2	0.8
2	1	0.95	0.4	0.6
3	1	0.9	0.6	0.4
4	1	1	0.8	0.2

Table 5.2: Toy example for which the approach in Pastor et al. [2002] does not provide the optimal solution to (OSDEA(p))

the outputs selected in iteration $p-1$ will also be selected in iteration p, for all p. The following is a toy example that illustrates that the approach in Pastor et al. [2002] does not provide, in general, the optimal solution to (OSDEA(p)).

Counterexample 2. *Consider* 4 *DMUs, each one described by a single input and three different outputs, as can be seen in Table 2. When performing the feature selection procedure, the results that we obtain are the following. In the case of selecting just one output, say* $p = 1$*, the procedure chooses "Output 1" and the obtained efficiencies are then just the same as the values of "Output 1" for each DMU. When two outputs are selected, the procedure chooses "Output 2" and "Output 3". These outputs make all the DMUs efficient. However, if either "Output 1" and "Output 2" or "Output 1" and "Output 3" were used instead, the efficiencies would be {0.85,0.9,0.95,1} and {1,1,0.927,1}, respectively.*

5.3.2 Alternative objective functions

In (OSDEA(p)), we maximize the average efficiency across all DMUs. In this section, we propose other objective functions $\phi()$ to select the outputs.

A straightforward generalization would be to consider the weighted average efficiency.

$$\phi^{(\mathrm{w})}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{z}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{o=1}^{M} \omega^{(k)} \beta_o^{(k)} y_o^{(k)}. \tag{5.23}$$

This is relevant if the DMUs are not equally important. If there is only one input, say cost in \$, we could for example use $w^{(k)} = x^{(k)}$, and (5.23) would correspond to

minimizing the total sector loss from inefficiency.

Instead of the weighted average efficiency, one could be interested in measuring how far each DMU is from efficiency. This can be measured with the following quadratic function

$$\phi^{(\mathrm{q})}(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z}) = \frac{1}{K}\sum_{k=1}^{K}(1-\sum_{o=1}^{M}\beta_o^{(k)}y_o^{(k)})^2. \tag{5.24}$$

Alternatively, our goal could have been maximizing the worst efficiency, i.e., the minimum one

$$\phi^{(\mathrm{m})}(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z}) = \min_{k=1,\dots,K}\sum_{o=1}^{M}\beta_o^{(k)}y_o^{(k)}. \tag{5.25}$$

This is relevant, for example, if outputs are selected with the aim of being Rawlsian fair towards all DMUs. Instead of the minimum, we could have optimized another π-percentile, $\pi = 1,\dots,100$, of the efficiency distribution. Assuming that the efficiencies are given in non-decreasing order, $\sum_{o=1}^{M}\beta_o^{(k)}\mathbf{y}_o^{(k)} \leq \sum_{o=1}^{M}\beta_o^{(k+1)}y_o^{(k+1)}$, for all k, we would have

$$\phi^{(\pi)}(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z}) = \sum_{o=1}^{M}\beta_o^{(k(\pi))}y_o^{(k(\pi))}, \tag{5.26}$$

with $k(\pi) = \lfloor K\,\frac{\pi}{100}\rfloor$.

The Output Selection for DEA problem where the goal is to maximize $\phi^{(\mathrm{w})}$ in (5.23), $(\mathrm{OSDEA}(p))^{(\mathrm{w})}$, can be formulated in the same fashion as $(\mathrm{OSDEA}(p))$.

The Output Selection for DEA problem where the goal is to maximize $\phi^{(\mathrm{q})}$ in (5.24), $(\mathrm{OSDEA}(p))^{(\mathrm{q})}$, can be formulated similarly to $(\mathrm{OSDEA}(p))$. While the feasible region remains the same, the objective function becomes quadratic and the goal is to minimize it, yielding a Mixed Integer Quadratic Programming formulation.

The Output Selection for DEA problem where the goal is to maximize the minimum efficiency $\phi^{(\mathrm{m})}$ in (5.25), $(\mathrm{OSDEA}(p))^{(\mathrm{m})}$ can be written as an MILP. Here, we need to define a new variable λ to rewrite the minimum in the objective function, and include the corresponding constraints to ensure that the new variable is well defined.

$$\lambda \leq \sum_{o=1}^{M}\beta_o^{(k)}y_o^{(k)} \quad k = 1,\dots,K, \tag{5.27}$$

and thus

$$\text{maximize}_{(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z},\lambda)}\ \lambda \tag{5.28}$$

s.t. $(\mathrm{OSDEA}(p))^{(\mathrm{m})}$

$$(5.15)-(5.18); (5.2)-(5.4); (5.27).$$

The Output Selection for DEA problem where the goal is to maximize the π-percentile $\phi^{(\pi)}$ in (5.26), $(\text{OSDEA}(p))^{(\pi)}$, can also be written as an MILP, similarly as in Benati [2015]. We need to define a new variable λ that is equal to the percentile, as well as include the corresponding constraints to ensure that the new variable is well defined. We also need a new binary variable, $\delta^{(k)}$, that is equal to 1 if the efficiency of DMU k, $\sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)}$, is at least λ and 0 otherwise.

$$\text{maximize}_{(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{z},\lambda,\delta)} \ \lambda \tag{5.29}$$

s.t. $(\text{OSDEA}(p))^{(\pi)}$

$$(5.15)-(5.18); (5.2)-(5.4)$$

$$\sum_{o=1}^{M} \beta_o^{(k)} y_o^{(k)} \geq \lambda - M_1'(1-\delta^{(k)}) \qquad \forall k = 1,\ldots,K \tag{5.30}$$

$$\sum_{k=1}^{K} \delta^{(k)} = \lfloor K \, \frac{\pi}{100} \rfloor \tag{5.31}$$

$$\delta^{(k)} \in \{0,1\} \qquad \forall k = 1,\ldots,K, \tag{5.32}$$

with M_1' a big constant.

5.4 Numerical section

In this section, we illustrate the models in previous sections using a real-world dataset in benchmarking of electricity DSOs Agrell and Bogetoft [2017, 2018]. Here, we have $K = 182$ DMUs, $M = 100$ outputs, and $N = 1$ input. As customary, each output has been normalized dividing it by the difference between the maximum and the minimum values of the output. Figure 5.1 displays the correlations between the outputs, with darker colours pointing at higher correlations. This matrix reveals subsets of outputs highly correlated with each other, such as outputs 23 to 31, where correlations are above 0.5, except for $\text{corr}(23, 27) = 0.37$ and $\text{corr}(27, 29) = 0.35$.

The experiments were run on a computer with an Intel® Core™ i7-6700 processor at 3.4 GHz using 16 GB of RAM, running Windows 10 Home. All the optimization problems have been solved using Python 3.5 interface Python Core Team [2015] with Gurobi 7.0.1 solver, Gurobi Optimization [2016].

We have solved $(\text{OSDEA}(p))$ for $p = 1,\ldots,10$, with M_1 equal to 1000. We have run the approach in Pastor et al. [2002] to provide $(\text{OSDEA}(p))$ with an initial solution. A time limit of 300 seconds has been imposed, although this is not binding for small values of p. Once $(\text{OSDEA}(p))$ has selected the p outputs, $p = 1,\ldots,10$, we calculate the efficiencies of the DSOs obtained with the chosen outputs. The results are summa-

p	min	max	mean	st. dev.	q_1	q_2	q_3	q_3-q_1	selected features
1	0.0000	1.0000	0.5555	0.1695	0.4743	0.5637	0.6300	0.1557	59
2	0.0006	1.0000	0.6553	0.1708	0.5772	0.6682	0.7482	0.1710	11 31
3	0.0009	1.0000	0.7118	0.1643	0.6391	0.7222	0.7839	0.1448	21 31 54
4	0.1161	1.0000	0.7487	0.1511	0.6645	0.7479	0.8404	0.1759	16 21 31 59
5	0.1161	1.0000	0.7812	0.1494	0.6962	0.7738	0.8911	0.1949	16 21 31 59 94
6	0.3105	1.0000	0.8082	0.1402	0.7222	0.8068	0.9305	0.2083	16 19 21 31 59 94
7	0.3105	1.0000	0.8290	0.1404	0.7474	0.8355	0.9545	0.2071	16 19 21 31 59 91 94
8	0.3105	1.0000	0.8462	0.1402	0.7658	0.8689	0.9802	0.2144	16 19 21 31 59 91 94 97
9	0.3105	1.0000	0.8610	0.1370	0.7789	0.8841	0.9972	0.2183	16 19 21 31 59 74 91 94 97
10	0.4576	1.0000	0.8732	0.1304	0.7902	0.9090	1.0000	0.2098	16 19 21 29 31 59 74 91 94 97

Table 5.3: Summary statistics for the distribution of efficiencies, for $p = 1, \ldots, 10$

rized in Table 5.4, and Figures 5.3 and 5.2, while the correlation matrix in Figure 5.4 highlights the correlations between the selected outputs.

Table 5.4 presents summary statistics of the distribution of the efficiencies, namely the minimum, the maximum, the average (i.e., $v(p)$), the standard deviation, the quartiles q_i, and the interquartile range (i.e., q_3-q_1). The last column of this table reports the selected outputs. Figure 5.2 displays the box-and-whiskers plots as well as the average efficiency $v(p)$, and Figure 5.3 the histograms of the distribution of the efficiencies. The average efficiency improves with the number of selected outputs, p, increasing from 0.5555 to 0.8732. Figure 5.2 shows that the marginal effect of increasing p to $p + 1$ is decreasing for this dataset. When looking at the quartiles, we can see that there is a substantial improvement too by increasing p. When the number of selected features, p, is small the chosen features give poor efficiencies to some of the DMUs. Indeed, for $p \leq 5$, the minimum efficiency is below 0.1200. As p increases, we can see that this minimum increases rapidly with p. The first quartile increases from 0.4743 to 0.7902. Similarly, for the median, we have 0.5637 to 0.9090, while for the third quartile 0.6300 to 1.0000. The maximum efficiency is already maximal, i.e., equal to 1.0000, for the smallest value of p, and remains like that for all values of p tested. In general, the standard deviation of the efficiencies decreases with p, while the interquartile range increases. In terms of the correlations, for small values of p, the selected outputs are highly correlated with each other. For instance, for $p = 2$, we choose outputs 11 and 31, for which $\text{corr}(11, 31) = 0.91$; while for $p = 3$, we choose outputs 21, 31, 54, with $\text{corr}(21, 31) = 0.89$, $\text{corr}(21, 54) = 0.88$ and $\text{corr}(31, 54) = 0.91$. Actually, for $p = 2, \ldots, 5$, the smallest correlation between the selected outputs is equal to 0.61. For $p = 10$, there are only two outputs, for which we can find correlations below 0.5, namely outputs 19 and 74.

In real-life applications, the DMUs may be consulted on the chosen outputs. This may be useful to ensure that the resulting choices make conceptual sense. However, such involvement is likely to lead strategic behavior. The evaluated DMUs may try to influence the choice of outputs in their own advantage. This may lead to a game

between the DMUs (the players), each of which has preferred selections of outputs (strategies), and the modeler, who is trying to make a reasonable selection based on the resulting efficiencies of all DMUs (the outcome).

To start investigating the challenges of such strategic behavior, we will consider a simple game or social choice problem. Without loss of generality, we assume that p is fixed. We assume that there are K players, the DMUs, and $K+1$ strategies or choices, namely the selection of outputs $z^k(p)$ according to the individual preferences as determined by (OSDEA$^{(k)}(p)$), $k = 1, \ldots, K$, and to the joint selection $z(p)$ as determined by (OSDEA(p)). We will think of the joint selection $z(p)$ as the default or status quo selection and the question is now if one of the individual selections $z^k(p)$ would be preferred by a large group of DMUs. If so, the modeler is likely to face strong opposition to his proposed selection of outputs.

For DMU k', the selection of outputs made with (OSDEA$^{(k')}(p)$), $z^{(k')}(p)$, is at least as attractive as the one made with (OSDEA(p)), $z(p)$, or in other words, the reported efficiency with $z^{(k')}(p)$ is at least as high as the one with $z(p)$. However, for any other DMU $k \neq k'$, it is not clear whether the selection $z^{(k')}(p)$ reports a higher efficiency for DMU k or not. One would like to know how many DMUs prefer the joint strategy $z(p)$ over the K individual strategies $z^{(k')}(p)$. The so-called cross-efficiency measures this preference. Let $\Delta^{(k)}(k') = E^{(k)}(\mathbf{z}^{(k')}(p)) - E^{(k)}(\mathbf{z}(p))$, with $k, k' = 1, \ldots, K$, be equal to the difference in reported efficiency for DMU k by the joint selection model and the individual selection model for DMU k'. We can define

$$\Pi(k') = \frac{100}{K} \text{cardinality}(\{k : \Delta^{(k)}(k') > 0\}),$$

i.e., the percentage of DMUs that prefer the individual strategy of DMU k' over the joint one.

In Figure 5.5, we illustrate the share of DMUs that prefer individual selections to the joint selection. As for the joint strategy, a time limit of 300 seconds has been imposed to (OSDEA$^{(k')}(p)$), although this is not binding for any value of p. We have binned the support to the individual selections in intervals of width 5%. The height of the first bar indicates how many individual strategies are preferred by [0%,5%) of the DMUs, the height of the second bar corresponds to [5%,10%) DMUs supporting it, etc. When p increases, we see that less individual strategies are preferred by many DMUs over the joint strategy. Indeed, for values of p above 5, the joint strategy is supported by at least 50% of the DMUs over any of the individual strategies, while for values of p above 7, this becomes at least 60%. We can therefore conclude that, in this simple game, as the model gets larger, it becomes less likely that a large group of DMUs will agree on alternative to the modeler's selection.

5.5 Chapter Summary

In this chapter, we have proposed a single-model approach for feature selection in DEA. When the objective is the average efficiency of the DMUs, the problem can be written as an MILP formulation. We have considered other objectives such as the squared distance to the ideal point, where all the DMUs are efficient, yielding a Mixed Integer Quadratic Programming formulation; and we have shown how to enrich the model to allow for situations where different features come with different costs, e.g., related to data collection or data quality, and where features can be grouped and restrictions can be placed on the use of different groups of features in the specific industrial application. Our numerical section illustrates that we can find good solutions in a reasonable amount of time for the case in which the average efficiency is the goal, which boils down to an MILP.

Our approach deviates from previous literature on feature selection in several ways. It is purely based on mathematical programming as opposed to a mixture of statistical and mathematical programming methods, where the desirable properties above can be modeled in a natural way. It works directly with the original features as opposed to dimensionality reduction techniques, which create artificial features. It focuses on the choice of features from a large set of potential candidate features. Finally, it can handle different objective functions to reflect the underlying objective of the modeler and the application context, e.g., the conflicts between different groups of DMUs in the evaluation.

In this chapter, we have also introduced an element of game theory. This is relevant since the evaluated parties in applied projects typically will try to influence the feature selection. It is therefore important to think about the conflict between choosing features from a joint and an individual point of view. We have shown how conflicts can be partially analyzed via the cross-efficiency matrix and we have illustrated the conflict between individual and joint perspectives in the numerical application.

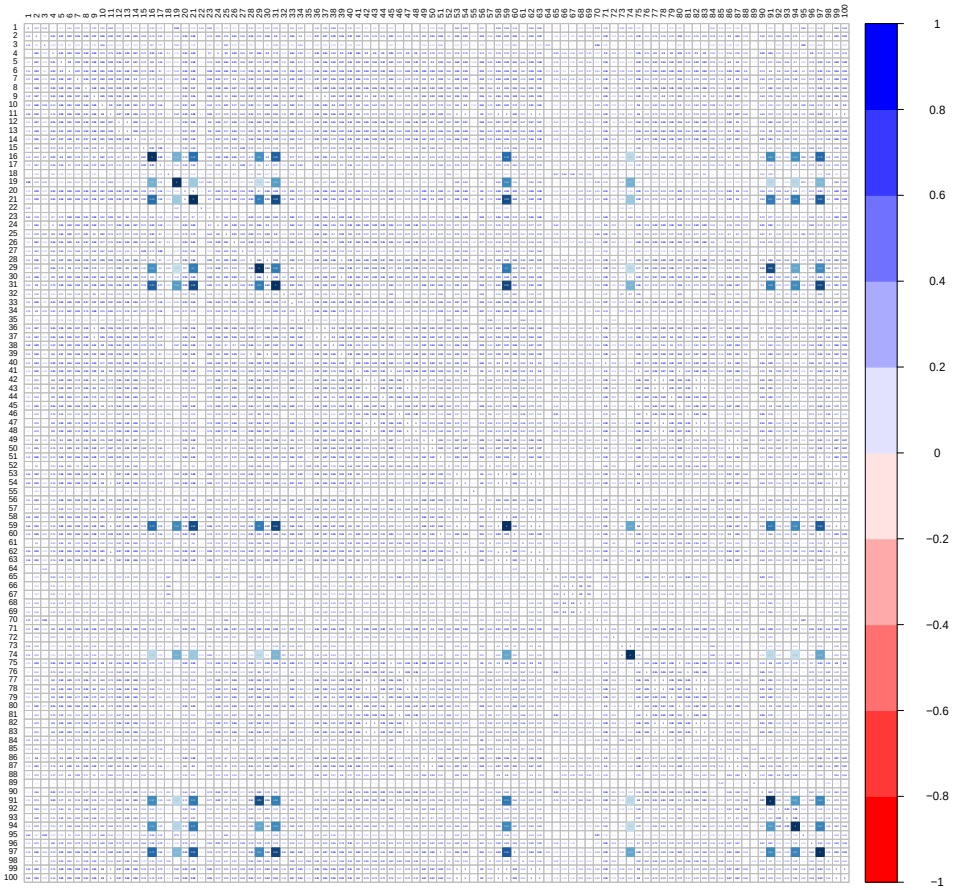


Figure 5.1: Correlation matrix for the outputs, highlighting the correlation between with the selected outputs for $p = 10$

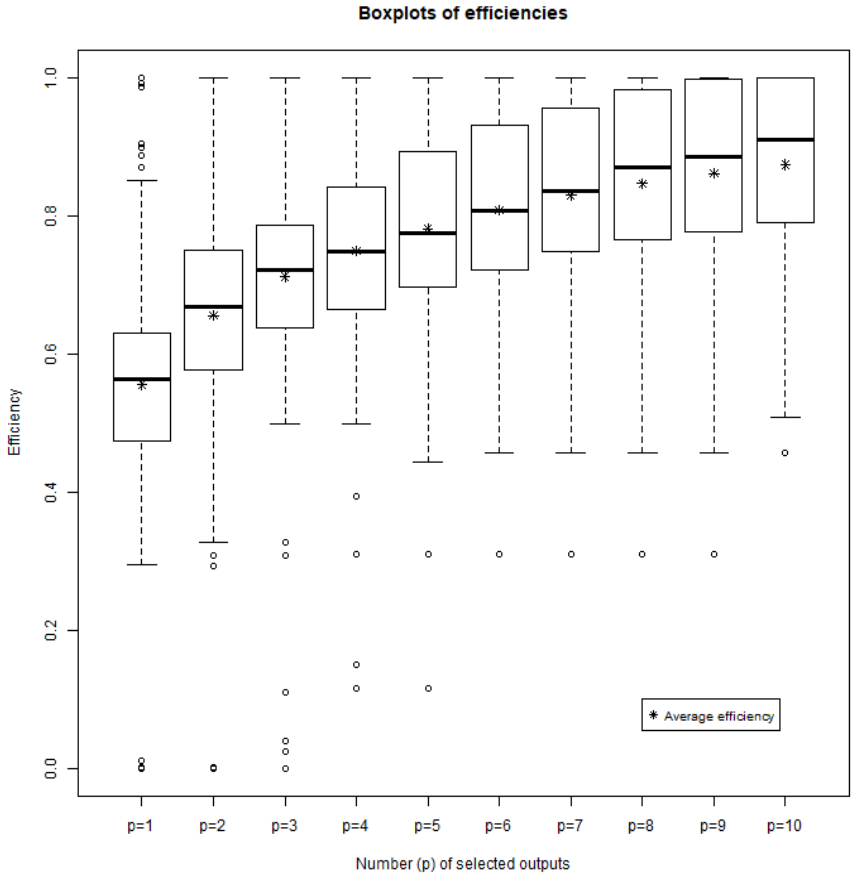


Figure 5.2: Box-and-whiskers plots of efficiencies, including average efficiency, for $p = 1, \ldots, 10$

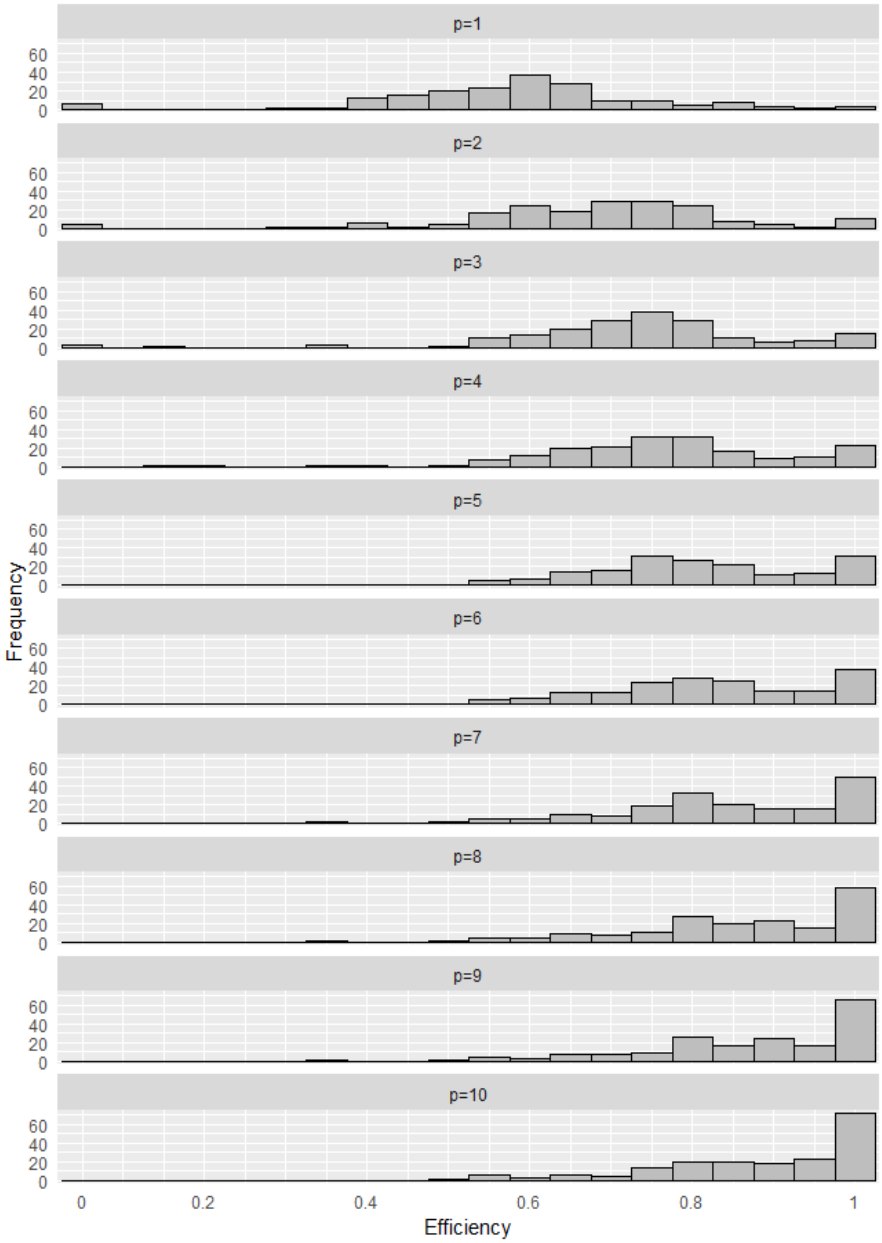


Figure 5.3: Histograms of the distribution of the efficiencies, $p = 1, \ldots, 10$

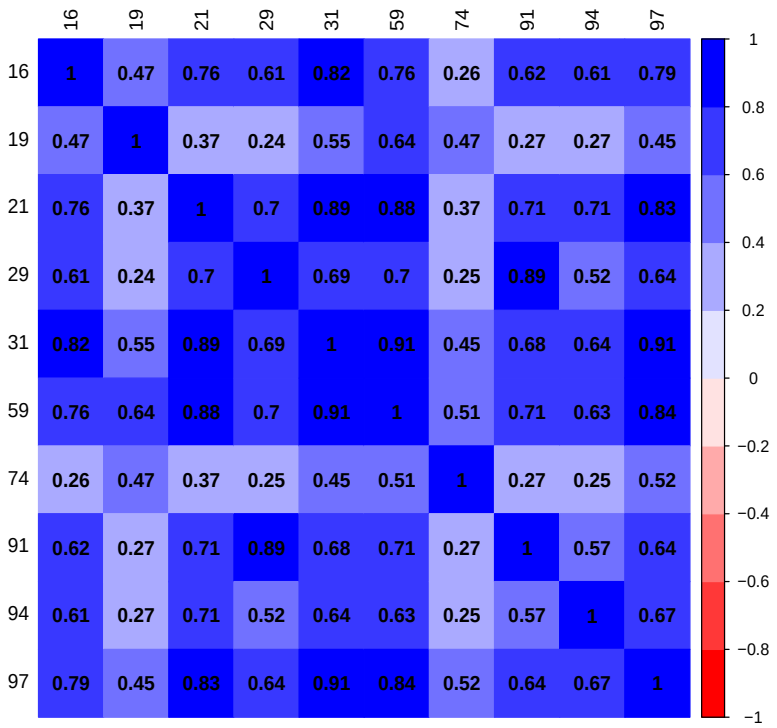


Figure 5.4: Correlation matrix for the selected outputs for $p = 10$

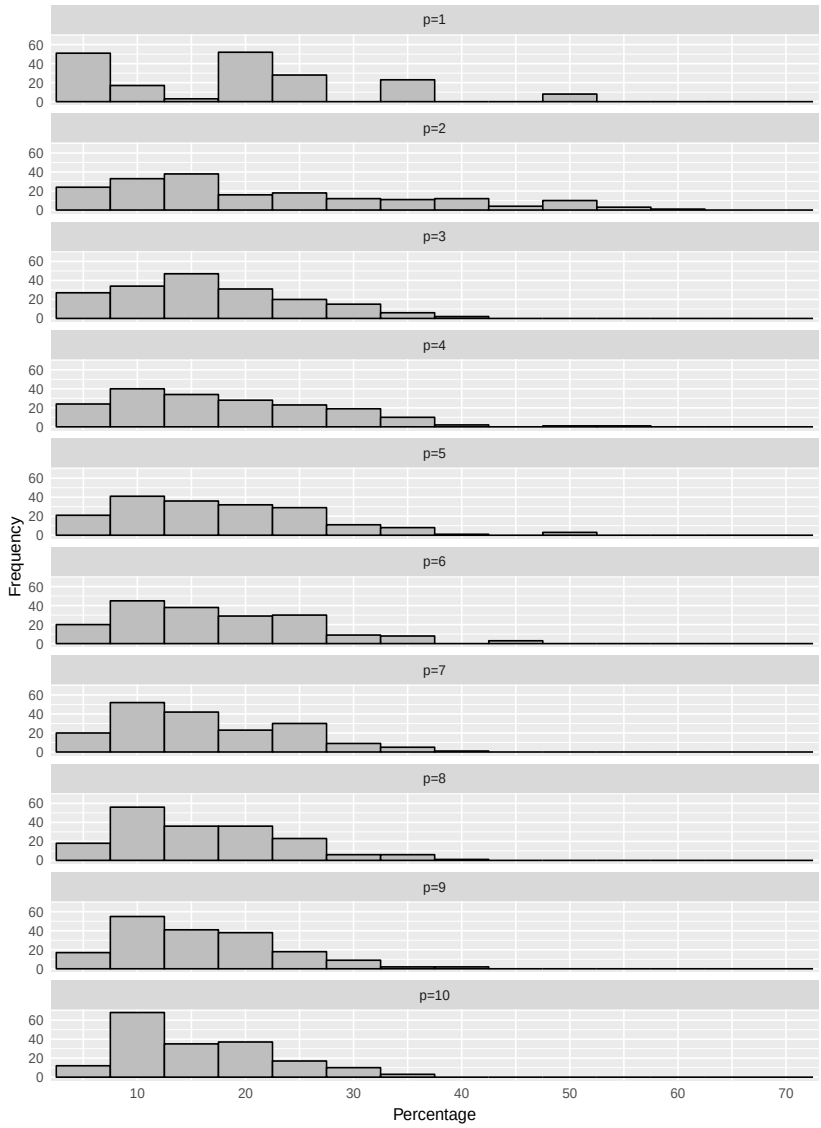


Figure 5.5: Histograms of the distribution of preferences of individual strategies over the joint strategy, $p = 1, \ldots, 10$

Chapter 6

General Conclusions and Further Work

In this PhD dissertation, Mathematical Optimization and Statistics have been used together as tools for the development of new models in Data Science. In particular, we focus on Support Vector Machines (SVM) (Chapters 2 to 4) and Benchmarking (Chapter 5). The works that has been presented here are motivated by the fact that in these recent years, the extraction of knowledge for both learning and interpreting information from datasets has became one of the most relevant tasks in Science and other fields of knowledge.

Regarding SVM, we have first proposed, in Chapter 2, a novel SVM model in which performance constraints are included in order to control the different misclassification costs. This is of particular importance in presence of imbalancedness in the dataset. In order to do that, threshold values for the misclassification rates are given to the model. Later, in Chapter 3, we have addressed the problem of obtaining class probabilities estimates for SVM. The method proposed is, contrary to existing proposals, distribution-free and cost-sensitive. Furthermore, it provides confidence intervals for both the score values and the posterior class probabilities. We close the SVM part by proposing a mathematical-optimization-based Feature Selection procedure for SVM in Chapter 4. Such a proposal consist on an embedded method, which also includes the idea of controlling the misclassification rates. Concerning Benchmarking, we propose in Chapter 5 a single-model approach for feature (inputs and outputs) selection in Data Envelopment Analysis (DEA). In this work, opposed to previous literature on feature selection in DEA, our method is purely based on mathematical programming instead of on a mixture of statistical and mathematical programming methods.

The research addressed in this dissertation poses new problems and challenges to be considered as future work.

A possible extension concerning the results of Chapter 2 would be expand the presented approach to the case when using more complex data as multi-class or *multi-way* arrays (Lyu et al. 2017) instead of two-ways data matrices and two-class problems as currently done. On the other hand, an alternative perspective for addressing the SVM regularization is to consider different norms (Yao and Lee 2014). Finally, another possible extension is to perform a feature selection which uses the proposed constraints in order to control the misclassification costs, as done in Benítez-Peña et al. [2019a] and also presented in Chapter 4. Such a process is an essential step in tasks such as high-dimensional microarray classification problems (Guo 2010).

A number of new problems deriving from Chapter 3 are stated next. First, as previously commented, the basis classifier used in this chapter is the SVM with linear kernel. However, more complex kernels (such as the RBF) can be used in order to obtain more powerful classifiers. On the other hand, traditional SVM can be used as a basis for addressing multiclass problems. How to extend properly our approach to such multiclass problems is an interesting research avenue which is now under investigation.

Several extensions of the approach presented in Chapter 4 are possible. In our opinion, they deserve further study. First, several classification and regression procedures based on optimization problems, such as Support Vector Regression, logistic regression or distance-weighted discrimination, are amenable to address, as done here, an integrated FS and classification or regression. The optimization problems obtained in this way have a structure which should be exploited to make the approach competitive and including cost-sensitivity in the FS procedure. Second, even within SVM, it should be observed that SVM is a tool for binary classification. For multiclass datasets, SVM classification is performed by solving a series of SVM problems, see Cristianini and Shawe-Taylor [2000]; Wang and Shen [2007]. When some classes are hard to identify, the basic multiclass strategies may yield discouraging results. Performing simultaneously feature selection and class fusion, as in Guo [2010], is an interesting nontrivial extension of our approach. To do this, problems $(P1)$, $(P2)$ and $(P3)$ need to be conveniently modified.

Finally, the results of Chapter 5 can be extended as follows. We have talked about the introduction of a game theory element and how to analyze different conflicts regarding the feature selection. Nevertheless, in the future, it would however be relevant to further explore better these issues. One limitation of our analysis above is that we only consider K specific alternatives to the modeler's joint selection. In theory, there are, of course, many more alternatives. Indeed, any subset of size p from the set of potential outputs M could potentially muster the support of many DMUs against the modeler's proposal. The strategic analysis of all possible alternatives is likely to become overwhelming. It may therefore be relevant to introduce some restrictions. One idea is to detect relevant clusters of DMUs and make the selection of features tailored to them. By looking at likely interest groups, the game theoretical analysis may be less complex. Groupings could, for example, refer to small versus large, start-up versus well-established, urban versus rural, and investor-owned versus cooperatively owned DMUs. Also, it would be interesting to add constraints to the feature selection model in order to guarantee the support of a number of DMUs, e.g., a majority of DMUs, hereby modeling more general game theoretical aspects, such as the scope for forming coalitions.

In general, there exist many challengers extensions of the works presented in this PhD dissertation as well as the study of alternative applications of Mathematical Optimization and Statistics to both the fields of Supervised Classification and Benchmarking.

List of Figures

List of Tables

Appendix

In this section we describe step by step how formulation (4.2) is built from equation (4.1). Hence, let us suppose first that we have the model

$$\begin{array}{lll} \min_{\omega,\beta,\xi,z} & \omega^\top\omega + C\sum_{i\in I}\xi_i & \\ s.t. & y_i(\omega^\top x_i+\beta) \geq 1-\xi_i, & i\in I \\ & 0\leq \xi_i \leq M_1(1-z_i) & i\in I \\ & \hat{p}_\ell \geq \hat{p}^*_{0\ell} & \ell\in L \\ & z_i\in\{0,1\} & i\in I. \end{array}$$

This one can be rewritten as

$$\begin{array}{llll} \min_z & & \min_{\omega,\beta,\xi} & \omega^\top\omega + C\sum\limits_{i\in I}\xi_i \\ \text{s.t.} & z_i\in\{0,1\} \quad i\in I & \text{s.t.} & y_i\left(\omega^\top x_i+\beta\right)\geq 1-\xi_i, \quad i\in I \\ & \hat{p}_\ell \geq \hat{p}^*_{0\ell} \quad \ell\in L & & 0\leq\xi_i\leq M_1(1-z_i) \quad i\in I \end{array}$$

If we assume that the binary variables z fixed, the Karush–Kuhn–Tucker (KKT) conditions for the inner problem are

$$\begin{array}{rcl} \omega & = & \sum\limits_{i\in I}\lambda_i y_i x_i \\ 0 & = & \sum\limits_{i\in I}\lambda_i y_i \\ 0 & \leq & \lambda_i \leq C/2 \quad i\in I. \end{array}$$

Substituting these expressions into the last optimization problem, the partial dual of such problem can be calculated, obtaining

$$\begin{array}{llll} \min\limits_z & & \min\limits_{\lambda,\beta,\xi} & \left(\sum\limits_{i\in I}\lambda_i y_i x_i\right)^\top\left(\sum\limits_{i\in I}\lambda_i y_i x_i\right) + C\sum\limits_{i\in I}\xi_i \\ \text{s.t.} & z_i\in\{0,1\} \quad i\in I & \text{s.t.} & y_i\left(\left(\sum\limits_{i\in I}\lambda_i y_i x_i\right)^\top x_i+\beta\right)\geq 1-\xi_i \quad i\in I \\ & \hat{p}_\ell\geq\hat{p}^*_{0\ell} \quad \ell\in L & & 0\leq\xi_i\leq M_1(1-z_i) \quad i\in I \\ & & & \sum\limits_{i\in I}\lambda_i y_i = 0 \\ & & & 0\leq\lambda_i\leq C/2 \quad i\in I \end{array}$$

As a last step, the kernel trick is used and the final formulation (4.2) is obtained.

References

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Adler, N. and Yazhemsky, E. (2010). Improving discrimination in data envelopment analysis: PCA–DEA or variable reduction. *European Journal of Operational Research*, 202(1):273–284.

Agarwal, S. (2011). Classification of countries based on macro-economic variables using fuzzy support vector machine. *International Journal of Computer Applications*, 27(6):41.

Agrell, P. and Bogetoft, P. (2017). Regulatory benchmarking: Models, analyses and applications. *Data Envelopment Analysis Journal*, 3(1–2):49–91.

Agrell, P. and Bogetoft, P. (2018). Theory, techniques, and applications of regulatory benchmarking and productivity analysis. In *The Oxford Handbook of Productivity Analysis*.

Aigner, D. J. and Chu, S.-F. (1968). On estimating the industry production function. *The American Economic Review*, 58(4):826–839.

Alcalá-Fdez, J., Sanchez, L., Garcia, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., et al. (2009). Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing*, 13(3):307–318.

Allen, R., Athanassopoulos, A., Dyson, R., and Thanassoulis, E. (1997). Weights restrictions and value judgements in data envelopment analysis: evolution, development and future directions. *Annals of Operations Research*, 73:13–34.

Allwein, E. L., Schapire, R. E., and Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141.

Aytug, H. (2015). Feature selection for support vector machines using Generalized Benders Decomposition. *European Journal of Operational Research*, 244(1):210–218.

Babatunde, O. H., Armstrong, L., Leng, J., and Diepeveen, D. (2014). A genetic algorithm-based feature selection.

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

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., and Vanthienen, J. (2003b). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the operational research society*, 54(6):627–635.

Balakrishnama, S. and Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. In *Institute for Signal and information Processing*, volume 18, pages 1–8.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Ben-Tal, A., Bhadra, S., Bhattacharyya, C., and Saketha Nath, J. (2011). Chance constrained uncertain classification via robust optimization. *Mathematical Programming*, 127(1):145–173.

Benati, S. (2015). Using medians in portfolio optimization. *Journal of the Operational Research Society*, 66(5):720–731.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., and Ramírez-Cobo, P. (2019a). Cost-sensitive feature selection for support vector machines. *Computers & Operations Research*, 106:169 – 178, `http://www.sciencedirect.com/science/article/pii/S0305054818300741`, doi:https://doi.org/10.1016/j.cor.2018.03.005.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., and Ramírez-Cobo, P. (2019b). On support vector machines under a multiple-cost scenario. *Advances in Data Analysis and Classification*, 13(3):663–682.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., and Ramírez-Cobo, P. (2021). Cost-sensitive probabilistic predictions for support vector machines. *https://www.researchgate.net/publication/341103637_Cost-sensitive_probabilistic_predictions_for_support_vector_machines*. Submitted.

Benítez-Peña, S., Bogetoft, P., and Morales, D. R. (2020). Feature selection in data envelopment analysis: A mathematical optimization approach. *Omega*, 96:102068.

Bertolazzi, P., Felici, G., Festa, P., Fiscon, G., and Weitschek, E. (2016). Integer programming models for feature selection: New extensions and a randomized solution algorithm. *European Journal of Operational Research*, 250(2):389–399.

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

Bertsimas, D. and King, A. (2016). OR Forum - An Algorithmic Approach to Linear Regression. *Operations Research*, 64:2–16.

Bertsimas, D., King, A., Mazumder, R., et al. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852.

Bertsimas, D., Mazumder, R., et al. (2014). Least quantile regression via modern optimization. *The Annals of Statistics*, 42(6):2494–2525.

Bertsimas, D. and Shioda, R. (2007). Classification and regression via integer optimization. *Operations Research*, 55(2):252–271.

Bertsimas, D. and Weismantel, R. (2005). *Optimization over integers*, volume 13. Dynamic Ideas Belmont.

Bewick, V., Cheek, L., and Ball, J. (2004). Statistics review 13: receiver operating characteristic curves. *Critical Care*, 8(6):508–512.

Biau, G. and Devroye, L. (2015). *Lectures on the Nearest Neighbor Method*. Springer Publishing Company, Incorporated, 1st edition.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271.

Bogetoft, P. (1996). Dea on relaxed convexity assumptions. *Management Science*, 42(3):457–465.

Bogetoft, P. (2013). *Performance benchmarking: Measuring and managing performance*. Springer Science & Business Media.

Bogetoft, P. and Otto, L. (2010). *Benchmarking with Dea, Sfa, and R*, volume 157. Springer Science & Business Media.

Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008). An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186 – 204. In Memory of George B. Dantzig.

Boţ, R. I. and Lorenz, N. (2011). Optimization problems in statistical learning: Duality and optimality conditions. *European Journal of Operational Research*, 213(2):395–404.

Boulesteix, A.-L., Lambert-Lacroix, S., Peyre, J., and Strimmer, K. (2011). plsgenomics: Pls analyses for genomics. *R package version*, pages 1–2.

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., and Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 131–136. Springer.

Bradley, P. S., Fayyad, U. M., and Mangasarian, O. L. (1999). Mathematical Programming for Data Mining: Formulations and Challenges. *INFORMS Journal on Computing*, 11(3):217–238.

Bradley, P. S., Mangasarian, O. L., and Street, W. N. (1998). Feature Selection via Mathematical Programming. *INFORMS Journal on Computing*, 10(2):209–217.

Burer, S. and Letchford, A. N. (2012). Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97 – 106.

Burges, C. J. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Camm, J. D., Raturi, A. S., and Tsubakitani, S. (1990). Cutting Big M Down to Size. *Interfaces*, 20(5):61–66.

Carrizosa, E., Martin-Barragan, B., and Romero Morales, D. (2008). Multi-group Support Vector Machines with Measurement Costs: A Biobjective Approach. *Discrete Applied Mathematics*, 156(6):950–966.

Carrizosa, E., Martín-Barragán, B., and Romero-Morales, D. (2011). Detecting relevant variables and interactions in supervised classification. *European Journal of Operational Research*, 213(1):260–269.

Carrizosa, E., Nogales-Gómez, A., and Romero-Morales, D. (2016). Strongly agree or strongly disagree?: Rating features in support vector machines. *Information Sciences*, 329:256–273.

Carrizosa, E., Nogales-Gómez, A., and Romero-Morales, D. (2017a). Clustering categories in support vector machines. *Omega*, 66:28–37.

Carrizosa, E., Olivares-Nadal, A. V., and Ramírez-Cobo, P. (2017b). A sparsity-controlled vector autoregressive model. *Biostatistics*, page kxw042.

Carrizosa, E. and Romero Morales, D. (2013). Supervised Classification and Mathematical Optimization. *Computers & Operations Research*, 40(1):150–165.

Caruana, R. and Freitag, D. (1994). Greedy attribute selection. In *Machine Learning Proceedings 1994*, pages 28–36. Elsevier.

Chan, A. B., Vasconcelos, N., and Lanckriet, G. R. G. (2007). Direct convex relaxations of sparse svm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 145–153, New York, NY, USA. ACM.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), `https://doi.org/10.1145/1961189.1961199`, doi:10.1145/1961189.1961199.

Charnes, A., Cooper, W., and Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6):429–444.

Chaudhuri, A. (2014). Support vector machine model for currency crisis discrimination. *arXiv preprint arXiv14030481*.

Chavda, A., Potika, K., Di Troia, F., and Stamp, M. (2018). Support vector machines for image spam analysis. In *ICETE (1)*, pages 597–607.

Coelli, T. J., Rao, D. S. P., O'Donnell, C. J., and Battese, G. E. (2005). *An introduction to efficiency and productivity analysis*. Springer Science & Business Media.

Cook, W., Ramón, N., Ruiz, J., Sirvent, I., and Zhu, J. (2019). DEA-based benchmarking for performance evaluation in pay-for-performance incentive plans. *Omega*, 84:45 – 54, `http://www.sciencedirect.com/science/article/pii/S0305048317311672`, doi:https://doi.org/10.1016/j.omega.2018.04.004.

Cook, W., Tone, K., and Zhu, J. (2014). Data envelopment analysis: Prior to choosing a model. *Omega*, 44:1–4.

Cook, W. D. and Zhu, J. (2006). *Modeling performance measurement: applications and implementation issues in DEA*, volume 566. Springer Science & Business Media.

Cooper, W. W., Seiford, L. M., and Tone, K. (2001). Data envelopment analysis: A comprehensive text with models, applications, references and dea-solver software. *Journal-operational Research Society*, 52(12):1408–1409.

Corne, D., Dhaenens, C., and Jourdan, L. (2012). Synergies between operations research and data mining: The emerging use of multi-objective approaches. *European Journal of Operational Research*, 221(3):469–479.

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

Coussement, K. (2014). Improving customer retention management through cost-sensitive learning. *European Journal of Marketing*.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Datta, S. and Das, S. (2015). Near-Bayesian Support Vector Machines for imbalanced data classification with equal or unequal misclassification costs. *Neural Networks*, 70:39–52.

Deo, R. C. (2015). Machine learning in medicine. *Circulation*, 132(20):1920–1930.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`, University of California, Irvine, School of Information and Computer Sciences.

Efron, B. (2000). The bootstrap and modern statistics. *Journal of the American Statistical Association*, 95(452):1293–1296.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75.

Emrouznejad, A. and Yang, G.-L. (2018). A survey and analysis of the first 40 years of scholarly literature in DEA: 1978–2016. *Socio-Economic Planning Sciences*, 61:4–8.

Fernandez-Palacin, F., Lopez-Sanchez, M., and Munõz-Márquez, M. (2018). Stepwise selection of variables in DEA using contribution loads. *Pesquisa Operacional*, 38(1):31–52.

Fethi, M. D., Jackson, P. M., and Weyman-Jones, T. G. (2001). European airlines: a stochastic dea study of efficiency with market liberalisation.

Fodor, I. K. (2002). A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab., CA (US).

Franc, V., Zien, A., and Schölkopf, B. (2011). Support vector machines as probabilistic models. In *Proceedings of the 28th International Conference on Machine Learning*, pages 665–672, Madison, WI, USA. International Machine Learning Society.

Freitas, A., Costa-Pereira, A., and Brazdil, P. (2007). Cost-Sensitive Decision Trees Applied to Medical Data. In *Data Warehousing and Knowledge Discovery: 9th International Conference, DaWaK 2007, Regensburg Germany, September 3-7, 2007. Proceedings*, pages 303–312, Berlin, Heidelberg. Springer Berlin Heidelberg.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Fung, G. M. and Mangasarian, O. L. (2004). A Feature Selection Newton Method for Support Vector Machine Classification. *Computational Optimization and Applications*, 28(2):185–202, http://dx.doi.org/10.1023/B:COAP.0000026884.66338.df, doi:10.1023/B:COAP.0000026884.66338.df.

Gambella, C., Ghaddar, B., and Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807 – 828, http://www.sciencedirect.com/science/article/pii/S037722172030758X, doi:https://doi.org/10.1016/j.ejor.2020.08.045.

Ghaddar, B. and Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265(3):993 – 1004, http://www.sciencedirect.com/science/article/pii/S0377221717307713, doi:https://doi.org/10.1016/j.ejor.2017.08.040.

Ghatasheh, N., Faris, H., AlTaharwa, I., Harb, Y., and Harb, A. (2020). Business analytics in telemarketing: cost-sensitive analysis of bank campaigns using artificial neural networks. *Applied Sciences*, 10(7):2581.

Golany, B. and Roll, Y. (1989). An application procedure for DEA. *Omega*, 17(3):237–250.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999a). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, http://science.sciencemag.org/content/286/5439/531, doi:10.1126/science.286.5439.531.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999b). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537.

Gonen, M., Tanugur, A. G., and Alpaydin, E. (2008). Multiclass posterior probability support vector machines. *IEEE Transactions on Neural Networks*, 19(1):130–139, doi:10.1109/TNN.2007.903157.

Green, R., Doyle, J., and Cook, W. (1996). Preference voting and project ranking using dea and cross-evaluation. *European Journal of Operational Research*, 90(3):461–472.

Greene, W. H. (1990). A gamma-distributed stochastic frontier model. *Journal of Econometrics*, 46(1):141 – 163, http://www.sciencedirect.com/science/article/pii/030440769090052U, doi:https://doi.org/10.1016/0304-4076(90)90052-U.

Greene, W. H. (2003). *Econometric analysis*. Pearson Education India.

Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K. (2003). Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer.

Guo, J. (2010). Simultaneous variable selection and class fusion for high-dimensional linear discriminant analysis. *Biostatistics*, 11(4):599, doi:10.1093/biostatistics/kxq023.

Gurobi Optimization, I. (2016). Gurobi Optimizer Reference Manual. http://www.gurobi.com.

Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Harman, H. (1976). *Modern Factor Analysis*. University of Chicago Press, 3rd edition.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, https://www.sciencedirect.com/science/article/pii/0095069678900062, doi:https://doi.org/10.1016/0095-0696(78)90006-2.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. In *Advances in neural information processing systems*, pages 507–513.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

He, H. and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, Inc.

Herbrich, R., Graepel, T., and Campbell, C. (1999). *Bayesian learning in reproducing kernel Hilbert spaces*. Leiter der Fachbibliothek Informatik, Sekretariat FR 5-4.

Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30.

Horn, D., Demircioğlu, A., Bischl, B., Glasmachers, T., and Weihs, C. (2016). A comparative study on large scale kernelized support vector machines. *Advances in Data Analysis and Classification*.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

Huang, B., Kechadi, M. T., and Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1):1414–1425.

Huang, X., Wu, L., and Ye, Y. (2019). A review on dimensionality reduction techniques. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(10):1950017.

J.-Y. Cai, C.-Y. L. (2016). LASSO variable selection techniques in data envelopment analysis. In *The 17th Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS 2016)*, Taipei, Taiwan.

Jiang, C. and Lin, W. (2015). DEARank: a data-envelopment-analysis-based ranking method. *Machine Learning*, 101(1–3):415–435.

Joro, T. and Korhonen, P. (2015). Data envelopment analysis. In *Extension of Data Envelopment Analysis with Preference Information*, pages 15–26. Springer.

Karatzoglou, A., Meyer, D., and Hornik, K. (2006). Support vector machines in r. *Journal of Statistical software*, 15(9):1–28.

Keramati, A. and Ardabili, S. M. (2011). Churn analysis for an iranian mobile operator. *Telecommunications Policy*, 35(4):344–356.

Kim, S., Yu, Z., Kil, R. M., and Lee, M. (2015). Deep learning of support vector machines with class probability output networks. *Neural Networks*, 64:19 – 28, `http://www.sciencedirect.com/science/article/pii/S0893608014002172`, doi:https://doi.org/10.1016/j.neunet.2014.09.007. Special Issue on "Deep Learning of Representations".

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Stanford, CA.

Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109.

Kruppa, J., Liu, Y., Biau, G., Kohler, M., König, I. R., Malley, J. D., and Ziegler, A. (2014). Probability estimation with machine learning methods for dichotomous and multicategory outcome: Theory. *Biometrical Journal*, 56(4):534–563, https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.201300068, doi:https://doi.org/10.1002/bimj.201300068.

Kudva, V. and Prasad, K. (2018). Pattern classification of images from acetic acid–based cervical cancer screening: A review. *Critical Reviews™ in Biomedical Engineering*, 46(2).

Kwok, J. T.-Y. (1998). Integrating the evidence framework and the support vector machine. In *ESANN*, volume 99, pages 177–182.

Kwok, J. T.-Y. (1999). Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 10(5):1018–1031.

Lal, T. N., Chapelle, O., Weston, J., and Elisseeff, A. (2006). Embedded methods. In *Feature extraction*, pages 137–165. Springer.

Land, K. C., Lovell, C. K., and Thore, S. (1993). Chance-constrained data envelopment analysis. *Managerial and decision economics*, 14(6):541–554.

Landete, M., Monge, J., and Ruiz, J. (2017). Robust DEA efficiency scores: A probabilistic/combinatorial approach. *Expert Systems with Applications*, 86:145–154.

Le Thi, H. A., Le, H. M., and Dinh, T. P. (2015). Feature selection in machine learning: an exact penalty approach using a Difference of Convex function Algorithm. *Machine Learning*, 101(1):163–186.

Lee, C.-Y. and Cai, J.-Y. (2020). LASSO variable selection in data envelopment analysis with small datasets. *Omega*, 91:102019.

Li, Y., Shi, X., Yang, M., and Liang, L. (2017a). Variable selection in data envelopment analysis via akaike's information criteria. *Annals of Operations Research*, 253(1):453–476.

Li, Z., Crook, J., and Andreeva, G. (2017b). Dynamic prediction of financial distress using Malmquist DEA. *Expert Systems with Applications*, 80:94–106.

Lichman, M. (2013). UCI Machine Learning Repository. http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences.

Lin, H.-T., Lin, C.-J., and Weng, R. C. (2007). A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276.

Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning*, 46(1-3):191–202.

Lorena, A. and de Carvalho, A. (2008). Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing*, 71:3326–3334.

Lovell, C. K. et al. (1993). Production frontiers and productive efficiency. *The measurement of productive efficiency: Techniques and applications*, 3:67.

Luo, Y., Bi, G., and Liang, L. (2012). Input/output indicator selection for DEA efficiency evaluation: An empirical study of Chinese commercial banks. *Expert Systems with Applications*, 39(1):1118–1123.

Lyu, T., Lock, E. F., and Eberly, L. E. (2017). Discriminating sample groups with multi-way data. *Biostatistics*.

Maldonado, S., Domínguez, G., Olaya, D., and Verbeke, W. (2021). Profit-driven churn prediction for the mutual fund industry: A multisegment approach. *Omega*, 100:102380, `https://www.sciencedirect.com/science/article/pii/S0305048320307349`, doi:https://doi.org/10.1016/j.omega.2020.102380.

Maldonado, S., Pérez, J., and Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research*, 261(2):656 – 665, `http://www.sciencedirect.com/science/article/pii/S0377221717301595`, doi:https://doi.org/10.1016/j.ejor.2017.02.037.

Maldonado, S. and Weber, R. (2009). A wrapper method for feature selection using support vector machines. *Information Sciences*, 179(13):2208 – 2217, `http://www.sciencedirect.com/science/article/pii/S0020025509000917`, doi:https://doi.org/10.1016/j.ins.2009.02.014. Special Section on High Order Fuzzy Sets.

Maldonado, S., Weber, R., and Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences*, 181(1):115 – 128, `http://www.sciencedirect.com/science/article/pii/S0020025510004287`, doi:https://doi.org/10.1016/j.ins.2010.08.047.

Mangasarian, O. L., Street, W. N., and Wolberg, W. H. (1995). Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577.

Mansouri, K., Ringsted, T., Ballabio, D., Todeschini, R., and Consonni, V. (2013). Quantitative structure–activity relationship models for ready biodegradability of chemicals. *Journal of Chemical Information and Modeling*, 53(4):867–878.

Margineantu, D. D. (2002). Class probability estimation and cost-sensitive classification decisions. In Elomaa, T., Mannila, H., and Toivonen, H., eds., *Machine Learning: ECML 2002*, pages 270–281, Berlin, Heidelberg. Springer Berlin Heidelberg.

Marron, J. S., Todd, M. J., and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271.

Meisel, S. and Mattfeld, D. (2010). Synergies of Operations Research and Data Mining. *European Journal of Operational Research*, 206(1):1–10.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A*, 209:415–446.

Milgram, J., Mohamed Cheriet, and Sabourin, R. (2005). Estimating accurate multi-class probabilities with support vector machines. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 3, pages 1906–1911 vol. 3.

Moffett, S., Anderson-Gillespie, K., and McAdam, R. (2008). Benchmarking and performance measurement: a statistical analysis. *Benchmarking: An International Journal*.

Murphy, K. P. (2012). *Machine learning, a probabilistic perspective*. The MIT Press.

Nataraja, N. and Johnson, A. (2011). Guidelines for using variable selection techniques in Data Envelopment Analysis. *European Journal of Operational Research*, 215(3):662–669.

Nunamaker, T. (1985). Using data envelopment analysis to measure the efficiency of non-profit organizations: A critical evaluation. *Managerial and Decision Economics*, 6(1):50–58.

Olesen, O. B. and Petersen, N. (1995). Chance constrained efficiency evaluation. *Management science*, 41(3):442–457.

Panagopoulos, O. P., Pappu, V., Xanthopoulos, P., and Pardalos, P. M. (2016). Constrained subspace classifier for high dimensional datasets. *Omega*, 59:40–46.

Parsons, L. J. (2002). Using stochastic frontier analysis for performance measurement and benchmarking. *Advances in Econometrics*, 16:317–350.

Pastor, J., Ruiz, J., and Sirvent, I. (2002). A statistical test for nested radial DEA models. *Operations Research*, 50(4):728–735.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Petersen, N. (2018). Directional Distance Functions in DEA with Optimal Endogenous Directions. *Operations Research*, 66(4):1068–1085.

Plastria, F. and Carrizosa, E. (2012). Minmax-distance approximation and separation problems: geometrical properties. *Mathematical Programming*, 132(1):153–177.

Platt, J. C. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers*, pages 61–74. MIT Press.

Podinovski, V. (2016). Optimal weights in DEA models with weight restrictions. *European Journal of Operational Research*, 254(3):916–924.

Prati, R. C., Batista, G. E., and Silva, D. F. (2015). Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems*, 45(1):247–270.

Python Core Team (2015). Python: A dynamic, open source programming language. Python Software Foundation. `{https://www.python.org}`.

Qin, Z. and Song, I. (2014). Joint Variable Selection for Data Envelopment Analysis via Group Sparsity. *ArXiv e-prints arXiv:1402.3740*.

Ramón, N., Ruiz, J., and Sirvent, I. (2010). On the choice of weights profiles in cross-efficiency evaluations. *European Journal of Operational Research*, 207(3):1564–1572.

Richtárik, P. and Takáč, M. (2016). Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484.

Ruiz, J. and Sirvent, I. (2016). Common benchmarking and ranking of units with DEA. *Omega*, 65:1 – 9, `http://www.sciencedirect.com/science/article/pii/S0305048315002510`, doi:https://doi.org/10.1016/j.omega.2015.11.007.

Ruiz, J. L. and Sirvent, I. (2019). Performance evaluation through DEA benchmarking adjusted to goals. *Omega*, 87:150–157.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517.

Sánchez, B. N., Wu, M., Song, P. X. K., and Wang, W. (2016). Study design in high-dimensional classification analysis. *Biostatistics*, 17(4):722, doi:10.1093/biostatistics/kxw018.

Sarveniazi, A. (2014). An actual survey of dimensionality reduction. *American Journal of Computational Mathematics*, 4(2):55–72.

Seeger, M. (2000). Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In *Advances in neural information processing systems*, pages 603–609.

Sexton, T., Silkman, R., and Hogan, A. (1986). Data envelopment analysis: Critique and extensions. *New Directions for Program Evaluation*, 1986(32):73–105.

Shen, X., Tseng, G. C., Zhang, X., and Wong, W. H. (2003). On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734.

Silva, A. P. D. (2017). Optimization approaches to Supervised Classification. *European Journal of Operational Research*, 261(2):772–788.

Sirvent, I., Ruiz, J., Borrás, F., and Pastor, J. (2005). A Monte Carlo evaluation of several tests for the selection of variables in DEA models. *International Journal of Information Technology & Decision Making*, 4(03):325–343.

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

Soleimani-Damaneh, M. and Zarepisheh, M. (2009). Shannon's entropy for combining the efficiency results of different DEA models: Method and application. *Expert Systems with Applications*, 36(3, Part 1):5146–5150.

Sollich, P. (2002). Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine learning*, 46(1):21–52.

Steinberg, D. and Colla, P. (2009). Cart: classification and regression trees. *The top ten algorithms in data mining*, 9:179.

Swan, A. L., Mobasheri, A., Allaway, D., Liddell, S., and Bacardit, J. (2013). Application of machine learning to proteomics data: Classification and biomarker identification in postgenomics biology. *OMICS: A Journal of Integrative Biology*, 17(12):595–610, `https://doi.org/10.1089/omi.2013.0017`, doi:10.1089/omi.2013.0017. PMID: 24116388.

Tan, F., Fu, X., Zhang, Y., and Bourgeois, A. G. (2008). A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2):111–120.

Tang, Y., Zhang, Y. Q., Chawla, N. V., and Krasser, S. (2009). SVMs Modeling for Highly Imbalanced Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288.

Tao, Q., Wu, G.-W., Wang, F.-Y., and Wang, J. (2005). Posterior probability support vector machines for unbalanced data. *IEEE Transactions on Neural Networks*, 16(6):1561–1573.

Tarca, A. L., Carey, V. J., Chen, X.-w., Romero, R., and Drăghici, S. (2007). Machine learning and its applications to biology. *PLoS Comput Biol*, 3(6):e116.

Thomas, L., Crook, J., and Edelman, D. (2017). *Credit scoring and its applications*. SIAM.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Timofeev, R. (2004). Classification and regression trees (cart) theory and applications. *Humboldt University, Berlin*, pages 1–40.

Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244.

Van Rossum, G. and Drake, F. L. (2011). *An Introduction to Python*. Network Theory Ltd.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Wagner, J. and Shimshak, D. (2007). Stepwise selection of variables in Data Envelopment Analysis: Procedures and managerial perspectives. *European Journal of Operational Research*, 180(1):57–67.

Wahba, G. (1992). Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In *Santa Fe Institute Studies in the Sciences of Complexity-Proceedings Volume-*, volume 12, pages 95–95. Addison-Wesley Publishing Co.

Wahba, G. et al. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87.

Walker, M. G. and Olshen, R. (1992). Probability estimation for biomedical classification problems. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 451. American Medical Informatics Association.

Wang, L. and Shen, X. (2007). On L1-Norm Multiclass Support Vector Machines. *Journal of the American Statistical Association*, 102(478):583–594, doi:10.1198/016214506000001383.

Wehrens, R., Putter, H., and Buydens, L. M. (2000). The bootstrap: a tutorial. *Chemometrics and intelligent laboratory systems*, 54(1):35–52.

Wei, Y., Fang, S., and Wang, X. (2019). Automatic modulation classification of digital communication signals using svm based on hybrid features, cyclostationary, and information entropy. *Entropy*, 21(8):745.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001). Feature Selection for SVMs. In Leen, T. K., Dietterich, T. G., and Tresp, V., eds., *Advances in Neural Information Processing Systems 13*, pages 668–674. MIT Press.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Xu, X., Liang, T., Zhu, J., Zheng, D., and Sun, T. (2019). Review of classical dimensionality reduction and sample selection methods for large-scale data processing. *Neurocomputing*, 328:5–15.

Yao, Y. and Lee, Y. (2014). Another look at linear programming for feature selection via methods of regularization. *Statistics and Computing*, 24(5):885–905.

Zhang, L., Zhu, J., and Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269.

Zhang, S. (2020). Cost-sensitive knn classification. *Neurocomputing*, 391:234–242.

Zhu, Q., Wu, J., and Song, M. (2018). Efficiency evaluation based on data envelopment analysis in the big data context. *Computers & Operations Research*, 98:291–300.