Programa de doctorado "Matemáticas"

PHD DISSERTATION

COMPUTATIONAL METHODS FOR THE ANALYSIS OF COMPLEX DATA

Author

Mª Remedios Sillero Denamiel

Supervisors

Prof. Dr. *Rafael Blanquero Bravo*

Prof. Dr. *Emilio Carrizosa Priego*

Prof. Dra. *Pepa Ramírez Cobo*

A mis padres y mi hermana.
A mis abuelos.

Agradecimientos

Y al final todo llega. Qué gran verdad. Siempre había visto este momento como algo muy lejano, intentando disfrutar al máximo del camino y ya, hoy, se cierra esta maravillosa etapa. Recuerdo perfectamente el día en el que mi director Emilio me brindó la oportunidad de trabajar en un proyecto de transferencia, junto con mi también director Rafa, y mi compañera y amiga Asun. Esto cambió todos mis planes, y de qué manera. Gracias por la confianza que depositaste en mí.

Emilio, Rafa, Pepa, qué suerte teneros como directores. Me iniciasteis en la investigación, un camino que requiere mucha dedicación y constancia, pero que gratifica más aún. Gracias por enseñarme tanto, por vuestro tiempo y apoyo durante estos años. No siempre ha sido fácil, pero vosotros me habéis ayudado a continuar. Emilio, gracias por transmitirme tu pasión por tu trabajo sin darte cuenta, por tu dedicación y por animarme siempre que lo he necesitado (y siempre que no, también). Rafa, gracias por tu paciencia infinita cuando las cosas no salían, encontrando una solución para todo y siempre con una sonrisa. Pepa, has estado al pie del cañón, y así me lo has hecho saber. Hemos compartido mucho más que esta tesis, eres espectacular. Gracias por siempre estar. *Las personas con las que trabajo son, sin duda, lo mejor de este trabajo.*

Poder viajar y vivir en otros países mientras aprendes de grandes profesionales es un regalo. Loli, cómo me gusta Copenhague, tu segunda ciudad. Gracias por hacerme sentir como en casa durante mis estancias, aun estando a tres mil kilómetros de mi familia. Todavía nos quedan muchos cafés juntas, y muchas exposiciones por descubrir. Mike, thank you for always welcoming me with a smile, for your patience teaching me and for the moments of chatting in the canteen while having coffee and tea.

También me gustaría agradecer al Departamento de Estadística e Investigación Operativa de la Universidad de Sevilla, al IMUS y a sus equipos de Administración, por facilitarme siempre todo lo que he necesitado. Y, por supuesto, a los proyectos y a los IP de los proyectos que han financiado esta tesis, 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).

Cuántos momentos compartidos con mis compañeros del IMUS. Más que compañeros,

amigos. No los he podido tener mejores. Habéis sido cómplices en los buenos momentos y un apoyo fundamental en los no tan buenos. ¡Me siento tan orgullosa de formar parte de esta familia! No sé dónde acabaré, pero le habéis dejado el listón demasiado alto a mis futuros compañeros. Y, aunque suene a despedida, esto no queda aquí, aún nos queda mucho por disfrutar juntos. Gracias por tanto.

Mis amigas de toda la vida, las que siempre están. Lorena, Eva, Virginia, sois lo más. No tengo palabras de agradecimiento a tantos años de amistad de la buena. Daniela, qué suerte que nos cruzáramos en el camino, eres única y no sabes cuánto nos enseñas. Marisa, gracias por hacer las veces de hermana mayor y por todos tus consejos. Vicente, al final aquellas tardes de estudio han valido la pena. Siempre habéis creído en mí, incluso mucho más que yo misma. Esta tesis también es vuestra.

Mamá, papá, qué os voy a decir. No hay día que no me mostréis lo orgullosos que estáis de mí, pero yo sí que estoy orgullosa de vosotros, y de la familia que somos. Todo lo que he conseguido y consiga es y será gracias a vosotros y a vuestro apoyo incondicional. Rocío, mi hermana y confidente, siempre contaremos la una con la otra. Abuela Aurelia, eres fortaleza y un pilar fundamental para mí. Abuelo Higinio, abuela Francisca, abuelo Andrés, sé que desde arriba estáis celebrándolo conmigo. Os quiero.

Gracias a todos, de corazón.

Resumen

Esta tesis combina las disciplinas de Investigación Operativa y Estadística con el fin de desarrollar nuevos métodos computacionales para extraer información de datos complejos. En este estudio, *datos complejos* se refiere a conjuntos de datos con un número elevado de muestras y/o variables, con diferentes tipos de variables, con estructuras de dependencia entre las variables, recogidos de diferentes fuentes (heterogéneos), posiblemente con clases desbalanceadas, con diferentes costes de clasificación incorrecta o caracterizados por valores extremos (datos de cola pesada), entre otros.

La complejidad de los datos y las nuevas exigencias de los usuarios (modelos interpretables, modelos sensibles a costes de errores de predicción o modelos eficientes en tiempo de ejecución) implica un reto desde la perspectiva científica. Las principales contribuciones de esta tesis se engloban en tres marcos teóricos diferentes: Regresión, Clasificación e Inferencia Bayesiana. Repecto al primero, consideramos modelos de regresión lineal, donde una variable respuesta continua se pronostica a partir de un conjunto de variables predictoras. Por un lado, buscando soluciones interpretables en datos heterogéneos, proponemos una nueva versión del Lasso en la que se controla el rendimiento del modelo en los grupos de interés. Por otro lado, aplicamos técnicas de optimización matemática para proponer un modelo de regresión lineal diseñado específicamente para conjuntos de datos con variables predictoras categóricas y jerárquicas. En lo que se refiere a Clasificación, en esta tesis se ha explorado en profundidad el clasificador Naïve Bayes. Este método se ha adaptado para obtener una solución sparse (es decir, expreseada en términos de un subconjunto de variables predictoras). Además, el método se ha modificado para tratar con datos en los que se debe tener en cuenta los diferentes costes de clasificación incorrecta. En ambos problemas, se presentan nuevas estrategias para reducir los tiempos de ejecución. Finalmente, la última contribución de esta tesis es relativa a la inferencia Bayesiana. En particular, se considera un enfoque de estimación Bayesiano semiparamétrico para estimar la distribución Elíptica, la cual generaliza la distribución normal multivariante permitiendo colas más pesadas.

La estructura de esta tesis es la siguiente. El Capítulo 1 revisa los conceptos teóricos necesarios para desarrollar los capítulos posteriores. En concreto, se revisan dos áreas de investigación principales: el aprendizaje automático (sparse y sensible a costes de errores de predicción) y la estadística Bayesiana.

En el Capítulo 2 se propone un método basado en el Lasso en el que se han añadido restricciones cuadráticas para controlar los errores de predicción en individuos de interés. Este modelo de regresión, sparse y con restricciones, se define mediante un problema de optimización no lineal. El método resulta de especial interés cuando se trabaja con muestras heterogéneas donde los datos provienen de diferentes fuentes, como resulta habitual en muchos contextos biomédicos.

El Capítulo 3 estudia modelos de regresión para variables predictoras categóricas con estructura jerárquica. El modelo es flexible en el sentido en que el usuario decide el nivel de detalle en la información que el modelo debe utilizar, atendiendo a razones de privacidad y confidencialidad. Para modelar el equilibrio entre el rendimiento del modelo y su complejidad, se define un problema cuadrático convexo con variables enteras y restricciones lineales.

En el Capítulo 4, se introduce una versión sparse del clasificador Naïve Bayes, caracterizada por las tres propiedades siguientes. Primero, la selección de variables se realiza teniendo en cuenta la estructura de correlación de las variables predictoras. Segundo, se pueden usar diferentes medidas de rendimiento al seleccionar los subconjuntos de variables. Finalmente, el modelo permite incluir restricciones de rendimiento en los grupos de individuos de mayor interés. Mediante el diseño de una búsqueda inteligente, el método consigue tiempos de ejecución competitivos.

El enfoque introducido en el Capítulo 2 también se explora en el Capítulo 5 con el fin de mejorar la predicción del clasificador Naïve Bayes en las clases de mayor interés para el usuario. A diferencia de la versión tradicional del clasificador basado en dos etapas (estimación primero y clasificación después), la nueva metodología integra ambas a la vez. El método se formula como un problema de optimización donde se maximiza la función de verosimilitud con restricciones para las tasas de clasificación de los grupos de interés.

El Capítulo 6 aborda la estimación estadística desde la perspectiva Bayesiana de una clase general de distribuciones de cola pesada, las distribuciones Elípticas. Esta familia de distribuciones se han propuesto a nivel teórico en problemas asociados a datos extremos, típicos en contextos financieros o en teoría de riesgo. El enfoque que adoptamos es semiparamétrico basado en los procesos Dirichlet.

Finalmente, el Capítulo 7 cierra esta tesis con conclusiones generales y futuras líneas de investigación.

Abstract

This PhD dissertation bridges the disciplines of Operations Research and Statistics to develop novel computational methods for the extraction of knowledge from complex data. In this research, *complex data* stands for datasets with many instances and/or variables, with different types of variables, with dependence structures among the variables, collected from different sources (heterogeneous), possibly with non-identical population class sizes, with different misclassification costs, or characterized by extreme instances (heavy-tailed data), among others.

Recently, the complexity of the raw data in addition to new requests posed by practitioners (interpretable models, cost-sensitive models or models which are efficient in terms of running times) entail a challenge from a scientific perspective. The main contributions of this PhD dissertation are encompassed in three different research frameworks: Regression, Classification and Bayesian inference. Concerning the first, we consider linear regression models, where a continuous outcome variable is to be predicted by a set of features. On the one hand, seeking for interpretable solutions in heterogeneous datasets, we propose a novel version of the Lasso in which the performance of the method on groups of interest is controlled. On the other hand, we use mathematical optimization tools to propose a sparse linear regression model (that is, a model whose solution only depends on a subset of predictors) specifically designed for datasets with categorical and hierarchical features. Regarding the task of Classification, in this PhD dissertation we have explored in depth the Naïve Bayes classifier. This method has been adapted to obtain a sparse solution and also, it has been modified to deal with cost-sensitive datasets. For both problems, novel strategies for reducing high running times are presented. Finally, the last contribution of this dissertation concerns Bayesian inference methods. In particular, in the setting of heavy-tailed data, we consider a semi-parametric Bayesian approach to estimate the Elliptical distribution.

The structure of this dissertation is as follows. Chapter 1 contains the theoretical background needed to develop the following chapters. In particular, two main research areas are reviewed: sparse and cost-sensitive statistical learning and Bayesian Statistics.

Chapter 2 proposes a Lasso-based method in which quadratic performance constraints to bound the prediction errors in the individuals of interest are added to Lasso-based objective functions. This constrained sparse regression model is defined by a nonlinear optimization problem. Specifically, it has a direct application in heterogeneous samples where data are

collected from distinct sources, as it is standard in many biomedical contexts.

Chapter 3 studies linear regression models built on categorical predictor variables that have a hierarchical structure. The model is flexible in the sense that the user decides the level of detail in the information used to build it, having into account data privacy considerations. To trade off the accuracy of the linear regression model and its complexity, a Mixed Integer Convex Quadratic Problem with Linear Constraints is solved.

In Chapter 4, a sparse version of the Naïve Bayes classifier, which is characterized by the following three properties, is proposed. On the one hand, the selection of the subset of variables is done in terms of the correlation structure of the predictor variables. On the other hand, such selection can be based on different performance measures. Additionally, performance constraints on groups of higher interest can be included. This smart search integrates the flexibility in terms of performance for classification, yielding competitive running times.

The approach introduced in Chapter 2 is also explored in Chapter 5 for improving the performance of the Naïve Bayes classifier in the classes of most interest to the user. Unlike the traditional version of the classifier, which is a two-step classifier (estimation first and classification next), the novel approach integrates both stages. The method is formulated via an optimization problem where the likelihood function is maximized with constraints on the classification rates for the groups of interest.

When dealing with datasets of especial characteristics (for example, heavy tails in contexts as Economics and Finance), Bayesian statistical techniques have shown their potential in the literature. In Chapter 6, Elliptical distributions, which are generalizations of the multivariate normal distribution to both longer tails and elliptical contours, are examined, and Bayesian methods to perform semi-parametric inference for them are used.

Finally, Chapter 7 closes the thesis with general conclusions and future lines of research.

Contents

Chapter 1

Introduction

The huge assorment of datasets describing timely and relevant real-world situations requires revisiting and updating decision methods and combine them with data analysis techniques, to yield Data-Driven Decision Making. Contemporary data are characterized by many instances and/or variables, different types of variables (quantitative, categorical, ordinal, clustered), dependence structures among the variables or extreme values (heavy-tailed data). Also, datasets can be characterized by asymmetric conditions (non-identical population classes size, different misclassification costs), where different errors may have different consequences (as happens in the context of medical diagnosis). Complexities can also be caused by the hierarchical structure of the data when, for instance, national statistics need to work with the partition of the country into regions, provinces, municipalities or neighbourhoods; or when a retailing company aims to group their products to be able to predict at the aggregate level. Recently, the complexity of the raw data in addition to new requests posed by practitioners (interpretable models, fair models or models which are efficient in terms of running times or memory demand for prediction) entail a challenge from a scientific and technological perspective.

The aim of this thesis is to propose novel computational methods for the extraction of knowledge from the above-mentioned complex data. In fact, this thesis bridges the disciplines of Operations Research and Statistics to lead to new approaches that outperform current methods.

1.1 Sparse and cost-sensitive statistical learning

The challenge of interpreting information and learning from complex data is at the core of the current Statistical Science. The aim of Statistical Learning theory is the estimation of a function $f(\mathbf{X})$ for predicting the response variable Y given the set of predictors $\mathbf{X}$, a random vector of dimension p. We refer to *regression* problem when the response Y is quantitative, whereas the term *classification* is employed in the case of a categorical response.

1.1.1 Sparsity in regression and classification

Current datasets are usually characterized by a large number of features (that is, the dimension of the feature space p is large). This fact may have negative consequences in terms of the comprehensibility of solutions and, thus, Statistics and Operations Research fields are continually adapting to tackle this matter [Friedman et al., 2001; Hastie et al., 2015]. For example, a clear interpretation of the solutions is of crucial importance for some specific medical or credit scoring related problems, where the interest is to find the key variables that determine if a patient is sick or healthy, or those for predicting a bank's customer as potentially defaulting, see Hand and Henley [1997]. The search for more interpretable and parsimonious solutions, common in multivariate contexts such as regression [Cai et al., 2009; Lin et al., 2011; Benítez-Peña et al., 2021], clustering [Maldonado et al., 2015], time series analysis [Carrizosa et al., 2017b]

or visualization [Carrizosa and Guerrero, 2014], has recently led to the development of sparse multivariate techniques, see Hastie et al. [2015].

Excessive computational costs, redundant variables and noise are other drawbacks associated with high-dimensional data. Besides the identification of significant predictors, which provides a good interpretation of the model reducing the computational costs and noise, another fundamental criterion for evaluating the model's performance is the accuracy of prediction. It could be said that looking for sparse models helps to obtain low prediction errors, since it avoids overparametrized models and thus, overfitting [Carrizosa et al., 2016]. For those reasons, sparsity is a desirable property that regression and classification models should satisfy.

Regarding parametric regression contexts, a solution is said to be sparse if only a subset of coefficients are non-zero. In particular, in some parts of this dissertation, we will focus on linear regression models, where the response variable is expressed as a linear combination of the predictors. The conventional linear regression procedure to estimate the coefficients, the Ordinary Least Squares (OLS), is known not to be sparse and, as a consequence, new penalization techniques appeared in the literature, see Gui et al. [2017] and Li et al. [2020] for a detailed overview. An example is the *best-subset* selection [Garside, 1965], which achieves sparser solutions, but suffers from high variability and computational difficulties [Fan and Li, 2001]. In contrast, the *ridge regression* [Hoerl and Kennard, 1970], a continuous shrinkage method that adds to the objective function an l_2-norm penalty over the coefficients to be estimated, achieves its better accuracy prediction through a bias-variance tradeoff. Nevertheless, ridge regression is known not to be able to render a parsimonious solution. To overcome those shortcomings, Tibshirani [1996] proposed the Lasso regularization technique, which includes an l_1-norm penalization term instead and, thus, achieves both estimation and selection of relevant predictors simultaneously by construction. One of the advantages of the Lasso is that the entire path of solutions can be found thanks to the LARS algorithm [Efron et al., 2004]. In addition, it is well-known that, under some conditions, the Lasso enjoys good theoretical and statistical properties [Donoho et al., 1995; Friedman et al., 2001; Bühlmann and Van-De Geer, 2011].

To visualize the effect of the penalty term in the Lasso formulation, consider the well-known `prostate` database [Stamey et al., 1989], which consists of the measurements of 8 predictors and one response variable (clinical measures) on 97 men who were about to receive a radical prostatectomy. If the goal is to minimize the overall mean squared error (MSE), the parameter vector can be estimated by a fitting procedure as OLS. The results obtained under the OLS are shown in the first two rows of Table 1.1, where $3/4$ of the total set has been used to fit the model (training set) and the remaining samples (testing set) to the assessment of the generalization error of the resulting model. The overall MSE, as well as the number of coefficients involved in the model, are presented. The third and fourth rows in Table 1.1 provide the results obtained under the Lasso for the `prostate` dataset. In this case, in comparison with OLS results, a sparser and therefore, a more interpretable solution has been obtained at

Method		*Overall MSE*	*Non-Zero Coefficients*
OLS	Training set	0.344	8
	Testing set	0.373	
Lasso	Training set	0.365	5
	Testing set	0.408	

Table 1.1: Results obtained using `prostate` dataset

the expense of slightly worsening the MSE values.

Many different variants of the Lasso have been proposed. For example, in Zou [2006] adaptive weights for penalizing different coefficients in the l_1 penalty are included as a way for fitting sparser models under more general conditions. Moreover, in the presence of highly correlated predictor variables (as is usual in microarray studies) or when predictors are structurally grouped (e.g. dummy variables), the Lasso sometimes does not perform well and, as a consequence, the *elastic net* of Zou and Hastie [2005] and the *group lasso* [Yuan and Lin, 2006; Simon et al., 2011] were proposed. They combine l_2 and l_1 penalties to try to select (or remove) the correlated or structured predictor variables together. See Hastie et al. [2015] for an extensive review about the Lasso problem and generalizations.

In classification, sparsity is closely linked to the concepts of *Variable Selection* and *Feature Selection* [George and McCulloch, 1993; Zou and Hastie, 2005; Lin et al., 2011; Carrizosa et al., 2016], whose aim is to identify the relevant variables within a set of many predictors so that classification accuracy (the percentage of class labels predicted correctly) is not reduced. First works published on feature selection handled datasets of a few predictors [Blum and Langley, 1997], but the size of data got larger and a number of variable selection techniques were proposed. There exist two main groups of methods that select features: filters [Guyon et al., 2006; Saeys et al., 2007] and wrappers [Kohavi and John, 1997; Saeys et al., 2007]. Broadly speaking, whereas the selection of features in the former group is based on the intrinsic properties of the features in addition to the correlation with the target variable, the latter tries to find a subset of features for training the model, adding or removing features from the subset according to the results drawn from the model. For instance, the filter *Correlation based Feature Selection* (CFS), see Hall [2000], is based on the assumption that a good subset of attributes should be highly correlated with the response variable but, on the other hand, there should exhibit low dependency among them. Other example can be the wrapper *Boruta* [Kursa and Rudnicki, 2010], which is in principle designed using a Random Forest strategy [Breiman, 2001], but can be modified and adapted to any classifier. There also exists an alternative to the use of the filters and wrappers, which has been recently considered in the literature. These are the so-called embedded methods, which embed the feature selection process into the classifier construction. In particular, there are some specific examples of embedded methods proposed for achieving sparse versions of the well-known Support Vector Machine (SVM) classifier [Cortes and Vapnik, 1995], see Aytug [2015], Maldonado et al. [2017], Ghaddar and Naoum-Sawaya [2018],

Blanquero et al. [2019] and Benítez-Peña et al. [2019], whereas Blanquero et al. [2020] look for interpretability for random forests. Concerning the Naïve Bayes (NB) classifier [Hand and Yu, 2001], some works have addressed different strategies for variable reduction. For example, McCallum and Nigam [1998] and Feng et al. [2015] base their feature selection approaches on the univariate correlations between features and the class. In this sense, Tang et al. [2016b] and Tang et al. [2016a] aim to rank the features according to their capacity for classification or a specific feature selection criterion, respectively. See Chandrashekar and Sahin [2014] for a survey of the differences among well-known filter, wrapper and embedded methods.

1.1.2 Cost-sensitive procedures

Most traditional classification approaches search for maximizing accuracy, and do not take into account the differences between types of misclassification errors. However, in many real-world problems, such as those mentioned previously (medical diagnosis, credit card fraud detection), it is more important to achieve better classification rates for the individuals of interest (ill people, defaulting customers), since the consequences of wrong predictions across the classes may be very different.

Cost-sensitive learning methods [Elkan, 2001; Zadrozny and Elkan, 2001] take into consideration different cost matrices that describe the costs for misclassification [Turney, 2000]. These approaches turn out to be very convenient for unbalanced datasets, where the minority class may be the worst classified (and the most critical one). In the case of binary classification (positive (1) and negative (0) classes), the cost matrix has the structure in Table 1.2. Whereas c_{10} is the cost of a false positive (actual negative but predicted as positive), c_{01} is the cost of a false negative. In contrast, c_{00} and c_{11} are associated with correct predictions. It seems natural that the cost of labeling a new individual incorrectly should always be greater than the cost of labeling it correctly.

	actual negative	actual positive
predict negative	c_{00}	c_{01}
predict positive	c_{10}	c_{11}

Table 1.2: Cost matrix for binary classification

In classification contexts, Y identifies the class. Then, the observed dataset is formed by pairs $(y, \mathbf{x})$ from an unknown joint distribution. Given the cost matrix, the classifier will assign a new individual $\mathbf{x}$ to the class that has the minimum expected cost. Thus, if $p(j \mid \mathbf{x})$ is the probability of classifying $\mathbf{x}$ into class j, a new individual $\mathbf{x}$ will be assigned to class 1 if and only if (iif)

$$p(0 \mid \mathbf{x})c_{10} + p(1 \mid \mathbf{x})c_{11} \leq p(0 \mid \mathbf{x})c_{00} + p(1 \mid \mathbf{x})c_{01},$$

or equivalently,

$$p(0 \mid \mathbf{x})(c_{10} - c_{00}) \leq p(1 \mid \mathbf{x})(c_{01} - c_{11}).$$

As can be observed, the optimal decision is unchanged if a constant is added to each column of the cost matrix, which implies a simpler matrix where c_{00} and c_{11} are equal to 0. Finally, as $p(0 \mid \mathbf{x}) = 1 - p(1 \mid \mathbf{x})$, the classifier will assign the positive class to the new individual iif

$$p(1 \mid \mathbf{x}) \geq \frac{c_{10}}{c_{10} + c_{01}}. \tag{1.1}$$

Likewise, this procedure can be extended when more than two classes are considered, where more rows and columns are adding to the cost matrix.

The classification threshold in (1.1) can be used by those classifiers which produce probability estimates, thus becoming cost-sensitive. Otherwise, misclassification costs can be directly introduced into the classifier construction. See Bradford et al. [1998], Freitas et al. [2007], Carrizosa et al. [2008], Sun et al. [2009], Datta and Das [2015] and Lee et al. [2017] for more details and applications. As some examples, consider Datta and Das [2015], Carrizosa et al. [2008] and Lee et al. [2017], which focus on the SVM classifier. In Datta and Das [2015] the decision boundary shift is combined with unequal misclassification penalties. On the other hand, in Carrizosa et al. [2008] a biobjective problem, which simultaneous minimizes the misclassification rates, is performed. In Lee et al. [2017], the authors propose a new weight adjustment factor that is applied to a weighted SVM. In the context of decision trees, Freitas et al. [2007]; Ling et al. [2004] introduce tree-building strategies which choose the splitting criterion by minimizing the misclassification costs, whereas Bradford et al. [1998] performs the pruning of a subtree following the cost information. Cost-sensitive versions of neural networks for unbalanced data classification have also been studied in the literature [Cao et al., 2013; Zhi-Hua Zhou and Xu-Ying Liu, 2006]. Other approaches can be found, for example in Peng et al. [2014], where a new version of the so-called data gravitation-based classification model is proposed.

Although the term *cost-sensitive learning* has been mainly exploited in classification contexts [He and Ma, 2013; Prati et al., 2015], it can also be extended to regression contexts, where the response variable is quantitative but the whole sample is splitted into different groups (or classes). In particular, one can think of a Lasso model whose objective function includes weighted penalties over the MSE of each group (quadratic penalties). Those penalties would play the role of the costs introduced in Table 1.2. In fact, new versions of cost-sensitive regression models have been recently proposed in the literature (see Ollier and Viallon [2017] and reference therein).

Under the umbrella of cost sensitive learning, it is generally assumed that misclassification costs are given and known. Unfortunately, fixing precise values for such misclassification costs may be problematic in real-world applications. In addition, in this way only an indirect control on misclassification rates is obtained. The application of mathematical optimization tools, the

approach undertaken in this thesis, seems to be promising [Carrizosa and Romero Morales, 2013] and not fully explored: one overall criterion is to be optimized, while constraints are introduced in the model to demand admissible values for the efficiency measures under consideration.

1.2 The Bayesian Paradigm

Different disciplines, such as Statistics, are required to obtain theoretical results for the effectiveness of sparse and cost-sensitive statistical learning methods. Bayesian decision theory [Berger, 2013], involves an assortment of inference techniques that can be used for estimating probabilities of interest associated with the properties of learning algorithms. This section is devoted to introduce the Statistical Learning through a Bayesian decision theory perspective.

1.2.1 Bayesian Statistical Learning

To select the best function f to predict the response Y in terms of the predictor variables $\mathbf{X}$, the loss function $L(f(\mathbf{X}), Y)$ is used. To characterize the performance of f, the expected value of $L(f(\mathbf{X}), Y)$, which is called the risk, is calculated. Given the observed sample $\mathbf{X} = \mathbf{x}$, the expression

$$R(f) = E_{\mathbf{X}}[E_{Y|\mathbf{X}}(L(f(\mathbf{X}), Y) \mid \mathbf{X})],$$

is known as the posterior expected loss or risk. Then, the best function f is that that minimizes the posterior expected loss. Such solution is known as the Bayes estimator under the loss function L. In the case of quantitative Y, the most common choice for L is the squared error loss, $L(f(\mathbf{X}), Y) = (f(\mathbf{X}) - Y)^2$, which results in the regression function $f(\mathbf{X}) = E(Y \mid \mathbf{X} = \mathbf{x})$. Otherwise, when dealing with categorical output Y, a different loss function for penalizing prediction errors is needed.

If, instead of a regression problem, we are interested in a classification problem with K possible classes in $\mathcal{C}$, then a matrix $\mathcal{L} \in \mathcal{M}_K(\mathbb{R})$ is introduced. Whereas its diagonal will be zero, extra-diagonal values, which represent the price paid for wrong predictions, will usually be equal to one. Then, the risk is defined as

$$R(f) = E_{\mathbf{X}}\left[\sum_{k=1}^{K} \mathcal{L}[C_k, \hat{C}(\mathbf{X})] p(C_k \mid \mathbf{X})\right],$$

where C_k means the class k, $k \in \{1, \ldots, K\}$, and the estimate $\hat{C}$ assumes values in $\mathcal{C}$. Under the 0-1 loss function choice,

$$\hat{C}(\mathbf{x}) = \underset{c \in \mathcal{C}}{arg\,max}\ p(c \mid \mathbf{X} = \mathbf{x}),$$

or equivalently

$$\hat{C}(\mathbf{x}) = C_k \text{ if } p(C_k \mid \mathbf{X} = \mathbf{x}) = \max_{c \in \mathcal{C}} p(c \mid \mathbf{X} = \mathbf{x}).$$

This method, which classifies each instance $\mathbf{x}$ in the most probable class via the conditional distribution, is known as the *Bayes classifier*.

The computation of $p(C_k \mid \mathbf{x})$ may be cumbersome if the number of features p is large. However, the use of the Bayes theorem eases the previous computation since

$$p(C_k \mid \mathbf{x}) = \frac{\pi(C_k)p(\mathbf{x} \mid C_k)}{p(\mathbf{x})},$$

where $\pi(C_k)$ is the prior distribution for the class, $p(\mathbf{x} \mid C_k)$ is the likelihood function of the data and $p(\mathbf{x})$ is the so-called evidence. Since the evidence is the same for all the classes, in practice, the interest is in computing the numerator.

Among the assortment of current classification techniques, the NB classifier has played a prominent role because of its simplicity, tractability and efficiency, see Hand and Yu [2001]. The method is based on the assumption of conditional independence of the features to the class,

$$p(\mathbf{x} \mid C_k) = p(x_1, \ldots, x_p \mid C_k) = p(x_1 \mid C_k) \ldots p(x_p \mid C_k), \tag{1.2}$$

which notably simplifies the computation of the class probability.

One of the advantages of the NB is that it usually estimates fewer parameters than other renowned classifiers, so it is less prone to make overfitting [Domingos and Pazzani, 1997; Hand and Yu, 2001]. As a consequence, a number of applications of the NB in real contexts can be found, for example, in medicine [Wolfson et al., 2015], genetics [Minnier et al., 2015], reliability [Turhan and Bener, 2009], risk [Minnier et al., 2015] or document analysis [Guan et al., 2014], among others.

1.2.2 Bayesian Inference

As commented at the beginning of this section, Statistical Learning methods can be viewed from a Bayesian paradigm. Furthemore, when dealing with datasets of especial characteristics (for example, heavy tails in contexts as Economics and Finance), Bayesian statistical techniques have shown their potential in the literature [Owen and Rabinovitch, 1983; Andrews et al., 1993; Ramírez-Cobo et al., 2010; Fortunati et al., 2020].

The Bayesian inference scheme can be briefly described as follows. Let X be the random variable whose distribution depends on the unknown parameter, $\theta \in \Theta$, to be estimated. Given a sample x, the model or density function is denoted by $f(x \mid \theta)$, which, as a function of θ, $L(\theta)$, is called the likelihood function. In the Bayesian paradigm, the parameter θ is treated as a random variable, and thus is associated with a (prior) distribution function. Once the sample is observed, the likelihood function helps to update the prior distribution into the posterior distribution, which is the final target of the inference process. Two main advantages of performing Bayesian data analysis versus the classical (frequentist) framework should be pointed

out. First, the available prior information about θ can be coherently incorporated into the statistical model, that is, experts can input prior knowledge in the process of modeling. Second, given the observed data, the posterior distribution, encompasses a more extensive information than that of a puntual estimation made in frequentist way.

Fundamentals

As commented before, the unknown parameter θ is considered a random variable with prior distribution $\pi(\theta)$, which describes uncertainty related to this unknown parameter before data are observed. In addition, in the case the prior distribution is specified up to other parameters, they are called the hyperparameters.

Once the data have been observed ($X = x$) the posterior distribution for θ is denoted by $\pi(\theta \mid x)$. Thanks to Bayes rule, $\pi(\theta \mid x)$ is calculated dividing the joint distribution by the marginal distribution, $m(x)$, that is,

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\pi(\theta)}{m(x)} = \frac{f(x \mid \theta)\pi(\theta)}{\int_\Theta f(x \mid \theta)\pi(\theta)d\theta}. \tag{1.3}$$

If the posterior distribution remain in the same family as the prior distribution, we will say that we have a conjugate structure. In fact, computationally speaking, this case is the most convenient. However, since simple conjugate analysis has a limited modeling capability of real-life data, to achieve a closed form for the posterior is not a rule. Therefore, sophisticated techniques in the literature, such as Markov Chain Monte Carlo (MCMC) methods, are required for Bayesian computation, see[Gelfand and Smith, 1990] and the following section for further details.

Posterior simulation

The two main difficulties of Bayesian inference are the prior elicitation and the computation of the posterior distribution (1.3). However, in practice, it does not have a closed form and, therefore, advanced techniques are required. As commented previously, MCMC methods allow for sampling from an unknown distribution. Robert and Casella [2013] gives the following definition:

> A *Markov Chain Monte Carlo (MCMC) method* for the simulation of a distribution f is any method producing an ergodic Markov chain $(X^{(t)})$ whose stationary distribution is f.

Therefore, these methods are based on the construction of a Markov chain, say $\theta^{(j)}$, whose stationary distribution is equal to the (posterior) distribution of interest.

One of the most popular MCMC algorithms is the Metropolis-Hastings algorithm. Assume that the unknown parameter is indeed a multivariate parameter, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$. Therefore,

the objective is to find the posterior distribution of $\boldsymbol{\theta}$. Given a *proposal* or conditional distribution $q(\tilde{\boldsymbol{\theta}} \mid \boldsymbol{\theta}^{(j)})$, which generates a candidate at iteration $j+1$ based on the previous accepted value $\boldsymbol{\theta}^{(j)}$, the candidate is accepted with certain probability. A summary of the Metropolis-Hastings algorithm is described in Algorithm 1.

Algorithm 1: Metropolis-Hastings

1. Set initial values $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$.
2. $j = 1$
3. Generate $\tilde{\boldsymbol{\theta}} \sim q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(j)})$.
4. Calculate $\rho = \rho(\boldsymbol{\theta}^{(j)}, \tilde{\boldsymbol{\theta}}) = \min\left\{ \dfrac{\pi(\tilde{\boldsymbol{\theta}}) q(\boldsymbol{\theta}^{(j)} \mid \tilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{\theta}^{(j)}) q(\tilde{\boldsymbol{\theta}} \mid \boldsymbol{\theta}^{(j)})}, 1 \right\}$
5. Set $\boldsymbol{\theta}^{(j+1)} = \begin{cases} \tilde{\boldsymbol{\theta}} & \text{with probability} \quad \rho \\ \boldsymbol{\theta}^{(j)} & \text{with probability} \quad 1-\rho \end{cases}$
6. Increase $j = j+1$ and return to 3.

This algorithm always accepts values with a higher likelihood ratio than the previous value. Note that for a given target distribution π, the *proposal* q must satisfy that π is contained in the union of supports of all conditional distributions $q(. \mid \boldsymbol{\theta})$. Necessary conditions on the *proposal* distribution q, to ensure the convergence to the limiting distribution of the chain (π), can be studied in depth in Chapter 6 of Robert and Casella [2013].

The Gibbs sampler [Gelfand, 2000], which is used for obtaining samples from a joint density function, is a special case of Metropolis-Hastings algorithm when the *proposal* distribution q is fixed as the conditional distribution $f(\boldsymbol{\theta}_i \mid \boldsymbol{\theta}_{-i})$, where $\boldsymbol{\theta}_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p)$, that are assumed to be known. Algorithm 2 describes how Gibbs sampling works.

Algorithm 2: Gibbs sampler

1. Set initial values $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})$.
2. $j = 1$
3. Generate $\theta_1^{(j+1)} \sim \theta_1 \mid \boldsymbol{\theta}_{-1}^{(j)}$. Update $\boldsymbol{\theta}^{(j)}$.
4. Generate $\theta_2^{(j+1)} \sim \theta_2 \mid \boldsymbol{\theta}_{-2}^{(j)}$. Update $\boldsymbol{\theta}^{(j)}$.
5. $\vdots$
6. Generate $\theta_p^{(j+1)} \sim \theta_p \mid \boldsymbol{\theta}_{-p}^{(j)}$. Update $\boldsymbol{\theta}^{(j)}$.
7. $j = j+1$. Go to 3.

Obviously, at each step j, $\rho(\theta_i^{(j)}, \boldsymbol{\theta}_{-i}^{(j)}, \tilde{\theta}) = 1$, which implies that each update in the Gibbs algorithm is accepted. Discussion about how to fix updating order can be found in Gilks et al. [1994].

In order to solve complex problems, a mixture of different MCMC algorithms (hibrid methods) are usually implemented. For instance, Müller [1991] proposes using Gibbs sampler

steps when the conditional distributions of the involved parameters provide for easy simulation and Metropolis-Hasting sampler, otherwise.

The reader is referred to Insua et al. [2012], Gelman et al. [2013] and Robert and Casella [2013] for a complete review of benchmark posterior simulation techniques. It should be noted that, in recent years and due to the computing capacities, these simulation methods (although established on the same basis) have been computationally enhanced yielding advanced models such as Hamiltonian Monte Carlo, among others (see Barbu and Zhu [2020] for a complete review).

Parametric, Non parametric and semi-parametric paradigms

A full Bayesian analysis of an experiment requires a precise choice of the prior distributions for each of the parameters in the model. Parametric inference is based on known distributions with unknown parameters which must be inferred (Normal, Gamma, Poisson, Beta, and so forth). Nevertheless, when we deal with complex experiments, it becomes more complicated to attribute the generated data to any well-known distribution. That is why nonparametric inference is coming into play. Its main idea is to draw inference on an unknown distribution function, leading to models on function spaces and dramatically changing the methodologies used until then. The insertion of nonparametric distributions for some of the unspecified priors in parametric models, whereas the rest are known distribution functions with possibly unknown parameters, yields the semi-parametric inference, which is commonly used for survival analysis [Lawless, 1982].

A plethora of nonparametric Bayes methodologies are proposed in the literature [Müller and Quintana, 2004], but one of the most popular is the Dirichlet Process (DP) prior [Ferguson, 1973, 1974], which is also the first prior developed for spaces of distribution functions.

Definition 1. *Let F be a specified distribution function and α a positive scalar parameter. For k-dimensional $\boldsymbol{\theta} \mid F \overset{i.i.d.}{\sim} F$, the DP prior on the distribution F, which is denoted by*

$$F \mid \alpha, \boldsymbol{\eta} \sim DP(\alpha F_{\boldsymbol{\eta}}),$$

is a prior on the set of probability distributions on $\mathbb{R}^k$, where $F_{\boldsymbol{\eta}}$ is a specified parametric probability distribution.

1.3 Contributions of this thesis

This dissertation is devoted to design novel computational methods to deal with datasets of today by means of two powerful tools such as Mathematical Optimization and Bayesian Data Analysis. Whereas chapters 2 and 3 are devoted to regression problems, chapters 4 and 5 deal with the Naïve Bayes classifier. Finally, Chapter 6 considers Bayesian inference for the general

class of Elliptical distribution. In what follows we briefly introduce and motivate the problems addressed.

Chapter 2 is based on the work Blanquero et al. [2021b]. The Lasso has become a benchmark data analysis procedure, and numerous variants have been proposed in the literature. Although the Lasso formulations are stated so that overall prediction error is optimized, no full control over the accuracy prediction on certain individuals of interest is allowed. In this chapter we propose a novel version of the Lasso in which quadratic performance constraints are added to Lasso-based objective functions, in such a way that threshold values are set to bound the prediction errors in the different groups of interest (not necessarily disjoint). As a result, a constrained sparse regression model is defined by a nonlinear optimization problem. This cost-sensitive constrained Lasso has a direct application in heterogeneous samples where data are collected from distinct sources, as it is standard in many biomedical contexts. Both theoretical properties and empirical studies concerning the new method are explored. In addition, two illustrations of the method on biomedical and sociological contexts are considered.

Chapter 3 is based on the work Carrizosa et al. [2020]. In this chapter, we study linear regression models built on categorical predictor variables that have a hierarchical structure, with their categories arranged as a directed tree. While the categories in the leaf nodes give the highest granularity in the representation of these variables, the user may decide to go upstream the tree and consolidate individuals at ancestor nodes, sharing the same coefficient. This reduced model, with fewer coefficients to be estimated, is easier to interpret, and hopefully does not damage the accuracy. We study the mathematical optimization problem that trades off the accuracy of the reduced linear regression model and its complexity, measured as a cost function of the level of granularity of the representation of the hierarchical categorical variables. We show that finding non-dominated outcomes for this problem boils down to solving a Mixed Integer Convex Quadratic Problem with Linear Constraints. We illustrate our approach in two real-world datasets, as well as a synthetic one, where our methodology finds a much less complex model with a very mild worsening of the accuracy.

The Naïve Bayes has proven to be a tractable and efficient method for classification in multivariate analysis. However, features are usually correlated, a fact that violates the Naïve Bayes' assumption of conditional independence, and may deteriorate the method's performance. Moreover, datasets are often characterized by a large number of features, which may complicate the interpretation of the results as well as slow down the method's execution. In Chapter 4 we propose a sparse version of the Naïve Bayes classifier that is characterized by three properties. First, the sparsity is achieved taking into account the correlation structure of the covariates. Second, different performance measures can be used to guide the selection of features. Third, performance constraints on groups of higher interest can be included. Our proposal leads to a smart search, which yields competitive running times, whereas the flexibility in terms of performance measure for classification is integrated. Our findings show that, when compared against well-referenced feature selection approaches, the proposed sparse

Naïve Bayes obtains competitive results regarding accuracy, sparsity and running times for balanced datasets. In the case of datasets with unbalanced (or with different importance) classes, a better compromise between classification rates for the different classes is achieved.

As commented before, the consequences of misclassifications may be rather different in different classes, making it crucial to control misclassification rates in the most critical and, in many real-world problems, minority cases, possibly at the expense of higher misclassification rates in less problematic classes. One traditional approach to address this problem in NB classification consists of assigning misclassification costs to the different classes and applying the Bayes rule, by optimizing a loss function. However, fixing precise values for such misclassification costs may be problematic in real-world applications. In Chapter 5 we address the issue of misclassification for the NB classifier. Instead of requesting precise values of misclassification costs, threshold values are used for different performance measures. This is done by adding constraints to the optimization problem underlying the estimation process. Our findings show that, under a reasonable computational cost, indeed, the performance measures under consideration achieve the desired levels yielding a user-friendly constrained classification procedure.

Elliptical distributions are generalizations of the multivariate normal distribution to both longer tails and elliptical contours. Elliptically contoured distributions were introduced in Schoenberg [1938]; Lord [1954] in order to preserve the symmetry of the normal distributions but to permit the incorporation of different tail behaviour. A particular sub-class of the elliptical distributions, which includes many of the most well known models such as normal, Student's t, contaminated normal and slash distributions, are the normal/independent (NI) distributions introduced in Andrews and Mallows [1974]. In Chapter 6, we examine how we can use Bayesian methods to perform semi-parametric inference for elliptical and NI distributions using Dirichlet process mixture models.

Finally, in Chapter 7 some conclusions and open problems are briefly discussed.

Chapter 2

A cost-sensitive constrained Lasso

In this chapter a novel version of the Lasso, in which quadratic performance constraints are added to Lasso-based objective functions, is proposed. Threshold values are set to bound the prediction errors in the different groups of interest (not necessarily disjoint). As a result, a constrained sparse regression model is defined by a nonlinear optimization problem. Theoretical properties as well as empirical studies concerning the new method are explored.

2.1 Introduction

Let $(Y, \mathbf{X})$ be a random vector, where $\mathbf{X} = (X_1, \ldots, X_p)$ is the vector of p predictors and Y identifies the continuous response variable. Given the observed response vector $\mathbf{y} = (y_1, \ldots, y_n)'$, $n > p$, and the related observed predictors, $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})'$, $j = 1, \ldots, p$, the linear regression model predicts $\mathbf{y}$ by

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \hat{\beta}_1 \mathbf{x}_1 + \ldots + \hat{\beta}_p \mathbf{x}_p.$$

Consider again the well-known `prostate` database, which consists of the measurements of $p = 8$ predictors and one response variable (clinical measures) on $n = 97$ men who were about to receive a radical prostatectomy. Further, assume that the dataset is divided into two groups: *Group 1*, corresponding to *young* individuals (aged less than 65) and *Group 2*, related to the population older than 65. As commented in Chapter 1, if the goal is to minimize the overall mean squared error (MSE), the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$ can be estimated by the fitting procedure OLS, yielding $\hat{\boldsymbol{\beta}}^{ols}$. The results obtained under the OLS are shown in the first two rows of Table 2.1, with the training and testing sets as in Chapter 1. The overall MSE, the prediction errors over the two groups as well as the number of coefficients involved in the model are presented.

Method		*Overall MSE*	*Group 1 MSE*	*Group 2 MSE*	*Non-Zero Coefficients*
OLS	Training set	0.344	0.355	0.333	8
	Testing set	0.373	0.380	0.367	
Lasso	Training set	0.365	0.397	0.335	5
	Testing set	0.408	0.414	0.403	
CSCLasso	Training set	0.355	0.357	0.352	6
	Testing set	0.393	0.399	0.388	

Table 2.1: Results obtained using `prostate` dataset

Once the model is fitted, there are two fundamental criteria for evaluating its performance: the accuracy of prediction and the identification of significant predictors, which provides a good interpretation of the solution. It is well-known that $\hat{\boldsymbol{\beta}}^{ols}$ is not sparse, as can be observed from Table 2.1 where the eight predictor variables have been used by the model. To overcome those shortcomings, the Lasso regularization technique can be used. Given $\mathcal{X} = [\mathbf{1} \mid \mathbf{x}_1 \mid \ldots \mid \mathbf{x}_p]$

the predictor matrix; then, the Lasso solution can be defined as

$$\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 + \lambda\|(\beta_1, \ldots, \beta_p)\|_1 \tag{2.1}$$

where $\lambda \geq 0$ is a tuning parameter and $\|.\|_1$ is the l_1 norm. To visualize the effect of the penalty term in the Lasso formulation, consider the third and fourth rows in Table 2.1, which provide the results obtained under the Lasso for the `prostate` dataset.

However, the Lasso presents some limitations; in particular, the literature related to the Lasso has not undertaken the problem of fully controlling the accuracy prediction on certain individuals of interest. In the previous `prostate` database, assume for instance that we are interested in fitting a sparse regression model to the dataset where, apart from obtaining a small overall mean squared error, also the prediction error for the *young* individuals should not exceed a given threshold. In this work we propose a Lasso-based model that allows for such aim, namely the cost-sensitive constrained Lasso, denoted from now on as CSCLasso. The results obtained for the `prostate` database under the CSCLasso, whose definition and main properties shall be discussed in Section 2.2 and 2.3, are shown in the last two rows of Table 2.1. A threshold for the mean squared error over *Group 1* is set equal to 0.357, which represents an improvement of 10% over the prediction error of the Lasso (0.397). Note that in the training set the MSE satisfies the imposed constraint, as expected. Also note that the improvement in *Group 1* is at the expense of slightly increasing the prediction error over *Group 2*. In terms of sparsity, the CSCLasso model has needed an additional predictor variable comparing to Lasso in order to comply with the constraint.

As it will be seen in Section 2.2, the novel approach is set up by adding convex quadratic constraints to the Lasso formulation, and aims to control the performance measure on certain groups of interest. Other approaches have considered constrained versions of the Lasso before, see for example James et al. [2020], Gaines et al. [2018], Torres-Barrán et al. [2018], Hu et al. [2015] and references therein. In such works, equality or/and inequality linear constraints are considered for imposing prior knowledge and structure onto the coefficient estimates. In our approach instead, quadratic convex constraints are formulated and thus, our approach and results generalize those previously obtained in the literature.

Not only constrained versions of the Lasso can be found in the literature. Indeed, many different variants have been proposed. In addition to those introduced in Chapter 1, e.g. the *elastic net* or the *group lasso*, other extension is to consider

$$\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \arg\min_{\boldsymbol{\beta}} \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1. \tag{2.2}$$

instead of (2.1), where $\mathcal{A}$ is a fixed matrix (see Tibshirani and Taylor [2011]). If $\mathcal{A} = (0|\mathcal{I}_p)$, then the Lasso objective function is obtained; however, other forms of $\mathcal{A}$ different from the identity can be found in the literature, see for example Ollier and Viallon [2017]. In fact,

various choices of $\mathcal{A}$ in (2.2) define problems that are already well-known in the literature as the *fused lasso* [Tibshirani et al., 2005].

This chapter is structured as follows. In Section 2.2, the cost-sensitive constrained Lasso (CSCLasso) is introduced and some key issues are discussed. Section 2.3 considers theoretical properties of the CSCLasso, as the existence and uniqueness of solution, limit behaviour (in terms of the penalty parameter) and consistency. Section 2.4 presents a detailed numerical analysis with both simulated and real datasets, and finally, some conclusions are provided in Section 2.5. Technical proofs are relegated to Appendix A.

2.2 The cost-sensitive constrained Lasso: definition and key aspects

This section presents the cost-sensitive constrained Lasso, which, as will be seen, is defined through an optimization problem with constraints related to prediction errors for individuals of interest. In addition, some computational details, as well as different key aspects concerning the tuning parameters of our proposal, are presented.

2.2.1 Definition

The proposed CSCLasso is a novel variant of the Lasso where we shall demand that the prediction errors for the groups of interest are below certain threshold values,

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \\ \text{s.t.} \quad & \frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}\|^2 - f_1 \leq 0, \\ & \qquad \vdots \\ & \frac{1}{n_L}\|\mathbf{y}_L - \mathcal{X}_L\boldsymbol{\beta}\|^2 - f_L \leq 0. \end{aligned} \tag{2.3}$$

In the previous formulation, $(\mathbf{y}_0, \mathcal{X}_0)$ is the set of observations used to build the sparse model with overall minimum MSE, which can be the complete dataset $(\mathbf{y}, \mathcal{X})$, or a subset of smaller size. Additionally, let $(\mathbf{y}_l, \mathcal{X}_l)$, $l = 1, \ldots, L$, define groups of interest (not necessarily disjoint), where the MSE predictions are to be controlled. Then, n_l is the number of instances related to group l. Finally, $f = (f_1, \ldots, f_L)$ contains the different threshold values for the MSE on the different groups. The solution of optimization problem (2.3) will be denoted by $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$. From the formulation (2.3) it is natural to wonder whether running a Lasso on just the groups of interest is more advantageous. However, if a single Lasso is run on the groups of interest, dramatically bad predictions can be obtained when the resulting model is applied to new observations outside those groups, which is not the case for our approach. The same issue arises when a different Lasso model is built on each group of interest, but, in addition, new

observations are not given with their group of origin. Contrary to what happens with our novel approach (2.3), the L predictions obtained through the L different estimated Lasso models may not be suitable to give a final prediction for such new samples.

The proposed method can be formulated as a Lasso with weighted quadratic penalties in the objective function associated with the different groups, but finding real meaning to their parameters (one per group) to be chosen is not an easy task (see Carrizosa and Romero Morales [2001] and the references therein) and the full control over the accuracy prediction on certain individuals of interest would disappear. However, the parameters $f = (f_1, \ldots, f_L)$ involved in our model have a clear interpretation and, in addition, this formulation enables us to bound the prediction errors in the different groups of interest.

As an example, in the illustration of the method in Section 2.1 related to the `prostate` dataset, whereas the training set was used in the objective function with $\mathcal{A} = (0|\mathcal{I}_8)$, the prediction error over the *young* population of the training set, $(\mathbf{y}_1, \mathcal{X}_1)$, is controlled through a performance constraint ($f_1 = 0.357$). In a real application, once the L groups of interest are selected by the user, threshold values $f_1, \ldots, f_L$ have to be fixed. Note that these thresholds will depend directly on the dataset in question and the considered groups of interest. As a first option, they could be fixed by the user according to her demand, but therefore unfeasibility problems may appear when solving the CSCLasso problem (2.3). For that reason, in Section 2.2.3 two procedures for determining such threshold values so that (2.3) is feasible are given.

Next, some other aspects related to the formulation of the CSCLasso and its resolution will be discussed.

2.2.2 Computational details

The CSCLasso problem as defined by (2.3) is a non-differentiable convex optimization problem with quadratic and convex constraints. However, if we rewrite the non-differentiable term in (2.3) as

$$\mathcal{A}\boldsymbol{\beta} = \mathbf{u}^+ - \mathbf{u}^-,$$

where $\mathbf{u}^+ = (u_1^+, \ldots, u_p^+)$ and $\mathbf{u}^- = (u_1^-, \ldots, u_p^-)$ are new vectors of positive auxiliary variables, a differentiable version for the CSCLasso problem (2.3) is obtained in a straightforward manner as

$$
\begin{aligned}
\min_{\boldsymbol{\beta},\mathbf{u}^+,\mathbf{u}^-} \quad & \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}\|^2 + \lambda\sum_{j=1}^{p} u_j^+ + \lambda\sum_{j=1}^{p} u_j^- \\
\text{s.t.} \quad & \frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}\|^2 - f_1 \leq 0, \\
& \qquad\vdots \\
& \frac{1}{n_L}\|\mathbf{y}_L - \mathcal{X}_L\boldsymbol{\beta}\|^2 - f_L \leq 0, \\
& \mathcal{A}\boldsymbol{\beta} = \mathbf{u}^+ - \mathbf{u}^-, \\
& \mathbf{u}^+, \mathbf{u}^- \geq 0.
\end{aligned}
$$

This previous smooth formulation for the CSCLasso eases its resolution notably, since efficient solvers for quadratically constrained programming problems, such as Gurobi [Gurobi Optimization, 2018], are available. In particular, the Gurobi R interface will be used in this work to obtain all numerical results.

Another remark concerning the formulation of the CSCLasso is that, instead of using the sum of squared deviations, least absolute deviations could have been considered. Then, (2.3) would be reduced to a regression problem under linear inequality constraints, as those described in James et al. [2020], Gaines et al. [2018] and Hu et al. [2015]. Nevertheless, to cope the non-differentiability of the absolute value function, a huge number of constraints and new auxiliary variables, which would depend on n, should have been added. Consequently, these constrained approaches are likely to face severe numerical difficulties in practice for large datasets.

2.2.3 The choice of threshold values

As commented in Section 2.2.1, threshold values $f_1, \ldots, f_L$ could be fixed by the user. If the user is too demanding, imposing very low MSE threshold values for (some of) the different groups, the optimization problem may become unfeasible. Although a try-and-error procedure may be used, it would be very helpful to have strategies yielding feasible solutions. Here we propose two procedures for determining $f_1, \ldots, f_L$ in such a way that (2.3) is feasible.

First, we propose a choice of the threshold values so that they are close to the OLS results,

$$
f_l = (1+\tau)\text{MSE}_l(\hat{\boldsymbol{\beta}}^{ols}),\ l = 1, \ldots, L, \tag{2.4}
$$

where $\text{MSE}_l(\boldsymbol{\beta}) = \dfrac{1}{n_l}\|\mathbf{y}_l - \mathcal{X}_l\boldsymbol{\beta}\|^2$, $l = 1, \ldots, L$ and $\tau \geq 0$ is a small parameter whose meaning is the percentage of worsening with respect to the OLS prediction error. For the numerical example in Section 2.1, we could have imposed the threshold for the MSE over *Group 1* equal to 0.391, which is a 10% ($\tau = 0.1$) more than $\text{MSE}_1(\hat{\boldsymbol{\beta}}^{ols}) = 0.355$. The choice (2.4) deals with the heterogeneity coming from the variability of the different groups

(MSE_l is different across groups). Nevertheless, when heterogeneity related to the importance of each group is also considered, the parameter τ can be replaced in (2.4) by τ_l, $l = 1, \ldots, L$.

Next, we shall compute the minimum value of τ, τ_{min}, so as to (2.3) is feasible. That is, the minimum τ so that there exists $\boldsymbol{\beta}^*$ satisfying

$$\left(\max_l \frac{\text{MSE}_l(\boldsymbol{\beta}^*)}{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{ols})} \right) - 1 \leq \tau,$$

and, therefore, τ_{min} will be given as

$$\tau_{min} = \left(\max_l \frac{\text{MSE}_l(\boldsymbol{\beta}^*)}{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{ols})} \right) - 1.$$

Such τ_{min} can be found as the optimal value of the following linear problem with convex and quadratic constraints

$$\begin{aligned} \min_{\boldsymbol{\beta}, z} \quad & z \\ \text{s.t.} \quad & z \geq \frac{\text{MSE}_l(\boldsymbol{\beta})}{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{ols})} - 1, \ \forall l = 1, \ldots, L. \end{aligned} \tag{2.5}$$

The feasible version of the CSCLasso optimization problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{n_0} \|\mathbf{y}_0 - \mathcal{X}_0 \boldsymbol{\beta}\|^2 + \lambda \|\mathcal{A}\boldsymbol{\beta}\|_1 \\ \text{s.t.} \quad & \frac{1}{n_1} \|\mathbf{y}_1 - \mathcal{X}_1 \boldsymbol{\beta}\|^2 - (1+\tau)\text{MSE}_1(\hat{\boldsymbol{\beta}}^{ols}) \leq 0, \\ & \qquad \vdots \\ & \frac{1}{n_L} \|\mathbf{y}_L - \mathcal{X}_L \boldsymbol{\beta}\|^2 - (1+\tau)\text{MSE}_L(\hat{\boldsymbol{\beta}}^{ols}) \leq 0, \end{aligned} \tag{2.6}$$

where $\tau \geq \tau_{min}$.

Finally, note that if τ is big enough, then solving (2.6) is equivalent to solve the unconstrained problem. Indeed, it is possible to find the value of τ, $\tau_{max}(\lambda)$, such that both the constrained and unconstrained problems are equivalent

$$\tau_{max}(\lambda) = \max_{l \in \{1,\ldots,L\}} \frac{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{Lasso}(\lambda))}{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{ols})} - 1. \tag{2.7}$$

A second possible choice for the threshold values follows an analogous approach but, instead of considering the results of the OLS, we shall consider the mean squared error of the

Lasso, as in the numerical example introduced in Section 2.1. For each $l = 1, \ldots, L$,

$$f_l = (1-\gamma)\text{MSE}_l(\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)),\ l = 1, \ldots, L, \tag{2.8}$$

where $\gamma \geq 0$ is related to the desired percentage of improvement over the Lasso solution ($\gamma = 0.1$ in the numerical example of Section 2.1). In this case, we will compute the maximum value of γ, γ_{max}, in such a way that (2.3) is feasible under (2.8), and the linear problem associated with γ_{max} is

$$\begin{aligned} \max_{\boldsymbol{\beta}, z} \quad & z \\ \text{s.t.} \quad & 1 - \frac{\text{MSE}_l(\boldsymbol{\beta})}{\text{MSE}_l(\hat{\boldsymbol{\beta}}^{Lasso}(\lambda))} \geq z,\ \forall l = 1, \ldots, L. \end{aligned} \tag{2.9}$$

Thus, another possible feasible version of the CSCLasso optimization problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \\ \text{s.t.} \quad & \frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}\|^2 - (1-\gamma)\text{MSE}_1(\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)) \leq 0, \\ & \qquad \vdots \\ & \frac{1}{n_L}\|\mathbf{y}_L - \mathcal{X}_L\boldsymbol{\beta}\|^2 - (1-\gamma)\text{MSE}_L(\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)) \leq 0, \end{aligned} \tag{2.10}$$

where $\gamma \leq \gamma_{max}$.

Note that the two choices previously described for selecting the threshold values are not unique. Indeed, instead of using the MSE, another statistical measure as the R-squared can be considered. Further details about how they perform in numerical applications are described in Sections 2.2.4 and 2.4.

2.2.4 The role of the tuning parameters

The CSCLasso, as defined by (2.6) or (2.10), is stated in terms of two tuning parameters, λ and τ or λ and γ, respectively. The first one, λ, is related to the sparsity of the solution, and the second one is linked to the user's demanding level, since the degree of requirement increases as $\tau \to \tau_{min}$ (or $\gamma \to \gamma_{max}$). In this section we investigate how the solution of the CSCLasso changes when λ and τ jointly vary (analogous results are obtained if λ and γ are analyzed instead). With this purpose, consider again the experimental setting as in the example of Section 2.1 related to `prostate` dataset with $\mathcal{A} = (0|\mathcal{I}_p)$ (Lasso objective function), but in this occasion assume that the prediction errors of both groups (the *young* and the *elderly* people) shall be controlled.

The interval of variation of the parameter λ is set to $I_\lambda = [0, 30]$. Moreover, according to (2.5), the smallest value of τ such that the CSCLasso optimization problem (2.6) is feasible is $\tau_{min} = 0.055$. On the other hand, following (2.7), $\tau_{max} = \max_{\lambda \in [0,30]} \tau_{max}(\lambda) = 2.355$, although we will enlarge the interval of variation of τ to also visualize the unconstrained solution; such interval will be finally set as $I_\tau = [\tau_{min}, \tau_{max} + 2] = [0.055, 4.355]$. Figure 2.1 represents, via a heat map, the solution for $\hat{\beta}_1^{CSCLasso}(\lambda)$ for the different values of (λ, τ) in a grid contained in $I_\lambda \times I_\tau$.

Some conclusions can be drawn from the figure. Consider first the cases where τ and λ are big enough. Since, in this case, $\tau \geq \tau_{max}$, then, as commented at the end of the previous section, solving (2.6) is equivalent to solving the Lasso. Therefore, $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda) = \hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \mathbf{0}$ will be the optimal solution, provided that λ is big enough. Analogously, if $\tau \geq \tau_{max}$ but λ is small, then $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda) = \hat{\boldsymbol{\beta}}^{Lasso}(\lambda)$, which will be equal to zero or not depending on the importance of the variable. When τ is small, the constraints are demanding and, even for large values of λ, it might happen that $\hat{\beta}_1^{CSCLasso}(\lambda) \neq \hat{\beta}_1^{Lasso}(\lambda) = 0$, as it is the case.

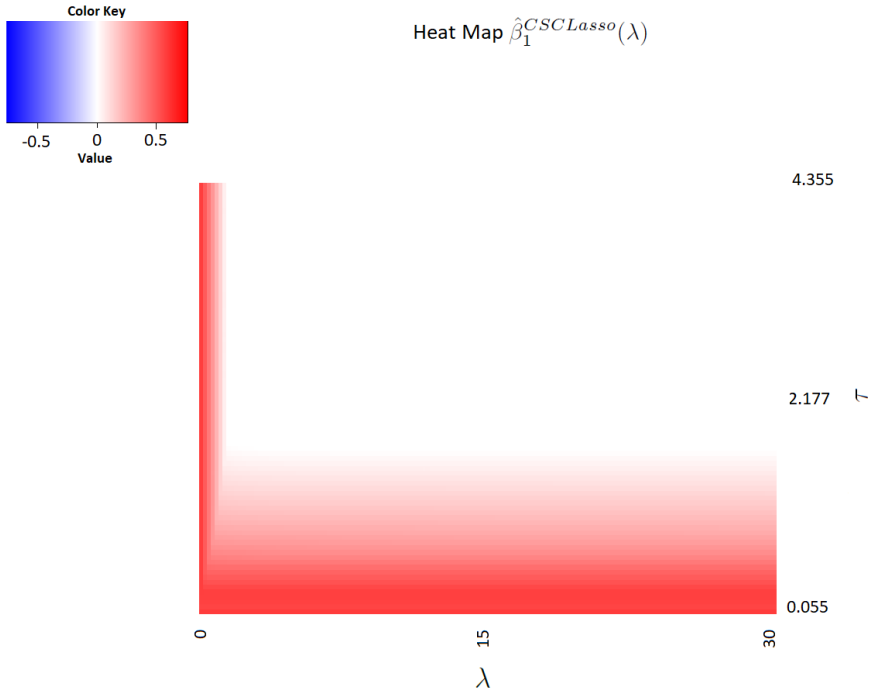


Figure 2.1: Heat map of $\hat{\beta}_1^{CSCLasso}(\lambda)$ using `prostate` dataset

Figure B.1 (see Appendix B for further results) represents the analogous heat maps concerning $\hat{\beta}_2^{CSCLasso}(\lambda), \ldots, \hat{\beta}_8^{CSCLasso}(\lambda)$. A similar discussion as with $\hat{\beta}_1^{CSCLasso}(\lambda)$ is applicable to these figures. An interesting remark to be made concerns the importance of each variable: while variable 1 is the only one selected for the Lasso, the CSCLasso returns a less sparse solution in this case, since predictor variables 1, 2, 4 and 5 turn out to be significant.

However, this is not the rule, since there are examples where the level of sparsity is higher for the CSCLasso, as will be shown in Section 2.4.

2.3 Theoretical properties

In this section we discuss some theoretical results concerning the CSCLasso model. In Section 2.3.1, the existence of a unique optimal solution to Problem (2.3) is proven for a fixed value of $\lambda \geq 0$. Section 2.3.2 deals with the limit behavior of the solution when λ approaches infinity. Finally, some consistency properties of the CSCLasso solution are derived in Section 2.3.3 from the *Sample Average Approximation* theory (see Shapiro et al. [2009]).

2.3.1 Existence and uniqueness of solution

For constrained versions of the Lasso in the literature, as in James et al. [2020], it is not possible to obtain the path of solutions and therefore approximations are made with the use of numerical algorithms. For the CSCLasso problem, a closed form solution of expression $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ is not available. However, an implicit characterization of the CSCLasso solution (with only one constraint) can be found as the following result states.

Proposition 1. *Consider the CSCLasso problem with one constraint,*

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \\ \textit{s.t.} \quad & \frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}\|^2 - (1+\tau)\mathit{MSE}_1(\hat{\boldsymbol{\beta}}^{ols}) \leq 0, \end{aligned} \tag{2.11}$$

where $\mathcal{A} = (0|\mathcal{I}_p)$ and assume that $\mathcal{X}_0$ and $\mathcal{X}_1$ are maximum rank matrices. Then

$$\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda) = \left(\frac{1}{n_0}\mathcal{X}_0'\mathcal{X}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathcal{X}_1\right)^{-1}\left(\frac{1}{n_0}\mathcal{X}_0'\mathbf{y}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathbf{y}_1\right) - \frac{1}{2}\left(\frac{1}{n_0}\mathcal{X}_0'\mathcal{X}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathcal{X}_1\right)^{-1}\mathbf{b}(\lambda) \tag{2.12}$$

where $\eta(\lambda)$ is the Lagrange multiplier of the constraint and the component s, $s = 0, 1, \ldots, p$, of the vector $\mathbf{b}(\lambda)$ is given by

$$b_s(\lambda) = \begin{cases} \lambda, & \textit{if } \hat{\beta}_s^{CSCLasso}(\lambda) > 0, \\ -\lambda, & \textit{if } \hat{\beta}_s^{CSCLasso}(\lambda) < 0, \\ 0 & \textit{else}. \end{cases}$$

From the previous proposition, it is clear that a closed form solution is hard to be obtained, even in the simplest scenario.

Nevertheless, given a fixed value of λ, the CSCLasso problem can be solved using quadratic programming via any of the standard solvers available in the literature. As an example, Figure 2.2 depicts the path of solutions for the `prostate` example introduced in Section 2.1, for an assortment of values of λ in a grid (see Section 2.3.2 for details). Each line represents a component of $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$, $\hat{\beta}_j^{CSCLasso}(\lambda)$ with $j = 1, \ldots, 8$. It can be observed from the figure that, contrary to what happens in the Lasso path of solutions (top panel of Figure 2.2), the CSCLasso path of solutions is not piecewise linear (bottom panel of Figure 2.2). Such non-linearity (due to the quadratic constraints) hinders the application of an iterative algorithm to obtain the path of solutions as those given in papers James et al. [2020] and Gaines et al. [2018].

Also note from Figure 2.3 that as a consequence of the performance constraints, the solution is stabilized when λ increases, but does not shrink to 0, as with Lasso. This is detailed in Section 2.3.2.

Even without having the expression of the general solution of (2.3), we next prove that, under full rank assumptions, the solution is unique. First, in order to simplify the formulation of (2.3), henceforth its feasible set will be denoted by $\boldsymbol{B}$, which is convex and closed. This is also true (and the results which follow remain valid) if, on top of the performance constraints, one adds linear constraints modeling other aspects of interest (for example, the sign of a certain coefficient can be fixed to be positive or negative depending on the known relation between the corresponding predictors with the response variable). In the same vein, henceforth, $(\mathbf{y}_0, \mathcal{X}_0) = (\mathbf{y}, \mathcal{X})$ is used to minimize the overall MSE. In this way, the CSCLasso problem (2.3) is rewritten as

$$\min_{\boldsymbol{\beta} \in \boldsymbol{B}} \quad \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1. \tag{2.13}$$

The following result guarantees that the solution of problem (2.13) is unique.

Theorem 1. *Consider Problem (2.13) where $\mathcal{X}$ is assumed to be a maximum rank matrix and its feasible region $\boldsymbol{B}$ is a convex and closed set in $\mathbb{R}^{p+1}$. Then, Problem (2.13) has a unique optimal solution.*

2.3.2 Asymptotic behaviour

One of the key points when dealing with Lasso-type problems is the choice of the regularization parameter λ. In the case of the Lasso, such choice is straightforward since the entire path of solutions is known to be piecewise linear, shrinking to 0. In particular, it is known that there exists a value of λ, λ^*, such that the solution $\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \mathbf{0}$ is optimal for all values $\lambda \geq \lambda^*$.

The following result provides explicitly the value of such λ^*.

Proposition 2. *Consider the Lasso model (2.2). Define λ^* as the optimal value of the linear*

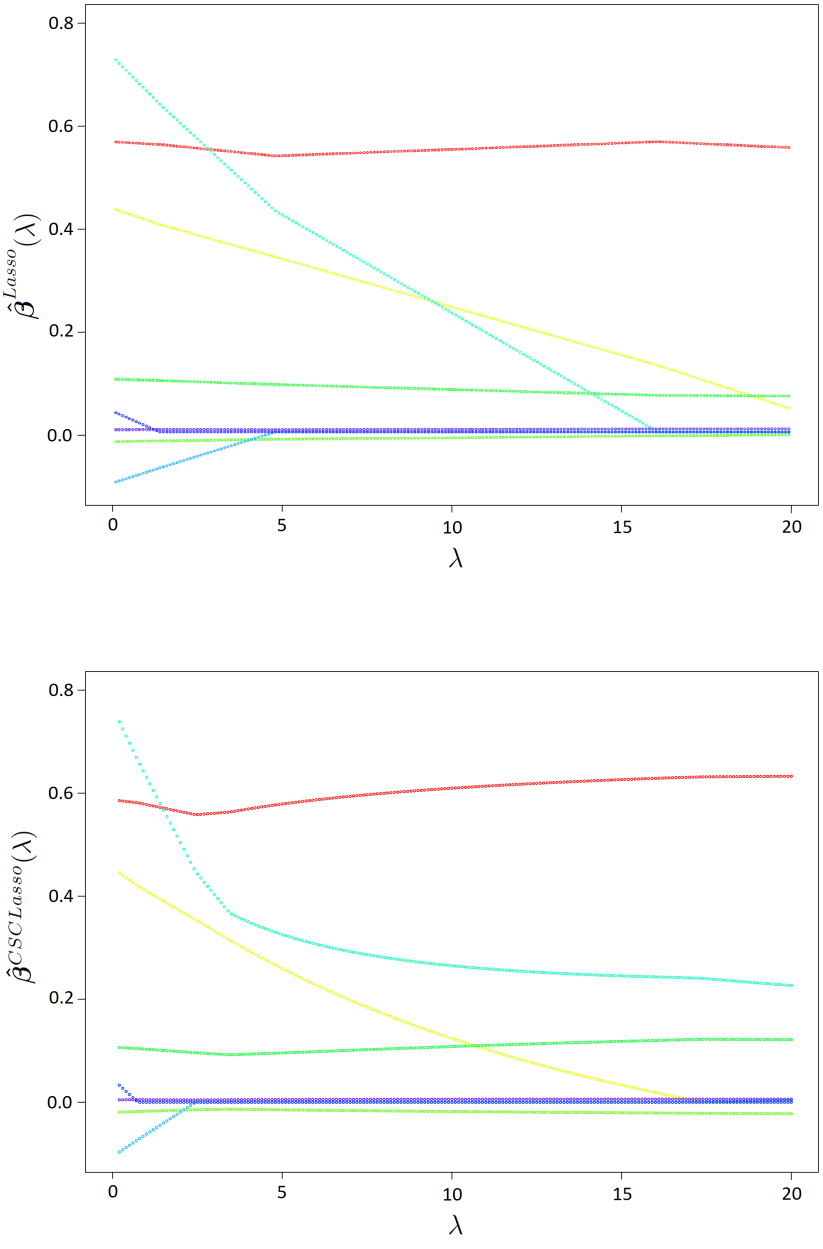


Figure 2.2: Path of solutions under Lasso (top) and CSCLasso (bottom) for `prostate` dataset

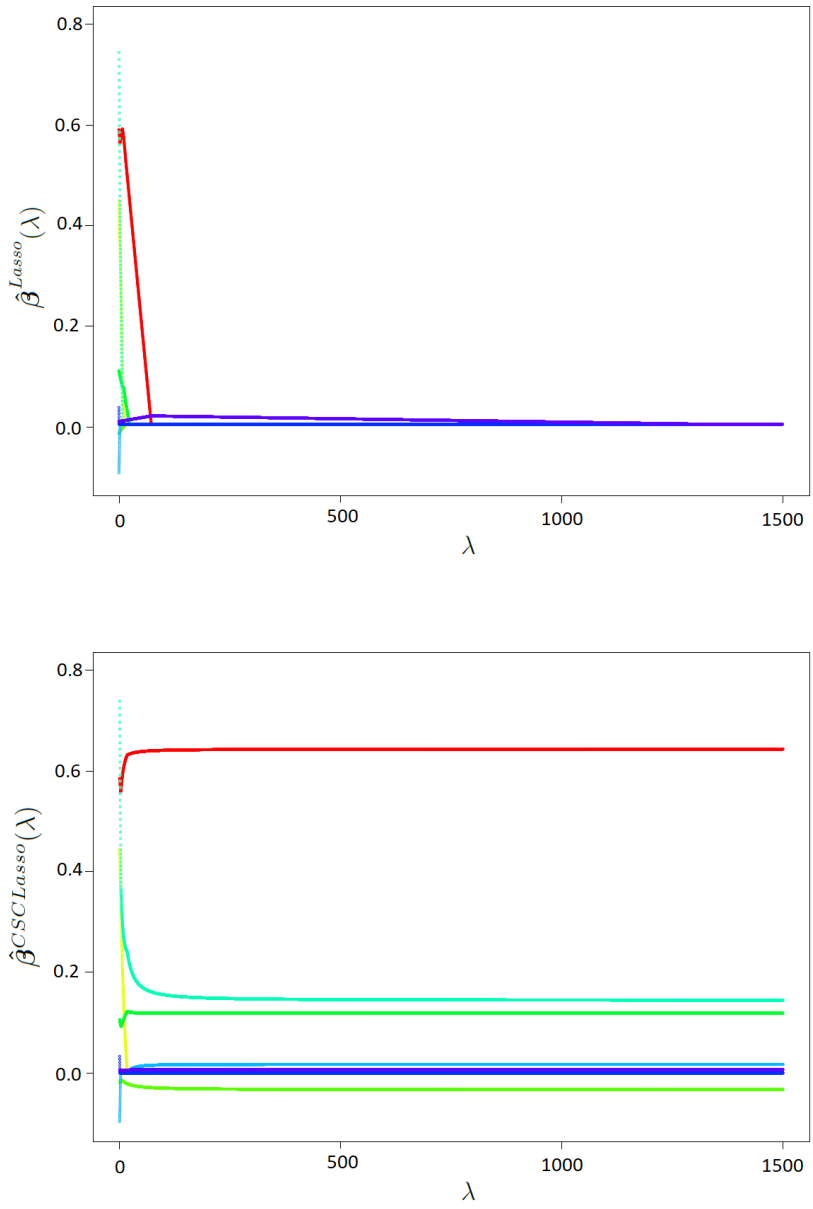


Figure 2.3: Path of solutions under Lasso (top) and CSCLasso (bottom) for `prostate` dataset when λ increases

programming problem

$$\begin{aligned} \min_{z,\mathbf{t}} \quad & z \\ \textit{s.t.} \quad & \frac{2}{n}\mathcal{X}'\mathbf{y} = \mathcal{A}'\lambda\mathbf{t}, \\ & -z \leq \lambda t_s \leq z, \quad s = 0, 1, \ldots, p. \end{aligned}$$

Then, $\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \mathbf{0}$ *for all* $\lambda \geq \lambda^*$. *In particular, for* $\mathcal{A} = (0|\mathcal{I}_p)$,

$$\lambda^* = \left\| \frac{2}{n}\mathcal{X}'\mathbf{y} \right\|_\infty$$

Some works dealing with (linear) constrained versions of the Lasso (as Gaines et al. [2018] and James et al. [2020]) have developed efficient algorithms to build the associated solutions path. Consider now the general problem (2.13). As commented in Section 2.3.1, the expression of $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ is not available in closed form and, consequently, the entire path cannot be computed. In this case, when λ tends to $+\infty$, the solution $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ stabilizes around $\hat{\boldsymbol{\beta}}^{CSCLasso}(+\infty) = \underset{\boldsymbol{\beta}\in\boldsymbol{B}}{\arg\min}\|\mathcal{A}\boldsymbol{\beta}\|_1$. This idea is also used in Gaines et al. [2018] and James et al. [2020], where, in order to find an initialization for the algorithms, the proposed constrained problems are solved by only considering the penalty term in the objective function. Such a limit solution is obtained by solving an optimization problem with a linear objective function and convex quadratic constraints, namely

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{u}^+,\boldsymbol{u}^-} \quad & \sum_{s=0}^{p}(u_s^+ + u_s^-) \\ \text{s.t.} \quad & \boldsymbol{\beta} \in \boldsymbol{B} \\ & \mathcal{A}\boldsymbol{\beta} = \boldsymbol{u}^+ - \boldsymbol{u}^- \\ & \boldsymbol{u}^+, \boldsymbol{u}^- \geq \mathbf{0}. \end{aligned}$$

A grid search is carried out in the general CSCLasso problem (2.13) to obtain suitable values of λ. In order to fix the grid, we propose the following dynamic approach to find an approximate maximum value of λ, λ^* (see Algorithm 3).

Once the maximum value λ^* is found, then the grid ranges from 0 to λ^* with the desired step. Note that the previous algorithm already provides an initial grid of the form $(2^{-5}, 2^{-4}, \ldots, 2^0, \ldots, \lambda^*)$.

2.3.3 Consistency properties in the CSCLasso

The purpose of this section is to prove some results related to the consistency of both the solution and the objective value for CSCLasso problem (2.13). To do that, the theory of *Sample Average Approximation* (SAA) [Shapiro et al., 2009] will be applied. Consider the following

Algorithm 3: Dynamic approach for selecting λ^* in the CSCLasso

1. Fix $\varepsilon > 0$ and $\boldsymbol{c} = (2^{-5})$. Fix $i = 1$ and compute $\hat{\boldsymbol{\beta}}^{CSCLasso}(\boldsymbol{c}[i])$.
2. Compute $\hat{\boldsymbol{\beta}}^{CSCLasso}(+\infty) = \arg\min_{\boldsymbol{\beta}\in\boldsymbol{B}} \|\mathcal{A}\boldsymbol{\beta}\|_1$.
3. While $\|\hat{\boldsymbol{\beta}}^{CSCLasso}(\boldsymbol{c}[i]) - \hat{\boldsymbol{\beta}}^{CSCLasso}(+\infty)\| > \varepsilon$, repeat
   a) $i = i + 1$
   b) $\boldsymbol{c} = (\boldsymbol{c}, 2\boldsymbol{c}[i-1])$
   c) compute $\hat{\boldsymbol{\beta}}^{CSCLasso}(\boldsymbol{c}[i])$
4. $\lambda^* = \boldsymbol{c}[i]$

stochastic programming problem

$$\min_{\boldsymbol{\beta}\in\boldsymbol{B}} \quad f(\boldsymbol{\beta}) := E[F(\boldsymbol{\beta}, (Y, \mathbf{X}))], \tag{2.14}$$

where $\boldsymbol{B}$ is a nonempty closed subset of $\mathbb{R}^{p+1}$, $(Y, \mathbf{X})$ is an absolutely continuous random vector whose probability distribution P is supported on a set $\Xi \subset \mathbb{R}^{p+1}$ and $F : \boldsymbol{B} \times \Xi \to \mathbb{R}$. In Shapiro et al. [2009], under some conditions, the *true* problem (2.14) can be estimated by the SAA:

$$\min_{\boldsymbol{\beta}\in\boldsymbol{B}} \quad \hat{f}_n(\boldsymbol{\beta}) := \frac{1}{n}\sum_{i=1}^{n} F(\boldsymbol{\beta}, (y_i, \mathbf{x}_i)), \tag{2.15}$$

where $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ip})'$, and $\{(y_i, \mathbf{x}_i)\}_{i=1,\ldots,n}$ is a realization of the n random vectors $\{(Y_i, \mathbf{X}_i)\}_{i=1,\ldots,n}$, which are independent and identically distributed (i.i.d.) as the random vector $(Y, \mathbf{X})$. Note that the CSCLasso problem as in (2.13) takes the form of (2.15) as

$$\min_{\boldsymbol{\beta}\in\boldsymbol{B}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \tag{2.16}$$

and the *true* CSCLasso problem equivalent to (2.14) is

$$\min_{\boldsymbol{\beta}\in\boldsymbol{B}} E[(Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1]. \tag{2.17}$$

Before proving the main result on the consistency of the CSCLasso, we first show the uniqueness of the solution of such a problem.

Proposition 3. *The optimal solution of the* true *CSCLasso problem (2.17) is unique.*

Denote by $\nu^{CSCLasso}(\lambda)$ and $\boldsymbol{\beta}^{CSCLasso}(\lambda)$, respectively, the optimal value and the optimal solution of problem (2.17). Analogously, let $\hat{\nu}^{CSCLasso}(\lambda)$ and $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ be the

optimal value and the optimal solution, respectively, of the SAA CSCLasso problem (2.16). The following result shows the consistency of the SAA values to the *true* values.

Theorem 2. *Assume that* $E[\|\mathbf{X}\|^2] < \infty$, $E[Y^2] < \infty$, $E[\|Y\mathbf{X}\|] < \infty$. *Then,* $\hat{\nu}^{CSCLasso}(\lambda)$ *converges to* $\nu^{CSCLasso}(\lambda)$ *and* $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ *converges to* $\boldsymbol{\beta}^{CSCLasso}(\lambda)$ *with probability one (w.p. 1).*

Finally, note that the theoretical results that have been studied in this work are also applicable to other versions of constrained Lasso as long as the feasible set is convex and closed, as is the case with the above-mentioned works James et al. [2020], Gaines et al. [2018] and Hu et al. [2015].

2.4 Numerical experiments

In this section, the behaviour and performance of our approach is illustrated throughout an extensive empirical study. In particular, using both simulated and real datasets, the aim of the experiments shall be to improve the prediction errors of the Lasso in one or more groups of interest. Or in other words, threshold values shall be fixed as in (2.10). Since our proposal is a novel extension of the Lasso, we will also show the results under the Lasso, not only for those groups that are controlled (for which obviously, the Lasso performs worse) but also for the non-controlled groups. In this way the CSCLasso can be better inspected in comparison to the Lasso. Other aspects as the overall MSE and the percentage of non-zero coefficients in the regression model, among others, will be explored. Such measures will be estimated through median values using a 5-fold cross validation approach. To this end, the dataset will be split at each fold into three sets: the so-called training, validation and testing sets. The training set is used to fit the model, the validation set is used to estimate prediction error for model selection and the testing set is used for assessment of the generalization error of the final chosen model.

2.4.1 A simulation study

The generation of the synthetic datasets in this section follows that of Ollier and Viallon [2017], where an overparameterized regression model is considered to cope with stratified data. A number of groups $K = 20$ is set and two different sample sizes per group are considered, $n_k = \{150, 500\}$, for $k = 1, \ldots, K$. The number of predictors p will be chosen from $\{20, 100, 500\}$. The matrix of predictor values $\mathcal{X}$ is generated according to a multivariate normal distribution with zero mean and covariance matrix Σ being a Toeplitz matrix with element (i, j) equal to $0.5^{|i-j|}$. Regarding the response vector, a set of 20 predictors are randomly selected (with indexes included in a set P_0), while the rest of predictors are noise (that is, $\beta_j = 0$ for $j \notin P_0$). The coefficients of the significant 20 predictors are chosen as follows. First, consider 10 random predictors out of the 20 selected. For such predictors, if the group $k > 6$ then $\beta_j = 1$ and, otherwise, $\beta_j = 1 + K^{\frac{1}{2}}$. For the other 10 predictors, $\beta_j = 1$ if $k \leq 6$ and $\beta_j = 1 + K^{\frac{1}{2}}$

otherwise. In this way, the predictors behave differently depending on the group. Finally, the response vector for each group is generated according to the standard linear regression model with normal error.

Once the synthetic dataset is built and its response and predictor variables have been standardized, the CSCLasso with $\mathcal{A} = (0|\mathcal{I}_p)$ is run with constraints imposed over the first six groups. The choice of λ will change at each fold. A grid in λ is built as in Section 2.3.2, and, the value of λ which leads to the lower overall MSE in the validation set is selected. Table 2.2 shows the median prediction errors per group k (MSE_k), $k = 1, \ldots, 6$, obtained by the Lasso (rows in grey color) and the corresponding values obtained under the CSCLasso for different threshold values f.

			$n_k = 150$						$n_k = 300$					
p			Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6
20	MSE_k(*Lasso*)		1.084	0.904	0.768	0.925	0.944	0.674	0.801	0.770	0.966	0.937	0.776	0.938
	Improv. 3%	f	1.052	0.877	0.745	0.898	0.916	0.653	0.777	0.747	0.937	0.909	0.753	0.910
		MSE_k	0.933	0.722	0.695	0.859	0.822	0.574	0.752	0.689	0.810	0.859	0.744	0.870
	Improv. 5%	f	1.030	0.859	0.730	0.879	0.897	0.640	0.761	0.732	0.918	0.890	0.737	0.891
		MSE_k	0.921	0.708	0.689	0.848	0.804	0.564	0.734	0.682	0.787	0.844	0.736	0.859
	Improv. 7%		-	-	-	-	-	-	0.745 0.717	0.717 0.677	0.899 0.768	0.871 0.834	0.722 0.721	0.873 0.847
	Improv. 10%		-	-	-	-	-	-	-	-	-	-	-	-
	Improv. 15%		-	-	-	-	-	-	-	-	-	-	-	-
	Improv. 20%		-	-	-	-	-	-	-	-	-	-	-	-
100	MSE_k(*Lasso*)		1.491	1.139	0.875	1.496	1.151	0.759	1.104	1.151	0.932	0.981	1.311	1.142
	Improv. 3%	f	1.447	1.105	0.848	1.451	1.116	0.736	1.070	1.117	0.904	0.951	1.272	1.108
		MSE_k	1.234	0.948	0.781	1.322	0.945	0.708	1.044	1.145	0.911	0.963	1.306	1.139
	Improv. 5%	f	1.417	1.082	0.831	1.421	1.093	0.721	1.048	1.094	0.885	0.932	1.245	1.085
		MSE_k	1.226	0.941	0.777	1.317	0.937	0.706	1.025	1.123	0.897	0.947	1.311	1.142
	Improv. 7%	f	1.387	1.059	0.813	1.391	1.070	0.706	1.026	1.071	0.866	0.912	1.219	1.062
		MSE_k	1.219	0.933	0.773	1.311	0.929	0.704	1.005	1.111	0.883	0.934	1.309	1.141
	Improv. 10%	f	1.342	1.025	0.787	1.346	1.035	0.683	0.993	1.036	0.838	0.883	1.180	1.028
		MSE_k	1.207	0.921	0.767	1.303	0.917	0.702	0.978	1.094	0.864	0.915	1.311	1.131
	Improv. 15%	f	1.268	0.968	0.743	1.271	0.978	0.645	0.938	0.979	0.792	0.834	1.114	0.971
		MSE_k	1.128	0.903	0.765	1.250	0.919	0.695	0.936	1.061	0.833	0.882	1.274	1.087
	Improv. 20%	f	1.193	0.911	0.700	1.196	0.920	0.607	0.883	0.921	0.745	0.785	1.049	0.914
		MSE_k	1.155	0.909	0.771	1.251	0.900	0.700	0.895	1.036	0.803	0.855	1.227	1.046
500	MSE_k(*Lasso*)		1.306	1.133	1.318	1.261	1.246	1.473	1.151	1.204	1.171	1.148	1.278	1.068
	Improv. 3%	f	1.267	1.099	1.279	1.223	1.208	1.429	1.116	1.168	1.136	1.114	1.240	1.035
		MSE_k	1.270	1.133	1.314	1.248	1.246	1.453	1.029	1.077	1.047	1.022	1.186	0.961
	Improv. 5%	f	1.241	1.076	1.252	1.198	1.183	1.400	1.093	1.144	1.113	1.091	1.214	1.014
		MSE_k	1.257	1.133	1.308	1.245	1.245	1.437	1.025	1.060	1.032	1.004	1.165	0.945
	Improv. 7%	f	1.215	1.053	1.226	1.173	1.158	1.370	1.070	1.119	1.089	1.068	1.189	0.993
		MSE_k	1.246	1.133	1.312	1.241	1.246	1.425	1.018	1.042	1.023	0.987	1.144	0.929
	Improv. 10%	f	1.176	1.019	1.186	1.135	1.121	1.326	1.036	1.083	1.054	1.033	1.150	0.961
		MSE_k	1.231	1.132	1.313	1.230	1.246	1.401	1.005	1.016	1.006	0.961	1.114	0.905
	Improv. 15%	f	1.110	0.963	1.120	1.072	1.059	1.252	0.978	1.023	0.995	0.976	1.086	0.907
		MSE_k	1.207	1.114	1.271	1.193	1.241	1.376	0.971	0.977	1.039	0.923	1.060	0.866
	Improv. 20%	f	1.045	0.906	1.054	1.009	0.996	1.179	0.921	0.963	0.937	0.919	1.022	0.854
		MSE_k	1.184	1.081	1.266	1.150	1.230	1.354	0.952	0.972	0.995	0.908	1.057	0.857

Table 2.2: Median errors over testing sets for synthetic datasets

In particular, the values of f have been set as improvement percentages over the Lasso values, where the improvement levels are 3%, 5%, 7%, 10%, 15% and 20% (γ equal to 0.03, 0.05, 0.07, 0.15 and 0.20, respectively). The results are obtained for different combinations

(p, n_k), where, as commented before, p is chosen from the set $\{20, 100, 500\}$ and n_k from $\{150, 500\}$. For example, if $n_k = 150$ and $p = 20$, the median of the mean squared error for the Lasso in *Group 1* was equal to 1.084. If the goal is to achieve an improvement of 3%, f must be chosen equal to 1.052. The median of the mean squared errors for the CSCLasso is equal to 0.933 in this case (results obtained on the testing sample). It is important to remark that, for some levels of improvement, the CSCLasso problem is unfeasible due to the fact that γ_{max} for such datasets is smaller than the required γ, and, therefore, those cases are represented as empty spaces in the table. It must be noted that γ_{max} will also depend on each fold in the cross-validation because it is associated with the partition of the data, since the MSE_l in (2.9) depends on such partition ($l = 1, \ldots, L$). It is also worth mentioning that the constraints will always be satisfied on the training set but not necessarily on the testing set, see, for example, the case $k = 3$, $n_k = 150$, $p = 100$ with improvement level equal to 15%. This phenomenon is particularly common as p increases and n_k decreases (see for example the values corresponding to $p = 500$ and $n_k = 150$ in Table 2.2).

We next investigate how the improvement in the prediction errors of the groups of interest affects the prediction errors in the rest of the groups, the overall prediction error and the sparsity level. Figure 2.4 represents the percentage of non-zero (NZ) coefficients and the overall prediction error for different sample sizes, different levels of improvement and for $p = 100$. Lasso results are also included. For instance, when $n_k = 300$ (black squares in Figure 2.4), the NZ percentage for Lasso is 39.60 with an associated overall MSE of 0.734; whereas running the CSCLasso demanding a 3% of improvement over the first six groups, we achieve a NZ percentage of 38.61, and an overall MSE of 0.735. In general terms, it can be seen that the sparsity of the solution decreases with the improvement level: smaller squares, which represent smaller imposed improvement percentages, are on the left of bigger squares (which are associated with demanding percentage of improvement). Then, if the user is very demanding in predicting a specific group, this implies, in the majority of the cases, a less sparse solution. Notwhitstanding, when no level of improvement is imposed (Lasso problem), the solution can be less sparse than in the CSCLasso, as in the case of $n_k = 300$. This also occurs when $p = 500$ and $n_k = 150$ (see the bottom graphic of Figure B.4 in Appendix B). Furthermore, the overall prediction error slightly worsens with the improvement level, due to the worsening of the predictions in the uncontrolled groups.

Figure 2.5 represents the prediction errors over the groups that are not controlled as well as the overall mean squared error, for different improvement levels, $n_k = 150$ (top figure) and $n_k = 300$ (bottom) when $p = 100$. In the figure, Lasso values are also shown. From the figure, it can be concluded that the Lasso performs better in the uncontrolled groups, since the prediction errors worsen under our proposal. However, the overall mean squared error remains almost constant (since the improved errors compensate the more deteriorated ones). Similar conclusions can be drawn under the choices $p = 20$ and $p = 500$ (see Figures B.2 and B.3 in Appendix B, respectively).

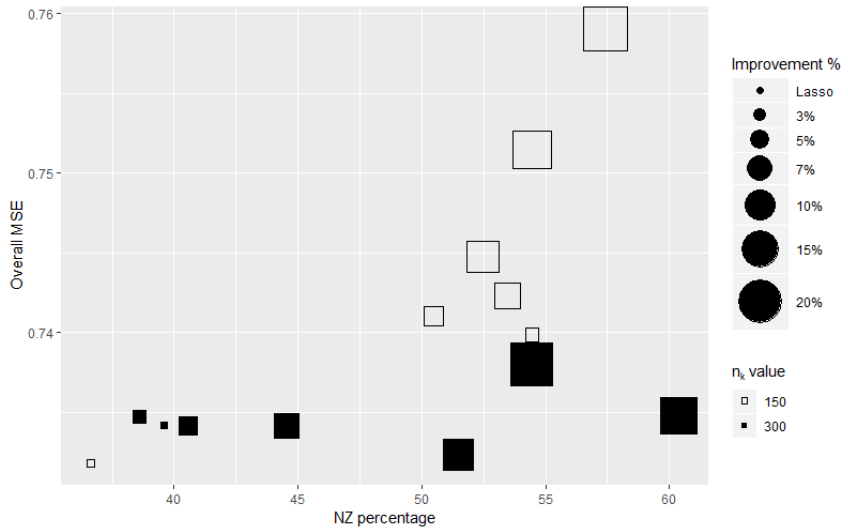


Figure 2.4: Median overall MSE over the testing sets and NZ percentage under the choice $p = 100$

Next, we test how the solution of the CSCLasso behaves with respect to other global performance measures as the l_2 distance, false positive and negative rates, which are defined not in terms of prediction errors, but on the correct fitting of the generator process (see Yu and Liu [2016]). In particular, the l_2 distance is defined as $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$, where $\boldsymbol{\beta}$ is the vector of coefficients that generated the datasets (described at the beginning of this section), and $\hat{\boldsymbol{\beta}}(\lambda)$ are the estimators. In addition, the false positive rate (FPR) and false negative rate (FNR) are calculated as follows:

$$\text{FPR} = \frac{|\ j : \beta_j = 0\ \&\ \hat{\beta}_j(\lambda) = 0\ |}{|\ j : \beta_j = 0\ |},$$

$$\text{FNR} = \frac{|\ j : \beta_j \neq 0\ \&\ \hat{\beta}_j(\lambda) = 0\ |}{|\ j : \beta_j \neq 0\ |},$$

where $j = 1, \ldots, p$. The median of these three measures as well as the median of the overall MSE (already shown in Figures 2.4-2.5 and Figures B.2, B.3 and B.4 in Appendix B), are presented in Table 2.3. For the choices where $p = 20$, the FPR values are not given since all the predictors have associated non-zero coefficients when the datasets were created. From this table it can be deduced that similar or even better results, comparing with those of the Lasso, are obtained in the majority of the cases across the four different measures.

A final remark concerning the computational cost of the CSCLasso when comparing with Lasso is as follows. The median user time required to solve the problem with the largest dataset considered in this study ($n_k = 300$ and $p = 500$) is 0.85 seconds when the Lasso is run on Intel(R) Core(TM) i7-7500U CPU at 2.70GHz 2.90GHz with 8.0 GB of RAM; whereas the CSCLasso requires 6.60 seconds. Nevertheless, to better understand how the computation time behaves depending on p value, a grid in this parameter has been inspected, while n_k is set to

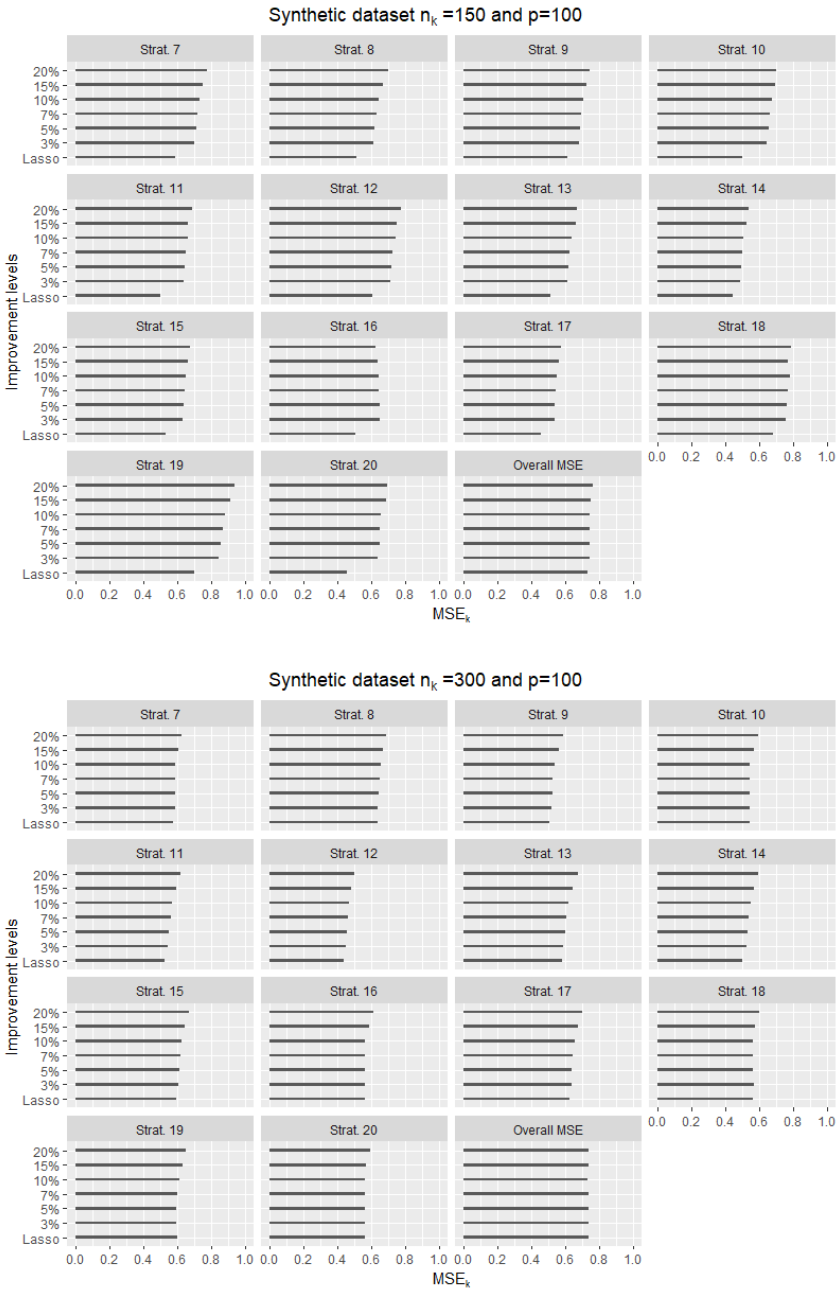


Figure 2.5: Median MSE_k over the testing sets for $k = 7, \ldots, 20$ under $p = 100$ features and, $n_k = 150$ (top) and $n_k = 300$ (bottom). Each subgraph represents one group and the Y-axis shows the different percentages of improvement

300. Figure 2.6 depicts the logarithm of the user times in seconds obtained under Lasso and CSCLasso models when $n_k = 300$ and p changes. Then, under a reasonable computational cost, the desired results are achieved. Further analyses regarding the computational times are shown in the Appendix B (Figure B.5).

		$n_k = 150$				$n_k = 300$			
p		Overall MSE	l_2 distance	FPR	FNR	Overall MSE	l_2 distance	FPR	FNR
20	*Lasso*	0.667	4.149	-	0.050	0.666	4.146	-	0.000
	Improv. 3%	0.691	4.108	-	0.000	0.673	4.099	-	0.000
	Improv. 5%	0.695	4.104	-	0.000	0.683	4.089	-	0.000
	Improv. 7%	-	-	-	-	0.693	4.077	-	0.000
	Improv. 10%	-	-	-	-	-	-	-	-
	Improv. 15%	-	-	-	-	-	-	-	-
	Improv. 20%	-	-	-	-	-	-	-	-
100	*Lasso*	0.732	2.931	0.275	0.250	0.734	2.729	0.250	0.150
	Improv. 3%	0.740	2.857	0.388	0.100	0.735	2.721	0.163	0.150
	Improv. 5%	0.741	2.855	0.375	0.100	0.734	2.721	0.163	0.100
	Improv. 7%	0.742	2.853	0.400	0.100	0.734	2.718	0.163	0.100
	Improv. 10%	0.745	2.849	0.400	0.100	0.732	2.704	0.275	0.100
	Improv. 15%	0.751	2.844	0.425	0.100	0.735	2.678	0.288	0.050
	Improv. 20%	0.759	2.835	0.463	0.100	0.738	2.655	0.288	0.100
500	*Lasso*	0.772	2.778	0.135	0.300	0.744	2.750	0.065	0.300
	Improv. 3%	0.772	2.779	0.040	0.350	0.744	2.658	0.063	0.250
	Improv. 5%	0.772	2.775	0.050	0.300	0.745	2.653	0.075	0.250
	Improv. 7%	0.772	2.769	0.042	0.350	0.746	2.647	0.077	0.250
	Improv. 10%	0.772	2.760	0.056	0.350	0.748	2.632	0.104	0.250
	Improv. 15%	0.771	2.754	0.063	0.300	0.752	2.640	0.129	0.250
	Improv. 20%	0.774	2.735	0.069	0.350	0.760	2.607	0.148	0.200

Table 2.3: Median performance measures over testing sets for synthetic datasets

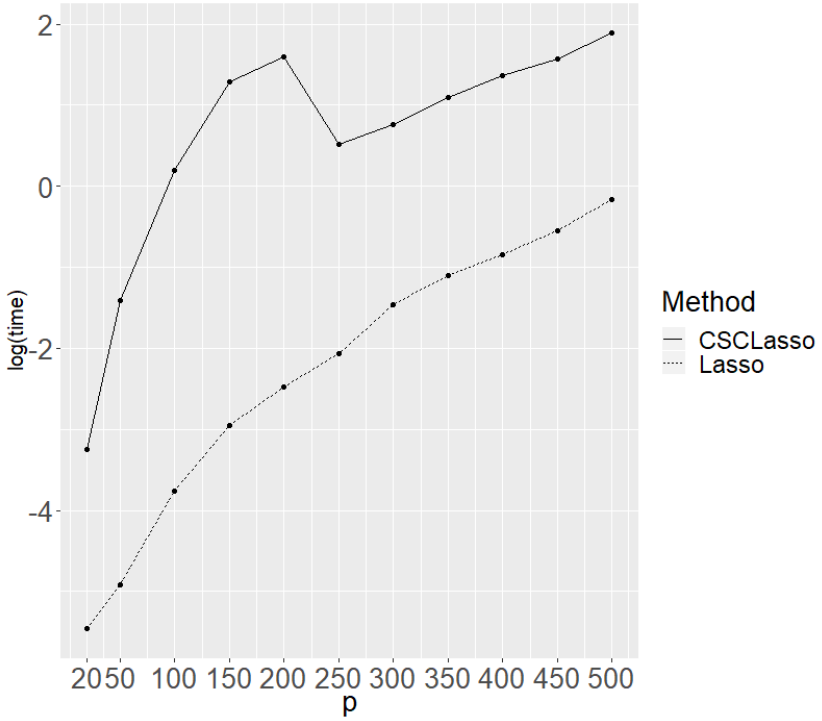


Figure 2.6: A two-dimensional graph of the logarithm of the user times in seconds for $n_k = 300$ as p increases

2.4.2 Leukemia dataset: a gene expression dataset

The real stratified dataset described in Kouno et al. [2013] is explored here. The data contain information related to myeloid monocytic leukemia cells undergoing differentiation to macrophages. In particular, the dataset is formed by expression levels of 45 transcription factors (response and predictor variables) measured at 8 distinct times (groups) of the differentiation process. As in Ollier and Viallon [2017], the aim is to predict the EGR2 transcription factor in terms of the other $p = 44$ factors. The sample size per group is equal to 120. Similarly as in Section 2.4.1, the Lasso was run and the overall prediction errors, individual prediction errors per group and percentage of non-zero coefficients are recorded. The records in *Group 1* yield the best MSE using Lasso model. Therefore, we may be interested in obtaining an even better fitting for such data. The CSCLasso problem is solved with threshold values smaller than the Lasso error, which turns out to be 0.370. Table 2.4 shows the obtained median results for an assortment of improvement levels, namely, 5%, 7%, 10% and 15% or, equivalently, γ is equal to 0.05, 0.05, 0.07 and 0.15, respectively. From that table, it can be seen how the prediction error of interest (corresponding to *Group 1*) decreases with the improvement level, as expected. Similarly as in Section 2.4.1, the overall mean squared error does not exhibit significant changes, while the prediction errors in the uncontrolled groups do not exhibit the same behaviour. Some of them slightly improve (*Group 6*), others slightly worsen (as the *Group 5* and *Group 8*) and others remain constant (as *Group 2*). Finally, in regards to the sparsity of the solution, for this dataset, less sparse solutions are obtained by CSCLasso in comparison with the Lasso ones.

	f_1	*Overall* MSE	MSE_1	MSE_2	MSE_3	MSE_4	MSE_5	MSE_6	MSE_7	MSE_8	NZ
Lasso	-	0.620	**0.370**	0.417	0.902	0.496	0.480	0.535	0.496	0.685	53.33
Improv. 5%	**0.352**	0.623	**0.357**	0.417	0.918	0.500	0.555	0.512	0.497	0.719	73.33
Improv. 7%	**0.344**	0.626	**0.348**	0.418	0.915	0.504	0.565	0.514	0.497	0.722	66.67
Improv. 10%	**0.333**	0.630	**0.335**	0.418	0.911	0.510	0.574	0.513	0.498	0.728	66.67
Improv. 15%	**0.315**	0.636	**0.331**	0.415	0.903	0.523	0.591	0.512	0.502	0.737	73.33

Table 2.4: Median errors over testing set for gene expression dataset. Constraints imposed over *Group 1*

2.4.3 Communities and Crime dataset

In this section, a real dataset from the UCI Machine Learning Repository [Lichman, 2013] will be analyzed. In particular, the so-called *Communities and Crime Unnormalized Data Set* shall be considered. The dataset is about communities within the United States and has already been inspected in the literature (see Redmond and Baveja [2002]). This dataset combines crime information from the FBI databases [U.S. Department, 1995] as well as socio-economic and law enforcement data from U.S. Department [1992a] and U.S. Department [1992b], respectively. The dataset is formed by $p = 124$ predictors, 23 of which present missing values,

and $n = 2215$ instances, where the response variable measures the number of murders per 100K population. The predictor variables with missing values are not consider for the next experiments. As such, we finally consider $p = 101$ predictors. Additionally, for each instance (community), the region from which it comes is known. Thus, if we were interested in obtaining a good prediction in a certain region, say Midwest, we could control these communities by including a performance constraint. Table 2.5 shows the median errors over the testing set for *Group 1*, formed by the communities of Midwest, and over the rest of communities (*Group 2*). In terms of overall MSE and MSE over the two groups, similar conclusions as in Section 2.4.2 are drawn. Whereas different improvement levels are imposed, the MSE of interest (MSE_1) is getting smaller but the overall prediction error is almost not affected by the constraint. Lastly, regarding the sparsity of the solution, an analogous behaviour as that observed in the case of simulated data is obtained: the solution becomes less sparse with the improvement level.

	f_1	*Overall* MSE	MSE_1	MSE_2	NZ
Lasso	-	0.488	**0.433**	0.453	21.57
Improv. 5%	**0.411**	0.488	**0.422**	0.453	25.49
Improv. 7%	**0.403**	0.487	**0.420**	0.453	28.43
Improv. 10%	**0.390**	0.488	**0.416**	0.453	26.47
Improv. 15%	**0.368**	0.486	**0.403**	0.459	34.31

Table 2.5: Median errors over testing set for communities and crime dataset. Constraints imposed over *Group 1*

As previously commented, the groups of interest may overlap. As an illustration, assume that the interest is in controlling the prediction error in communities of Midwest or communities with a population density larger than or equal to the 75th percentile. Let *Group 1* denote the communities from Midwest, while *Group 2* represents the communities where the density of population is higher than the 75th percentile. For instance, if we aim to improve in a 7% the errors obtained by the Lasso model (equal to 0.513 and 0.442), then the CSCLasso results become 0.475 and 0.441, respectively.

2.5 Chapter summary

In this work a new version of the Lasso regression model that strives to control the performance rates associated with individuals of interest is proposed. The method has a significant application in the context of heterogeneous data, where it is common that certain sources are more reliable than others, or simply the prediction on some groups of data are of higher interest, and thus a better fit is sought for some data. In order to control the individuals of interest, performance constraints are included in the regression model. This approach leads to a novel method (CSCLasso) which is not reported in the literature previously, up to our knowledge. Theoretical results concerning this novel methodology have been discussed and, in addition, the CSCLasso has been tested on six synthetic datasets with different properties, on a well-referenced real

stratified biomedical dataset and on a real social sciences dataset. The numerical section shows that, with a low computational cost, the accuracy prediction errors for the groups of interest are controlled. This is done at the expense of reducing sparsity (if the regularization parameter is kept fixed) or the overall accuracy.

Chapter 3

On linear regression models with hierarchical categorical variables

In this chapter we study the mathematical optimization problem that trades off, in linear regression models, accuracy and model complexity, in the presence of categorical variables that have a hierarchical structure, with their categories arranged as a directed tree. In the literature, this kind of data appears in different fields of research, such as nested spatial data in Spatial Statistics [Gotway and Young, 2002], behavioral data in Retail Business Analytics [Griva et al., 2018], and economic activity data in Official Statistics [European Commission, 2008; Katz-Gerro and López Sintas, 2019].

3.1 Introduction

Let $\mathcal{J}'$ be the set of continuous and dummy predictor variables, whereas $\mathcal{J}$ the set of hierarchical categorical predictor variables. Then, consider the random vector $(Y, \mathbf{X}', \mathbf{X})$, where $\mathbf{X}'$ denotes the vector of the predictor variables in $\mathcal{J}'$, $\mathbf{X}$ denotes the vector of categorical predictor variables in $\mathcal{J}$, and Y denotes the response variable. In the real-world dataset `cancer-reg` [Rippner, 2017] used in the numerical section, with individuals from the United States of America (U.S.), *geography* is a categorical variable with a hierarchical structure. According to the *U.S. Department of Commerce Economics and Statistics Administration* and the *U.S. Census Bureau*, *geography* can be coded using the states (51 in total), which is the highest level of granularity for which information is available in the dataset. This means that 51 coefficients need to be estimated for this variable, where individuals in the same state share the same coefficient in the linear regression model. The variable *geography* can alternatively be coded using the subregions, such as *East-South Central*, *Middle Atlantic* and *New England*, where each state belongs to exactly one of the 9 subregions. Consolidating individuals at the subregions, sharing the same coefficient, yields a lower level of granularity for *geography*, where, instead of 51, only 9 coefficients need to be estimated and interpreted. The individuals can be further consolidated into 4 regions, namely *West*, *South*, *Mid-West* and *North-East*, where only 4 coefficients would be associated to *geography* in the reduced linear regression model. Using these regions, one has the least granular representation of *geography*. This work is devoted to trading off accuracy of the linear regression model and its complexity, measured as a cost function of the level of granularity used to represent each of the hierarchical categorical variables.

The categories of hierarchical categorical variable $j \in \mathcal{J}$ can be arranged as a directed tree $\mathcal{T}_j$, i.e., a directed graph with a root node, $r(\mathcal{T}_j)$, and a unique path from each node to $r(\mathcal{T}_j)$. In addition, let $\mathcal{V}(\mathcal{T}_j)$ denote the set of nodes in the tree and $\mathcal{L}(\mathcal{T}_j) \subset \mathcal{V}(\mathcal{T}_j)$ the set of leaf nodes. See Figure 3.1 for the tree associated with the categories of *geography*, where the leaf nodes correspond to the states, going upstream we find the subregions and then the regions, which, in turn, are directly connected with the root node. Let $(y_i, \mathbf{x}'_i, \mathbf{x}_i)$ be the vector associated with individual i, with $\mathbf{x}'_i = (x'_{ij'})$ and $\mathbf{x}_i = (x_{ijv})$, where x_{ijv} is equal to 1 if individual i belongs to category $v \in \mathcal{V}(\mathcal{T}_j)$ of variable $j \in \mathcal{J}$. If we were to use the most granular representation of the hierarchical categorical variables, we would need to use the categories associated with the

leaf nodes $l \in \mathcal{L}(\mathcal{T}_j)$, i.e.,

$$\hat{y}_i = \hat{\beta}'_0 + \sum_{j' \in \mathcal{J}'} \hat{\beta}'_{j'} x'_{ij'} + \sum_{j \in \mathcal{J}} \sum_{l \in \mathcal{L}(\mathcal{T}_j)} \hat{\beta}_{jl} x_{ijl}, \tag{3.1}$$

where β'_0 is the independent term, $\beta'_{j'}$ is the coefficient of variable $j' \in \mathcal{J}'$, whereas β_{jl} is the coefficient of category $l \in \mathcal{L}(\mathcal{T}_j)$ of hierarchical categorical variable $j \in \mathcal{J}$. In the OLS paradigm, the coefficients are obtained by minimizing the MSE. The corresponding OLS model reads as follows

$$\text{MSE}^*((\mathcal{T}_j)_{j \in \mathcal{J}}) = \min_{\beta'_0, (\beta'_{j'})_{j' \in \mathcal{J}'}, (\beta_{jl})_{l \in \mathcal{L}(\mathcal{T}_j), j \in \mathcal{J}}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta'_0 - \sum_{j' \in \mathcal{J}'} \beta'_{j'} x'_{ij'} - \sum_{j \in \mathcal{J}} \sum_{l \in \mathcal{L}(\mathcal{T}_j)} \beta_{jl} x_{ijl})^2, \tag{3.2}$$

where n is the sample size. In the `cancer-reg` dataset, with the most granular representation of *geography*, we have a MSE of 0.407 for the training sample. The question arises as to whether that level of granularity is necessary, or whether we can merge categories at the bottom of the tree into a broader category upstream in the tree. With this, we can eliminate the state information for all the individuals of same subregion, respectively from the same region, and report the subregion, respectively the region. We have done this for the states in the subregions *Middle Atlantic* and *New England*, yielding the subtree in Figure 3.3 (a) of the tree in Figure 3.1. All individuals in the descendants leaf nodes of *Middle Atlantic* are consolidated in its parent node *Middle Atlantic* and, therefore, they share the same coefficient in the linear regression model, and the same for *New England* node. With this representation, the MSE increases from 0.407 to 0.408. This mild worsening in accuracy corresponds to an improvement in the complexity of the linear regression model, with a reduction from 51 to 44 in the number of coefficients to be estimated and interpreted for the *geography* variable.

Reducing the granularity of the representation of hierarchical categorical variables has several advantages. First, and as illustrated above, it is a step towards enhancing the interpretability of the linear regression model, where fewer coefficients need to be estimated and interpreted [Carrizosa et al., 2017a]. Second, if the samples of individuals associated with categories are homogeneous enough, a very granular representation would yield an overparametrized model. Instead, we could merge these categories into a broader one upstream the tree, thus having more observations to estimate fewer coefficients. The homogeneity together with the increase in sample size ensure lower errors in the estimation of the coefficients of the broader categories [LeBlanc and Tibshirani, 1998]. Third, and again if the samples of individuals associated with categories are homogeneous enough, a very granular representation will yield higher data gathering costs [Carrizosa et al., 2008; Turney, 1995], if, for instance, the surveying costs are asymmetric. Indeed, we would need to ensure a large enough sample for each category in the representation, even though the cost of surveying may be high for some of these categories. By merging homogeneous categories into a broader one upstream the tree, we can sample from a

larger subpopulation lowering these data gathering costs. Fourth, our methodology can identify where j is an irrelevant predictor [Bertsimas et al., 2020; Blanquero et al., 2020; Carrizosa et al., 2017b] by consolidating individuals at the root node $r(\mathcal{T}_j)$. Finally, the consolidation of information is important when having data privacy considerations, [Li and Sarkar, 2009; Lu et al., 2014], since it is well-known that more detailed information is linked to confidentiality concerns [Baena et al., 2020].

The remainder of this chapter is structured as follows. In Section 3.2, we study the *constrained* problem, in which we minimize the accuracy of the reduced linear regression model, measured by its MSE, subject to a complexity constraint, where a threshold is imposed on the cost of granularity of the representation of the hierarchical categorical variables. This problem is then formulated as a Mixed Integer Convex Quadratic Problem with Linear Constraints. Section 3.3 illustrates our approach in two real-world datasets as well as in a synthetic one, where the entire set of non-dominated outcomes to the problem is obtained solving the constrained problem for the different values of the threshold. To end, some conclusions are provided in Section 3.4.

3.2 The constrained problem

In this section, we first model the two objectives under consideration when building the reduced linear regression model. We then provide a Mixed Integer Convex Quadratic formulation with Linear Constraints for the constrained problem. We end the section with a discussion on the values of the threshold parameter to find all possible non-dominated outcomes to our problem.

Consolidating the information of hierarchical categorical variables is equivalent to finding, for each $j \in \mathcal{J}$, a subtree $\mathcal{S}_j$ of $\mathcal{T}_j$, with the same root as $\mathcal{T}_j$, $r(\mathcal{S}_j) = r(\mathcal{T}_j)$. The accuracy of the reduced linear regression model, with individuals consolidated at the leaf nodes $\mathcal{L}(\mathcal{S}_j)$, will be measured by its MSE, while its complexity will be measured by

$$\mathrm{C}((\mathcal{S}_j)_{j\in\mathcal{J}}) = \sum_{j\in\mathcal{J}} \sum_{l\in\mathcal{L}(\mathcal{S}_j)} c_{jl}, \tag{3.3}$$

where $c_{jv} \geq 0$ represents the cost associated to node $v \in \mathcal{V}(\mathcal{T}_j)$.

With this, our problem reads as follows:

$$\min_{(\mathcal{S}_j)_{j\in\mathcal{J}}} (\mathrm{MSE}^*((\mathcal{S}_j)_{j\in\mathcal{J}}), \mathrm{C}((\mathcal{S}_j)_{j\in\mathcal{J}})), \tag{3.4}$$

where $\mathrm{MSE}^*((\mathcal{S}_j)_{j\in\mathcal{J}})$ is defined as in (3.2) with $\mathcal{L}(\mathcal{S}_j)$ replacing $\mathcal{L}(\mathcal{T}_j)$. Note that Problem (3.4) performs akin to the pruning of a regression tree [Sherali et al., 2009; Su et al., 2004]. In our case, we have one tree per hierarchical categorical predictor in the dataset, and the pruning of all these trees needs to be performed simultaneously to properly trade off the accuracy and the complexity of the reduced linear regression model.

Non-dominated outcomes to Problem (3.4) are obtained by solving the following constrained problem:

$$\begin{aligned} \min_{(\mathcal{S}_j)_{j\in\mathcal{J}}} \quad & \text{MSE}^*((\mathcal{S}_j)_{j\in\mathcal{J}}) \\ \text{s.t.} \quad & \text{C}((\mathcal{S}_j)_{j\in\mathcal{J}}) \leq c, \end{aligned} \tag{3.5}$$

where c is a threshold on the complexity of the model.

To formulate Problem (3.5) as a Mixed Integer Convex Quadratic Problem with Linear Constraints, we note that finding a subtree $\mathcal{S}_j$ of $\mathcal{T}_j$, with $r(\mathcal{S}_j) = r(\mathcal{T}_j)$, is equivalent to finding its leaf nodes. Therefore, we introduce binary decision variables $\mathbf{z} = (z_{jv})$, such that $z_{jv} = 1$ if the node associated with category v of the hierarchical categorical variable j is selected as leaf node of $\mathcal{S}_j$, and $z_{jv} = 0$ otherwise. If node v is selected, all individuals in its descendant leaf nodes are consolidated at v, and these individuals will share the same coefficient in the reduced linear regression model.

We need additional constraints to ensure that $\mathbf{z}$ is well defined. For this, we make use of the structural properties of the unique path $\mathcal{P}_{jl}$ in $\mathcal{T}_j$ from its root to leaf node $l \in \mathcal{L}(\mathcal{T}_j)$, $j \in \mathcal{J}$. It is easy to see that $\mathbf{z}$ is well defined if and only if there exists exactly one v such $z_{jv} = 1$ for each path $\mathcal{P}_{jl}$. With this, $\sum_{v\in\mathcal{V}(\mathcal{T}_j)} z_{jv}x_{ijv}$ represents the observed value for hierarchical predictor variable j in individual i, $\sum_{v\in\mathcal{V}(\mathcal{T}_j)} z_{jv}x_{ijv}\beta_{jv}$ is the contribution of j towards the predicted response for individual i, and $\sum_{v\in\mathcal{V}(\mathcal{T}_j)} c_{jv}z_{jv}$ is the contribution of j towards the cost in (3.3).

Therefore, Problem (3.5) can be formulated as follows:

$$\min_{\mathbf{z},\beta'_0,(\beta'_{j'})_{j'\in\mathcal{J}'},(\beta_{jv})_{v\in\mathcal{V}(\mathcal{T}_j),j\in\mathcal{J}}} \quad \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta'_0 - \sum_{j'\in\mathcal{J}'} x'_{ij'}\beta'_{j'} - \sum_{j\in\mathcal{J}}\sum_{v\in\mathcal{V}(\mathcal{T}_j)} z_{jv}x_{ijv}\beta_{jv})^2 \tag{3.6}$$

$$\text{s.t.} \quad \sum_{v\in\mathcal{P}_{jl}} z_{jv} = 1, \;\; l \in \mathcal{L}(\mathcal{T}_j), \;\; j \in \mathcal{J}, \tag{3.7}$$

$$\sum_{j\in\mathcal{J}}\sum_{v\in\mathcal{V}(\mathcal{T}_j)} c_{jv}z_{jv} \leq c, \tag{3.8}$$

$$z_{jv} \in \{0,1\}, \;\; \forall v \in \mathcal{V}(\mathcal{T}_j), \;\; j \in \mathcal{J}, \tag{3.9}$$

$$\beta'_0,\, \beta'_{j'},\, \beta_{jv} \in \mathbb{R}, \, \forall j' \in \mathcal{J}', \, \forall v \in \mathcal{V}(\mathcal{T}_j), \;\; j \in \mathcal{J}. \tag{3.10}$$

The objective function (3.6) is the MSE of linear models. The linear constraints (3.7) model that only one node is selected per path, becoming thus a leaf node of the subtree sought. Constraint (3.8) imposes the threshold c on the complexity of the reduced linear regression model. Constraints (3.9) and (3.10) impose the range of the decision variables.

Since the objective function (3.6) has semi-continuous variables, $z_{jv}\beta_{jv}$, a smooth formu-

lation can be obtained using big M constraints:

$$\begin{aligned}\min_{\mathbf{z},\beta'_0,(\beta'_{j'})_{j'\in\mathcal{J}'},(\beta_{jv})_{v\in\mathcal{V}(\mathcal{T}_j),j\in\mathcal{J}}} \quad & \frac{1}{n}\sum_{i=1}^{n}(y_i-\beta'_0-\sum_{j'\in\mathcal{J}'}x'_{ij'}\beta'_{j'}-\sum_{j\in\mathcal{J}}\sum_{v\in\mathcal{V}(\mathcal{T}_j)}x_{ijv}\tilde{\beta}_{jv})^2\\ \text{s.t.} \quad & (3.7)-(3.9),\\ & -Mz_{jv}\le\tilde{\beta}_{jv}\le Mz_{jv},\ \ \forall v\in\mathcal{V}(\mathcal{T}_j),\ \ j\in\mathcal{J},\\ & \beta'_0,\ \beta'_{j'},\ \tilde{\beta}_{jv}\in\mathbb{R},\ \forall j'\in\mathcal{J}',\ \forall v\in\mathcal{V}(\mathcal{T}_j),\ \ j\in\mathcal{J}.\end{aligned} \tag{3.11}$$

This is the formulation that will be used in the numerical section. Note that we can sharpen the value of M by imposing an upper bound on the coefficients of the categories of hierarchical variables. This can be seen as a regularization, thus preventing overfitting and allowing for sparser models [Carrizosa et al., 2016]. Other types of regularization can be easily incorporated into our model, such as those in Simon et al. [2011]; Yuan and Lin [2006].

We now discuss the choice of values for threshold c. It is easy to show that if c_{jv} are integer numbers, it is enough to consider integer values for c too. Moreover, it is easy to define lower ($c^{\min}:=|\mathcal{J}|$) and upper ($c^{\max}:=\mathrm{C}((\mathcal{T}_j)_{j\in\mathcal{J}})$) bounds on c. By varying the threshold value c among this finite set of values, we obtain the entire set of non-dominated outcomes to Problem (3.4).

Non-dominated outcomes to Problem (3.4) can also be obtained by solving the alternative constrained problem:

$$\begin{aligned}\min_{(\mathcal{S}_j)_{j\in\mathcal{J}}} \quad & \mathrm{C}((\mathcal{S}_j)_{j\in\mathcal{J}})\\ \text{s.t.} \quad & \mathrm{MSE}^*((\mathcal{S}_j)_{j\in\mathcal{J}})\le f,\end{aligned} \tag{3.12}$$

where f is the threshold value on the MSE of the reduced linear regression model. The advantage of constraining $\mathrm{MSE}^*((\mathcal{S}_j)_{j\in\mathcal{J}})$ is to have full control on the accuracy of the model and to allow the user to define meaningful values of f, [Blanquero et al., 2021b]. Therefore, this option is recommended when the constrained problem is solved only for a few values of f. A lower bound on f is

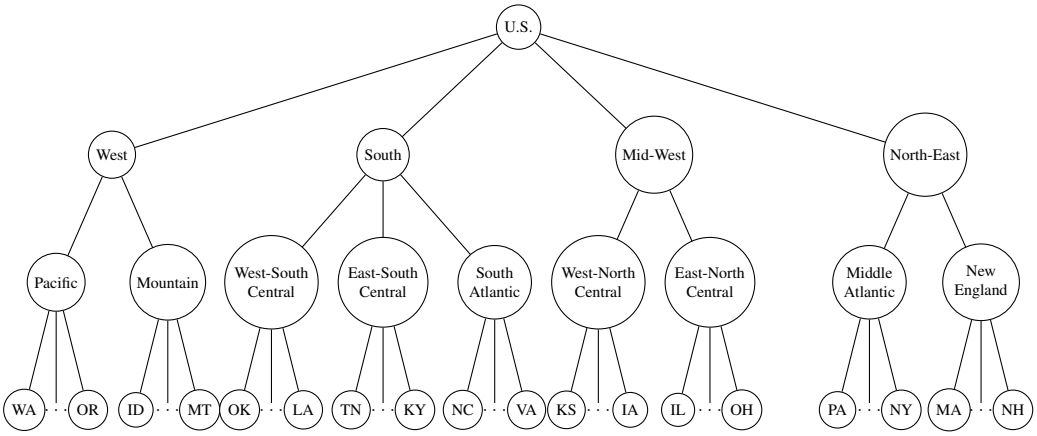
$$f^{\min}:=\mathrm{MSE}^*((\mathcal{T}_j)_{j\in\mathcal{J}}), \tag{3.13}$$

which is the MSE that we achieve for the highest level of granularity on all the hierarchical categorical variables. An upper bound on f is found by removing all the variables $j\in\mathcal{J}$. This corresponds to

$$f^{\max}:=\min_{\beta'_0,(\beta'_{j'})_{j'\in\mathcal{J}'}} \ \frac{1}{n}\sum_{i=1}^{n}(y_i-\beta'_0-\sum_{j'\in\mathcal{J}'}\beta'_{j'}x'_{ij'})^2, \tag{3.14}$$

where we consider the subtree with only the root node, i.e., $\mathcal{S}_j=\{r(\mathcal{T}_j)\}\ \forall j\in\mathcal{J}$. In this case, by varying the threshold value f in a grid of $[f^{\min},f^{\max}]$, we obtain a collection of non-dominated outcomes to Problem (3.4).

Figure 3.1: Tree representation of the variable *geography* in the `cancer-reg` dataset



3.3 Numerical experiments

In this section, we illustrate our approach using two real-world datasets and a synthetic one. Our aim is to depict the tradeoff between the accuracy of the reduced model and its complexity, measured by the number of coefficients to be estimated for the hierarchical categorical variables, which corresponds to $c_{jv} = 1$ in (3.3). To solve Problem (3.11) for all possible values of $c \in \{c^{\min}, \ldots, c^{\max}\}$, we use Gurobi, where M is set to 1000. The experiments have been run on Intel(R) Core(TM) i7-7500U CPU at 2.70 GHz 2.90 GHz with 8.0 GB of RAM.

3.3.1 Cancer trials dataset: a real-world dataset

Consider again the real-world dataset `cancer-reg` introduced in Section 3.1. This dataset aims to look for relationships between the socioeconomic status in U.S. and the mean per capita cancer mortality (response variable). It has a sample of size $n = 3047$ with 32 predictor variables: one hierarchical predictor variable ($|\mathcal{J}| = 1$) and 31 non-hierarchical predictor variables ($|\mathcal{J}'| = 31$), where continuous predictors have been standardized. This database was collected from the American Community Survey (`census.gov`), `clinicaltrials.gov` and `cancer.gov` sources. As mentioned in Section 3.1, the only hierarchical categorical variable is *geography*, see Figure 3.1, and contains information on the state linked to the individuals.

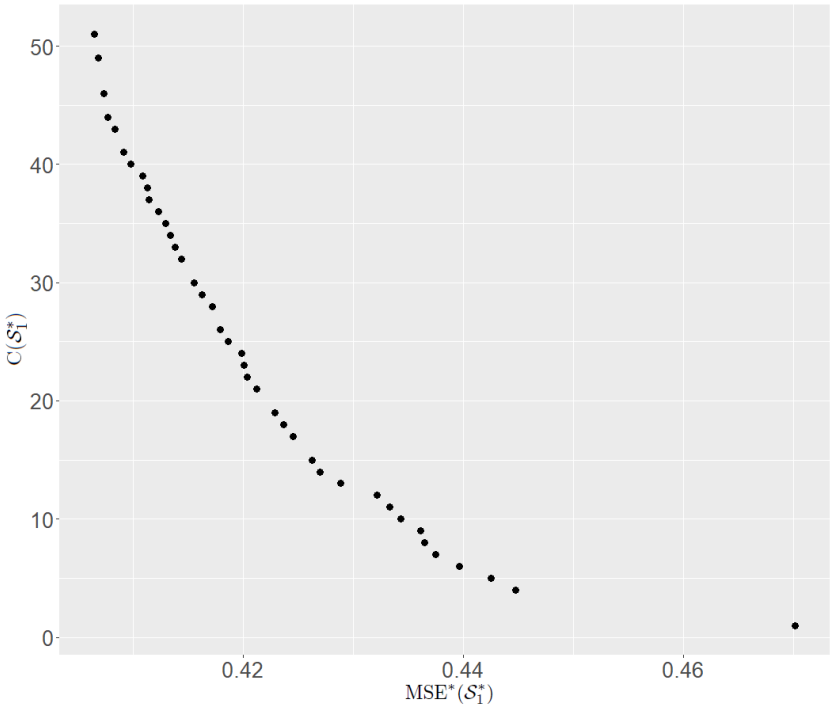
We solve Problem (3.11) for the 51 values of c in the set $\{1, \ldots, 51\}$. Figure 3.2 reports the pareto frontier for the MSE and the number of coefficients to be estimated in the reduced model for the hierarchical categorical variable. Clearly, our methodology can find a much less complex model with a very mild worsening of the accuracy, but it is ultimately the decision of the user as to which reduced model to choose.

Figure 3.3 plots the selected subtree $\mathcal{S}_1^*$ associated with *geography* for three of the solutions in Figure 3.2. In particular, Figure 3.3(a) is the representation associated with the model that

achieves the minimum Akaike information criterion (AIC) metric [Akaike, 1998], whereas Figure 3.3(c) the one with the minimum Bayesian information criterion (BIC) [Schwarz, 1978], which are two measures for model selection that compute the tradeoff between the fit in the training sample and the number of parameters involved.

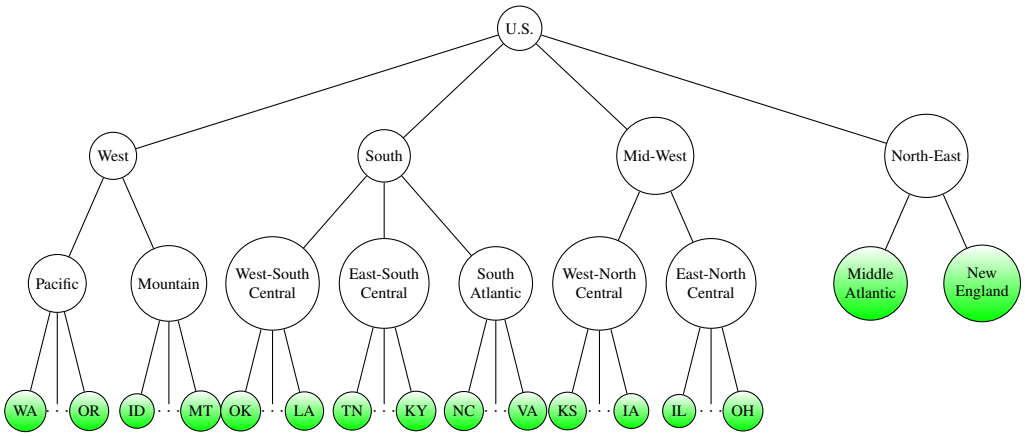
Table 3.1 presents the coefficients of *geography* for four of the solutions in Figure 3.2, namely the most complex model when all the leaf nodes in Figure 3.1 are considered, as well as the three reduced models with less granular representation of *geography* in Figure 3.3. We can see that when categories are merged into one upstream the tree, the single coefficient that needs to be estimated for that broader category is within the range of the coefficients obtained with the most granular representation.

Figure 3.2: Pareto frontier for MSE versus the number of coefficients to be estimated in the reduced model for the hierarchical categorical variable *geography* in the `cancer-reg` dataset

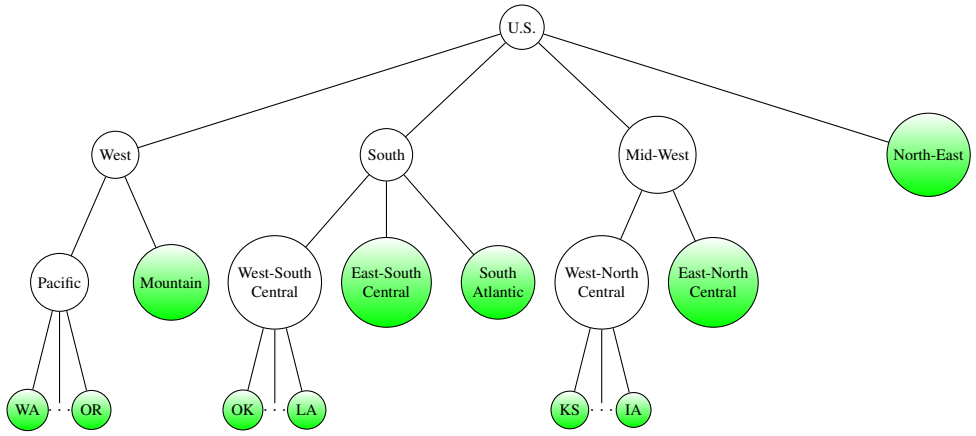


3.3.2 Boston Housing dataset

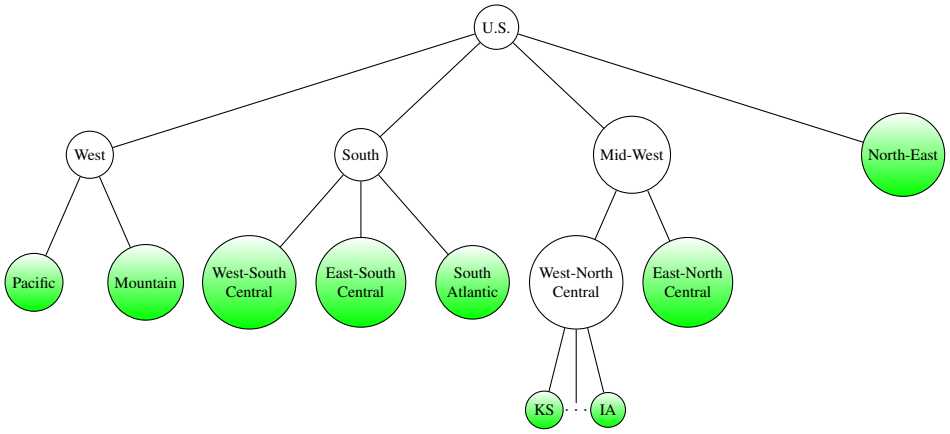
The well-known `housing` dataset [Harrison and Rubinfeld, 1978] contains information concerning the price of the houses in the area of Boston, which was collected from the U.S. Census Service. See Table 3.2 for a description of its predictor variables, as well as the response. It has a sample of size $n = 506$ with 13 predictor variables: 12 continuous, which have been discretized yielding 12 hierarchical predictor variables ($|\mathcal{J}| = 12$), and 1 binary one ($|\mathcal{J}'| = 1$). Figure 3.4 illustrates the discretization of *CRIM*, the first continuous variable. Similar ones have been implemented for the other 11 continuous variables. First, we split the observations



(a) $\mathcal{S}_1^*$ when $\text{MSE}^*(\mathcal{S}_1^*) = 0.408$ and $c = 44$



(b) $\mathcal{S}_1^*$ when $\text{MSE}^*(\mathcal{S}_1^*) = 0.421$ and $c = 21$

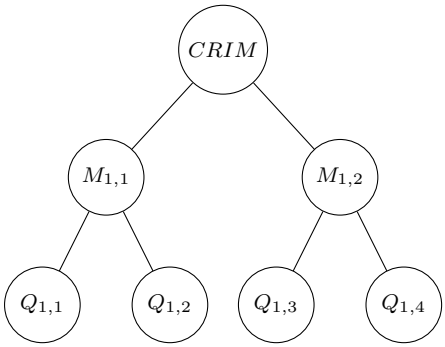


(c) $\mathcal{S}_1^*$ when $\text{MSE}^*(\mathcal{S}_1^*) = 0.427$ and $c = 14$

Figure 3.3: Less granular representations for the *geography* variable in the `cancer-reg` dataset

of *CRIM* into two groups: those whose values are below (node $M_{1,1}$) and above (node $M_{1,2}$) the median. Second, the quartiles are used to subdivide $M_{1,1}$ (nodes $Q_{1,1}$ and $Q_{1,2}$) and $M_{1,2}$ (nodes $Q_{1,3}$ and $Q_{1,4}$) into two nodes. This way we examine the thresholds of the continuous predictor variables required to predict the response variable.

Figure 3.4: Tree associated with the variable *CRIM* in the `housing` dataset after being discretized



When solving Problem (3.11) for the 37 values of c in the set $\{12, \ldots, 48\}$, we obtain the pareto frontier in Figure 3.5. The MSE of the model with the highest granularity for all hierarchical variables is 23.06. When we start reducing the granularity the MSE remains approximately the same. Actually, when c is reduced from 48 to 30, the accuracy is barely damaged but the complexity of the linear regression model is dramatically improved.

Figures 3.6-3.7 show the subtrees $\mathcal{S}_j^*$ for all $j \in \mathcal{J}$ for the solution in Figure 3.5 that achieves the minimum AIC. In this solution, we can observe how variables *INDUS*, *AGE* and *RAD* are eliminated from the linear regression model, as their root node is the only one selected with a coefficient equal to zero. By contrast, we require the highest level of granularity for *PTRATIO*, *B* and *LSTAT*. For *DIS*, the linear regression model only needs to know whether the predictor variable is below the median. For the remaining predictor variables, leaf as well as non-leaf nodes are selected.

3.3.3 The synthetic data

In this section we illustrate our approach on synthetic data. The data generating model is

$$y_i = \sum_{j \in \mathcal{J}} \sum_{l \in \mathcal{L}(\mathcal{T}_j)} \beta_{jl}^{\mathrm{S}} x_{ijl} + \varepsilon_i,\ i = 1, \ldots, n, \tag{3.15}$$

where $|\mathcal{J}| = 2$ and $\mathcal{J}' = \emptyset$. The values of the coefficients β_{jl}^{S}, $l \in \mathcal{L}(\mathcal{T}_j)$, are given in Figure 3.8. Note that the first two leaf nodes of $\mathcal{T}_1$ have the same coefficient, and the same holds for the other two leaf nodes. Therefore, the tree can be pruned to avoid unnecessary splits, yielding

Figure 3.5: Pareto frontier for MSE versus the number of coefficients to be estimated in the reduced model for the hierarchical categorical variables in the `housing` dataset

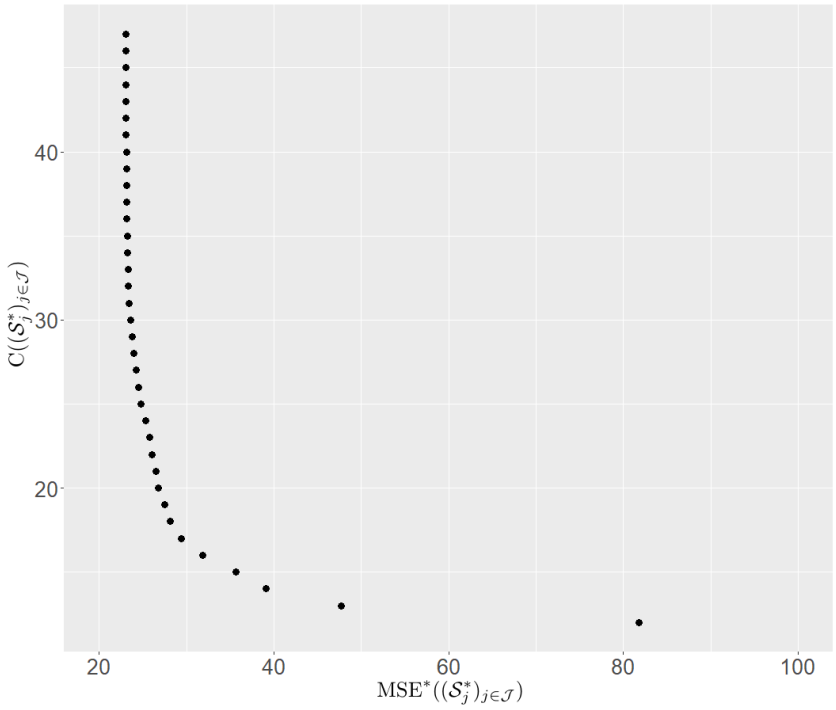


Figure 3.6: Less granular representations for the first six hierarchical categorical variables in the `housing` dataset for the solution in Figure 3.5 with $\text{MSE}^*((\mathcal{S}_j^*)_{j\in\mathcal{J}}) = 23.37$ and $c = 32$. Note that this is the solution that achieves the minimum AIC

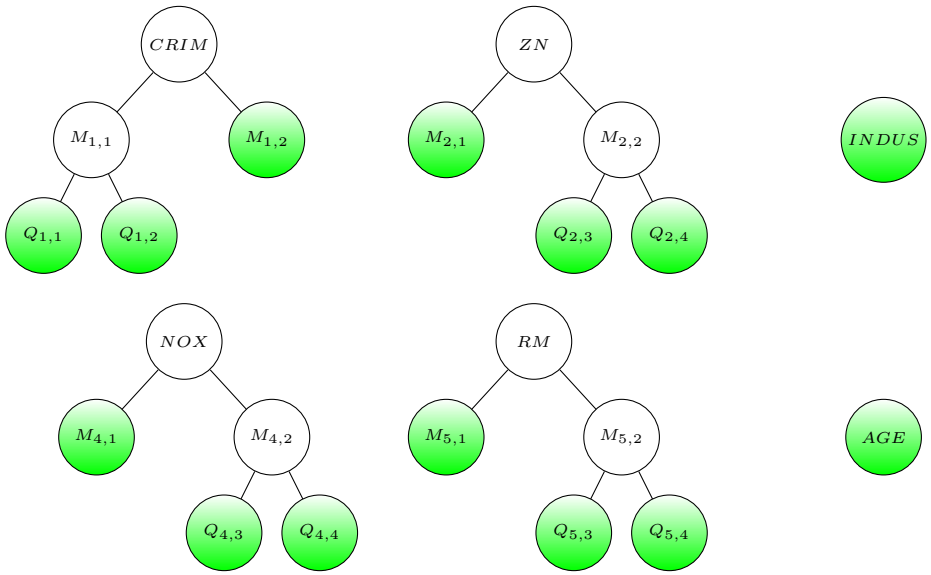
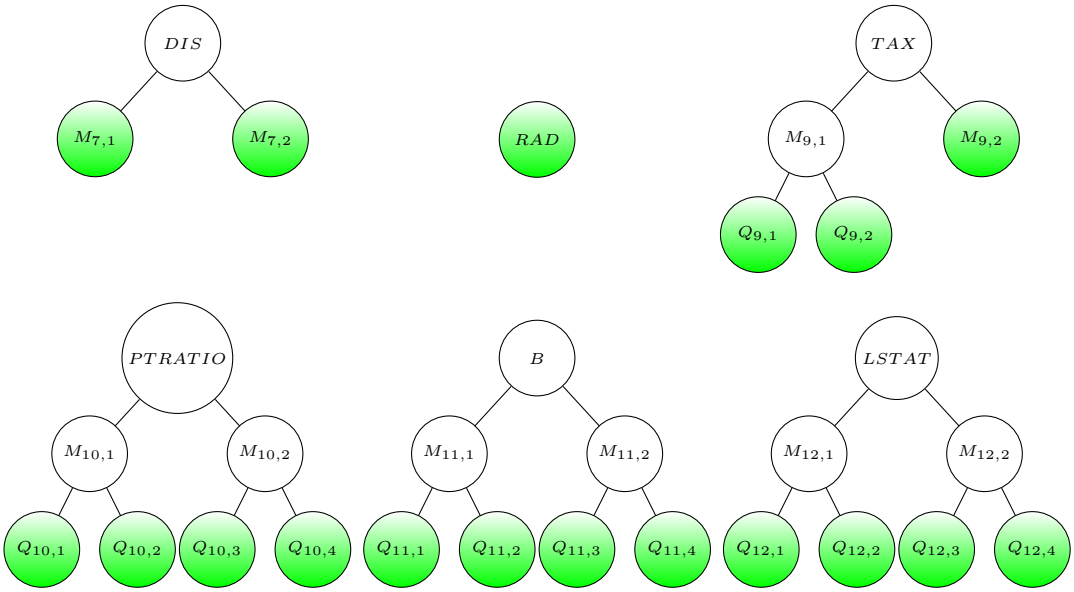


Figure 3.7: Less granular representations for the last six hierarchical categorical variables in the `housing` dataset for the solution in Figure 3.5 with $\text{MSE}^*((\mathcal{S}_j^*)_{j\in\mathcal{J}}) = 23.37$ and $c = 32$. Note that this is the solution that achieves the minimum AIC

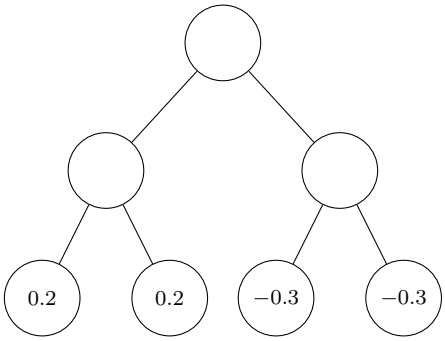


the subtree in Figure 3.9. The same holds for $\mathcal{T}_2$. The error is taken $\varepsilon_i \sim N(0, \sigma^2)$ for different values of σ^2 given below. We have $n = 3000$ individuals, evenly distributed across the different combinations of categories $l_1 \in \mathcal{L}(\mathcal{T}_1)$ and $l_2 \in \mathcal{L}(\mathcal{T}_2)$. The purpose of this section is twofold. First, we illustrate how our approach is able to recover the pruned tree underlying our synthetic data. Second, we carry out a study for assessment of the generalization error of the final chosen model.

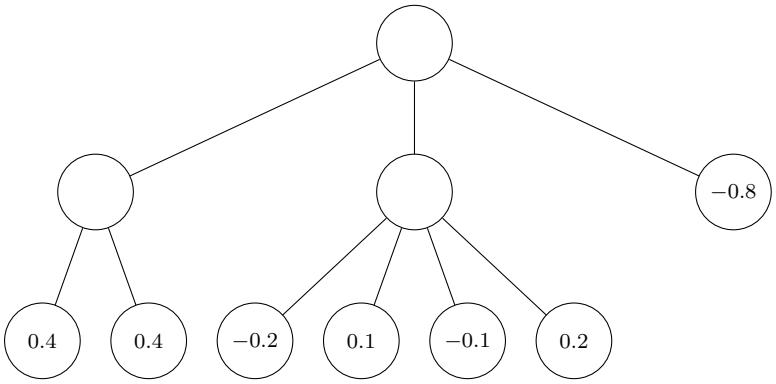
Let us consider $\sigma^2 = 0.04$ and solve Problem (3.11) for the 10 values of c in the set $\{2, \ldots, 11\}$. Figure 3.10(a) shows the pareto frontier for the number of coefficients to be estimated in the reduced model versus the MSE. For small values of MSE, the chosen nodes are the 8 green leaf nodes in Figure 3.9, which implies that our methodology is able to successfully detect the pruned tree underlying each hierarchical categorical variable in our data. Similar conclusions can be drawn when $\sigma^2 = 0.16$ (Figure 3.10(b)) and $\sigma^2 = 0.36$ (Figure 3.10(c)).

To end the numerical section, we provide an estimation for the MSE and the complexity of the reduced model using a 10-fold cross validation approach, showing that our procedure works properly with the available individuals (training sample), but also for future individuals (testing sample). For each fold, the training set is used to solve Problem (3.11) and get $\mathcal{S}_j^*$, $j \in \mathcal{J}$. Once the subtrees are found, and thus the reduced linear regression model, we calculate its MSE for the training and testing sets, which are plotted in Figures 3.11(a)-3.11(c) for the different values of σ^2. As can be observed, the MSE values for the training sets (red lines) are only slightly

Figure 3.8: Trees associated with the two hierarchical categorical variables in the synthetic dataset together with β_{jl}^{S}, $l \in \mathcal{L}(\mathcal{T}_j)$



(a) Tree $\mathcal{T}_1$ for hierarchical categorical variable $j = 1$



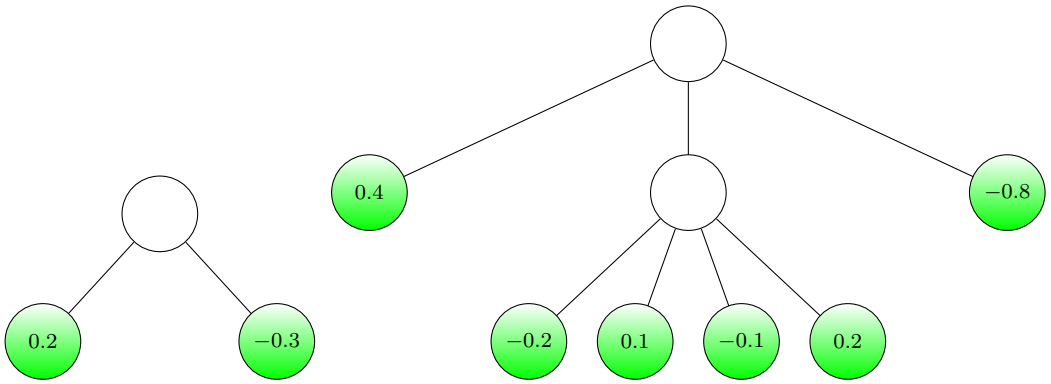
(b) Tree $\mathcal{T}_2$ for hierarchical categorical variable $j = 2$

smaller than those for the testing sets (blue lines). Then, in view of results, we can conclude that our methodology generalizes well.

3.4 Chapter summary

In this work a new methodology to deal with hierarchical categorical variables, i.e., categorical variables that can be measured at different levels of granularity, has been developed. Through a Mixed Integer Convex Quadratic Problem with Linear Constraints, we study the tradeoff between accuracy and model complexity. Our proposal has been tested on both real-world and synthetic datasets. The numerical section shows that much less granular representations for the hierarchical categorical variables can be found at the expense of slightly damaging the accuracy.

Figure 3.9: Pruned tree and less granular representation of the two hierarchical categorical variables in Figure 3.8 from the synthetic dataset



(a) Pruned tree for the first hierarchical categorical variable

(b) Pruned tree for the second hierarchical categorical variable

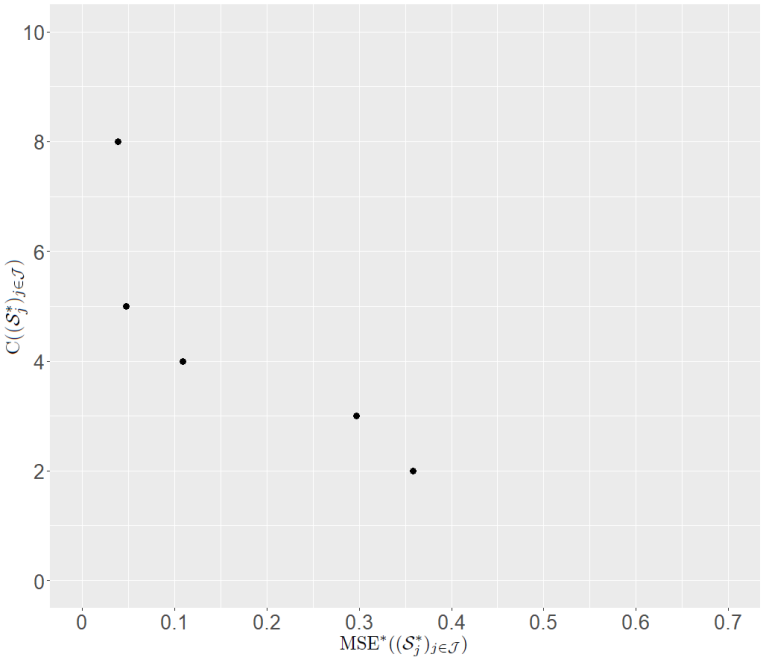
					Tree Figure 3.1	Optimal Tree Figure 3.3(a)	Optimal Tree Figure 3.3(b)	Optimal Tree Figure 3.3(c)
U.S.	West	Pacific		WA	0.058	0.065	0.086	0.109
				AK	0.720	0.716	0.651	
				CA	-0.149	-0.138	-0.114	
				HI	-0.978	-0.992	-0.990	
				OR	-0.015	-0.010	0.002	
		Mountain		ID	-0.302	-0.313	-0.199	-0.161
				WY	0.143	0.143		
				NV	0.481	0.499		
				UT	-0.459	-0.466		
				CO	-0.300	-0.303		
				AZ	-0.473	-0.459		
				NM	-0.273	-0.263		
				MT	-0.131	-0.144		
	South	West-South Central		OK	0.692	0.683	0.705	0.567
				AR	0.686	0.678	0.722	
				TX	0.416	0.408	0.432	
				LA	0.355	0.345	0.378	
		East-South Central		TN	0.529	0.525	0.575	0.631
				MS	0.545	0.539		
				AL	0.330	0.330		
				KY	0.681	0.683		
		South Atlantic		NC	0.099	0.098	0.306	0.368
				DE	0.043	0.056		
				FL	0.284	0.282		
				GA	0.129	0.121		
				MD	0.325	0.337		
				SC	0.373	0.369		
				WV	0.364	0.359		
				DC	0.350	0.393		
				VA	0.535	0.538		
	Mid-West	West-North Central		KS	0.638	0.659	0.538	0.580
				MN	0.302	0.327	0.201	0.230
				MO	0.581	0.575	0.574	0.611
				NE	0.069	0.073	0.069	0.100
				ND	0.123	0.132	0.097	0.103
				SD	0.019	0.019	0.016	0.014
				IA	-0.077	-0.072	-0.088	-0.057
		East-North Central		IL	0.201	0.211	0.291	0.336
				IN	0.527	0.526		
				MI	0.239	0.243		
				WI	0.145	0.148		
				OH	0.386	0.389		
	North-East	Middle Atlantic		PA	-0.099	-0.102	-0.076	-0.025
				NJ	0.074			
				NY	-0.183			
		New England		ME	0.327	0.121		
				VT	0.241			
				MA	-0.053			
				RI	0.102			
				CT	-0.311			
				NH	0.183			

Table 3.1: Coefficients associated with four different representations for *geography* variable in the `cancer-reg` dataset

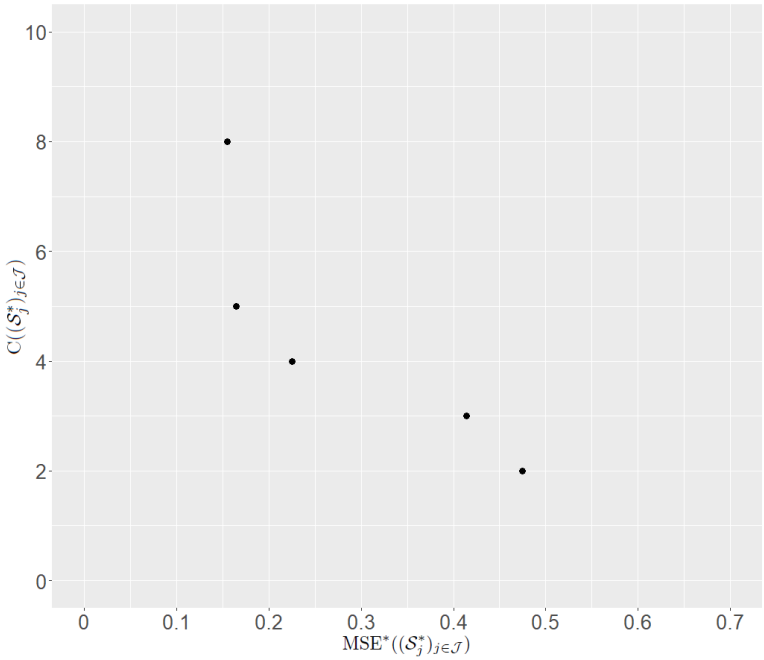
Variable	Name	Description	Type	Discretized
Predictor	CRIM	Crime rate by town	Continuous	Yes
	ZN	Proportion of residential land zoned for lots greater than 25.000 square feet	Continuous	Yes
	INDUS	Proportion of nonretail business acres per town	Continuous	Yes
	NOX	Nitrogen oxide concentrations	Continuous	Yes
	RM	Average number of rooms	Continuous	Yes
	AGE	Proportion of owner units built prior to 1940	Continuous	Yes
	DIS	Weighted distances to five employment centres	Continuous	Yes
	RAD	Index of accessibility to radial highways	Continuous	Yes
	TAX	Full value property tax rate ($/$10.000)	Continuous	Yes
	PTRATIO	Pupil-teacher ratio by town school district	Continuous	Yes
	B	Black proportion of population	Continuous	Yes
	LSTAT	Proportion of population that is lower status	Continuous	Yes
	CHAS	1 if tract bounds river; 0 otherwise	Binary	No
Response	MEDV	Median value of owner-occupied homes (in $1000's)	Continuous	No

Table 3.2: The predictor and the response variables in the `housing` dataset

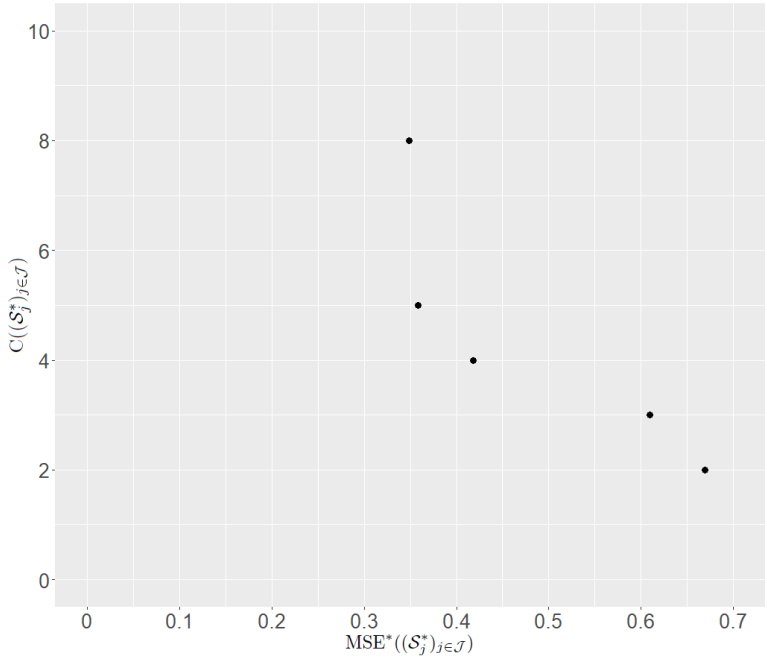
Figure 3.10: Pareto frontier for MSE versus the number of coefficients to be estimated in the reduced model for the synthetic dataset for different σ^2 values



(a) $\sigma^2 = 0.04$



(b) $\sigma^2 = 0.16$



(c) $\sigma^2 = 0.36$

Figure 3.11: Average MSE (10-fold CV) versus the imposed threshold c when σ^2 changes

(a) $\sigma^2 = 0.04$

(b) $\sigma^2 = 0.16$

(c) $\sigma^2 = 0.36$

Chapter 4

Variable selection for Naïve Bayes classification

In this chapter we propose an alternative sparse method for databases with dependent features. In particular, we embed a variable reduction algorithm within the NB's scheme to produce a sparse version of the classifier. Our aim is two-fold: on one hand, sparsity is pursued in the sense that only a subset of predictive features is used by the classifier's construction, making the so-obtained classifier more interpretable, and, on the other hand, we have a flexible framework to choose the accuracy measure to be optimized so that the classifier's performance does not worsen with respect to the classic NB. Our proposal leads to a smart search, which yields competitive running times. Numerical results in both balanced and unbalanced datasets show the competitive results our approach achieves when compared against well-referenced feature selection approaches.

4.1 Introduction

Some works have addressed different strategies for variable reduction for the NB. In Zhang et al. [2009], the use of the principal components technique and genetic algorithms to remove irrelevant and redundant features are examined. The Evolutional Naïve Bayes [Jiang et al., 2005] is a wrapper which also performs a genetic search to select a subset from the whole set, although it is sensitive to many parameters, which is disadvantageous in practice. Other studies which are also focused on *hard variable selection* approaches to reduce the number of redundant predictors are Bermejo et al. [2014] and Mukherjee and Sharma [2012]. In this sense, Langley and Sage [1994] define the *selective Naïve Bayes* (SNB) classifier, which is based on a wrapper approach [Kohavi and John, 1997]. However, due to the complexity of the involved search algorithm and its tendency to make overfitting, the SNB does not perform well on large datasets [Boullé, 2007]. Therefore, a Bayesian approach - defined as SNB(MAP) - is considered in Boullé [2007] to improve the performance of the SNB so that a compromise between the performance of the classifier and the sparsity is found. Another example can be found in "ann" Ratanamahatana and Gunopulos [2003], which proposes a method that combines NB and decision trees.

However, as pointed out by Boullé [2007], it is important to "*exploit multivariate preprocessing methods in order to circumvent the Naïve Bayes assumption*". In this chapter, we adopt this scheme and propose a *hard variable selection* process which is motivated by the conditional independence assumption of the NB. It is known that the NB is Bayes-optimal (that is, it guarantees the minimum classification error), when the predictors are independent conditioned to the class [Kuncheva, 2006]. On the other hand, it is also well documented in the literature that conditional independence is a sufficient condition but not necessary to get the optimal NB [Domingos and Pazzani, 1997; Hand and Yu, 2001; Hastie et al., 2001]. Even if the fact that features are conditionally independent might not make a significant difference with respect to the situation where features are correlated, such slight difference in the NB performance may be crucial for some real contexts (cancer diagnosis, for example). The sparse version of the NB

proposed in this work, which is suitable for dealing with correlated patterns in datasets, is obtained by integrating a variable reduction method in such a way that only certain combinations of features, chosen according to their degree of dependence, are considered. Other papers have considered before correlations among the features as is the case of Hall [2000], Jiang et al. [2019] and Rezaei et al. [2018]. The former is the filter CFS, which was introduced in Section 1.1.1 and is based on the assumption that a good subset of attributes should be highly correlated with the response variable but there should exist few dependencies among them. This hypothesis is also used in Jiang et al. [2019], where a correlation-based feature weighting filter for NB is developed. In Rezaei et al. [2018], clustering is used to detect groups of correlated features and select only a small number of attributes. In particular, the optimal number of clusters stems from the mean silhouette score, which measures how similar a variable is to its own cluster compared to other clusters.

Additionally, the novel strategy can be implemented using the most adequate performance measure given the properties of the datastets. Minimizing the overall misclassification rate is always an option, but, for example, if datasets are unbalanced, the AUC (area under the ROC curve) may be preferred, since it is sensitive to class imbalance and, therefore, achieves a better compromise among the correct classification rates for the different classes. Recent works have considered different alternative performance measures, [Jiang et al., 2012, 2019; Zhang et al., 2020]. For instance, the Randomly Selected Naïve Bayes [Jiang et al., 2012] considers the classification accuracy (ACC), AUC or conditional log likelihood; whereas in Jiang et al. [2019]; Zhang et al. [2020], two class-specific attribute weighted Naïve Bayes versions are defined.

Not only our method establishes the sparsity in terms of the correlation among the covariates and is flexible so that the most convenient classification measure can be used, but also it is a cost-sensitive classifier. In particular, the inclusion of constraints on the proportions of correctly classified instances of groups at risk may be convenient for having direct control over their misclassification rates and obtaining adequate results for them [Benítez-Peña et al., 2019; Blanquero et al., 2021a,b]. That is, whereas the global performance criterion is optimized, further control can be added via performance constraints on the groups of interest in each case. As it will be detailed, the sparse NB defined in this work is able to integrate such performance constraints.

This chapter is organized as follows. In Section 4.2, a brief review of the NB is done, the notation is introduced, and some performance measures typically used in classification are reviewed. A numerical example motivating our approach for a sparse NB is presented next. In Section 4.3, the proposed version of sparse NB is described. Section 4.4 illustrates the new sparse classifier. Synthetic datasets as well as ten well documented real databases with different properties will be thoroughly analyzed, considering different performance measures and/or adding performance constraints in groups of interest. A complete discussion concerning the performance results, sparsity and running times of the proposed methodology in comparison

with benchmark approaches will be given. Finally, some conclusions to this work are described in Section 4.5. Further information concerning the properties of the considered datasets and the choice of the tuning parameters will be described at the Supplementary Material (Appendix C).

4.2 Preliminaries

4.2.1 The Naïve Bayes classifier and performance measures

Consider a classification problem with a set of p features $(X_1, \ldots, X_p)$ and K possible classes. Given a new observation $\mathbf{x} = (x_1, \ldots, x_p)$, the aim is to assign $\mathbf{x}$ to one of the K classes. As introduced in Section 1.2.1, the key assumption of the NB is the independence of the features conditioned to the class, which implies that

$$p(\mathbf{x}|C_k) = p(x_1, \ldots, x_p|C_k) = p(x_1 \mid C_k) \ldots p(x_p \mid C_k) \tag{4.1}$$

and therefore, the probabilities of interest $p(C_k \mid \mathbf{x})$ are computed in a straightforward manner as proportional to (4.1). Note that, in (4.1), a probability distribution for the features conditioned to the class $X_i \mid C_k$ needs to be chosen by the user and estimated by some statistical method as, for example, a maximum likelihood criterion.

Several measures can be used to study a classifier's performance, see for example Sokolova and Lapalme [2009]. In real contexts, besides good overall classification rates, high classification rates for specific classes may be sought. For this reason, throughout this work, we shall consider the classic *Recall of each class k* (*Recall*$_k$) for $k = 1, \ldots, K$, and also, the *accuracy (ACC)* and the *precision*, which are defined as follows,

$$\textit{Recall}_k = \frac{(\textit{True Class } k) \times 100}{\textit{Number of individuals in class } k}, \tag{4.2}$$

$$\text{ACC} = \frac{(\sum_k \textit{True Class } k) \times 100}{\textit{Total number of individuals}}, \tag{4.3}$$

$$\textit{precision}_k = \frac{\textit{True Class } k}{(\textit{True Class } k) + (\textit{False Class } k)}, \tag{4.4}$$

as well as the AUC.

4.2.2 The independence assumption: a numerical example

The effect of the independence assumption over the performance of the NB when correlated features are analyzed, has been studied in the literature, see Domingos and Pazzani [1996, 1997]; Hand and Yu [2001]; Hastie et al. [2001]; Zhang [2004]. As commented in Section 4.1, the conclusion is that, even though the independence assumption is not satisfied, the classifier's performance may not be considerably altered. However, using just a properly chosen subset of

the variables may make the independence assumption less violated, and the accuracy improved (on top of the fact that a model with less variables is more explainable).

In order to illustrate how the violation of the independence assumption may affect the performance of the NB, consider the next numerical example. A sample of size 2000 of a random vector (X_1, X_2, X_3, X_4) is simulated for two classes from a multivariate Normal distribution in such a way that the random variables are independent conditioned to the classes except for X_1 and X_2 which are correlated according to a Pearson coefficient of 0.95. A Gaussian NB classifier (that is, $X_i \mid C_k \sim N\left(\mu_{i,k}, \sigma^2_{i,k}\right)$ for $i = 1, 2, 3, 4$, $k = 1, 2$, where $\mu_{i,k}$ were randomly selected in the interval [1,7]) was run using all possible subsets of features and the results are shown in Table 4.1. The accuracy when all the variables are used is equal to 78.28, a value that is improved if the set $\{X_1, X_3, X_4\}$ is considered (accuracy equal to 79.94).

Table 4.1: Performance rate for all possible combinations of features in a multivariate normal simulated example.

Combination of variables	X_1	X_2	X_3	X_4	X_1, X_2	X_1, X_3	X_1, X_4	X_2, X_3	X_2, X_4	X_3, X_4	X_1, X_2, X_3	X_1, X_2, X_4	X_1, X_3, X_4	X_2, X_3, X_4	X_1, X_2, X_3, X_4
ACC	68.08	68.51	68.55	69.01	68.38	74.99	75.08	74.86	75.25	75.68	73.58	73.85	79.94	79.84	78.28

Having illustrated that using just a subset of the features may improve accuracy, we face the combinatorial problem of finding the adequate set of features to be used. The previous *brute force* procedure, where all possible combinations of features are examined, turns out infeasible in practice, especially for large databases. Instead, in this work we propose a variable reduction method in which only certain combinations of features are sampled and evaluated. Such combinations, as will be seen in Section 4.3, shall be chosen by considering the dependencies among the features.

4.3 A sparse Naïve Bayes

As commented in Section 4.2.2, considering all possible combinations of features to determine the best one is hard from a computational point of view, especially for large datasets since a total of $2^p - 1$ sets should be evaluated. The aim of this section is to describe an efficient methodology to guide the search of the subset of features, by inspecting only some subsets selected in terms of the dependence among features. As a result, a sparse, computationally tractable NB is obtained.

4.3.1 Description of the method

The variable reduction strategy proposed in this section is based on a clustering of features made in terms of their dependencies. As commented in Section 4.2.1, the key assumption of the Naïve Bayes is the independence of the features conditioned to the class. The novel method presented in this work aims to preserve the independence assumption without damaging the predictive power of the classic NB. In other words, our methodology helps to select variables that are as independent as possible while provides good classification accuracy. To do that, we consider a dependence measure between random variables X and Y, which increases with the degree of dependence between the variables. First, consider for $i, j \in \{1, \ldots, p\}$ and $k = 1, \ldots, K$, the dependence between feature X_i and feature X_j conditioned on class C_k. In order to have a unique, summarized measure of dependence between X_i and X_j, let $\mathcal{M}$ be the matrix whose elements ($\mathcal{M}_{ij}$) represent the maximum dependence among all classes, between X_i and X_j. Note that such a choice represents the worst case scenario. A number of dependence measures proposed in the literature can be selected: Pearson correlation coefficient, Spearman's rank-order correlation coefficient, Hoeffding D statistic (see Hoeffding [1948]), the mutual information coefficient (MI) [Linfoot, 1957], the Maximal Information coefficient (MIC) [Reshef et al., 2011] or the distance correlation coefficient [Székely et al., 2007], among others. We tried using these different measures and similar results were obtained (see Section 4.4 and the Supplementary Material in Appendix C). Therefore, since the mutual information measure enables us to work with both continuous and categorical variables and has been widely used in the literature [Kinney et al., 2010; Sharpee et al., 2004], we will select this measure. This coefficient quantifies the information about one variable X provided by a different variable Y, and it is defined as

$$I(X, Y) = \int_Y \int_X p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) dxdy,$$

in the case of continuous variables. In the categorical case, the previous formula can be rewritten in terms of sums. The previous dependence measure can be computed by the function `mutinformation` from the `infotheo` package of the Statistical software environment R [R Core Team, 2017]. An illustration of the matrix $\mathcal{M}$ for the real dataset *Statlog (Australian Credit Approval)* (`australian`) from the UCI Machine Learning Repository [Lichman, 2013] is represented by Figure 4.1. The dataset concerns credit card applications, and is formed by 14 variables and two classes (+/-). Moreover, to visualize the different correlation patterns of the real-life datasets used throughout this work, at the end of Appendix C, the associated matrices $\mathcal{M}$ using the MI measure are represented via heatmaps.

Next, with the aim of performing a cluster analysis in terms of the degree of association among the features, a dissimilarity matrix $\mathcal{H}$ of dimension $p \times p$ is defined in terms of matrix

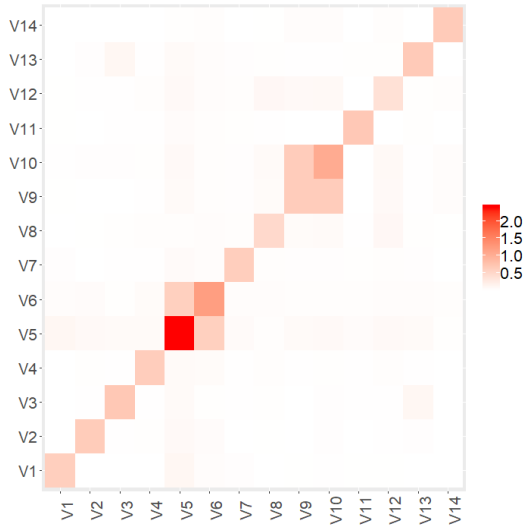


Figure 4.1: Heatmap associated to matrix $\mathcal{M}$ (based on MI correlation) corresponding to the `australian` dataset

$\mathcal{M}$ by the elements

$$\mathcal{H}_{ij} = \frac{M^* - \mathcal{M}_{ij}}{M^*}, \tag{4.5}$$

where

$$M^* = \max_{i,j \in \{1,\dots,p\}} \mathcal{M}_{ij}.$$

Note that, under the previous definition, the elements of $\mathcal{H}$ are bounded below by zero, where this value represents the maximum degree of dependence. Moreover, the upper-bound of the elements of $\mathcal{H}$ is one, which represents the minimum degree of dependence. Therefore, the higher the values of $\mathcal{H}_{ij}$ are, the less dependence exists between X_i and X_j, according to the selected dependence measure.

Note that, as described in the first section of Appendix C, the results obtained are rather robust regarding the dependence measure. Once the dependence measure is set, the classifier's performance measure to be maximized in the embedded Variable Selection strategy can be chosen, among the previously described measures in Section 4.2.1, according to the user's convenience and the properties of the dataset. Generally, the ACC is selected, but in some cases, e.g. for unbalanced datasets or when there exist critical classes, our proposal replace ACC with AUC, *precision* or a certain *Recall*. Thus, the novel method turns out specially advisable for datasets where the classes are unbalanced and/or of different importance. The selection of the dependence and the classifier's performance measures is the first step of our algorithm (see Algorithm 4).

Once we have chosen a dependence measure, and the elements of the matrix $\mathcal{H}$ are computed, we perform a hierarchical cluster analysis of features according to the dissimilarity matrix $\mathcal{H}$ (step 2 of the algorithm).

In the obtained dendrogram, the vertical axis represents the degree of dissimilarity. The higher the value of the height is, the less dependent the variables are, according to the dependence measure. For example, the dendrogram corresponding to the `australian` dataset is given by Figure 4.2. Such a dendrogram has been obtained using the routine `hclust` of the Statistical software environment R. In this case, V_5 and V_6 are highly dependent, as well as V_9 and V_{10}. However, the rest of variables are almost independent, since they cluster at heights between 0.9 and 1.

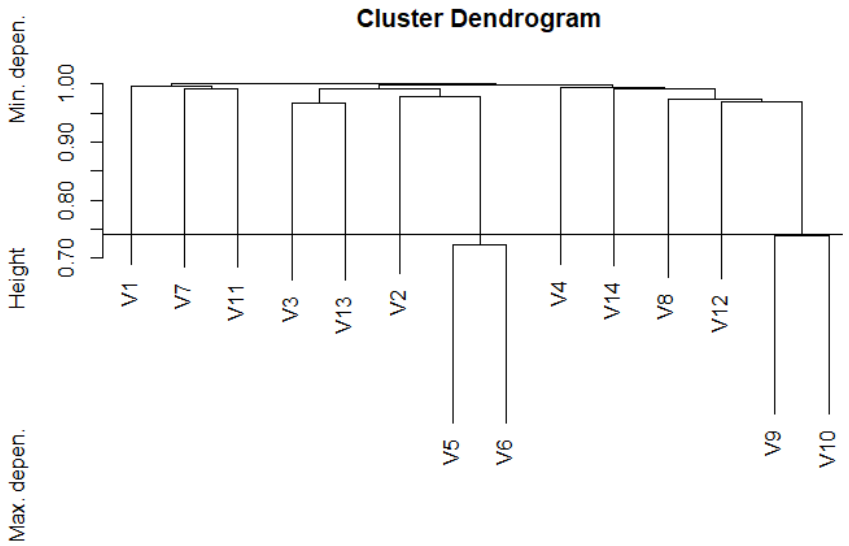


Figure 4.2: Cluster dendrogram (based on MI correlation) corresponding to the `australian` dataset

Once the dendrogram is built, a (not necessarily regular) grid of a specified number C of cuts along the height is fixed. The basic idea underlying the variable reduction strategy is to examine at each cut (or threshold) of the grid several combinations of features, in such a way that only one feature is selected per cluster since all elements in a cluster are assumed to be strongly dependent. As an example, consider Figure 4.2 and assume that one of the C cuts is $c = 0.74$ (horizontal line). Then, we consider that there are 12 clusters: 10 clusters formed by only one feature and the clusters $\{V_5, V_6\}$ and $\{V_9, V_{10}\}$. And, therefore, four independent combinations would be selected at this threshold: $(V_1, V_7, V_{11}, V_3, V_{13}, V_2, \boldsymbol{V_5}, V_4, V_{14}, V_8, V_{12}, \boldsymbol{V_9})$, $(V_1, V_7, V_{11}, V_3, V_{13}, V_2, \boldsymbol{V_5}, V_4, V_{14}, \ V_8, V_{12}, \boldsymbol{V_{10}})$, $(V_1, V_7, V_{11}, V_3, V_{13}, V_2, \boldsymbol{V_6}, V_4, V_{14}, V_8, V_{12}, \boldsymbol{V_9})$ and $(V_1, V_7, V_{11}, V_3, \ V_{13}, V_2, \boldsymbol{V_6}, V_4, V_{14}, V_8, V_{12}, \boldsymbol{V_{10}})$. Note that the higher (lower) the value of the cut, the more likely we are to choose independent (dependent) variables.

Although the previous strategy reduces the computational cost of the *brute force* approach, it still may be costly for large datasets that originate a complex dendrogram with many combinations per threshold. In addition, removing some of the features from the combinations may

lead up to sparser and more accurate models, since it might happen that the (independent) variables selected in the combinations have a very low predictive power. In order to strive to avoid such inconveniences, we propose a refinement of the strategy as follows. First, a maximum number S of combinations per threshold is set (if the total number of possibilities for a given threshold c, $nc(c)$, does not exceed S, then all of them will be considered). In the previous example, $nc(0.74) = 4$. We should point out that parameters C and S are used to alleviate the computational burden, since, as commented before, C fixes the number of cuts along the height in the dendrogram, and S the maximum number of combinations per threshold to be evaluated. Therefore, the higher C and S are, the higher the computer time is. For this reason, we will fix reasonable values for these parameters in Section 4.4.3. Second, for each cluster of variables to be examined, a value q representing the probability of selecting this cluster for extracting randomly a variable to be included in the combination is also set. If we fix $q = 0.4$, the previous four combinations become $(V_1, V_7, V_8, V_{13}, V_{14})$, $(V_1, V_4, V_6, V_8, V_9, V_{12}, V_{14})$, (V_2, V_8) and $(V_4, V_6, V_{10}, V_{11})$, respectively. The parameter q is directly related to the sparsity degree: the lower the value of q is, the less variables are inspected (the expected number variables to be considered is equal to $q \times p$). The choice of the values $\{C, S, q\}$ will be discussed in Section 4.4.

Once the set of combinations of features to be evaluated is reduced, the NB would be implemented and, its performance and feasibility on the constrains considered, evaluated for each combination. This is summarized in step 3 of the Algorithm 4.

Finally, the feasible combination yielding the highest performance measure (accuracy, AUC or whatever chosen measure) would be considered the best, taking also into account the whole set of variables in this comparison (step 4). For `australian`, if no constraint is imposed, the features selected by our model are $(V_1, V_4, V_6, V_8, V_9, V_{12}, V_{14})$, which achieve an ACC of 86.76, whereas the whole set of variables returns 85.29. According to the results, it can be deduced that our model has kept the important features, using only a half of the total set. However, in this dataset, the positive class (the load is granted) is the most risky. Then, if we impose e.g. that *Recall* + > 92, the combination of features (V_2, V_8) would be the selected one.

A summary concerning the strategy for the sparse NB is given by Algorithm 4.

4.4 Numerical Illustrations

In this section, the behaviour and performance of our approach is illustrated throughout an extensive empirical study, using both simulated and real datasets. In the first case, a synthetic data set is simulated in order to test how the performance and level of sparsity of the proposed sparse NB changes with the level of dependence among the features. Second, ten real datasets from the UCI Machine Learning Repository [Lichman, 2013], presenting different correlation patterns, different degrees of unbalancedness and some of them combining both continuous

Algorithm 4: Pseudo-code of the sparse NB

1. Select the dependence and the classifier's performance measures.
2. Perform cluster analysis and build the dendrogram.
3. Variable reduction strategy: set specific values for the parameters $\{C, S, q\}$ and initialize $\mathcal{F} = \emptyset$.

for $c = 1, \ldots, C$ **do**
  for $s = 1, \ldots, \min\{nc(c), S\}$ **do**
    (a) Obtain the $s-$th combination of features. For each cluster only one variable is randomly selected with probability q, and none with probability $1 - q$.
    (b) Construct the classifier for the $s-$th combination of features.
    (c) Evaluate the selected classifier's performance measure and if feasible, add it to $\mathcal{F}$.
  end
end

4. Variable Selection: select the combination of variables leading to the best performance, among those in $\mathcal{F}$.

and categorical variables, will be analyzed under the sparse NB described in Section 4.3. In the experiments, the performance rates of the classifier shall be estimated according to an 10 runs $10-$fold cross validation procedure. At each fold, the dataset is split into three sets, the so-called training, validation and testing sets. A tenth of the dataset is used as testing set, and the remaining nine tenths are for training set $\left(\frac{2}{3} \times \frac{9}{10}\right)$ and validation set $\left(\frac{1}{3} \times \frac{9}{10}\right)$. Steps `2, 3(a)` and `3(b)` of Algorithm 4 are implemented on the training set. The different classifiers built in this way are compared according to their performance results (step `3(c)`) on the validation set. The classifier (combination of features) with the highest performance measure on the validation set is chosen, and its average performance rates on the testing set are reported. Special emphasis will be made on the performance behavior and sparsity of the solutions of the proposed method.

4.4.1 Parameters setting

The probability distribution for the features conditioned to the class $X_i \mid C_k$, for $i = 1, \ldots, p$, $k = 1, \ldots, K$, needs to be selected. It is well-known in the literature that the performance of the NB classifier improves when features are categorized using any discretization method [Liu et al., 2002; Boullé, 2004; Boullé, 2006]. Therefore, instead of imposing a specific probability distribution (such as the Gaussian), we adopted the discretization method based on an entropy criterion (see Dougherty et al. [1995]) and used the `mdlp` routine [Fayyad and Irani, 1993] from the `discretization` package of R.

Now, we discuss the choice of the parameters $\{C, S, q\}$ and the performance criterion.

Choice of the parameter C

The value of C, which represents the number of cuts in the vertical axis of the dendrogram, is critical for a proper sampling. As a default value, we propose to select the points of the

grid where features are clustered. When the routine `hclust` of R is used to generate the dendrogram (as in this work), one has $C = p - 1$, where p is the number of features. In addition, `hclust` specifies where to make the cuts. However, a large value of C may slow down the execution of the algorithm notably and, on the other hand, it may lead to overfitting. For this reason, C will be defined as $\min\{p-1, 100\}$. As will be seen next, in Section 4.4.2, such a choice yields a right balance between the performance and the computational time for the considered datasets.

Choice of the parameter S

Regarding the value of S, which represents the maximum total of combinations for each cut, we tested several possible values for this parameter (see Appendix C), and settled on the final choice $S = 25$. Note that under the previous choices of C and S a total of $\max\{25 \times (p-1), 25 \times 100\}$ combinations of features will be evaluated under the proposed sparse NB in contrast to the total number of possible combinations, equal to $2^p - 1$.

Choice of the parameter q

Small values of q are associated with more sparsity (since fewer variables would be included in the combinations to be examined). Therefore, q should be selected in such a way that it provides a compromise between the classifier's performance and the sparsity of the solution. In particular, in Appendix C, different experiments to evaluate how the choice of this parameter affects the results can be found. Here, the selection of this parameter has been addressed according to the dependence matrix $\mathcal{M}$, which is defined in Section 4.3. In particular, when 20% of the matrix elements are higher than 0.1 (that is, from moderate to strong dependence cases), we fix $q = 0.4$ (which implies a sparser solution). Otherwise (few dependent features), $q = 0.6$ will be set.

4.4.2 Simulation study

In this section we analyze how the performance of the sparse NB varies as dependence among features increases. In particular, we simulate data following Witten et al. [2014] and according to the model $\mathbf{y} = \mathcal{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $p \in \{100, 200, 300, 400, 500\}$. The errors $\varepsilon_1, \ldots, \varepsilon_n$ are iid from a $N(0, 2.5^2)$ distribution. The observations (rows of $\mathcal{X}$) are iid from a $N_p(0, \Sigma)$ distribution, where Σ is a $p \times p$ block diagonal matrix, with elements as follows:

$$\Sigma_{ij} = \begin{cases} 1 & \text{if } i = j, \\ \rho & \text{if } i \leq \frac{p}{4},\ j \leq \frac{p}{4},\ i \neq j, \\ \rho & \text{if } \frac{p}{4} + 1 \leq i \leq p,\ \frac{p}{4} + 1 \leq j \leq p,\ i \neq j, \\ 0 & \text{otherwise} \end{cases}$$

We explored various values of ρ, ranging from 0.1 to 0.9. Furthermore, $\beta_i \sim$ Unif[0.9, 1.1] for $1 \leq i \leq \lfloor \frac{p}{4} \rfloor$ and $\beta_i \sim$ Unif[-$\frac{1}{3} - 0.1$, $-\frac{1}{3} + 0.1$] otherwise. In other words, there are two sets of $\frac{p}{4}$ and $\frac{3p}{4}$ correlated features, respectively, and all the features are associated with the response. Finally, two classes are defined according to the sign of y_n, $n = 1, \ldots, 2000$.

The results in Table 4.2 have been obtained using the Mutual Information dependence measure (MI), $S = 25$ and values of q fixed as in Section 4.4.1. Moreover, the performance measure considered for these simulated experiments is the accuracy, and its average performance rates as in (4.3) on the testing set are reported in Table 4.2.

Table 4.2: Average accuracy and sparsity (10 runs 10-fold CV) for simulated datasets.

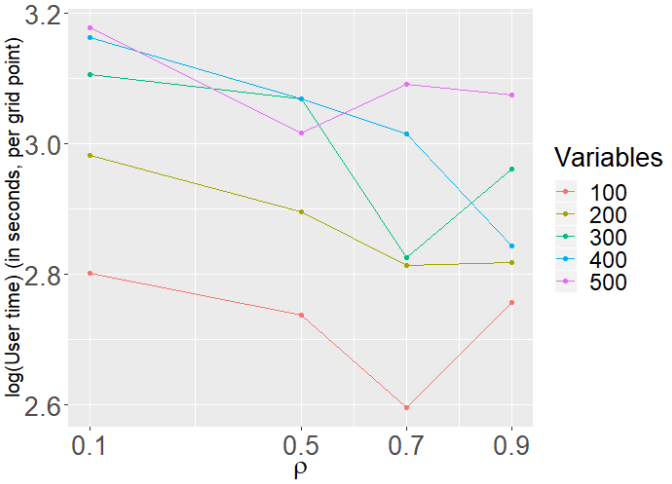
p	*Method*	$\rho = 0.1$		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
		ACC	*Sparsity*	ACC	*Sparsity*	ACC	*Sparsity*	ACC	*Sparsity*
100	**Sparse NB**	88.20	66	93.30	37	93.20	24	95.05	21.30
	Classic NB	89.50	100	87.05	100	87.10	100	86.95	100
200	**Sparse NB**	90.10	113.5	93.40	52.50	95.70	30.50	96.55	20.10
	Classic NB	90.10	200	87.30	200	87.65	200	86.80	200
300	**Sparse NB**	89.85	168.60	92.40	79.70	94.70	37.30	95.95	18.20
	Classic NB	90.15	300	87.00	300	87.15	300	87.85	300
400	**Sparse NB**	91.90	216.30	91.45	94	92.35	46.80	93.65	17
	Classic NB	89.65	400	87.25	400	87.25	400	87.50	400
500	**Sparse NB**	91.90	283.20	90.70	102.70	93.85	29.60	91.90	5.90
	Classic NB	90.15	500	87.05	500	87.45	500	87.55	500

Some conclusions can be drawn. On the one hand, in terms of sparsity levels, the sparse NB returns better results in the presence of moderate to strong dependence cases. For datasets where the dependences among features are weak, $\rho = 0.1$, our sparse strategy is able to remove around one third of the total number of variables whereas, in some cases, the ACC is slightly reduced with regards to the classic NB. While ρ increases, our proposal is able to significantly reduce the number of variables considered, achieving better ACC results than the classic NB, as the curse of dependency is alleviated. On the other hand, Figure 4.3 reports the logarithm of the average user times (in seconds) when the sparse NB is run on Intel(R) Core(TM) i7-7500U CPU at 2.70GHz 2.90GHz with 8.0 GB of RAM. The X-axis shows the ρ values whereas each line represents the number of variables of the dataset (p). Overall, for weak dependences among features, the behaviour of runing time is monotonous respect to the number of variables, but this changes when ρ increases.

4.4.3 Datasets and benchmark approaches

The so-called *Breast Cancer Wisconsin (Diagnostic) Data Set*, *Wine Data Set*, *Mushroom*, *Waveform Database Generator Data Set* (version 2), *ISOLET Data Set*, *Multiple Features Data Set*, *SPECTF Heart Data Set*, *German Credit*, *Page Blocks Classification Data Set* and *Statlog (Australian Credit Approval)* shall be considered. They are described in Table 4.3, whose first three colums report the dataset name, the number of instances and the class split. The number of

Figure 4.3: Scalability



continuous variables (L) and categorical variables (L') are presented in the last two columns. Three of the ten datasets, `SPECTF`, `german` and `page blocks`, are unbalanced datasets, due to the very different sizes of the classes.

Table 4.3: Datasets description

Name	*Instances*	*Class split in %*	*L*	*L'*
`breast cancer`	569	63(*Benign*)/37(*Malignant*)	30	0
`wine`	178	33(*Class* 1)/40(*Class* 2)/27(*Class* 3)	13	0
`mushroom`	8124	51.8(*edible*)/48.2(*poisonous*)	0	22
`waveform`	5000	33.33(*Class* 0)/33.33(*Class* 1)/33.33(*Class* 2)	40	0
`ISOLET`	7797	26 equiprobable classes (0.04)	617	0
`Multiple Features`	2000	9 equiprobable classes (0.11)	649	0
`SPECTF`	267	79(*Abnormal*)/21(*Normal*)	44	0
`german`	1000	70(*Class* 1)/30(*Class* 2)	7	13
`page blocks`	5473	90(*Negative*)/10(*Positive*)	10	0
`australian`	690	55.5(*Negative*)/44.5(*Positive*)	6	8

We aim to compare the novel method with alternative, well-known strategies for feature selection. In this study, we focus on techniques which perform *hard variable selection* and, in consequence, feature weighting approaches as in Jiang et al. [2019] have not been considered here. Specifically, we selected one filter and one wrapper that are well referenced in the literature and that can be easily adapted to the NB classifier to make a fair comparison. Our choice was the filter CFS and the wrapper *Boruta*, both introduced in Section 1.1.1. These methods are widely spread and can be computed by the routines `cfs` and `Boruta`, from R packages `FSelector` and `Boruta`, respectively. In order to adapt the wrapper *Boruta* to the NB classifier, we have used the function `filterVarImp` in R package `caret` as the function that returns the importance of the attributes, instead of the default `getImpRfZ`, which is based on the Random Forest classifier. It is important to highlight that the time limit is not an input parameter of the `cfs` and `Boruta` routines and therefore, differences in the computational

costs were found (to be discussed later). Apart from the previous feature selection methods, that can be applied to any classifier, there are works that specifically deal with variable reduction for the NB. In particular, Boullé [2007] proposes a straightforward Bayesian modern-style approach, the *MAP Approach for Variable Selection* (noted SNB(MAP)), where the conditional probabilities are formulated according to

$$p(C_k|\mathbf{x}) \propto \pi(C_k) \prod_{i=1}^{p} p(x_i|C_k)^{a_i}, \;\; k = 1, \ldots, K. \tag{4.6}$$

In Eq. (4.6), the values $\{a_i\}_{i=1}^p$ are either 1 or 0, depending on whether or not feature i is included in the model. Then, the posterior distribution of the different models (resulting from different choices of $\{a_i\}_{i=1}^p$) is evaluated using a shrinkage prior so that parsimoniuos models are favoured. In the same paper, a search heuristic that performs a fast forward backward selection is described and therefore, it has been implemented in this work to run the different experiments. However, when the number of variables increases, note that the time required to run this method is excessive. For that reason, a time limit of eight hours for the two biggest considered real datasets (`ISOLET` and `Multiple Features`) was fixed. Finally, we have also compared with the Lasso approach for classification (see Vincent and Hansen [2014]), whose goal is precisely the same: obtain good classification performance while selecting few features. The routine `fit` in R package `msgl` has been used.

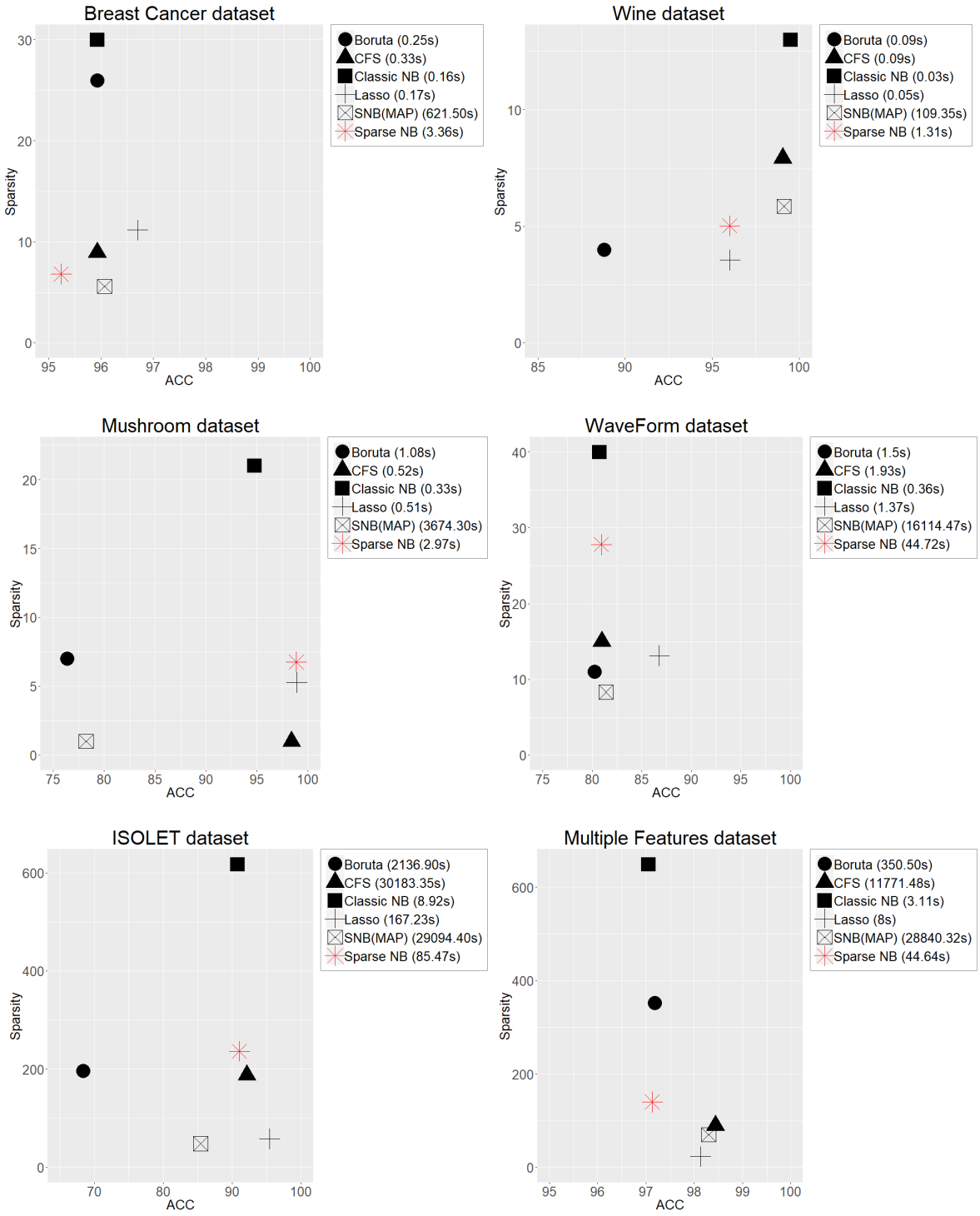
Next, we will break the results down depending on the datasets are balanced or unbalanced. As we will show below, if necessary and motivated by the properties of the dataset, our proposal can be easily adapted in terms of the performance criterion to be optimized and the required constraints on groups of interest. To make a fair comparison, we do not impose any additional constraints and therefore only the performance criterion will change accordingly throughout these sections. However, an illustrative example where constraints are imposed is also included at the end of Section 4.4.5.

4.4.4 Results for balanced datasets

For comparison purposes, consider the same parameters setting than in Section 4.4.2, where for `waveform` dataset, q is equal to 0.6 and, for the remaining balanced databases, $q = 0.4$. We next analyze the performance and sparsity of the method, as well as the running times. The average accuracy, number of variables in the selected combinations and the CPU time in seconds for 1 fold-CV execution are shown by Figure 4.4. Moreover, a comparison between the sparse NB with the above-mentioned feature selection methods is made. The results under the classic NB, CFS, *Boruta*, SNB(MAP) and Lasso methods are also shown.

Several conclusions can be drawn at this point. Note that the performance rates under the sparse NB are comparable to the classic NB using between a half and one third of the variables, except for `waveform`. As commented before, the novel approach is intended to

Figure 4.4: Average accuracy, sparsity and CPU time (10 runs 10-fold CV) for `breast cancer`, `wine`, `mushroom`, `waveform`, `ISOLET` and `Multiple Features` datasets



address databases with correlated patterns and, for this reason, the outperformance of the sparse NB improves with the dependence among the features. Therefore, since the variables of the `waveform` dataset are almost independent, it is expected that the novel sparse strategy does not yield a significant enhancement in this sense, as Figure 4.4 shows.

With regards to the five feature selection methods considered in this study, the next conclusions can be drawn from the figure. Whereas the proposed method achieves competitive ACC and sparsity results, it performs in between the other methods in these two measures. Also, it can be concluded that SNB(MAP) is computationally slower than the sparse NB. In addition, *Boruta* and CFS are less computationally costly than the sparse NB, but when the number of features increase, it turns out to be exceptionally low.

In summary, it can be deduced that, for balanced datasets with dependencies among the features, the proposed sparse NB leads to a significant reduction in the number of features while keeping the power prediction. Also, it can be concluded that in general, for this kind of datasets, our method and the Lasso seem to achieve the best compromise between accuracy, sparsity and running times.

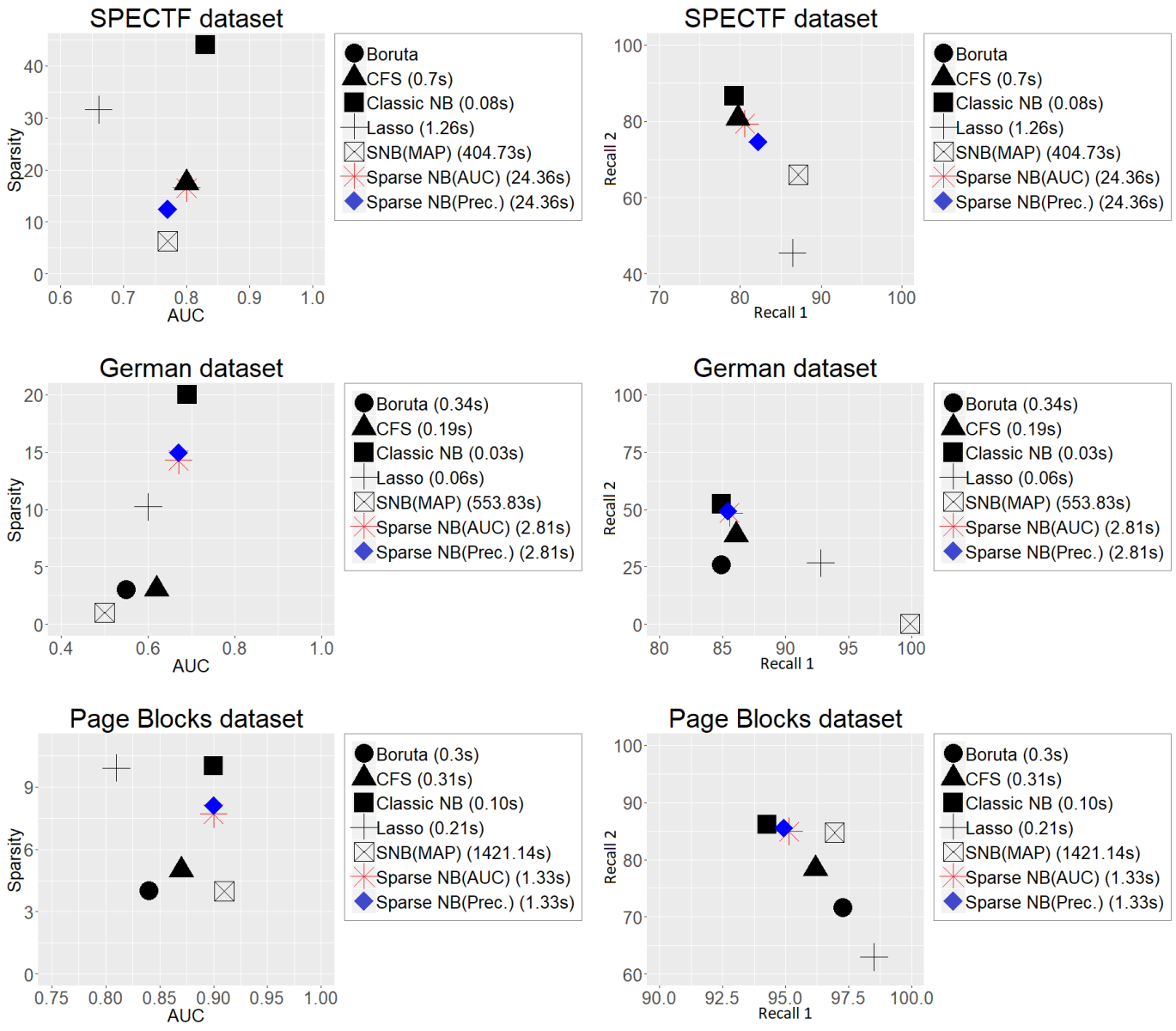
4.4.5 Results for unbalanced datasets

In this section we deal with three unbalanced datasets. The `SPECTF`, `german` and `page blocks`, which are unbalanced according to classes. It implies that the use of the ACC, defined by (4.3), as the performance criterion may not be a sensible choice because of the difference between the classes sizes. Therefore, for these cases, the area under the curve (AUC) as well as the *precision* of the majority class (*Class 1*), calculated by (4.4), will be the measures to be maximized. The former measure, $precision_1$, leads to good $Recall_2$, since will minimize the *False Class 1*. In addition, the performance at each class, will be inspected via the so-called *Recall*. We have considered the previous two performance measures when selecting the set of variables via sparse NB, and the obtained results are shown in red and blue (respectively) in Figure 4.5. Finally, $q = 0.6$ for `german` and `page blocks`, whereas is equal to 0.4 in the case of `SPECTF`.

Again, the performance results, the sparsity results and the running times are reported in Figure 4.5. For each dataset, two graphics are shown. The images on the left represent the AUC versus the sparsity, while the *Recall* of the majority and minority classes ($Recall_1$ and $Recall_2$, respectively) are drawn on the right side. Note that the *Boruta* results for `SPECTF` database are not reported since this dataset does not satisfy the technical requirements of the implementation of that method.

The performance rates under the sparse NB are comparable to the results obtained with all the features, since the AUC (respectively, the *precision*) has been used as performance criterion and the novel approach keeps at least the area under the curve (or *precision*) obtained by the classic NB. The sparse NB is able to reduce to less than half the number of variables in the case of `SPECTF` dataset; it removes one fourth of the variables of `german` dataset

Figure 4.5: Average performance, sparsity and CPU time (10 runs 10-fold CV) for `SPECTF`, `german` and `page blocks` datasets



and one third in `page blocks`. Now, if we compare to CFS, *Boruta*, *SNB(MAP)* and the Lasso, it can be observed how, although they tend to be sparser, they increase significantly the misclassification rate on the minority class ($Recall_2$), since, in general, they tend to increase the correct classification for the majority class ($Recall_1$) and to decrease the minority one. The latest results assert the need to choose an appropriate performance measure according to the properties of the dataset.

Therefore, with regards to the unbalanced databases, the sparse NB provides more balanced *Recall* values, in the sense that the performance of the least frequent class is not so reduced. Another illustration is given by Table 4.4, where the `australian` is considered. As commented before, in this case, the positive class (the load is granted) is the most risky. For these cases, the sparse NB would be the most suitable choice, not only because the performance criterion to be used can be easily adapted but also because while optimizing such criterion, constraints on acceptable performance measures can be included. The second row of Table 4.4 shows the results for the sparse NB if the ACC is considered as performance criterion and no additional constraints are imposed. However, the ACC can be optimized whereas a performance constraint on the *Recall* of the positive class is considered (*Recall* + $>$ 85), as can be seen in the third row of Table 4.4. As a final example, we are interested in maximizing the *Recall* + instead. Note that the improvement in the positive class will be at the expense of reducing the *Recall* - and therefore admissible values for it have been imposed via a threshold value to avoid worsening it, say *Recall* - $>$ 60 (last row).

Table 4.4: Average performance and sparsity (10 runs 10-fold CV) for `australian` dataset using the sparse NB with different performance measures to select the set of variables

Method	*Recall* -	*Recall* +	ACC	*Sparsity*
Classic NB	91.10	78.78	85.61	14
Sparse NB (ACC)	84.59	85.83	85.15	5.78
Sparse NB (ACC); *Recall* + $>$ 85	84.14	86.48	85.19	5.53
Sparse NB (*Recall* +); *Recall* - $>$ 60	79.93	92.35	85.46	1.4

To sum up, for unbalanced datasets with dependent variables, the considered benchmark methods tend to be sparser than our approach but at the cost of damaging unpredictably the performance of the classifier and, in particular, the *Recall* of the least frequent class. In contrast, the novel method allows the user to set the performance measure that best suits it as well as admissible values for specific performance measures, which turns out advantageous for unbalanced datasets or for cases in which misclassification costs are strongly class-dependent.

4.5 Chapter summary

In this work, a new version of the NB classifier for dealing with datasets with correlated patterns is proposed with the aim of improving the sparsity of the solution. In order to achieve sparsity, a variable reduction technique is embedded into the classifier. Such a variable reduc-

tion strategy is based on clustering the features in terms of their dependence degree, and it selects combinations of features that, being as independent as possible, lead to a good performance rate. The performance measure used in the algorithm can be given by the out-of-sample accuracy, or more generally, an estimate of the expected misclassification cost, among others. The proposed methodology has been tested on synthetic datasets and ten real datasets of different sizes and properties. The numerical results show that not only sparse solutions are attained, but also the performance rates are comparable or better than those achieved under the classic version of the NB, where all features are taken into account for classifying. In addition, when compared with benchmark approaches, the novel method turns out especially advisable for datasets where the classes are unbalanced and/or of different importance. This fact stems from the flexibility of our method in the selection of the performance measure and the ability to include constraints on certain performance measures for feature selection, which does not occur with the feature selection approaches proposed in the literature.

Chapter 5

Constrained Naïve Bayes with application to unbalanced data classification

The approach introduced in Chapter 2, one overall criterion is optimized while constraints to demand admissible values for the efficiency measures under consideration are introduced in the model, is explored for improving the NB performance in the classes of most interest to the user. It will be seen that unlike the traditional NB, which is a two-step classifier (estimation first and classification next), the novel approach integrates both stages. In particular, maximum likelihood estimates are replaced here by constrained maximum likelihood estimates, where the constraints control the *Recall* values of the classes of interest.

5.1 Introduction

In this chapter we propose a novel way of controlling misclassification rates, that do not call for using misclassification costs which may be hard to choose and are not usually given [Sun et al., 2007, 2009]. In particular, a new version of the NB is obtained by modeling performance contraints where the *Recall* (proportion of instances of a given class correctly classified) for the classes of interest is forced to be lower-bounded by certain thresholds. In this way, the user is allowed to assign different importance to the different classes according to her preferences. For example, in the previously considered `breast cancer` dataset, it may be desiderable to increase the *Recall* for the *Malignant* class, which is equal to 88.41. As it will be shown in Section 5.3, for this case such rate can be increased up to 91.88. Other example where performance constraints are useful is when fair classification is a requirement as a social criterion, and then the sensitive groups should be protected to avoid the discrimination against race, or other sensitive data [Romei and Ruggieri, 2014]. Acceptable values for the *Recall* of groups at risk could be fixed via the proposed method in this work.

The problem of cost imbalance has been addressed in the literature from two different perspectives: Data-Level techniques and Algorithm-Level approches, see Leevy et al. [2018]. Whereas the former include data sampling methods and feature selection, the latter encompass cost-sensitive and hybrid/ensemble methods which adapt the base classifier to overcome the imbalance. Particularly, our approach can be seen as a cost-sensitive method. Cost-sensitive approaches have already been considered in the literature for well-known classifiers, such as those introduced in Section 1.1.2. However, there is a lack of methodology that allows the user to have full control on the different performance measures of interest at the same time, which is the what we explore in this work.

This chapter is organized as follows. In Section 5.2 the notation is introduced and the proposed version of constrained NB (CNB from now on) is described. Section 5.3 illustrates the usefulness of our novel approach. Eight real databases with different sampling properties are thoroughly analyzed, and a detailed discussion concerning the *Recall* values of the proposed approach compared with the classic NB is given. Some conclusions are considered in Section 5.4.

5.2 The constrained Naïve Bayes

In our approach, the estimation is performed by solving a constrained maximum likelihood estimation problem, constraints being related with thresholds on the *Recall* values for different classes. The aim of this section is to describe the associated optimization problem. As a result, a computationally tractable classifier that allows the user to control its performance is obtained.

5.2.1 Preliminaries on NB classification: notation

Consider again the random vector $(Y, \mathbf{X})$, where $\mathbf{X} = (X_1, \ldots, X_p)$ contains p features and Y identifies the class label. Assume that we have a classification problem with K classes. Then, for each class $k \in \{1, \ldots, K\}$, let π_k denote the prior probability of the class, and assume that $X_j|(Y = k)$ has a probability density function $f_{\theta_{jk}}(x)$, where $\theta_{jk} \in \Theta_{jk}$. For $k = 1, \ldots, K$, define $\boldsymbol{\theta}_k = (\theta_{1k}, \ldots, \theta_{pk})$.

Let $\mathbf{x} = (x_1, \ldots, x_p)$ be a new observation. The aim is to label it on one of the K classes. Then, under the 0-1 loss function, Bayesian Decision Theory establishes that $\mathbf{x}$ is classified in the most probable class according to the conditional distribution. The estimation of the associated parameters may be cumbersome if the number of features p is large. However, the use of the Bayes theorem, in addition to the assumption of independence (conditioned to the class) ease the previous estimation process. As it is well known, the latter assumption implies that the joint density function can be expressed as

$$\begin{aligned} f(x_1, \ldots, x_p, k) &= \pi_k f(x_1, \ldots, x_p \mid k) \\ &= \pi_k f_{\boldsymbol{\theta}_k}(\mathbf{x}) \\ &= \pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j), \end{aligned}$$

and thus the estimation process is reduced to estimate the parameters of each marginal distribution. Then, the NB classifier performs by assigning $\mathbf{x}$ to class k satisfying

$$\pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j) \geq \pi_l \prod_{j=1}^{p} f_{\theta_{jl}}(x_j) \quad \forall l = 1, \ldots, K. \tag{5.1}$$

Given a training sample of size n_1, $(k_1, \mathbf{x}_1), \ldots, (k_{n_1}, \mathbf{x}_{n_1})$, then $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$ is estimated in NB via maximum likelihood [Hogg et al., 2005], and therefore computed as the solution of the optimization problem:

$$\max_{\boldsymbol{\theta}} \sum_{i=1}^{n_1} \log f_{\boldsymbol{\theta}_{k_i}}(\mathbf{x}_i) \tag{5.2}$$

Therefore, the classic NB can be seen as a two-step classifier, where the model parameter is first estimated as $\hat{\boldsymbol{\theta}}$ from a training sample, and then (5.1) is applied under $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

5.2.2 A novel formulation with performance constraints

In order to calibrate the performance of a classifier, many measures have been defined in the literature, see Sokolova and Lapalme [2009]. In particular, the so-called *Recall*$_k$, for $k = 1, \ldots, K$, is defined as the sample fraction of individuals in class k which are correctly classified.

Given a validation sample of size n_2, where $n_2 = \sum_k n_{2,k}$ and $n_{2,k}$ is the size of class k in such a validation sample, $(k, \mathbf{x}_1^{(k)}), \ldots, (k, \mathbf{x}_{n_{2,k}}^{(k)})$, then the *Recall* for class k can be expressed as functions of $\hat{\boldsymbol{\theta}}$,

$$\mathit{Recall}_k(\hat{\boldsymbol{\theta}}) = \frac{1}{n_{2,k}} \sum_{i=1}^{n_{2,k}} C_k(\hat{\boldsymbol{\theta}},\ \mathbf{x}_i^{(k)}),\ k = 1, \ldots, K, \tag{5.3}$$

where

$$C_k(\hat{\boldsymbol{\theta}},\ \mathbf{x}_i^{(k)}) = \begin{cases} 1 & \text{if the individual } \mathbf{x}_i^{(k)} \text{ is classified in class } k \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Unlike the classic NB, based on a two-step approach, the CNB proposed in this work integrates the performance of the classifier (according to expression (5.3)) within the estimation step. In particular, the pursued aim is to estimate $\boldsymbol{\theta}$ as the solution of an optimization problem where the objective function is given using a training sample of size n_1 as in (5.2) and, to prevent overfitting, constraints on (5.3) are imposed on an independent sample (validation set) of size $n_2 = \sum_{k=1}^{K} n_{2,k}$,

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n_1} \log f_{\boldsymbol{\theta}_{k_i}}(\mathbf{x}_i) \\ \text{s.t.} \quad & \frac{1}{n_{2,k}} \sum_{i=1}^{n_{2,k}} C_k(\boldsymbol{\theta},\ \mathbf{x}_i^{(k)}) \geq \alpha_k,\ k = 1, \ldots, K \end{aligned} \tag{CNB}$$

In the previous CNB optimization problem, $\alpha_k \in (0, 1)$ is a threshold, a lower-bound value close to 1, for $k = 1, \ldots, K$, which is fixed by the user according to her requirements about the classification in the different classes. From the point of view of optimization, we assume that the function $f_{\boldsymbol{\theta}_{k_i}}$ is smooth with respect to the parameter $\boldsymbol{\theta}_{k_i}$. Regarding the constraints, they are not smooth and therefore, gradient methods cannot be applied in order to solve Problem (CNB). This fact makes the resolution of (CNB) to be slow, especially for large datasets. However, a proxy version of (CNB) can be written in a more tractable way if the constraints are reformulated in terms of smooth functions as

$$\widetilde{C}_k(\boldsymbol{\theta}, \mathbf{x}^{(k)}; \lambda) = \prod_{l=1, l \neq k}^{K} F(y_{kl}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda), \tag{5.5}$$

where $F(y;\lambda) = \frac{1}{1+e^{-\lambda y}}$ is the sigmoid function and

$$y_{kl}(\boldsymbol{\theta}, \mathbf{x}) = \pi_k \prod_{j=1}^{p} f_{\theta_{jk}}(x_j) - \pi_l \prod_{j=1}^{p} f_{\theta_{jl}}(x_j). \tag{5.6}$$

On the one hand, from the definition of the sigmoid function, it can be seen that $\lim_{\lambda\to\infty} \widetilde{C}_k(\boldsymbol{\theta}, \mathbf{x}^{(k)}; \lambda) = C_k(\boldsymbol{\theta}, \mathbf{x}^{(k)})$, since for large λ values $F(y_{kl}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda)$ will only take the values 0 or 1 depending on the sign of $y_{kl}(\boldsymbol{\theta}, \mathbf{x}^{(k)})$. Then, λ is a hyperparameter big enough so that C and $\widetilde{C}$ are as close as possible. On the other hand, the reason why we use the product function to define $\widetilde{C}$ is explained below. Note that if any class l has associated a density much greater than class k, then y_{kl} will take a large negative value which makes $F(y_{kl}(\boldsymbol{\theta}, \mathbf{x}^{(k)}); \lambda)$ close to 0 and therefore $\widetilde{C}_k(\boldsymbol{\theta},\ \mathbf{x}^{(k)}; \lambda)$ will also be close to 0. From the previous discussion, a differentiable version of the CNB problem is obtained as

$$\begin{aligned} \max_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{n_1} \log f_{\boldsymbol{\theta}_{k_i}}(\mathbf{x}_i) \\ \text{s.t.} \quad & \frac{1}{n_{2,k}} \sum_{i=1}^{n_{2,k}} \tilde{C}_k(\boldsymbol{\theta},\ \mathbf{x}_i^{(k)}) \geq \alpha_k,\ k = 1, \ldots, K. \end{aligned} \tag{SCNB}$$

The smooth formulation (SCNB) can be solved using efficient solvers for nonlinear constrained programming (see, e.g. Birgin and Martínez [2008]). From now on, we refer to (SCNB) as our optimization problem.

Two important remarks need to be made at this point. The first one regards the feasibility of the (SCNB). In a real application, threshold values $\alpha_1, \ldots, \alpha_K$ have to be fixed. As a first option, they could be fixed by the user according to her demand, but it might be the case that (SCNB) is unfeasible. For that reason, we propose a procedure for determining the thresholds in such a way that (SCNB) is always feasible. If we consider a dataset with K different classes, let $\boldsymbol{\theta}^*$ be the model parameter associated with (5.2) and k_0 be the critical class or the class where the method performs the worst. Suppose that the aim is to improve the *Recall* for such class k_0, say

$$\alpha_{k_0} = \frac{1}{n_{2,k_0}} \sum_{i=1}^{n_{2,k_0}} \tilde{C}_{k_0}(\boldsymbol{\theta}^*,\ \mathbf{x}_i^{(k_0)}) + \Delta,$$

with $\Delta > 0$. Then, in order to know the maximum threshold τ for the other classes $k \neq k_0, k \in \{1, \ldots, K\}$, the next optimization problem can be solved:

$$\begin{aligned}\max_{\boldsymbol{\theta},\tau}\ & \tau \\ \text{s.t.}\ & \frac{1}{n_{2,k_0}}\sum_{i=1}^{n_{2,k_0}}\tilde{C}_{k_0}(\boldsymbol{\theta},\ \mathbf{x}_i^{(k_0)}) \geq \frac{1}{n_{2,k_0}}\sum_{i=1}^{n_{2,k_0}}\tilde{C}_{k_0}(\boldsymbol{\theta}^*,\ \mathbf{x}_i^{(k_0)}) + \Delta \\ & \frac{1}{n_{2,k}}\sum_{i=1}^{n_{2,k}}\tilde{C}_{k}(\boldsymbol{\theta},\ \mathbf{x}_i^{(k)}) \geq \tau, \quad \forall k \neq k_0.\end{aligned}$$

This way we search the estimates $\boldsymbol{\theta}$ such that in the relevant class k_0 the *Recall* is improved in at least Δ with respect to the *Recall* in the traditional Bayes estimate and maximize the minimum *Recall* in the remaining classes.

The second remark concerns the solutions of (SCNB), which are not maximum likelihood estimates any more, but maximum constrained likelihood estimates instead. On the contrary, the problem yields a solution with the highest sample likelihood fulfilling the constraints on performance on the independent sample. Up to our knowledge, this is a breaking approach that has never been considered in NB models.

5.3 Numerical results

In this section, eight data sets from the UCI Machine Learning Repository and KEEL open source [Alcalá-Fdez et al., 2011, 2009] diverse, in both in the number of classes, sizes and imbalance ratio shall be analyzed. The description of the datasets can be found in Section 5.3.1 and the numerical experiments and obtained results will be considered in Section 5.3.2 and 5.3.3, respectively.

5.3.1 Datasets

The datasets `breast cancer`, `SPECTF`, `page-blocks`, `abalone`, `yeast`, `Satimage`, `RCV1` and `letter` will be considered. From all the available versions of the datasets, we have chosen those described in Table 5.1. The colums report the dataset name, the number of instances and features and finally, the class split of the eight considered datasets (`page-blocks`, `abalone`, `yeast`, `Satimage` and `RCV1` are significantly more unbalanced than the other three).

5.3.2 Design of experiments

Probability distributions setting and resolution of the optimization problem

As comented in Section 5.2.1, a probability model needs to be selected for the features conditioned to the class. If the feature is continuous, in this chapter we will assume the normal

Name	*Intances*	*Features*	*Class split (%)*
`breast cancer`	569	30	(*Benign*, *Malignant*) (63, 37)
`SPECTF`	267	44	(*Abnormal*, *Normal*) (79, 21)
`page-blocks`	5473	10	(*text*, *horiz. line*, *graphic*, *vert. line*, *picture*) (89.8, 6, 0.5, 1.6, 2.1)
`abalone`	4177	8	(*1-5*, *6-10*, *11-15*, *16-29*) (4.52, 28.39, 6.25, 60.83)
`yeast`	1484	8	(*CYT*, *EXC*, *ME1*, *ME3*, *MIT*, *NUC*, *POX*, *VAC*) (32.42, 2.45, 3.08, 11.41, 17.09, 30.04, 1.40, 2.10)
`Satimage`	6435	36	(*1*, *2*, *3*, *4*, *5*, *7*) (23.82, 10.92, 21.10, 9.73, 10.99, 23.43)
`RCV1`	18758	21531	(*C15*, *CCAT*, *E21*, *ECAT*, *GCAT*, *M11*) (23.70, 20.12, 5.73, 9.54, 22.43, 5.61)
`letter`	20000	16	From *A* to *Z* (26 classes) *Equally distributed*

Table 5.1: Datasets description

distribution. From the point of view of the optimization, (SCNB) will be solved using solvers for smooth optimization. In particular, `auglag` and `mma` functions from R package `nloptr` will be used in this work to obtain all numerical results.

Estimation of the performance rates

The performance of the proposed classifier will be estimated using a classic 10 Monte-Carlo cross validation, since in this chapter we deal with highly unbalanced datasets. The dataset will be split into three sets, the so-called training set, validation set and testing set. One-third of the dataset is used as testing set, and the remaining two-thirds for training set $\left(\frac{2}{3} \times \frac{2}{3}\right)$ and validation set $\left(\frac{1}{3} \times \frac{2}{3}\right)$. As explained in Section 5.2, the objective function will be optimized on the training set while the constraints will be evaluated on the validation set. Once the SCNB problem is solved, *Recall* values are estimated on the testing set. It must be highlighted that at each run, the training sample is built in a stratified way so that the proportion of samples per class is similar to the proportions depicted by Table 5.1. Finally, regarding the hyperparameter λ, after an extensive simulation study considering a wide grid of values, the choice $\lambda = 2^3$ is set in the experiments since it provides a good match between C and $\tilde{C}$ as in (5.4) and (5.5).

Pre-processing for large datasets

Problem SCNB turns out computationally costly for large datasets as the considered `RCV1` dataset. As it is common in the literature (see Leevy et al. [2018] and references therein), we suggest to pre-process such datasets in a way that irrelevant variables are removed in a first step previous to the resolution of (SCNB). Specifically, in this work the importance of the predictor variables composing `RCV1` were measured using the R function `information.gain` from

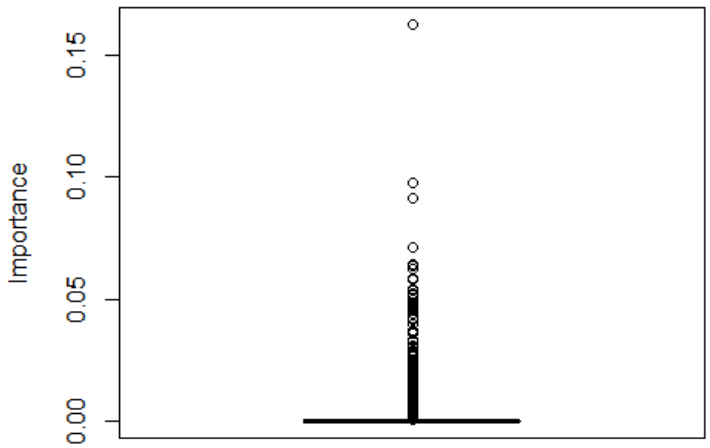


Figure 5.1: Boxplot of the importance of the variables for `RCV1` dataset.

`FSelector`. Figure 5.1 shows the boxplot of the importance associated with the variables of the `RCV1` dataset. As it can be observed, most of them have an associated importance close to 0 and, then, only 392 of the total are going to be kept when solving (SCNB).

The choice of thresholds

In order to select the threshold values α_k in Problem (SCNB), the classic NB classifier (5.2) was first run. Table 5.2 shows the *Recall* estimates for each class. In particular, from Table 5.2 and in order to improve the rate of the classes where the classic NB performs the worst, the set of thresholds to be tested in the numerical experiments shall be given by Table 5.3.

Note that, according to the previous values, not only better rates for the classes with the worst associated *Recall* are imposed, but also, admissible values for the rest of classes are fixed.

5.3.3 Results

The estimated rates are reported in Tables 5.4-5.11. The first row shows the results for the classic NB, when no thresholds are imposed. The first column shows the imposed thresholds for the *Recall* of each class, whereas the column and thresholds in bold correspond to the class where the classic method performs the worst. For example, in Table 5.5, it is required that the *Recall* of *Normal* class is at least 80, while over the *Abnormal* class the threshold varies from 67 to 70. The remaining columns provide the average *Recall* values measured on the test set.

Name	*Recall Classic NB*
`breast cancer`	(*Benign*, *Malignant*) (96.15, 88.41)
`SPECTF`	(*Abnormal*, *Normal*) (65.40, 91.29)
`page-blocks`	(*text*, *horiz. line*, *graphic*, *vert. line*, *picture*) (90.61, 68.06, 67.14, 94.14, 40.11)
`abalone`	(*1-5*, *6-10*, *11-15*, *16-29*) (92.58, 61.23, 22.09, 53.97)
`yeast`	(*CYT*, *EXC*, *ME1*, *ME3*, *MIT*, *NUC*, *POX*, *VAC*) (2.37, 62.73, 73.57, 19.81, 54.13, 48.79, 53.33, 31.11)
`Satimage`	(*1*, *2*, *3*, *4*, *5*, *7*) (79.79, 89.59, 89.60, 66.70, 73.52, 74.21)
`RCV1`	(*C15*, *CCAT*, *E21*, *ECAT*, *GCAT*, *M11*) (87.55, 5.53, 74.99, 23.16, 75.12, 88.14)
`letter`	From *A* to *Z* (26 classes) First row in Table 5.11. The letter *S* is the worst classified.

Table 5.2: Average *Recall* of classic NB (10 Monte-Carlo cross validation)

`breast cancer`: 85 (*Benign*), 90/91.5/93/94.5 (*Malignant*)
`SPECTF`: 67/68.5/70 (*Abnormal*), 80 (*Normal*)
`page-blocks`: 80 (*text*), 58 (*horiz. line*), 57 (*graphic*), 84 (*vert. line*), 43/44.5 (*picture*)
`abalone`: 70 (*1-5*), 50 (*6-10*), 24/26/28/30 (*11-15*), 40 (*16-29*)
`yeast`: 5/10/15 (*CYT*), 50 (*EXC*), 60 (*ME1*), 10 (*ME3*), 40 (*MIT*), 30 (*NUC*), 40 (*POX*), 20 (*VAC*)
`Satimage`: 70 (*1*), 80 (*2*), 80 (*3*), 68/70 (*4*), 60 (*5*), 60 (*7*)
`RCV1`: 60 (*C15*), 6/7/8 (*CCAT*), 60 (*E21*), 10 (*ECAT*), 60 (*GCAT*), 60 (*M11*)
`letter`: 24.5/26/27.5 (*S*), Twenty-five percent less than the results of the classic NB (*The rest of classes*)

Table 5.3: Tested thresholds

As expected, the results under the constrained NB version differ from the results provided by the classic NB. For example, for the `breast cancer` dataset, the *Recall* values under the classic NB are 96.15 and 88.41, for *Benign* and *Malignant* class, respectively (Table 5.2). As commented in Section 5.1, it may be of interest to increase the *Recall* of the *Malignant* class. According to Table 5.4, if the minimum 94.50 is imposed for the *Malignant* class, the final rates change from 88.41 to 91.88. It is important to highlight two different facts concerning the previous results. First, note that a better rate for the *Malignant* class has been obtained, but at the expense of slightly decreasing the rate of the *Benign* class. Second, note that even though a rate equal to 94.50 was imposed, such value was not finally obtained, but a smaller one (91.88). This is not surprising, since the constraints are imposed for one sample, and tested on an independent set.

From the results shown in Table 5.4-5.11, it can be concluded that the proposed approach allows the user to control the *Recall* values in such a way that the classes where the classic method performs the worst can be improved, even switching to the classes where the best rates are achieved (e.g., in `breast cancer` dataset) and changing thus the natural tendency of the classifier. Note that among the possible non-dominated solutions shown for each dataset, the

user could choose according to her interest and to what she is willing to lose in the less critical classes.

Thresholds (Benign/Malignant)	*Recall Benign*	*Recall Malignant*
Classic NB	**96.15**	**88.41**
85.00/**90.00**	94.62	**88.70**
85.00/**91.50**	94.10	**90.00**
85.00/**93.00**	92.39	**91.59**
85.00/**94.50**	91.45	**91.88**

Table 5.4: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `breast cancer`

Thresholds (Abnormal/Normal)	*Recall Abnormal*	*Recall Normal*
Classic NB	**65.40**	**91.29**
67.00/80.00	**67.09**	88.72
68.50/80.00	**67.37**	88.08
70.00/80.00	**69.63**	87.08

Table 5.5: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `SPECTF`

Thresholds (text/horiz. line/ graphic/vert. line/picture)	*Recall text*	*Recall horiz. line*	*Recall graphic*	*Recall vert. line*	*Recall picture*
Classic NB	**90.61**	**68.06**	**67.14**	**94.14**	**40.11**
80.00/58.00/57.00/84.00/**43.00**	89.34	68.97	70.23	93.23	**43.58**
80.00/58.00/57.00/84.00/**44.50**	90.15	69.21	73.56	94.14	**44.28**

Table 5.6: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `page-blocks`

Thresholds (1-5/6-10/11-15/16-29)	*Recall 1-5*	*Recall 6-10*	*Recall 11-15*	*16-29*
Classic NB	**92.58**	**61.23**	**22.09**	**53.97**
70.00/50.00/**24.00**/40.00	92.58	60.56	**24.65**	53.87
70.00/50.00/**26.00**/40.00	92.58	60.05	**25.47**	53.85
70.00/50.00/**28.00**/40.00	92.58	59.36	**26.28**	53.81
70.00/50.00/**30.00**/40.00	92.58	58.31	**28.26**	53.75

Table 5.7: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `abalone`

Thresholds (CYT/EXC/ME1/ME3/ MIT/NUC/POX/VAC)	*CYT*	*EXC*	*ME1*	*ME3*	*MIT*	*NUC*	*POX*	*VAC*
Classic NB	**2.37**	**62.73**	**73.57**	**19.81**	**54.13**	**48.79**	**53.33**	**31.11**
5.00/50.00/60.00/10.00/40.00/30.00/40.00/20.00	**3.88**	61.82	74.29	24.34	56.75	47.45	53.33	35.56
10.00/50.00/60.00/10.00/40.00/30.00/40.00/20.00	**6.18**	58.18	75.00	24.91	56.50	47.52	55.00	35.56
15.00/50.00/60.00/10.00/40.00/30.00/40.00/20.00	**14.87**	58.18	75.71	28.68	58.63	39.36	53.33	33.33

Table 5.8: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `yeast`

Thresholds (1/2/3/4/5/7)	*Recall 1*	*Recall 2*	*Recall 3*	*Recall 4*	*Recall 5*	*Recall 7*
Classic NB	**79.79**	**89.59**	**89.60**	**66.70**	**73.52**	**74.21**
70.00/80.00/80.00/**68.00**/60.00/60.00	79.47	89.59	90.67	**68.35**	72.12	73.64
70.00/80.00/80.00/**70.00**/60.00/60.00	79.63	89.68	91.25	**68.55**	71.56	70.22

Table 5.9: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `Satimage`

Thresholds (C15/CCAT/ E21/ECAT/GCAT/M11)	*Recall C15*	*Recall CCAT*	*Recall E21*	*Recall ECAT*	*Recall GCAT*	*Recall M11*
Classic NB	**87.55**	**5.53**	**74.99**	**23.16**	**75.12**	**88.14**
60.00/**6.00**/60.00/10.00/60.00/60.00	87.46	**7.85**	76.27	23.83	76.57	88.59
60.00/**7.00**/60.00/10.00/60.00/60.00	87.56	**7.94**	76.27	23.89	76.59	88.64
60.00/**8.00**/60.00/10.00/60.00/60.00	87.20	**9.14**	75.68	26.02	77.50	88.69

Table 5.10: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `RCV1` using 392 variables of the total

Thresholds	*Recall A*	*Recall B*	*Recall C*	*Recall D*	*Recall E*	*Recall F*	*Recall G*	*Recall H*
Classic NB	**88.46**	**65.71**	**76.67**	**64.85**	**36.98**	**67.81**	**48.90**	**30.50**
24.50	88.46	65.71	76.67	65.00	36.83	67.81	48.91	30.50
26.00	88.46	65.24	76.67	65.45	36.98	67.81	48.91	30.50
27.50	88.46	64.13	76.50	64.85	37.14	67.50	48.59	30.83
	Recall I	*Recall J*	*Recall K*	*Recall L*	*Recall M*	*Recall N*	*Recall O*	*Recall P*
Classic NB	**77.90**	**66.07**	**43.11**	**74.60**	**84.31**	**71.56**	**72.58**	**72.72**
24.50	77.90	66.23	43.11	74.60	84.31	71.72	72.58	72.73
26.00	77.90	65.74	43.11	74.60	84.31	71.72	72.74	72.88
27.50	77.58	65.57	42.46	74.60	84.31	72.34	73.06	72.88
	Recall Q	*Recall R*	*Recall S*	*Recall T*	*Recall U*	*Recall V*	*Recall W*	*Recall X*
Classic NB	**51.56**	**61.61**	**22.95**	**72.77**	**72.39**	**72.86**	**79.03**	**44.15**
24.50	51.56	61.45	**23.11**	72.46	72.24	73.02	78.87	44.00
26.00	51.41	61.45	**23.77**	72.46	72.39	73.17	78.87	44.00
27.50	51.25	61.29	**23.77**	72.46	71.79	71.90	79.03	43.69
	Recall Y	*Recall Z*						
Classic NB	**34.15**	**58.17**						
24.50	34.00	57.83						
26.00	33.85	58.17						
27.50	34.00	60.17						

Table 5.11: Average *Recall* values of SCNB (10 Monte-Carlo cross validation) for `letter`

Finally, to illustrate the computational cost of the optimization algorithm depending on the number of instances and features, we simulated data following Witten et al. [2014] with $\{500, 1000, 3000, 5000, 10000, 15000, 20000\}$ instances and $p \in \{10, 50, 100, 300, 500, 700, 900, 1000\}$. Figures 5.2 and 5.3 report the logarithm of the user times (in seconds) when the SCNB is run on Intel(R) Core(TM) i7-7500U CPU at 2.70 GHz 2.90 GHz with 8.0 GB of RAM, and the number of evaluations for the algorithm `auglag` is 100. The X-axis of Figure 5.2 shows the number of instances whereas each line represents the number of variables of the dataset (p). Figure 5.3 is the opposite. Overall, running time grows linearly respect to the number of instances, but not so smooth when p increases.

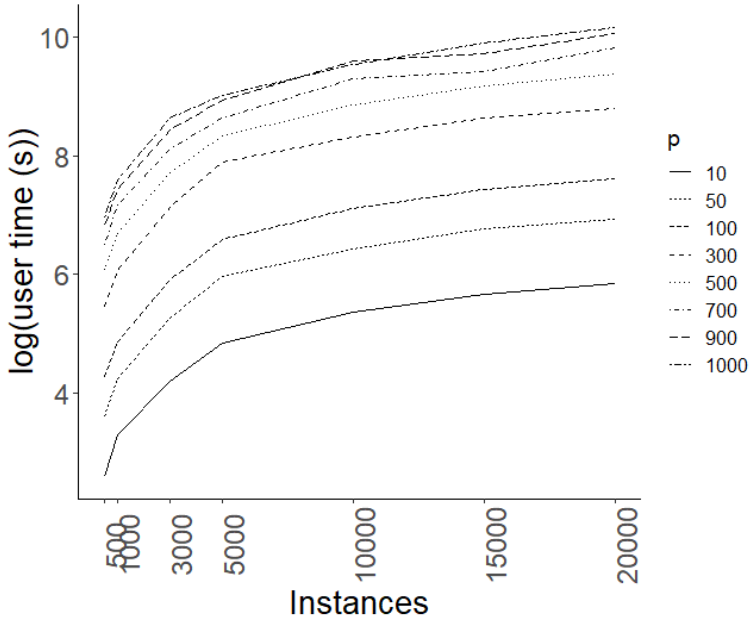


Figure 5.2: Scalability: X-axis represents the number of instances whereas each line the number of features.

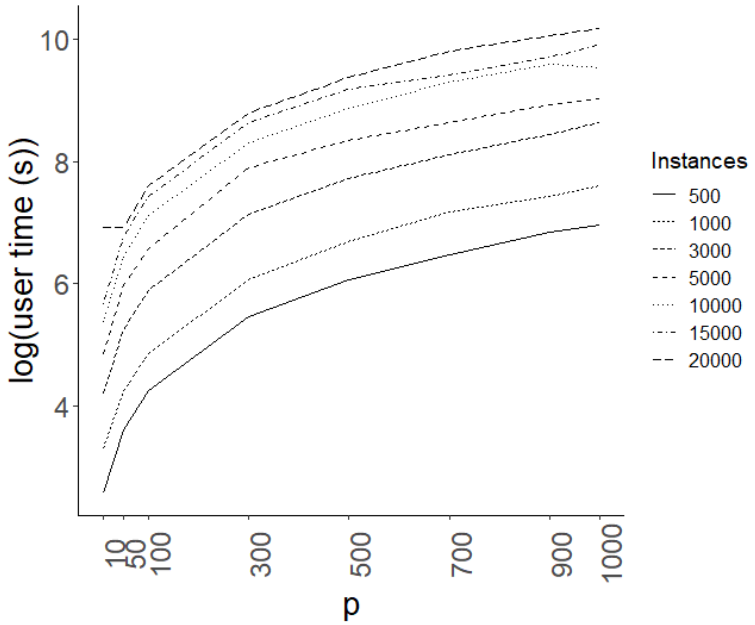


Figure 5.3: Scalability: X-axis represents the number of features whereas each line the number of instances.

5.4 Chapter summary

In this work a new version of the NB classifier is proposed with the aim of controlling misclassification rates in the different classes, avoiding the use of precise values of misclassification costs, which may be hard to choose. In order to achieve this goal, performance constraints are included into the optimization problem which estimates the involved parameters. The approach results in a novel method (SCNB) not reported in the literature previously, up to our knowledge. Unlike the classic NB, the (SCNB) integrates the performance rates in the parameters' estimation step. In fact, this novel approach allows the user to impose thresholds to assure the achievement in the measures of efficiency (in this case, the *Recall* values). The proposed methodology has been tested on eight real datasets with different sampling properties. The numerical results show that not only the classification rates of interest can be controlled, but also for some cases the worst classified class turns out the best classified class under the novel approach. This fact is of great interest in some medical, credit scoring or social contexts where some classes are more critical than others.

Chapter 6

A Bayesian semi-parametric approach to normal/independent and elliptical distributions

Although the normal distribution is a standard model for many phenomena in real life, it does not permit the modeling of heavy or light tails. In multivariate statistical analysis, elliptical (or elliptical-contoured) distributions, introduced in Schoenberg [1938]; Lord [1954], have provided a wide alternative to the standard normal models in a number of contexts such as Economics, Finance or Sociology (see for example Owen and Rabinovitch [1983]; Lindskog et al. [2003]; Abdous et al. [2005]; Gupta et al. [2013]; Frahm et al. [2003]; Jara et al. [2008]). In this chapter, we present an approach to semi-parametric inference for elliptical distributions using Dirichlet processes mixture models. The approach is illustrated on simulated and real-life datasets.

6.1 Introduction

Elliptical distributions, which constitute a generalization of the multivariate normal/independent (NI) family [Rogers and Tukey, 1972; Andrews and Mallows, 1974; Lange and Sinsheimer, 1993], have been studied from long time ago. Many of their properties can be found in Kelker [1970]; Cambanis et al. [1981]; Anderson and Fang [1982]; Dickey and Chen [1985]; Berkane and Bentler [1987, 1990]; Fang et al. [1990]; Anderson [2003]. However, the first works devoted to elliptical distributions are even older, see Schoenberg [1938]; Lord [1954]. Generally speaking, elliptical continuous distributions are those whose density functions are constant over ellipsoids as it is the case of normal distributions. Indeed, an elliptical distribution is an extension of the multivariate normal distribution in such a way that certain properties of the normal model are maintained (as the symmetry) but, at the same time, distributions with shorter or longer tails than those of the normal are also possible. Examples of elliptical (and normal/independent) distributions are the normal, Student-t, slash, contaminated-normal, Pearson type VII, Laplace or spherical distributions, see Lange and Sinsheimer [1993]; Gómez et al. [2007]; Gómez and Venegas [2008]; Berkane and Bentler [1987]; Anderson [1992]; Díaz-García [2005]; Kelker [1970].

The definition of the elliptical distribution is as follows. If $\mathbf{Y}$ a $k-$dimensional random vector, $\boldsymbol{\mu} \in \mathbb{R}^k$ and Σ represents a covariance matrix, then $\mathbf{Y}$ is distributed as an elliptical (contoured) distribution if, for a given sample $\mathbf{y}$, the joint density is

$$f_{\mathbf{Y}}(\mathbf{y} \mid \boldsymbol{\mu}, \Sigma, g) = c \mid \Sigma \mid^{-1/2} g\left((\mathbf{y}-\boldsymbol{\mu})'\Sigma^{-1}(\mathbf{y}-\boldsymbol{\mu})\right), \tag{6.1}$$

where $g(\cdot)$ (the *density generator* or *kernel*) is a non-negative function and c is a normalizing constant, see Fortunati et al. [2020]. From the previous definition the model can be understood as a semi-parametric model where g is the infinite-dimensional parameter.

Since the class of elliptical distributions have been known long ago, different estimation strategies have already been considered in the literature. The first works focused on parametric estimation approaches, see Fang and Anderson [1990]; Anderson [1992]. In Anderson et al.

[1986], the maximum-likelihood estimators of the model parameters are obtained and in Kano et al. [1993]; Berkane et al. [1994], a pseudo-maximum likelihood approach is considered. On the other hand, the Bayesian perspective has also been explored for the estimation of elliptical distributions, see for example Osiewalski and Steel [1993]; Fang and Li [1999]; Branco et al. [2000]; Niekerk et al. [2015]. Finally, Maruyama and Seo [2003] undertakes a moments-matching approach.

The semi-parametric estimation approach has also been explored. For example, Liebscher [2005] used kernel density estimation to provide a classical, semi-parametric estimator. Also, Fortunati et al. [2020] undertakes the estimation of the elliptical distribution in semi-parametric fashion which results in a robust, efficient estimator. In this chapter we adopt the Bayesian paradigm and propose a semi-parametric approach based on Dirichlet processes [Ferguson, 1973]. The stochastic representation of the elliptical distribution discussed in Section 6.2 will play a key role for the inference approach.

The remainder of the chapter is structured as follows. In Section 6.2, we formally define elliptical as well as NI distributions, and their main properties. Then, in Section 6.3 we illustrate how semi-parametric, Bayesian inference can be carried out for both NI and elliptical distributions. Our approach is illustrated in detail with simulated and a real dataset in Section 6.4. The chapter ends with some concluding remarks in Section 6.5.

6.2 Preliminaries on elliptical and NI distributions

According to Cambanis et al. [1981] the elliptical contoured distributions is defined as follows. As previously, let $\mathbf{Y}$ denote a $k-$dimensional random vector, $\boldsymbol{\mu} \in \mathbb{R}^k$ and let Σ be some $k \times k$ covariance matrix. If the characteristic function of $\mathbf{Y} - \boldsymbol{\mu}$, $\phi_{\mathbf{Y}-\boldsymbol{\mu}}(\mathbf{t})$ is a function of the quadratic form $\mathbf{t}'\Sigma\mathbf{t}$, that is, if

$$\phi_{\mathbf{Y}-\boldsymbol{\mu}}(\mathbf{t}) = \phi\left(\mathbf{t}'\Sigma\mathbf{t}\right),$$

then, $\mathbf{Y}$ follows an elliptical (contoured) distribution with location parameter $\boldsymbol{\mu}$, scale parameter Σ and scalar function ϕ. Note that when $\phi(u) = e^{-u/2}$, then the normal multivariate distribution with parameters $(\boldsymbol{\mu}, \Sigma)$ is recovered.

The elliptical distribution admits a stochastic representation as proven in Schoenberg [1938]; Cambanis et al. [1981]. This is an advantageous result which allows to simulate in straightforward way from the elliptical distribution and provides a natural scheme for designing the estimation method (as will be detailed in Section 6.3). Prior to the stochastic representation, let us introduce some concepts in relation to the uniform distribution on the unit sphere in $\mathbb{R}^k$. If Φ_k, $k \geq 1$ is the class of all functions $\phi : [0, \infty) \to \mathbb{R}$ such that $\phi(||\mathbf{t}||^2)$, $\mathbf{t} \in \mathbb{R}^k$, is a characteristic function, then $\phi \in \Phi_k$ if and only if

$$\phi(s) = \int_0^\infty \Omega_k(r^2 s) dF(r), \quad s \geq 0$$

for some distribution function F on $[0, \infty)$, where $\Omega_k(||\mathbf{t}||^2)$, $\mathbf{t} \in \mathbb{R}^k$, is the characteristic function of a $k-$dimensional random vector $\mathbf{S}_k$ which is uniformly distributed on the unit sphere in $\mathbb{R}^k$.

Remark 1. *If $\mathbf{S}_k$ is uniformly distributed on the unit sphere in $\mathbb{R}^k$, then $\mathbf{S}_k = \mathbf{Z}/|\mathbf{Z}|$, where $\mathbf{Z} \sim N(\mathbf{0}, \mathbf{I}_k)$ and $|\mathbf{Z}| = \sqrt{Z_1^2 + \ldots + Z_k^2}$.*

Theorem 3 (Schoenberg [1938]; Cambanis et al. [1981])**.** *A $k \times 1$ random vector $\mathbf{Y}$ follows an elliptical distribution with parameters $\boldsymbol{\mu}$, Σ and ϕ if and only if:*

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}\mathbf{S}_k U, \tag{6.2}$$

where $\mathbf{S}_k$ is uniformly distributed on the (k-dimensional) hypersphere, U is a non-negative, univariate random variable, independent of $\mathbf{S}_k$ and $\mathcal{A}$ is the unique, symmetric matrix, with non-negative diagonal elements such that $\mathcal{A}\mathcal{A} = \Sigma$.

The derivation of the density function as in (6.1) from the stochastic representation (6.2) can be found in Cambanis et al. [1981] (section 4).

For the purposes of this work, it is useful to introduce some reparametrizations of the elliptical distribution. Suppose that we have an elliptical variable defined as in Theorem 3. Then, we shall write:

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}\mathbf{S}_k\sqrt{G} \quad \text{where } G = U^2. \tag{6.3}$$

Given the stochastic formulation in (6.3), it is clear that an elliptical variable is alternatively defined by the location vector, $\boldsymbol{\mu}$, the scale matrix, Σ, and the distribution of the variable G, say $F_G(\cdot)$. Therefore, from now on we shall refer to an elliptical variable satisfying (6.3) as $\mathbf{Y} \sim EC(\boldsymbol{\mu}, \Sigma, F_G)$. Note that we have

$$G = (\mathbf{Y} - \boldsymbol{\mu})'\Sigma^{-1}(\mathbf{Y} - \boldsymbol{\mu}). \tag{6.4}$$

From Remark 1, we have that:

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}\mathbf{Z}/\sqrt{W}, \tag{6.5}$$

where $W = \frac{|\mathbf{Z}|^2}{G}$.

For the general elliptical distribution as in (6.5), W and $\mathbf{Z}$ are not necessarily independent. However, if this assumption is made, following Andrews and Mallows [1974], then $\mathbf{Y}$ is said to have a normal/independent (NI) distribution. A NI distribution is therefore defined by $\boldsymbol{\mu}$, Σ and the distribution of W, say $F_W(\cdot)$, so we shall write $\mathbf{Y} \sim NI(\boldsymbol{\mu}, \Sigma, F_W)$ to represent a NI variable satisfying (6.5).

In order to avoid identifiability problems is important to take into account that elliptical distributions are identified, up to a scale factor. If we consider an elliptical variable $\mathbf{Y}$, defined as in (6.5), we can write $\mathcal{A}^* = b\mathcal{A}$ and $W^* = b^2W$ for any non-negative constant, b, and then

we have an equivalent stochastic representation as:

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}^* \mathbf{Z}/\sqrt{W^*}.$$

Therefore, in order to ensure identifiability, throughout this work, we shall assume, for both the elliptical and NI models, that the diagonal elements of Σ sum up to one, that is:

$$\sum_{i=1}^{k} \Sigma_{ii} = 1, \tag{6.6}$$

where Σ_{ij} is the $(i,j)-$th element of matrix Σ, for $i, j = 1, \ldots, k$.

Concerning the first two moments of elliptical distributions, it is known that if $E(U) < \infty$, then $E(\mathbf{Y}) = \boldsymbol{\mu}$. Also, if $E(U^2) < \infty$, then Σ is proportional to the covariance matrix,

$$E\left[(\mathbf{Y} - \boldsymbol{\mu})(\mathbf{Y} - \boldsymbol{\mu})'\right] = k^{-1} E(U^2)\Sigma.$$

Finally, it is interesting to mention that if two random vectors are jointly elliptically distributed, then the conditional distribution of one given the other is elliptical.

6.3 Bayesian inference for NI and elliptical distributions

In order to implement Bayesian inference for either the NI or the general, elliptical models, we first need to define prior distributions for the location vector, $\boldsymbol{\mu}$ and the scale matrix Σ. Here, we shall assume a normal prior distribution:

$$\boldsymbol{\mu} \sim N(\mathbf{m}, \mathcal{V})$$

where $\mathbf{m} = (m_1, ..., m_k)'$ is a constant vector and $\mathcal{V}$ is a $k \times k$ variance-covariance matrix. In practice, we set $\mathbf{m}$ to be a zero vector and $\mathcal{V} = 1000\mathcal{I}_k$, where $\mathcal{I}_k$ is the $k \times k$ identity matrix, in order to provide a proper but relatively uninformative prior.

In order to define a prior distribution for Σ, we need to recall the identifiability restriction in (6.6). To do this, we shall first define:

$$\Sigma = \text{diag}(\boldsymbol{\sigma})\mathcal{R}\text{diag}(\boldsymbol{\sigma}),$$

where $\text{diag}(\boldsymbol{\sigma})$ is a diagonal matrix with diagonal elements $\sigma_i = \sqrt{\Sigma_{ii}}$ for $i = 1, ..., k$ and $\mathcal{R}$ is a matrix with elements

$$\mathcal{R}_{ij} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}} = \frac{\Sigma_{ij}}{\sigma_i \sigma_j}.$$

Now, we can define an implicit prior for Σ by setting prior distributions for $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_k^2)' =$

$(\Sigma_{11}, ..., \Sigma_{kk})'$ and $\mathcal{R}$. Firstly, we assume that $\boldsymbol{\sigma}^2$ has a Dirichlet distribution

$$\boldsymbol{\sigma}^2 \sim Dir(\mathbf{a}_\sigma),$$

where $\mathbf{a}_\sigma = (a_{\sigma 1}, ..., a_{\sigma k})'$. In practice, we set $a_{\sigma i} = 1$ for $i = 1, ..., k$.

For $\mathcal{R}$, we follow an approach of Zhang et al. [2006] and use a parameter extended inverse Wishart distribution. Thus, we set

$$\mathcal{M} = \mathcal{D}^{\frac{1}{2}} \mathcal{R} \mathcal{D}^{\frac{1}{2}}$$

where $\mathcal{D}$ is a diagonal matrix. Then we assume that $\mathcal{M}$ has an inverse Wishart prior distribution, that is

$$\mathcal{M} \sim IW(k+1, \mathcal{I}_k).$$

Under this structure, it can be shown that the implicit prior for $\mathcal{R}$ is the marginal uniform distribution of Barnard et al. [2000].

Finally, it is necessary to define prior distributions for the distributions, F_W, in the case of the NI model or F_G, in the elliptical case. Instead of using a parametric model, here we shall use Dirichlet process mixtures of gamma distributions as in Hanson [2006]. We outline the prior set up and the posterior inference procedures for the NI and elliptical models respectively in the following subsections.

6.3.1 Inference for the NI distribution

Assume that we observe a sample $\mathbf{y}_1, \ldots, \mathbf{y}_n$ from a NI distribution with stochastic representation as in (6.5),

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}\mathbf{Z}/\sqrt{W},$$

where $\mathbf{Z}$ and W are independent. As W is non-negative, a natural approach, which we shall take here, is to follow Hanson [2006] and model W using a Dirichlet process mixture of gamma distributions. Therefore, we assume that

$$\begin{aligned} W|\lambda, \eta &\sim Ga(\lambda, \eta) \\ \lambda, \eta|F &\sim F \\ F &\sim DP(\alpha F_0), \end{aligned} \tag{6.7}$$

where $DP(\cdot)$ represents a Dirichlet process with paramaters $\alpha > 0$ and $F_0(\cdot, \cdot)$ being a prior mean distribution for $F(\cdot, \cdot)$.

Also following Hanson [2006], we shall assume that F_0 is such that the associated density function is a product of exponential distributions

$$f_0(\lambda, \eta|a_\lambda, a_\eta) = a_\lambda e^{-a_\lambda \lambda} a_\eta e^{-a_\eta \eta} \quad \text{for } \lambda, \eta > 0, \text{ where } a_\lambda, a_\eta > 0,$$

and we shall place hyperprior distributions

$$\begin{aligned} \alpha &\sim Ga(a_\alpha, b_\alpha) \\ a_\lambda &\sim Ga(b_\lambda, c_\lambda) \\ a_\eta &\sim Ga(b_\eta, c_\eta), \end{aligned}$$

where $a_\alpha, b_\alpha, b_\lambda, c_\lambda, b_\eta, c_\eta > 0$. In the practical examples later, we set $a_\alpha = b_\alpha = 2$ and $b_\lambda = c_\lambda = b_\eta = c_\eta = 0.001$. Under this model, we have that integrating out over W, the implied distribution of $\mathbf{Y}$ is an infinite mixture of Student's t distributions, all with location parameter $\boldsymbol{\mu}$ and scale parameter proportional to Σ.

Given this prior formulation and supposing that we observe a sample, $\mathbf{y}_1, ..., \mathbf{y}_n$, then inference can be carried out using a Gibbs sampling scheme based on successively sampling the conditional posterior distributions. The relevant formulae are given below.
Firstly, the conditional posterior distributions of $\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathcal{D}, \mathcal{R}$ are as follows:

$$\begin{aligned} \boldsymbol{\mu}|\Sigma, \mathbf{w}, \mathbf{y} &\sim N(\mathbf{m}^*, \mathcal{V}^*) \quad \text{where} \\ \mathbf{m}^* &= \mathcal{V}^* \left(\mathcal{V}^{-1}\mathbf{m} + n\Sigma^{-1}\overline{w\mathbf{y}}\right) \quad \text{where } \overline{w\mathbf{y}} = \frac{1}{n}\sum_{i=1}^{n} w_i \mathbf{y}_i \\ \mathcal{V}^* &= \left(\mathcal{V}^{-1} + n\bar{w}\Sigma^{-1}\right)^{-1} \quad \text{where } \bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i \\ f(\boldsymbol{\sigma}^2|\boldsymbol{\mu}, \mathcal{R}, \mathbf{w}, \mathbf{y}) &\propto \prod_{i=1}^{n} N\left(\mathbf{y}_i|\boldsymbol{\mu}, \frac{1}{w_i}\Sigma\right) f(\boldsymbol{\sigma}^2) \\ f(\mathcal{R}, \mathcal{D}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{w}, \mathbf{y}) &\propto \prod_{i=1}^{n} N\left(\mathbf{y}_i|\boldsymbol{\mu}, \frac{1}{w_i}\Sigma\right) f(\mathcal{R}, \mathcal{D}) \end{aligned}$$

where, throughout, $\Sigma = \text{diag}(\boldsymbol{\sigma})\mathcal{R}\text{diag}(\boldsymbol{\sigma})$, $\mathbf{w} = (w_1, ..., w_n)'$, $N(\mathbf{y}_i|\boldsymbol{\mu}, \frac{1}{w_i}\Sigma)$ refers to a multivariate normal density function with parameters $\boldsymbol{\mu}$ and $\frac{1}{w_i}\Sigma$ evaluated at $\mathbf{y}_i$, whereas $f(\sigma^2)$ and $f(\mathcal{R}, \mathcal{D})$ are the prior distributions. The conditional posterior of $\boldsymbol{\sigma}^2$ can be sampled via a Metropolis-Hastings pass, using for example an adaptive logit sampler as in Director et al. [2017]. In order to sample the distribution of $\mathcal{D}, \mathcal{R}$, we follow Fang and Li [1999] and use a Wishart candidate for the composed matrix $\mathcal{M} = \mathcal{D}^{\frac{1}{2}}\mathcal{R}\mathcal{D}^{\frac{1}{2}}$.

The conditional posterior distribution of W_i is directly evaluable as

$$W_i|\boldsymbol{\mu}, \Sigma, \mathbf{y}_i \sim Ga\left(\frac{k}{2}, (\mathbf{y}_i - \boldsymbol{\mu})'\Sigma^{-1}(\mathbf{y}_i - \boldsymbol{\mu})\right).$$

Now, the conditional posterior distributions of the Dirichlet process parameters follow from Hanson [2006]. Firstly, we can generate values from the conditional posterior of λ_i, η_i given $\boldsymbol{\lambda}_{-i}, \boldsymbol{\eta}_{-i}, a_\lambda, a_\eta, w_i$ where $\boldsymbol{\lambda}_{-i} = (\lambda_1, ..., \lambda_{i-1}, \lambda_{i+1}, ..., \lambda_n)'$ and similarly for $\boldsymbol{\eta}$. We have

then that:

$$(\lambda_i, \eta_i) = (\lambda_j, \eta_j) \text{ with probability } p_j \text{ for } j \neq i \text{ in } 1, ..., n,$$

where $p_j \propto Ga(w_i|\lambda_j, \eta_j)$ and $Ga(\cdot|\cdot, \cdot)$ is a gamma density function. With probability p_i, where

$$p_i \propto \alpha \frac{a_\lambda a_\eta}{w_i(w_i + a_\eta)(a_\lambda - \log w_i + \log(w_i + a_\eta))^2},$$

it follows

$$\begin{aligned} \lambda_i &\sim Ga(2, a_\lambda - \log w_i + \log(w_i + a_\eta)) \\ \eta_i|\lambda_i &\sim Ga(\lambda_i + 1, w_i + a_\eta). \end{aligned}$$

Secondly, the hyperparameters are:

$$\begin{aligned} a_\lambda|\boldsymbol{\lambda} &\sim Ga\left(b_\lambda + m^*, c_\lambda + \sum_{j=1}^{m^*} \lambda_j^*\right) \\ a_\eta|\boldsymbol{\eta} &\sim Ga\left(b_\eta + m^*, c_\eta + \sum_{j=1}^{m^*} \eta_j^*\right), \end{aligned}$$

where there are m^* unique values, say $\{(\lambda_1^*, \eta_1^*), ..., (\lambda_{m^*}^*, \eta_{m^*}^*)\}$.

Finally, values from the distribution of α can be generated following Escobar and West [1995] by introducing a further latent variable, θ, such that:

$$\begin{aligned} \theta|\alpha, n &\sim Beta(\alpha + 1, n) \\ \alpha|\theta, m^* &\sim \begin{cases} Ga(a_\alpha + m^*, b_\alpha - \log\theta) & \text{with probability } \frac{a_\alpha + m^* - 1}{a_\alpha + m^* - 1 + n(b_\alpha - \log\theta)} \\ Ga(a_\alpha + m^* - 1, b_\alpha - \log\theta) & \text{otherwise.} \end{cases} \end{aligned}$$

6.3.2 Inference for the elliptical distribution

Now assume that we observe a sample $\mathbf{y}_1, \ldots, \mathbf{y}_n$ from (6.3). Then, we shall assume a Dirichlet process mixture prior:

$$\begin{aligned} G|\lambda, \eta &\sim Ga(\lambda, \eta) \\ \lambda, \eta|F &\sim F \\ F &\sim DP(\alpha F_0), \end{aligned}$$

where we shall use the same prior modeling for α, F_0 as previously in (6.7).

In order to undertake inference, we first use (6.5), so that

$$\mathbf{Y} \stackrel{d}{=} \boldsymbol{\mu} + \mathcal{A}\mathbf{Z}/\sqrt{W}$$

where $W = |\mathbf{Z}|^2/G$. In addition, note that $X \stackrel{d}{=} |\mathbf{Z}|^2 \sim \chi^2_k$. Also, we have that:

$$\begin{aligned} f(w_i|\lambda_i,\eta_i) &= \int_0^\infty g_i f_{X_i}(w_i g_i) f_{G_i}(g_i|\lambda_i,\eta_i)\, dg_i \\ &= \frac{1}{Beta\left(\frac{k}{2},\lambda_i\right)} \frac{\frac{w_i}{2}^{\frac{k}{2}-1}\eta_i^{\lambda_i}}{2^{\frac{k}{2}}\left(\frac{w_i}{2}+\eta_i\right)^{\frac{k}{2}+\lambda_i}}. \end{aligned} \tag{6.8}$$

This implies that if we define $B_i = W_i/(W_i + 2\eta_i)$, then:

$$B_i|\lambda_i,\eta_i \sim Beta\left(\frac{k}{2},\lambda_i\right).$$

Now, we have that the conditional distribution of G_i/x_i is $G_i|\lambda_i,\eta_i,x_i \sim Ga(\lambda_i,\eta_i x_i)$ so that, conditional on $X_i = x_i$, W_i has an inverse gamma distribution, that is

$$W_i|\lambda_i,\eta_i,x_i \sim Inv\ Gamma(\lambda,\eta_i x_i). \tag{6.9}$$

Recalling that $\mathbf{Z}_i \sim N(\mathbf{0},\mathcal{I}_k)$, we can calculate the posterior conditional distribution of $\mathbf{Z}_i$ via Bayes theorem by combining this with (6.9, 6.8) to give:

$$f(\mathbf{z}_i|w_i,\lambda_i,\eta_i) = \frac{(w_i+2\eta_i)^2\, N(\mathbf{z}_i \mid \mathbf{0},\mathcal{I}_k) Inv\ Gamma\left(w_i \mid \lambda_i,\eta_i x_i\right)}{2\eta_i Beta\left(b_i \mid \frac{k}{2},\lambda_i\right)}.$$

where $b_i = \frac{w_i}{w_i+2\eta_i}$, $x_i = |\mathbf{z}_i|^2$ and $N(\cdot|\cdot,\cdot)$, $Inv\ Gamma(\cdot|\cdot,\cdot)$ and $Beta(\cdot|\cdot,\cdot)$ represent the normal, inverse gamma and beta density functions respectively. Finally, we have that the density of $\mathbf{Y}_i$ is:

$$f(\mathbf{y}_i|\boldsymbol{\mu},\Sigma,\lambda_i,\eta_i,w_i) = w_i^{\frac{k}{2}}|\Sigma|^{-\frac{1}{2}} f(\mathbf{z}_i = \sqrt{w_i}\mathcal{A}^{-1}(\mathbf{y}_i - \boldsymbol{\mu}))$$

As previously, given sample data $\mathbf{y}_1,\mathbf{y}_2,...,\mathbf{y}_n$, then we can derive the relevant conditional posterior distributions needed to set up a Gibbs sampler as follows. Firstly, the conditional posterior distribution of $\boldsymbol{\mu}$ is

$$f(\boldsymbol{\mu}|\mathbf{w},\Sigma,\boldsymbol{\lambda},\boldsymbol{\eta},\mathbf{y}) \propto \prod_{i=1}^n f(\mathbf{y}_i|\boldsymbol{\mu},\Sigma,\lambda_i,\eta_i,w_i) N(\boldsymbol{\mu}|\mathbf{m},\mathcal{V}).$$

Values of $\boldsymbol{\mu}$ can be sampled using, for example a random walk sampler centred on the current value of $\boldsymbol{\mu}$.

In a similar way, we have:

$$\begin{aligned} f(\boldsymbol{\sigma}^2|\boldsymbol{\mu},\mathcal{R},\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\eta},\mathbf{y}) &\propto \prod_{i=1}^{n} f(\mathbf{y}_i|\boldsymbol{\mu},\Sigma,\lambda_i,\eta_i,w_i) f(\boldsymbol{\sigma}^2) \\ f(\mathcal{R},\mathcal{D}|\boldsymbol{\mu},\boldsymbol{\sigma}^2,\mathbf{w},\boldsymbol{\lambda},\boldsymbol{\eta},\mathbf{y}) &\propto \prod_{i=1}^{n} f(\mathbf{y}_i|\boldsymbol{\mu},\Sigma,\lambda_i,\eta_i,w_i) f(\mathcal{R},\mathcal{D}) \end{aligned}$$

where in these formulae, we recall that $\Sigma = \text{diag}(\boldsymbol{\sigma})\mathcal{R}\text{diag}(\boldsymbol{\sigma})$. In a similar way to the previous section, we can sample values of $\boldsymbol{\sigma}^2$ using a Metropolis Hastings pass via an adaptive logit sampler and we can sample from the distribution of $\mathcal{R}, \mathcal{D}$ following the approach of Fang and Li [1999].

The conditional posterior distribution of W_i has a much simpler form which can be sampled directly. We have:

$$W_i|\boldsymbol{\mu},\Sigma,\boldsymbol{\lambda},\boldsymbol{\eta},\mathbf{y}_i \sim Ga\left(\frac{k}{2},\frac{g_i}{2}\right)$$

where g_i is calculated as in (6.4).

Finally, the distributions for the Dirichlet process parameters have the same form as for the NI model, but replacing the values w_i with g_i for $i = 1, ..., n$, throughout and can thus be sampled as previously.

6.4 Numerical illustrations

In this section we show how the proposed methodology for estimating the NI and elliptical distribution performs on several simulated and real datasets. With respect to synthetic data, we simulate data from both the NI and elliptical (but non-NI) models and apply the fitting algorithms described in Section 6.3.

6.4.1 Simulated data from a NI distribution

In this example, we consider a sample of size $n = 500$ of a bi-dimensional NI distribution (Student-t with 4 degrees of freedom) with parameters $\boldsymbol{\mu} = (10, 1)$ and $\Sigma = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.4 \end{pmatrix}$. The algorithms were run a total of 250000 iterations with a burnin period of 10000 iterations.

Figure 6.1 depicts the convergence of the algorithms by showing the evolution of the average parameters μ_1 and Σ_{11}. Figure 6.2 shows the fits to the histograms of both components provided by the algorithm to fit an elliptical distribution (solid line) versus that to adjust a NI distribution (dashed line). And, Figure 6.3 shows the contours plots of the fitted elliptical (left panel) and NI (right panel) distributions.

Finally, Table 6.1 depicts the starting points for initializing the model parameters, 95% credible intervals when an elliptical and NI distributions are used to fit the data, according to the algorithms proposed in Section 6.3.

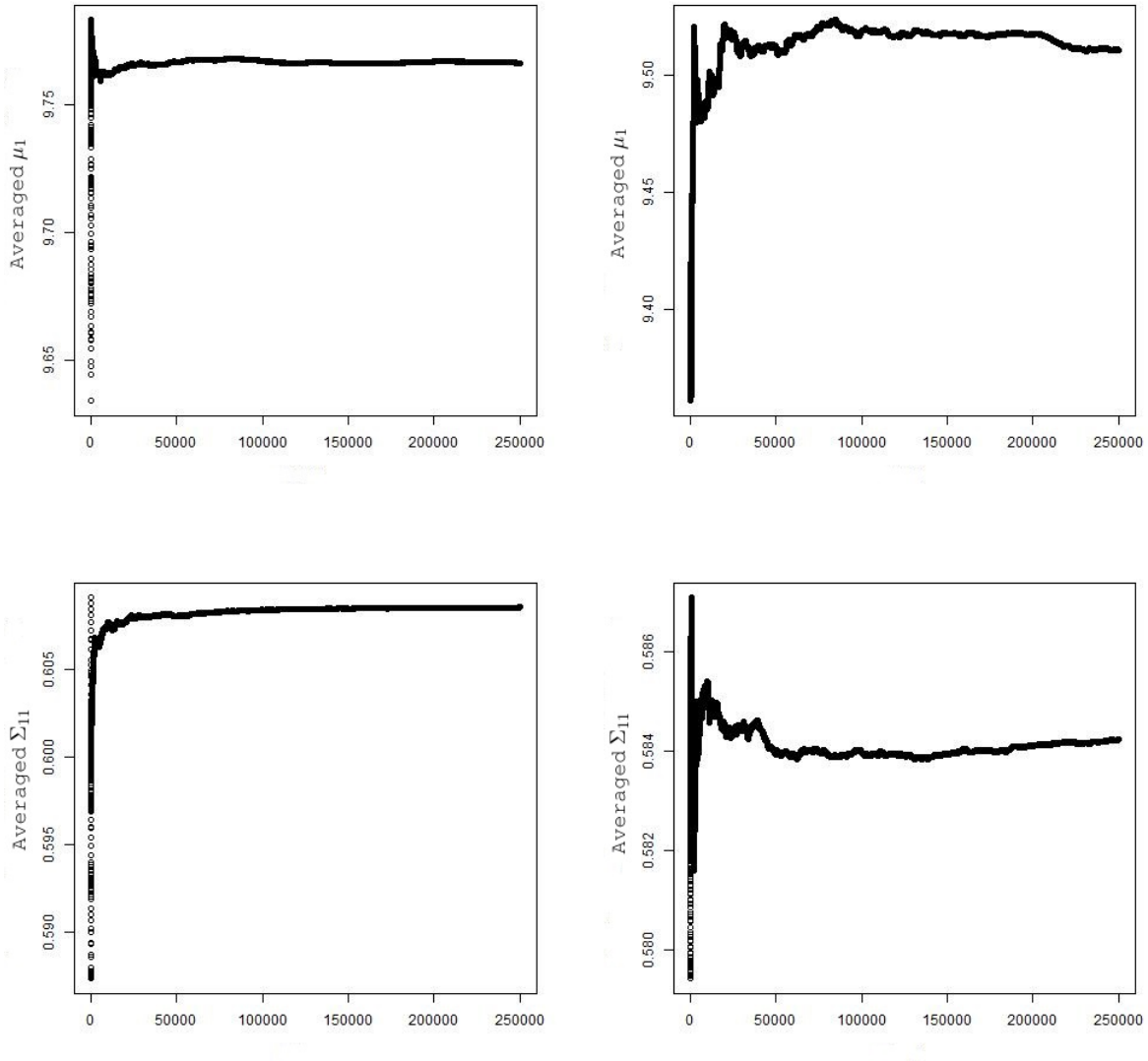


Figure 6.1: Averaged μ_1 (top) and Σ_{11} (bottom) from the total number of runs of MCMC for the NI simulated dataset. *Left panels*: NI fit. *Right panel*: Elliptical fit.

6.4.2 Simulated data from an elliptical distribution

In this section we consider an elliptical distribution with $G \sim Weibull(5, 2)$. The sample size n, and the parameters $\boldsymbol{\mu}$ and Σ are defined as in Section 6.4.1. Figure 6.4 shows the scatterplot of the data generated from the non-NI elliptical model. It can be observed that the elliptical dataset has a doughnut form. Figure 6.5 reports the plots of the two components of $\mathbf{Z}$ (left and right panel, respectively) versus the values of W. We can visualize how both variables, $\mathbf{Z}$ and W, seem to be quadratically dependent. Thus, these data follow an elliptical but non-NI distribution.

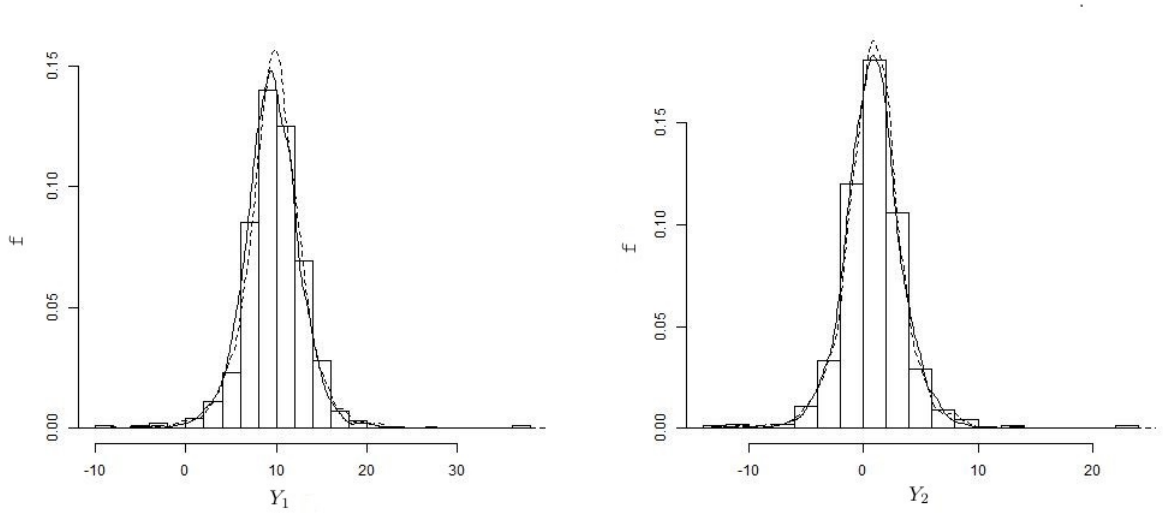


Figure 6.2: Fit to the histogram of the simulated NI data (t4) by an elliptical distribution (solid line) and a NI distribution (dashed line). *Left panel*: first component. *Right panel*: second component.

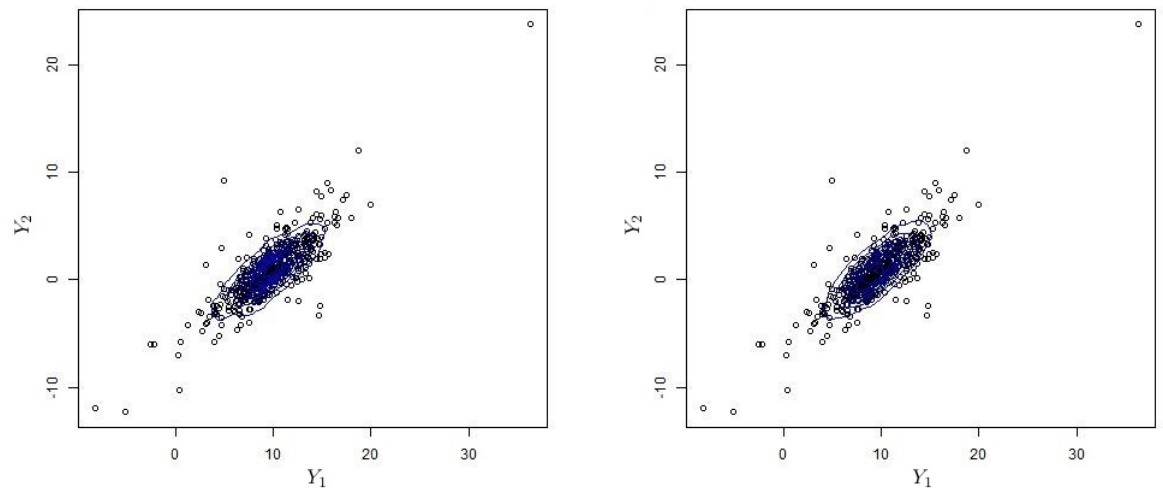


Figure 6.3: Contour plots from fitted joint distributions for the simulated NI data (t4). *Left panel*: Elliptical fit. *Right panel*: NI fit.

Having run a total of 250000 iterations and 10000 iterations for the burning period and having reached the convergence of the algorithms, Figure 6.6 shows the fits to the empirical distribution function of both components (black line) by the elliptical distribution (green line) as well as the NI distribution (red line). It is clear that these data cannot be modeled well via a NI distribution.

Parameters $\boldsymbol{\mu} = (10, 1)$, $\Sigma_{11} = 0.6$, $\Sigma_{12} = \Sigma_{22} = 0.4$	*Elliptical*	*Normal Independent*
$\boldsymbol{\mu}^{(0)}$	(9.6885677, 0.9422802)	(9.6885677, 0.9422802)
$\Sigma_{11}^{(0)}$	0.5959394	0.5959394
$\Sigma_{22}^{(0)}$	0.4040606	0.4040606
$\Sigma_{12}^{(0)}$	0.4065146	0.4065146
C_{μ_1}	[9.350291, 9.80222]	[9.551228, 9.978516]
C_{μ_2}	[0.7281338, 1.006025]	[0.691019, 1.038496]
$C_{\Sigma_{11}}$	[0.5655454, 0.6021779]	[0.5853332, 0.630645]
$C_{\Sigma_{22}}$	[0.3978221, 0.4344546]	[0.369355, 0.4146668]
$C_{\Sigma_{12}}$	[0.382758, 0.4117022]	[0.3943056, 0.4169515]

Table 6.1: Models comparison for the simulated NI (t4) dataset

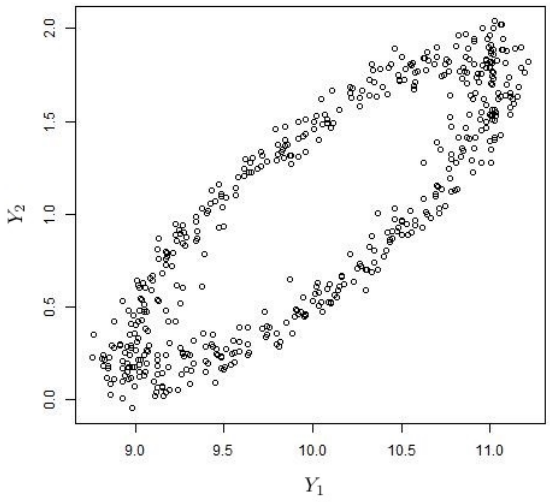


Figure 6.4: Scatterplot of the non-NI, elliptical data

Table 6.2 is similar to Table 6.1, where the starting points for initializing the model parameters as well as 95% credible intervals for both, the elliptical and the NI fittings, are shown. Whereas parameters $\boldsymbol{\mu}$ and Σ are well estimated by both of them, from Figure 6.6 it seems that the NI model is not able to recover the distribution F_G.

6.4.3 Real dataset

In this section we illustrate the performance of the approach for fitting a real dataset in two dimensions. The dataset used in this work is formed by the $n = 2000$ real parts of two channels from a mountain top radar facility. They were collected from a multiple element array, multiple coherent pulse instrumentation radar system which was designed to emulate a radar on an

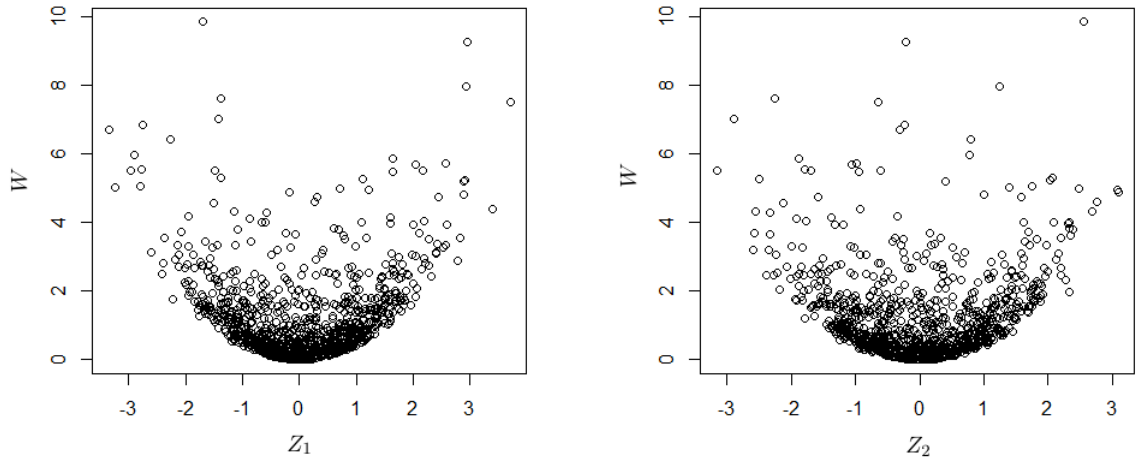


Figure 6.5: The X-axis shows the $\mathbf{Z}$ values (*Left panel*: First component. *Right panel*: Second component) whereas the Y-axis the variable W for the simulated non-NI elliptical dataset

Parameters $\boldsymbol{\mu} = (10, 1)$, $\Sigma_{11} = 0.6$, $\Sigma_{12} = \Sigma_{22} = 0.4$	*Elliptical*	*Normal Independent*
$\boldsymbol{\mu}^{(0)}$	(10.00289, 0.9613196)	(10.00289, 0.9613196)
$\Sigma_{11}^{(0)}$	0.5649796	0.5649796
$\Sigma_{22}^{(0)}$	0.3749756	0.3749756
$\Sigma_{12}^{(0)}$	0.3796576	0.3796576
C_{μ_1}	[9.989973, 10.01237]	[9.942711, 10.05671]
C_{μ_2}	[0.9933165, 1.01185]	[0.9427575, 1.033973]
$C_{\Sigma_{11}}$	[0.589549, 0.6026773]	[0.5805024, 0.6211319]
$C_{\Sigma_{22}}$	[0.3973227, 0.410451]	[0.3788681, 0.4194976]
$C_{\Sigma_{12}}$	[0.396274, 0.4029067]	[0.3975464, 0.4154377]

Table 6.2: Models comparison for the simulated elliptical (but non-NI) dataset

airborne moving platform. Data full description can be found at Titi and Marshall [1996] and data can be downloaded from:

`http://spib.linse.ufsc.br/radar.html`

Both the elliptical, NI distributions were fitted to the data in addition to the multivariate normal distribution. Figure 6.7 depicts the evolution of μ_1 and Σ_{11} from the total number of runs of MCMC, while Figure 6.8 shows the fitted histograms. From the histograms, it can be deduced that the multivariate normal model is not able to fit this dataset, whereas both approaches we present in this work are effectively modeling the data.

Finally, Table 6.3 is analogous to Tables 6.1 and 6.2 where, in addition, the results under a fitted multivariate normal distribution are also shown. From the obtained estimates, it can

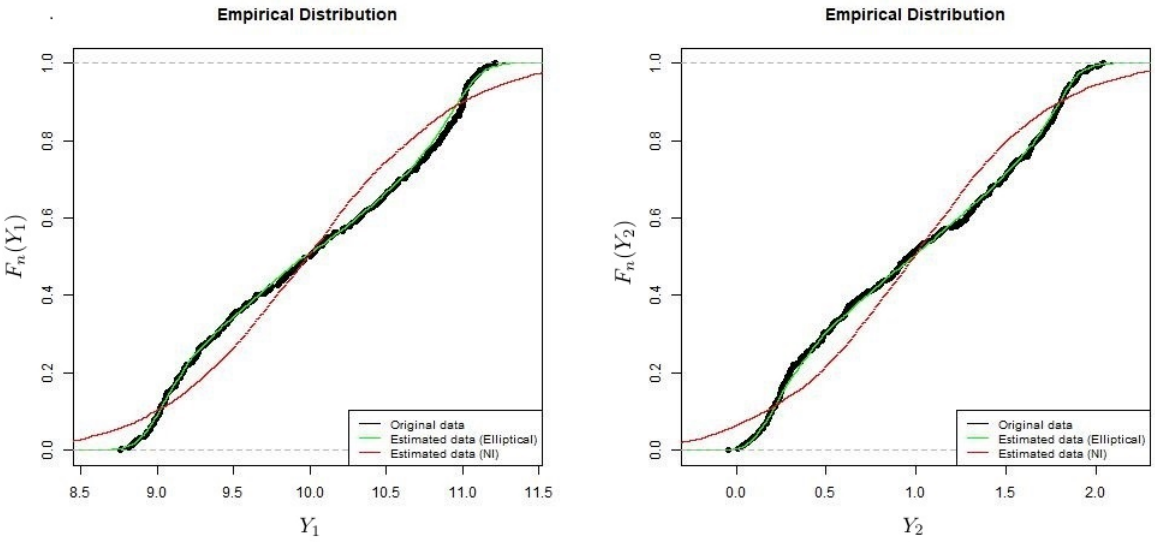


Figure 6.6: Fit to the empirical distribution function of the simulated non-NI elliptical dataset (black line) by the elliptical distribution (green line) and the NI distribution (red line). *Left panel*: first component. *Right panel*: second component.

be seen that the covariance matrix is similarly fitted by all models, but discrepant values are observed in regards the means vector. One possible reason for this behaviour is the fact that the sample exhibits a high variability (variation coefficient around 50). Note also that the credible intervals for μ_1 and μ_2 in the case of the elliptical model presents a lower amplitude than the benchmark approaches. This is in relation with the random walk step to sample for the posterior distribution. The algorithm obtains reasonable acceptance rates but the variance of the proposal distribution is small.

Parameters	*Elliptical*	*Normal Independent*	*Multivariate Normal (R package)*
$\boldsymbol{\mu}^{(0)}$	(11, -2.5)	(11, -2.5)	(11, -2.5)
$\Sigma_{11}^{(0)}$	0.4686852	0.4686852	0.4686852
$\Sigma_{22}^{(0)}$	0.5313148	0.5313148	0.5313148
$\Sigma_{12}^{(0)}$	-0.3048796	-0.3048796	-0.3048796
C_{μ_1}	[-1.1797, 1.072641]	[-19.55131, 15.06277]	[-38.04486, 148.4927]
C_{μ_2}	[-1.208301, 1.193307]	[-16.33051, 19.88458]	[-135.1396, 47.2029]
$C_{\Sigma_{11}}$	[0.4650191, 0.4908175]	[0.4534358, 0.4899288]	[0.4434851, 0.4926143]
$C_{\Sigma_{22}}$	[0.5091825, 0.5349809]	[0.5100712, 0.5465642]	[0.5043759, 0.5618458]
$C_{\Sigma_{12}}$	[-0.2733114, -0.2474081]	[-0.2950202, -0.2678692]	[-0.3266393, -0.2834512]

Table 6.3: Models comparison for the real `Mountain Top Radar` dataset

6.5 Chapter summary

In this chapter we have shown how semi-parametric inference can be carried out for both elliptical and NI distributions in a semi-parametric, Bayesian way, using Dirichlet process mixture models. The ability of the proposed approach for estimating the NI and elliptical distribution has been tested on simulated and real-life datasets.

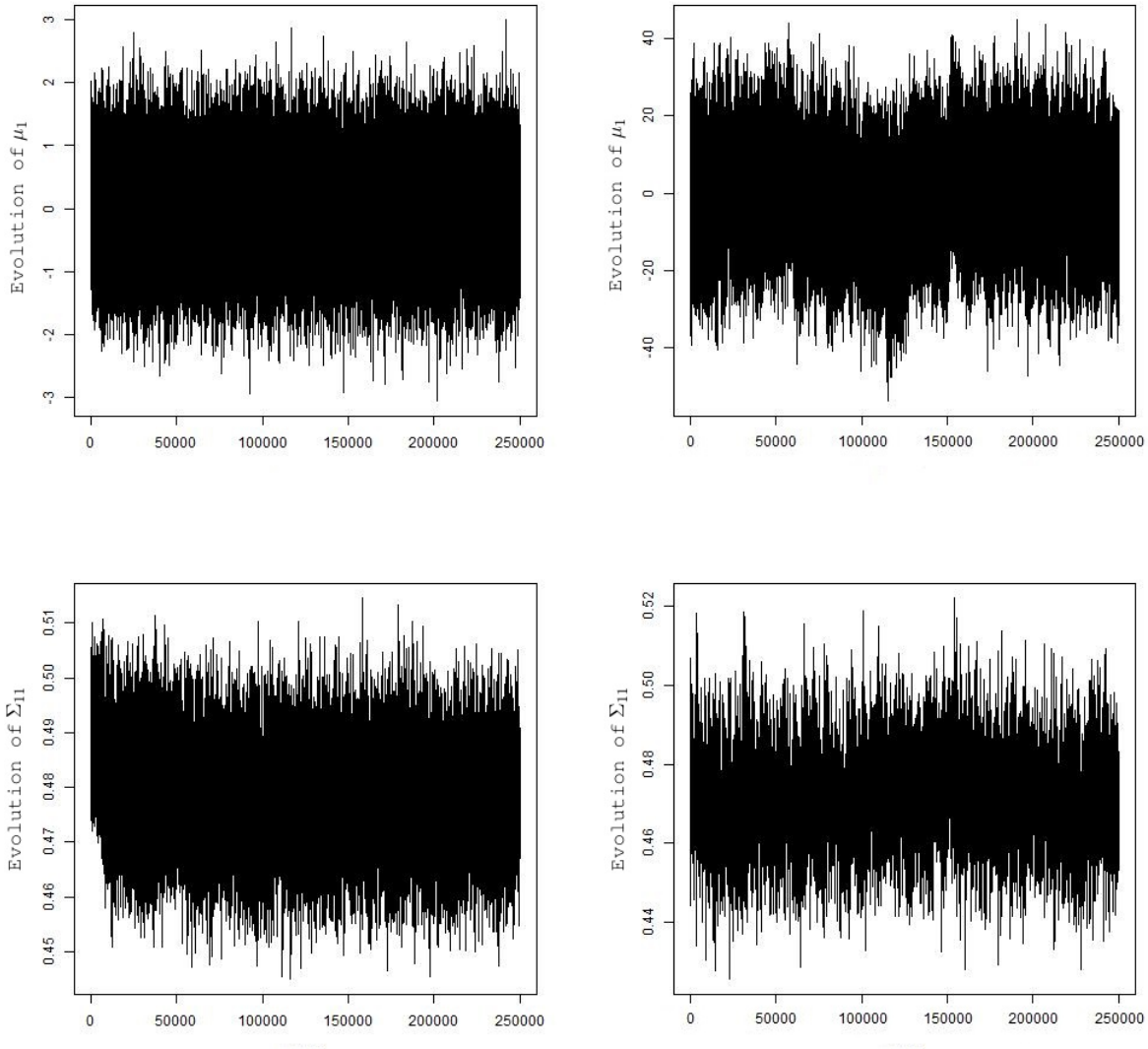


Figure 6.7: Evolution of μ_1 (top) and Σ_{11} (bottom) from the total number of runs of MCMC for the real dataset. *Left panel*: NI fit. *Right panel*: Elliptical fit.

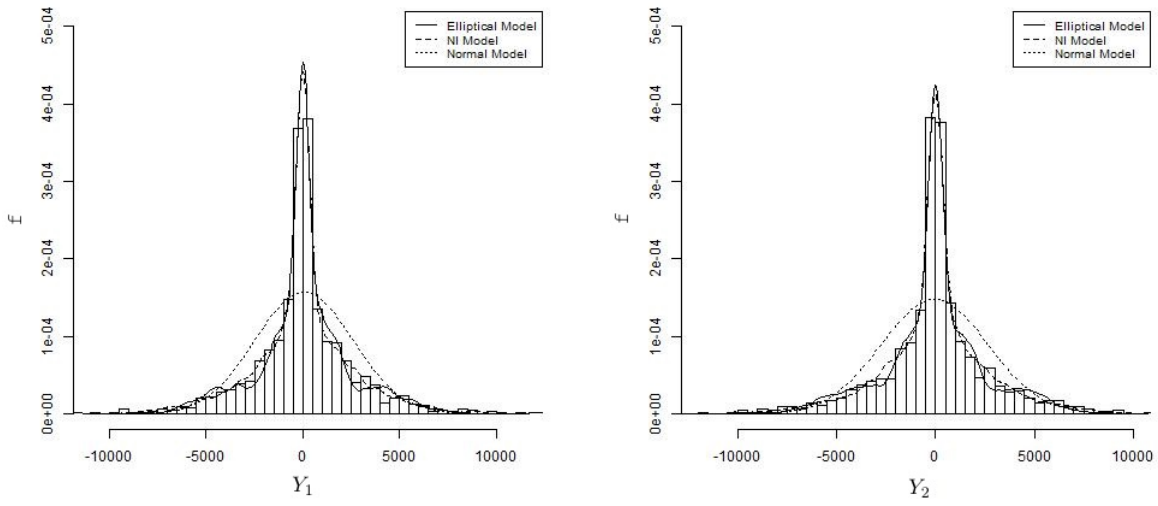


Figure 6.8: Fit to the histogram of the real dataset by an elliptical distribution (solid line), an NI distribution (dashed line) and a multivariate normal (dotted line). *Left panel*: first component. *Right panel*: second component.

Chapter 7

General conclusions and future work

In this PhD dissertation, the disciplines of Operations Research and Statistics constitute the common thread to propose novel solutions in real-life situations regarding with complex datasets. The challenge of learning and interpreting information from complex data is at the core of the current Statistical Science. New computational methods to deal with current datasets, by means of the powerful tools Mathematical Optimization and Bayesian Data Analysis, have been presented through the chapters forming this dissertation.

In recent years, datasets have been usually characterized by a large number of features. This fact may have negative consequences in terms of the comprehensibility of solutions and, thus, the search for more interpretable solutions has recently led to the development of sparse multivariate techniques. Specifically, chapters 2 and 3 demonstrate different tradeoffs between sparsity and predictive performance of linear regression models, whereas Chapter 4 seeks sparse Bayesian classification models.

Besides, when dealing with real-world applications where there exist groups at risk, it is more important to improve performance rates for such groups. To this aim, in this thesis, we apply mathematical optimization tools, which seem to be suitable. Then, in chapters 2 and 5 we propose two new constrained optimization models: one overall criterion is optimized, while constraints on the efficiency measures under consideration for the individuals of interest are introduced.

Additionally, when dealing with datasets with extreme instances (heavy-tailed data), Bayesian statistical techniques have proven useful, as Chapter 6 also confirms.

The works presented in this PhD dissertation can be extended. Next, we discuss some open problems.

- In Chapter 2 a novel version of the Lasso in which quadratic performance constraints are added to Lasso-based objective functions is introduced. A possible extension for this work could be to change the objective function. For example, for the sake of dealing with strongly correlated predictors, it may be of interest to change the objective function by that of the *elastic net*. Another non-straightforward extension could be to address classification problems (via the logistic regression) instead of regression problems. In this case, we would not address a quadratic formulation.

- A new methodology to deal with hierarchical categorical variables in linear regression models has been studied in Chapter 3. A number of extensions to this work are the following. Firstly, when the number of categories is large, instead of solving Problem (3.11) considering all the categories at once, a sequential pruning can be used instead. The main idea is to consider subtrees in $\mathcal{T}_j$ and try to compress their categories solving Problem (3.11) sequentially. Another option to deal with large number of categories is to cluster them based on a dissimilarity, see Carrizosa et al. [2017a]; Cerda et al. [2018] and references therein. Secondly, our methodology can be extended to generalized linear models [Tibshirani, 1996], where, instead of predicting the response variable as in (3.1), a non-

linear relationship between the response variable and the predictors is through a linkage function. However, this extension makes the optimization problem highly nonlinear and its resolution is very challenging.

- In Chapter 4 a novel feature selection methodology characterized by three main features is presented. Namely, (1) sparsity is achieved taking into account the correlation among the features, (2) different performance measures can be used to guide the selection of features and (3) performance constraints on groups of interest can be included. It has been explored in the case of the NB classifier due to its tractability and good performance, but other classifiers could have also been tested instead.

- Chapter 5 is devoted to introduce a constrained version of the classic NB classifier with the aim of improving classification rates in the different classes, avoiding the use of misclassification costs. Although distributional assumptions have been considered throughout this chapter, a possible extension to this work is to consider non parametric estimation for the density function for continuous attributes via kernel density estimation.

- Finally, elliptical distributions are examined in Chapter 6. In particular, semi-parametric inference has been carried out for both elliptical and NI distributions, using Dirichlet process mixture models. To continue this work we have to undertake model selection. For model selection purposes, one possibility would be to try to calculate the marginal likelihoods for the various models considered and then to use the Bayes factor to select an optimal model, following Basu and Chib [2003]. However, other simpler approaches which are appropriate for mixture models can also be explored [Spiegelhalter et al., 2002]. Additionally, a more extensive computation study with applications in Finance, where heavy-tailed distributions are commonly used, is planned to be carried out.

List of Figures

List of Tables

Appendix A

Proofs

Proof of Proposition 1

Given $\lambda \geq 0$, consider problem (2.11). If $\boldsymbol{\beta} = \boldsymbol{\beta}^+ - \boldsymbol{\beta}^-$ with $\boldsymbol{\beta}^+ \geq 0$ and $\boldsymbol{\beta}^- \geq 0$ and $\boldsymbol{\lambda} = (0, \lambda, \ldots, \lambda)'$, a vector whose length is $p+1$, then the differentiable version of that problem turns out to be

$$\begin{aligned} \min_{\boldsymbol{\beta}^+, \boldsymbol{\beta}^-} \quad & \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 + \boldsymbol{\lambda}'\boldsymbol{\beta}^+ + \boldsymbol{\lambda}'\boldsymbol{\beta}^- \\ \text{s.t.} \quad & \frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 - (1+\tau)\text{MSE}_1(\hat{\boldsymbol{\beta}}^{ols}) \leq 0, \\ & \boldsymbol{\beta}^+ \geq \mathbf{0} \Leftrightarrow -\boldsymbol{\beta}^+ \leq \mathbf{0}, \\ & \boldsymbol{\beta}^- \geq \mathbf{0} \Leftrightarrow -\boldsymbol{\beta}^- \leq \mathbf{0}. \end{aligned}$$

From the Karush-Kuhn-Tucker conditions,

$$\mathcal{L}(\boldsymbol{\beta}^+, \boldsymbol{\beta}^-, \boldsymbol{\theta}^+, \boldsymbol{\theta}^-, \eta) = \frac{1}{n_0}\|\mathbf{y}_0 - \mathcal{X}_0(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 + \boldsymbol{\lambda}'\boldsymbol{\beta}^+ + \boldsymbol{\lambda}'\boldsymbol{\beta}^- - (\boldsymbol{\theta}^+)'\boldsymbol{\beta}^+ - (\boldsymbol{\theta}^-)'\boldsymbol{\beta}^- +$$

$$+\eta\left(\frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 - (1+\tau)\text{MSE}_1(\hat{\boldsymbol{\beta}}^{ols})\right)$$

$$\frac{\partial}{\partial \boldsymbol{\beta}^+} : -\frac{2}{n_0}\mathcal{X}_0'(\mathbf{y}_0 - \mathcal{X}_0(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)) + \boldsymbol{\lambda} - \boldsymbol{\theta}^+ - \frac{2}{n_1}\eta\mathcal{X}_1'(\mathbf{y}_1 - \mathcal{X}_1(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)) = 0$$

$$\frac{\partial}{\partial \boldsymbol{\beta}^-} : \frac{2}{n_0}\mathcal{X}_0'(\mathbf{y}_0 - \mathcal{X}_0(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)) + \boldsymbol{\lambda} - \boldsymbol{\theta}^- + \frac{2}{n_1}\eta\mathcal{X}_1'(\mathbf{y}_1 - \mathcal{X}_1(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)) = 0$$

$$\boldsymbol{\theta}^+, \boldsymbol{\theta}^-, \eta \geq 0$$

$$(\boldsymbol{\theta}^+)'\boldsymbol{\beta}^+ = 0$$

$$(\boldsymbol{\theta}^-)'\boldsymbol{\beta}^- = 0$$

$$\eta\left(\frac{1}{n_1}\|\mathbf{y}_1 - \mathcal{X}_1(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 - (1+\tau)\text{MSE}_1(\hat{\boldsymbol{\beta}}^{ols})\right) = 0$$

Thus,

- if $\boldsymbol{\beta} > 0 \Rightarrow \boldsymbol{\beta}^+ > 0, \boldsymbol{\beta}^- = 0 \Rightarrow \boldsymbol{\theta}^+ = 0 \Rightarrow -\frac{2}{n_0}\mathcal{X}_0'(\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}) + \boldsymbol{\lambda} - \frac{2}{n_1}\eta\mathcal{X}_1'(\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}) = 0$
- if $\boldsymbol{\beta} < 0 \Rightarrow \boldsymbol{\beta}^+ = 0, \boldsymbol{\beta}^- > 0 \Rightarrow \boldsymbol{\theta}^- = 0 \Rightarrow \frac{2}{n_0}\mathcal{X}_0'(\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}) + \boldsymbol{\lambda} + \frac{2}{n_1}\eta\mathcal{X}_1'(\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}) = 0$

Therefore,

$$\frac{2}{n_0}\mathcal{X}_0'(\mathbf{y}_0 - \mathcal{X}_0\boldsymbol{\beta}) + \frac{2}{n_1}\eta(\lambda)\mathcal{X}_1'(\mathbf{y}_1 - \mathcal{X}_1\boldsymbol{\beta}) = \mathbf{b}(\lambda), \tag{A.1}$$

where $\eta(\lambda)$ is the Lagrange multiplier associated with the first constraint and $\mathbf{b}(\lambda)$ is a $(p+1)$-dimensional vector whose s-th component, $s = 0, 1, \ldots, p$, takes the following value

$$b_s(\lambda) = \begin{cases} \lambda, & \text{if } \beta_s > 0, \\ -\lambda, & \text{if } \beta_s < 0, \\ 0, & \text{else.} \end{cases}$$

Then, since $\mathcal{X}_0$ and $\mathcal{X}_1$ are maximum rank matrices, one obtains from (A.1) the following implicit expression for the solution $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ of Problem (2.11)

$$\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda) = \left(\frac{1}{n_0}\mathcal{X}_0'\mathcal{X}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathcal{X}_1\right)^{-1}\left(\frac{1}{n_0}\mathcal{X}_0'\mathbf{y}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathbf{y}_1\right) - \frac{1}{2}\left(\frac{1}{n_0}\mathcal{X}_0'\mathcal{X}_0 + \frac{1}{n_1}\eta(\lambda)\mathcal{X}_1'\mathcal{X}_1\right)^{-1}\mathbf{b}(\lambda).$$

Proof of Theorem 1

Consider the function $h : \boldsymbol{\beta} \mapsto \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 = \frac{1}{n}(\mathbf{y} - \mathcal{X}\boldsymbol{\beta})'(\mathbf{y} - \mathcal{X}\boldsymbol{\beta})$, where $\mathcal{X}$ is a maximum rank matrix by hypothesis. The matrix $\mathcal{X}$ is of maximum rank and therefore the Hessian matrix $H_h(\boldsymbol{\beta}) = \frac{2}{n}\mathcal{X}'\mathcal{X}$ is positive definite, from where we conclude that $h(\boldsymbol{\beta})$ is strictly convex, and hence, $h(\boldsymbol{\beta}) + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1$ is also a strictly convex function.

We next prove that $h(\boldsymbol{\beta})$ is a coercive function. Since $\mathcal{X}'\mathcal{X}$ is positive definite, its eigenvalues are all positive. In particular, the smallest eigenvalue, say γ_r, will be nonzero. Moreover, using the spectral decomposition of a symmetric matrix,

$$\begin{aligned} \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 = \frac{1}{n}(\mathbf{y} - \mathcal{X}\boldsymbol{\beta})'(\mathbf{y} - \mathcal{X}\boldsymbol{\beta}) = \frac{1}{n}\boldsymbol{\beta}'\mathcal{X}'\mathcal{X}\boldsymbol{\beta} - \frac{2}{n}\mathbf{y}'\mathcal{X}\boldsymbol{\beta} + \frac{1}{n}\mathbf{y}'\mathbf{y} = \\ = \frac{1}{n}\boldsymbol{\beta}'\mathcal{Q}'\mathcal{D}\mathcal{Q}\boldsymbol{\beta} - \frac{2}{n}\mathbf{y}'\mathcal{X}\boldsymbol{\beta} + \frac{1}{n}\mathbf{y}'\mathbf{y} \geq \\ \geq \frac{1}{n}\boldsymbol{\beta}'\mathcal{Q}'\mathcal{D}\mathcal{Q}\boldsymbol{\beta} - \left|\frac{2}{n}\mathbf{y}'\mathcal{X}\boldsymbol{\beta}\right| + \frac{1}{n}\mathbf{y}'\mathbf{y} \geq \frac{\gamma_r}{n}\|\mathcal{Q}\boldsymbol{\beta}\|^2 - \left\|\frac{2}{n}\mathbf{y}'\mathcal{X}\right\|\|\boldsymbol{\beta}\| + \frac{1}{n}\mathbf{y}'\mathbf{y} = \\ = \frac{\gamma_r}{n}\|\boldsymbol{\beta}\|^2 - \left\|\frac{2}{n}\mathbf{y}'\mathcal{X}\right\|\|\boldsymbol{\beta}\| + \frac{1}{n}\mathbf{y}'\mathbf{y}, \end{aligned}$$

where, in the second-to-last step, the Cauchy-Schwarz inequality has been used. As $\|\boldsymbol{\beta}\| \to +\infty$, then $h(\boldsymbol{\beta}) \to +\infty$ too, and thus $h(\boldsymbol{\beta})$ is a coercive function.

Now we show that (2.13) has optimal solution. Let $\boldsymbol{\beta}^* \in \boldsymbol{B}$. As $h(\boldsymbol{\beta})$ is coercive, then there exists $R > 0$ such that

$$\frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}\|^2 > \frac{1}{n}\|\mathbf{y} - \mathcal{X}\boldsymbol{\beta}^*\|^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}^*\|_1,$$

for all $\boldsymbol{\beta}$ such that $\|\boldsymbol{\beta}\| > R$. For that reason, the problem can be reduced to the feasible compact region $\boldsymbol{B} \cap \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\| \leq R\}$, which implies that the optimal solution is reached.

Finally, the uniqueness of the solution follows from the fact that $h(\boldsymbol{\beta}) + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1$ is strictly convex.

Proof of Proposition 2

Let us consider the optimization problem (2.2) and let $\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)$ denotes its optimal solution. The necessary and sufficient optimality condition is:

$$\nabla\frac{1}{n}\|\mathbf{y} - \mathcal{X}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)\|^2 + \lambda\partial\|\mathcal{A}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)\|_1 \ni \mathbf{0}. \tag{A.2}$$

From the properties of subdifferential (see *Theorem 23.9* of Rockafellar [1972]) it follows that

$$\partial\|\mathcal{A}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)\|_1 = \mathcal{A}'\partial\|.\|_{1_{|\mathcal{A}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)}},$$

which implies that (A.2) becomes

$$-\frac{2}{n}\mathcal{X}'(\mathbf{y} - \mathcal{X}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)) + \lambda\mathcal{A}'\partial\|.\|_{1_{|\mathcal{A}\hat{\boldsymbol{\beta}}^{Lasso}(\lambda)}} \ni \mathbf{0}. \tag{A.3}$$

Consequently, the necessary and sufficient condition (A.3) in $\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \mathbf{0}$ is

$$-\frac{2}{n}\mathcal{X}'\mathbf{y} + \lambda\{\mathcal{A}'\mathbf{t} : \|\mathbf{t}\|_\infty \leq 1\} \ni \mathbf{0},$$

since $\partial\|\mathbf{0}\|_1$ is the unit ball of the $\|.\|_\infty$. Equivalently,

$$\frac{2}{n}\mathcal{X}'\mathbf{y} \in \{\mathcal{A}'\lambda\mathbf{t} : \|\mathbf{t}\|_\infty \leq 1\}.$$

Therefore, the solution of the problem

$$\begin{aligned} \min_{\lambda,\mathbf{t}} \quad & \lambda \\ \text{s.t.} \quad & \frac{2}{n}\mathcal{X}'\mathbf{y} = \mathcal{A}'\lambda\mathbf{t}, \\ & \|\mathbf{t}\|_\infty \leq 1, \\ & \lambda \geq 0, \end{aligned} \tag{A.4}$$

will provide the minimum λ from which $\hat{\boldsymbol{\beta}}^{Lasso}(\lambda) = \mathbf{0}$ is the optimal solution. If $\mathbf{q} = \lambda\mathbf{t}$, then Problem (A.4) becomes

$$\begin{aligned} \min_{\lambda,\mathbf{q}} \quad & \lambda \\ \text{s.t.} \quad & \frac{2}{n}\mathcal{X}'\mathbf{y} = \mathcal{A}'\mathbf{q}, \\ & \|\mathbf{q}\|_\infty \leq \lambda. \end{aligned}$$

The constraint $\|\mathbf{q}\|_\infty$ is equivalent to $|q_s| \leq \lambda$, $s = 0, 1, \ldots, p$ and the result follows.

Proof of Proposition 3

The proof follows very closely that of Theorem 1. First of all, it shall be proven that $h : \boldsymbol{\beta} \mapsto E[(Y - \mathbf{X}'\boldsymbol{\beta})^2]$ is coercive. It is strictly convex on $\boldsymbol{\beta}$ since its Hessian matrix, $2E[\mathbf{X}\mathbf{X}']$, is positive definite due to X is an absolutely continuous p-dimensional random variable:

$$u'E[\mathbf{X}\mathbf{X}']u = E[u'\mathbf{X}\mathbf{X}'u] = E[(\mathbf{X}'u)^2] > 0,$$

since $P(\mathbf{X}'u = 0) = 0$. Moreover, $\lambda\|\mathcal{A}\boldsymbol{\beta}\|_1$ is a convex function on $\boldsymbol{\beta}$ and, therefore, $E[(Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1]$ is also a strictly convex function on $\boldsymbol{\beta}$.

On the one hand, the eigenvalues of the Hessian matrix are all positive and, in particular, the smallest eigenvalue, say γ_r, will be non-zero. On the other hand, using the spectral decomposition of a symmetric matrix,

$$E[(Y - \mathbf{X}'\boldsymbol{\beta})^2] = \boldsymbol{\beta}'E[\mathbf{X}\mathbf{X}']\boldsymbol{\beta} - 2E[Y\mathbf{X}]\boldsymbol{\beta} + E[Y^2] = \boldsymbol{\beta}'\mathcal{Q}'\mathcal{D}\mathcal{Q}\boldsymbol{\beta} - 2E[Y\mathbf{X}]\boldsymbol{\beta} + E[Y^2] \geq$$

$$\geq \boldsymbol{\beta}'\mathcal{Q}'\mathcal{D}\mathcal{Q}\boldsymbol{\beta} - \mid 2E[Y\mathbf{X}]\boldsymbol{\beta} \mid + E[Y^2] \geq \gamma_r\|\mathcal{Q}\boldsymbol{\beta}\|^2 - \|E[Y\mathbf{X}]\|\|\boldsymbol{\beta}\| + E[Y^2] =$$

$$= \gamma_r\|\boldsymbol{\beta}\|^2 - \|E[Y\mathbf{X}]\|\|\boldsymbol{\beta}\| + E[Y^2],$$

where, in the second-to-last step, the Cauchy-Schwarz inequality was used. As $\|\boldsymbol{\beta}\| \to +\infty$, then $E[(Y - \mathbf{X}'\boldsymbol{\beta})^2] \to +\infty$, that is, the quadratic function $h(\boldsymbol{\beta}) = E[(Y - \mathbf{X}'\boldsymbol{\beta})^2]$ is coercive. The next step in the proof is to transform the original *true* problem (2.17) into an equivalent one with a feasible compact region $\boldsymbol{B}^*$. Given $\boldsymbol{\beta}^* \in \boldsymbol{B}$, since $h(\boldsymbol{\beta}) = E[(Y - \mathbf{X}'\boldsymbol{\beta})^2]$ is coercive, there exists R such that

$$E[(Y - \mathbf{X}'\boldsymbol{\beta})^2] \; > E[(Y - \mathbf{X}'\boldsymbol{\beta}^*)^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}^*\|_1],$$

for all $\boldsymbol{\beta}$ with $\|\boldsymbol{\beta}\| > R$. For that reason, the problem (2.17) can be reduced to the feasible compact region $\boldsymbol{B}^* = \boldsymbol{B} \cap \{\boldsymbol{\beta} : \; \|\boldsymbol{\beta}\| \leq R\}$, which implies that the optimal solution is reached.

Finally, the uniqueness of solution is a consequence of the strict convexity of the objective

function.

Proof of Theorem 2

For the sake of simplicity, $\boldsymbol{\beta}^{CSCLasso}(\lambda)$ and $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda)$ will be denoted henceforth by $\boldsymbol{\beta}$ and $\hat{\boldsymbol{\beta}}$, respectively. In addition, let us consider the nonempty compact set $C = \boldsymbol{B} \cap \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\| \leq R\}$, where R is chosen according to the proof of Theorem 2.3.1.

Theorem 2 is a direct consequence of *Theorem 5.3* in Shapiro et al. [2009] under some technical conditions, namely:

C1. The expected value function $E[(Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1]$ is finite valued and continuous on C.

C2. $\frac{1}{n}\sum_{i=1}^{n}((y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1)$ converges to $E[(Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1]$ w.p. 1, as $n \to \infty$, uniformly in $\boldsymbol{\beta} \in C$.

Let us denote $F(\boldsymbol{\beta}, (Y, \mathbf{X})) = (Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1$. Then, the previous conditions C1 and C2 are consequences of *Theorem 7.48* in Shapiro et al. [2009] provided that

A1. for any $\boldsymbol{\beta} \in C$, the function $F(\cdot, (Y, \mathbf{X}))$ is continuous at $\boldsymbol{\beta}$ for almost every $(Y, \mathbf{X})$,

A2. the function $F(\boldsymbol{\beta}, (Y, \mathbf{X}))$, with $\boldsymbol{\beta} \in C$, is dominated by an integrable function,

A3. the sample is i.i.d.

Given $(Y, \mathbf{X})$, the function $(Y - \mathbf{X}'\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1$ is continuous at $\boldsymbol{\beta}$ for any $\boldsymbol{\beta} \in C$, and therefore A1 is fulfilled. The sample is i.i.d. by hypothesis, and thus A3 holds too. Finally, in order to prove A2, it is necessary to find a measurable function $g(Y, \mathbf{X}) > 0$ such that $E[g(Y, \mathbf{X})] < \infty$ and, for every $\boldsymbol{\beta} \in C$, $\mid F(\boldsymbol{\beta}, (Y, \mathbf{X})) \mid \leq g(Y, \mathbf{X})$ w.p. 1. Using the Cauchy-Schwarz inequality, one has,

$$\mid F(\boldsymbol{\beta}, (Y, \mathbf{X})) \mid = \mid (Y - \mathbf{X}\boldsymbol{\beta})^2 + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \mid =$$

$$= \mid Y^2 - 2Y\mathbf{X}'\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}\mathbf{X}'\boldsymbol{\beta} + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \mid \leq$$

$$\leq Y^2 + (\mathbf{X}'\boldsymbol{\beta})^2 + 2 \mid Y\mathbf{X}'\boldsymbol{\beta} \mid + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 =$$

$$= Y^2 + \mid \mathbf{X}'\boldsymbol{\beta} \mid^2 + 2 \mid Y\mathbf{X}'\boldsymbol{\beta} \mid + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1 \leq$$

$$\leq Y^2 + \|\mathbf{X}\|^2\|\boldsymbol{\beta}\|^2 + 2\|Y\mathbf{X}\|\|\boldsymbol{\beta}\| + \lambda\|\mathcal{A}\boldsymbol{\beta}\|_1.$$

Let M_1 and M_2 be given by

$$M_1 = \max_{\boldsymbol{\beta} \in C} \|\boldsymbol{\beta}\| \qquad M_2 = \max_{\boldsymbol{\beta} \in C} |\mathcal{A}\boldsymbol{\beta}|$$

which are well defined due to the compactness of C. Therefore, g can be chosen as

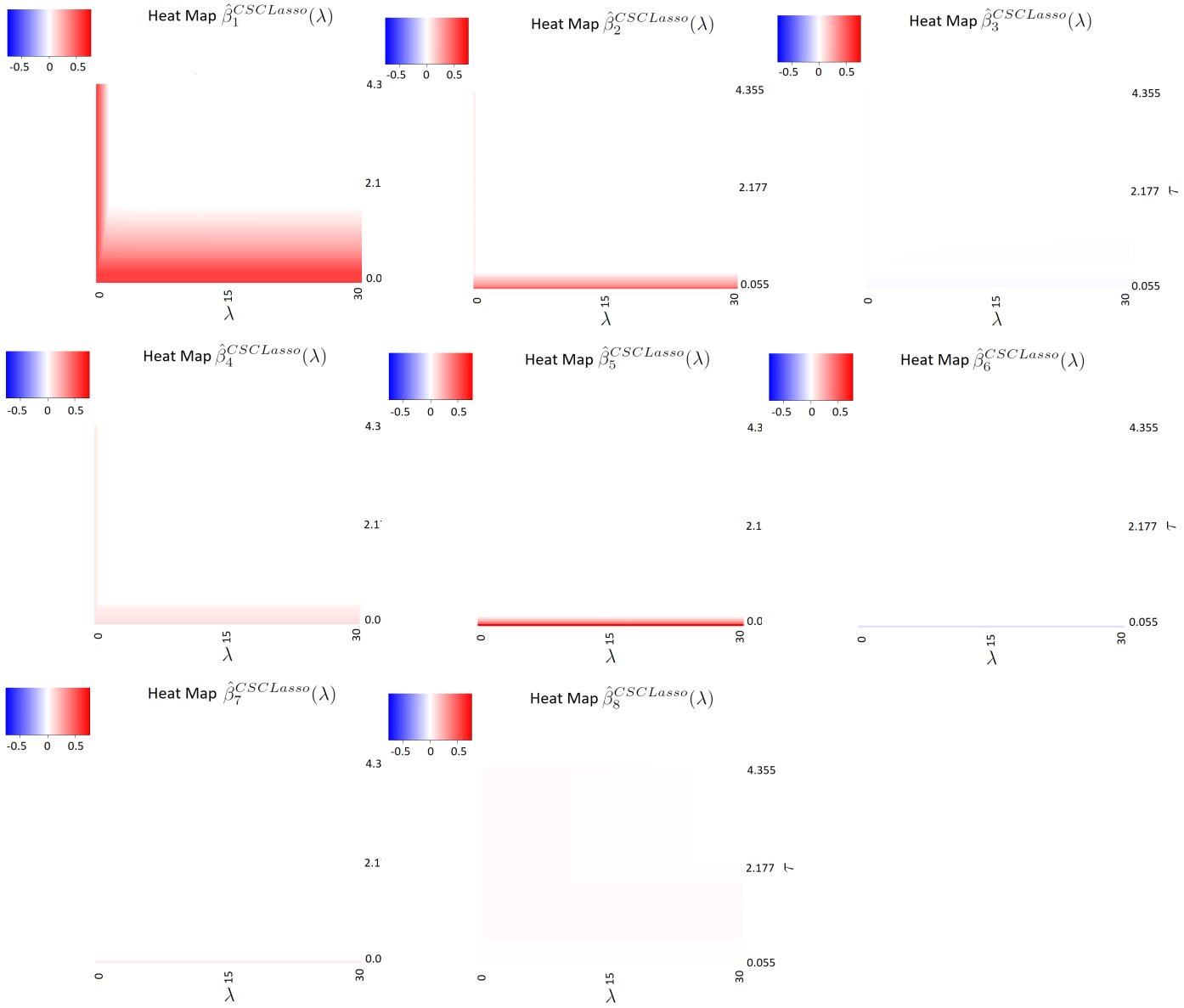
$$g(Y, \mathbf{X}) = Y^2 + M_1^2 \|\mathbf{X}\|^2 + 2M_1 \|Y\mathbf{X}\| + \lambda M_2,$$

which is positive and, since $E(\|\mathbf{X}\|^2) < \infty$, $E(Y^2) < \infty$, $E(\|Y\mathbf{X}\|) < \infty$, its expected value is finite. In consequence, A2 holds and the proof is concluded.

Appendix B

Further results

Figure B.1: Heat maps of $\hat{\boldsymbol{\beta}}^{CSCLasso}(\lambda) = (\hat{\beta}_1^{CSCLasso}(\lambda), \ldots, \hat{\beta}_8^{CSCLasso}(\lambda))$ using `prostate` dataset



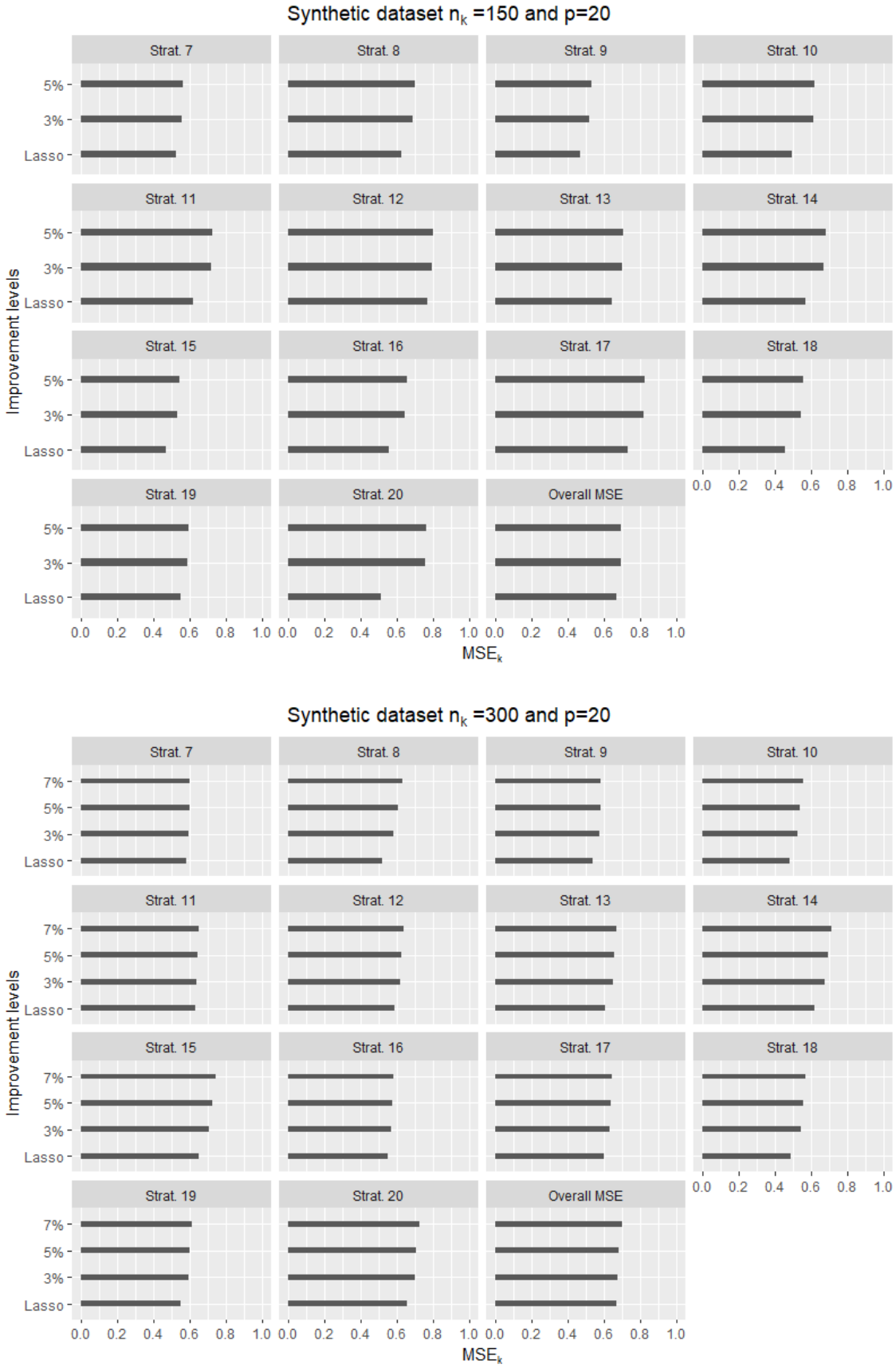


Figure B.2: Median MSE_k over the test sets for $k = 7, \ldots, 20$ under $p = 20$ features and the two n_k options. Each subgraph represents one group and the Y-axis shows the different percentages of improvement

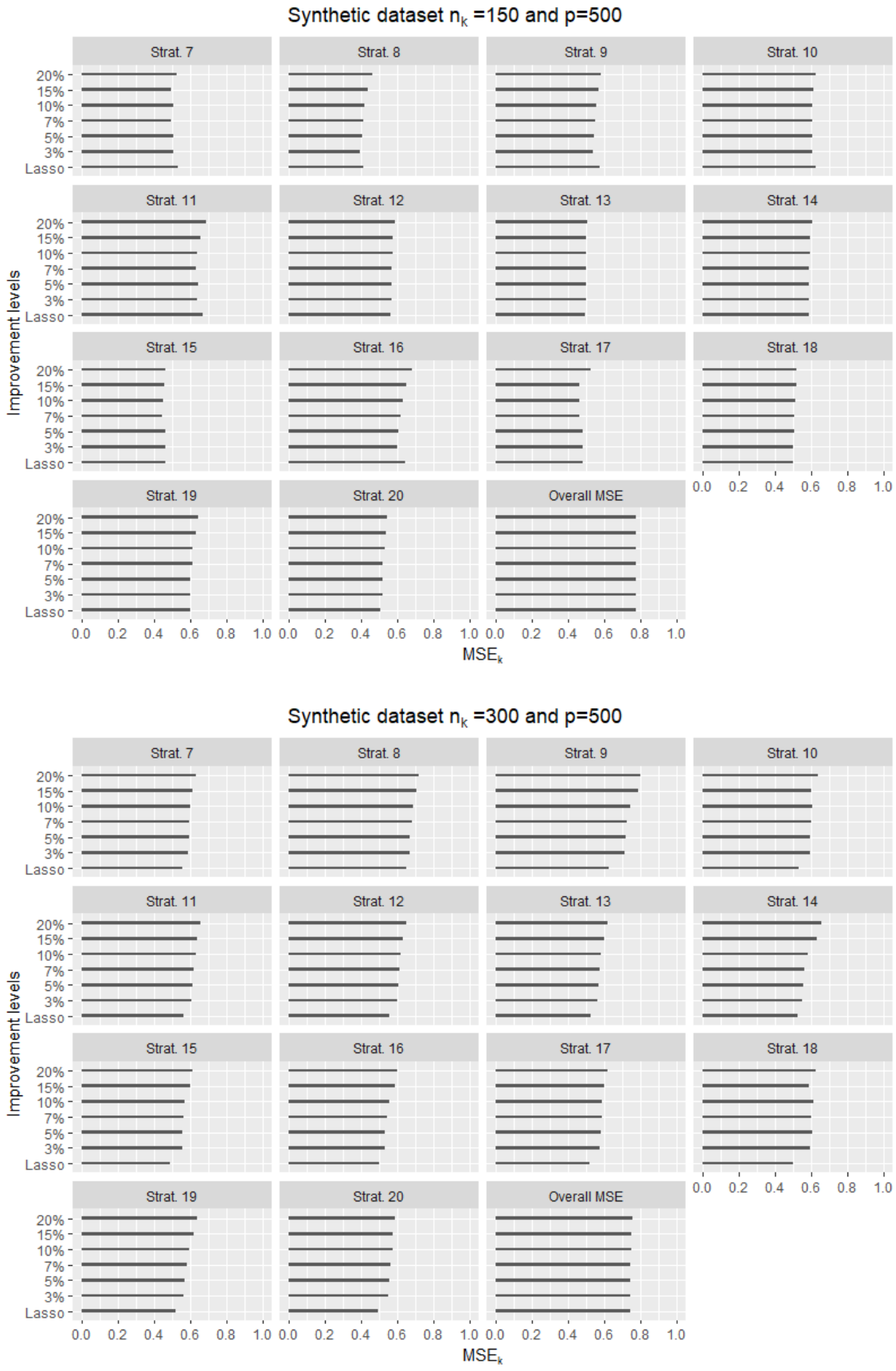


Figure B.3: Median MSE_k over the test sets for $k = 7, \ldots, 20$ under $p = 500$ features and the two n_k options. Each subgraph represents one group and the Y-axis shows the different percentages of improvement

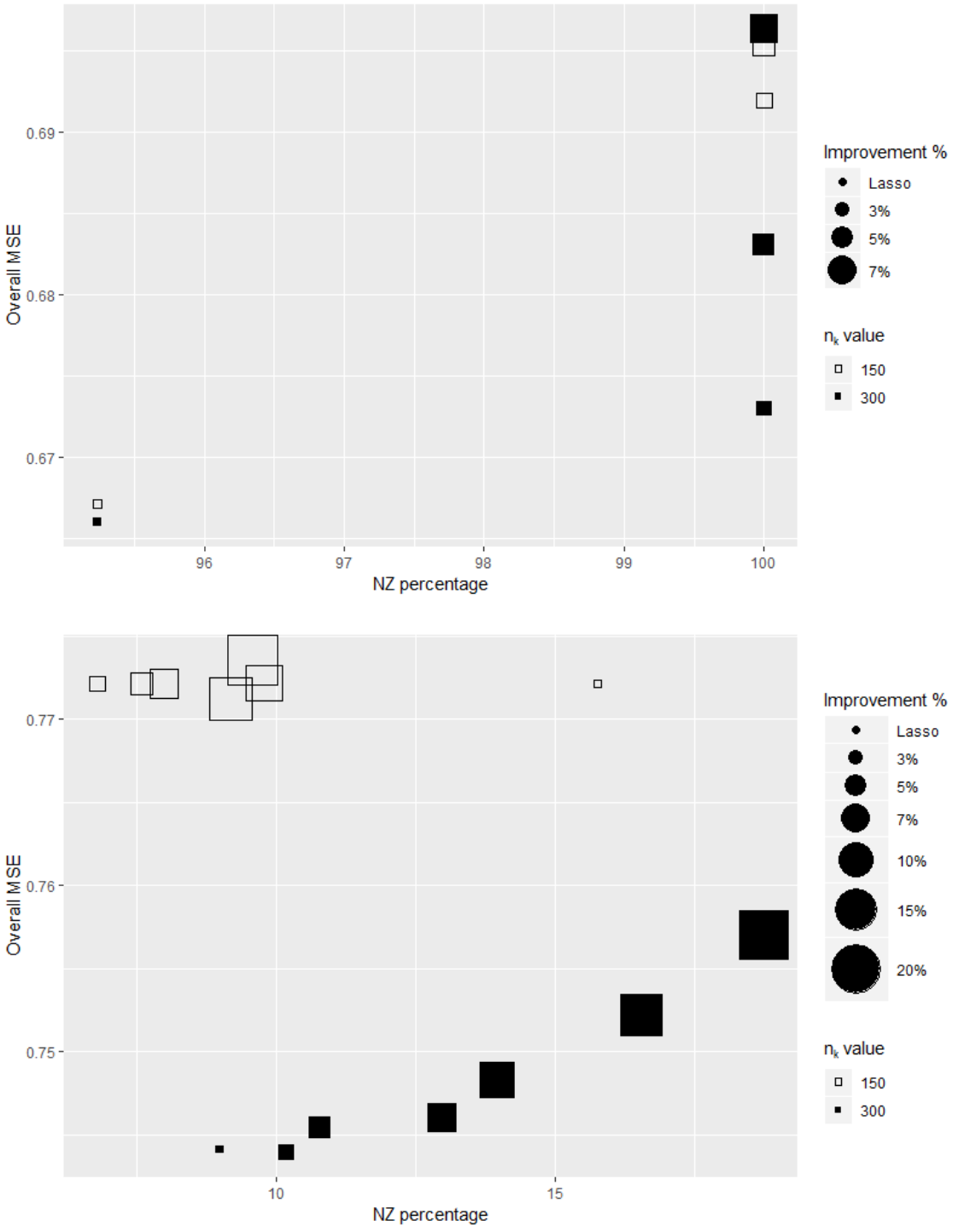


Figure B.4: Median overall MSE over the test sets and NZ percentage under the choice $p = 20$ (top) and $p = 500$ (bottom)

To fully understand how the computation time behaves depending on n_k and p values, a grid in both parameters have been inspected. Figure B.5 displays the logarithm of the user times in seconds obtained under Lasso and CSCLasso models when n_k and p change. The perspective drawn in the top left figure shows that Lasso model (bottom surface) is solved faster and in a smoother way. Besides, whereas smaller times are obtained for both methods when n_k and p are small, the biggest times are associated to $n_k = 300$ and $p = 500$.

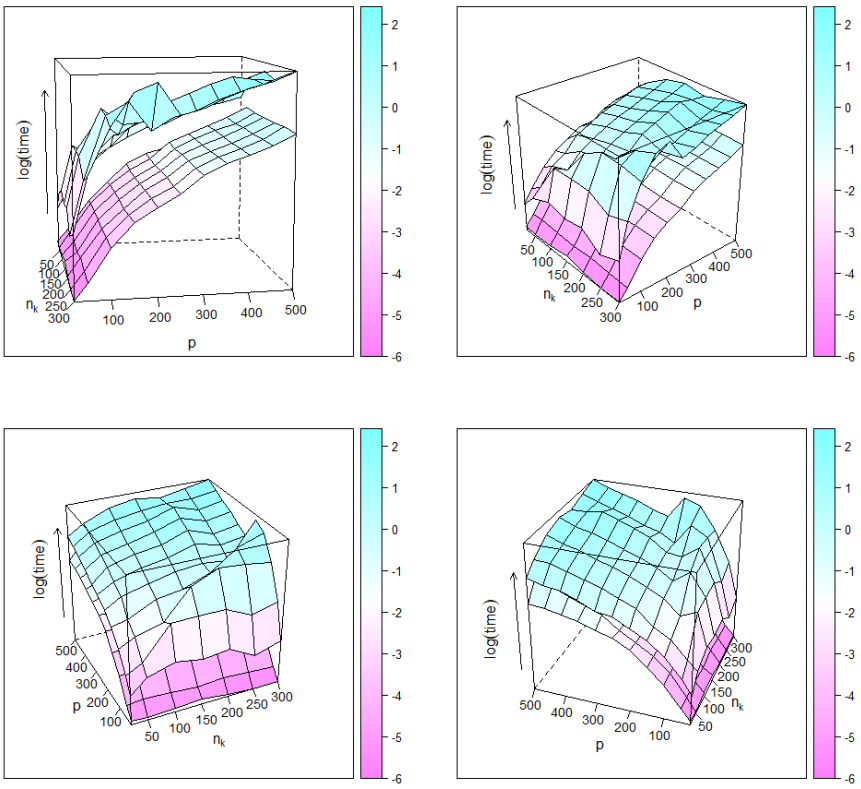


Figure B.5: Four perspectives of the logarithm of the user times in seconds for Lasso (bottom surface in the four graphics) and CSCLasso (top surfaces) models across a grid in n_k and p

Appendix C

Supplementary Material

The choice of the tuning parameters

The purpose of this section is to perform different experiments for setting the tuning parameters involved in our proposal. A number of dependence measures can be found in the literature and the authors have tested some of them. First, for linear dependencies, the (squared) Pearson correlation coefficient, whose evaluation is fast in practice, is the first option to examine. However, other type of relationships (nonlinear) may be present among the predictors, and therefore it also seems natural to consider alternative measures. Another possibility is the (squared) Spearman's rank-order correlation coefficient which is able to detect monotonic relationships between two random variables. In addition, more sophisticated, non-linear dependence coefficients can be explored. Hoeffding D statistic allows to test the independence of two continuous variables X and Y. This coefficient is calculated from a random sample and takes values from $-1/60$ to $1/30$. The Maximal Information coefficient (MIC), the distance correlation coefficient and the mutual information coefficient (MI) also measures nonlinear relationships among random variables. The previous dependence measures, which can be computed by the routines `hoeffd` (D statistic), `mutinfo` (MI), `mine` (MIC) and `dcor` (distance correlation), from the R packages `Hmisc`, `FNN`, `minerva`, and `energy`, respectively, have been tested.

Next, we discuss the choice of the parameters $\{C, S, q\}$. The parameter C will be defined as $\min\{p-1, 100\}$, as commented in Section 4.4.1. Regarding the value of S, which represents the maximum total of combinations for each cut, we tested three possible values, $S = 10,\ S = 25$ and $S = 100$. Finally, as regards the value of the probability q, it should be highlighted that small values of q are associated with more sparsity. Here, we tested the values $q = 0.4,\ q = 0.6$ and $q = 0.8$.

The so-called *Breast Cancer Wisconsin (Diagnostic) Data Set*, *Waveform Database Generator Data Set* (version 2) and *Wine Data Set* shall be considered. They are described in Table 4.3 of Chapter 4. For comparison purposes, the average accuracy (10-fold CV) has been obtained for the above-mentioned number of dependence measures, and values of S and q. The last row of Table C.1 shows the performance rates under the classic NB, that is, the rates obtained when all the features are considered for constructing the NB classifier.

Several conclusions are obtained. First, note that for `breast cancer` the performance rates under the sparse NB are slightly better than that of the classic NB. In addition, the sampling strategy using a value of $S = 10$ seems to produce similar results as if $S = 25$ or $S = 100$ were chosen; therefore, for computational reasons the use of small values is the best choice. Regarding the considered values for the probability q, not many significant differences can be drawn between the three chosen values, from a performance viewpoint. However, as will be seen, the choice of q is influential in the sparsity of the solution. In the case of `waveform` the rates are, for all values of q and S, comparable to those obtained when all features are considered. In the case of `wine` the accuracy under the sparse NB is slightly worse than the results obtained under the classic NB, although this fact is compensated by a better degree of sparsity

Table C.1: Accuracy of the sparse NB (10-fold CV) for `breast cancer`, `waveform` and `wine` datasets

		`breast cancer`			`waveform`			`wine`		
S	*Dependence measure*	$q = 0.4$	$q = 0.6$	$q = 0.8$	$q = 0.4$	$q = 0.6$	$q = 0.8$	$q = 0.4$	$q = 0.6$	$q = 0.8$
10	Pearson	94.62	94.87	95.04	80.14	80.06	81.42	94.50	92.14	95.71
10	Spearman	95.08	95.28	94.95	79.66	80.82	80.78	94.00	92.29	94.00
10	Hoeffding	93.85	94.16	94.71	80.22	80.66	81.06	93.79	93.86	92.86
10	MI	94.55	94.03	95.47	79.98	80.70	81.12	95.14	95.64	96.64
10	MIC	95.10	94.91	95.43	79.24	79.88	80.42	95.57	94.50	94.64
10	Distance Correlation	93.66	94.03	94.21	79.82	80.28	81.82	94.64	95.14	95.14
25	Pearson	94.36	94.73	95.12	80.44	80.06	81.24	93.43	95.14	93.29
25	Spearman	94.90	94.87	95.47	79.92	79.93	81.18	95.64	96.07	94.14
25	Hoeffding	94.74	93.50	94.73	79.90	79.66	80.70	94.07	93.36	93.36
25	MI	93.84	94.18	95.06	79.74	80.32	81.14	95.14	94.57	94.00
25	MIC	95.49	94.56	95.07	79.84	79.66	81.42	93.50	94.57	95.07
25	Distance Correlation	94.91	94.94	95.06	80.30	80.44	81.50	92.64	92.14	91.14
100	Pearson	95.06	94.52	94.05	80.12	80.48	80.68	95.14	91.71	93.43
100	Spearman	94.01	94.81	95.11	80.22	80.46	80.72	91.79	92.36	95.57
100	Hoeffding	96.34	95.61	95.96	80.38	80.98	81.66	93.36	90.00	92.14
100	MI	94.90	95.62	94.94	80.02	80.40	81.68	97.36	95.57	96.71
100	MIC	95.67	95.95	94.93	80.58	80.98	81.30	92.21	91.21	92.86
100	Distance Correlation	94.57	95.99	96.16	79.94	80.48	81.52	95.14	92.93	93.43
	Classic NB		**93.71**			**80.08**			**97.36**	

as will be shown by Table C.2.

It is interesting to note that despite the fact that the Pearson correlation coefficient is not able to detect nonlinear dependencies, the choice of the dependence measure does not substantially alter the performance results. A possible reason for this is that, in many cases, first-order linear approximations according to the Taylor expansion are very close to the nonlinear function. In order to look with more detail into this finding, the dendrograms of `wine` for the first fold in the cross validation procedure have been included below. Even though the dendrograms are different for the different measures, it can be seen how their structure is similar. The sampling strategy is based on a thin grid which does not vary with the dependence measure, and therefore, the resultant combinations per dendrogram do not differ notably. In other words, the results of the sampling strategy will be similar despite the different shape of the dendrograms. In fact, the selected variables across the different dependence measures tend to be the same, as is depicted in Figure C.1, which illustrates the number of times that each variable has been selected in the total of 10 best combinations (one combination per fold) under the choices $S = 10$ and $q = 0.6$. The color intensity of the circles varies with the magnitude: the more intense the color is, the higher the number of times such variable has been included in the best combinations.

We next analyze the sparsity results depending on the tuning parameters. The average number of variables in the selected combinations is shown by Table C.2. In the table, the total number of variables of the dataset is also shown. It can be deduced that the proposed sparse NB leads to a significant reduction in the number of features, especially for `breast cancer` since from the 30 variables characterizing the data, the algorithm selects around 4-7 variables (on average) yielding the most predictive combination. As commented before, for `wine` the algorithm halves the number of variables. The reduction in the case of `waveform` is not so

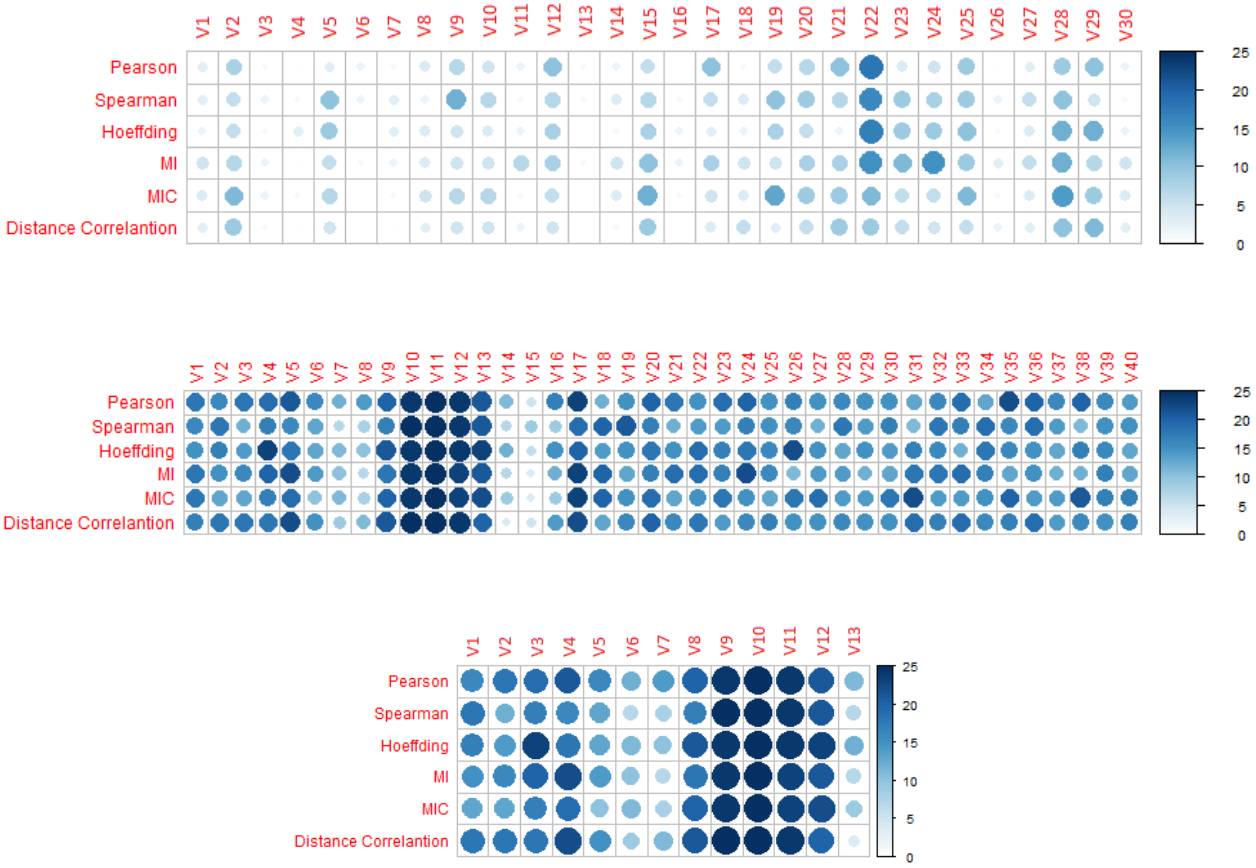


Figure C.1: Variable selection process for `breast cancer`, `waveform` and `wine`, respectively, under the choices of $S = 10$ and $q = 0.6$

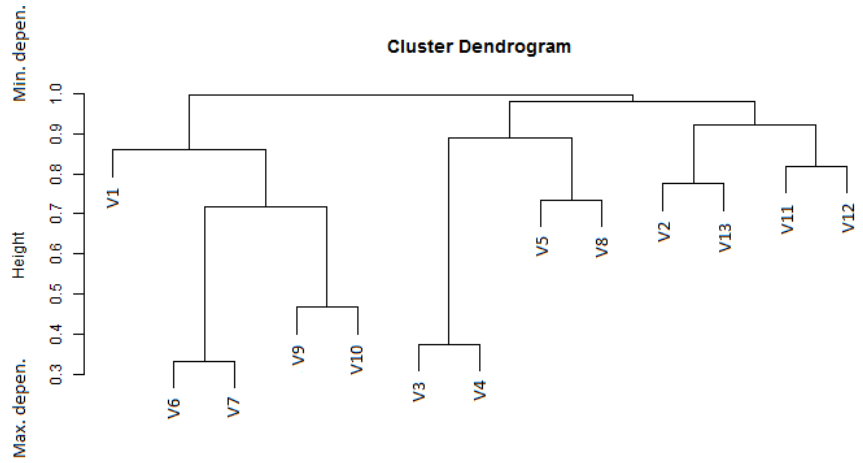
important since features are almost independent.

The effect of q differs between the datasets. In the case of `breast cancer`, a value of $q = 0.4$ produces, as expected, solutions slightly sparser than those obtained under $q = 0.6$ or $q = 0.8$. A choice of $q = 0.6$ also produces, for `waveform`, sparser solutions than under $q = 0.8$. However, the results for $q = 0.4$ are not sparser than those for $q = 0.6$, a phenomenon that can be explained as follows. The algorithm described in Section 4.3 computes the performance of the classifiers constructed from different combinations of features and selects the combination producing the best accuracy. In the case of `waveform`, $q = 0.4$ turns out a very restrictive value since the performance obtained under the different sampled combinations of features is worse than the performance of the combination formed by all features (corresponding to the cut at height equal to zero in the dendrogram). In consequence, since in many folds the selected combination contains all features, the average number of variables increases with respect to the choice $q = 0.6$.

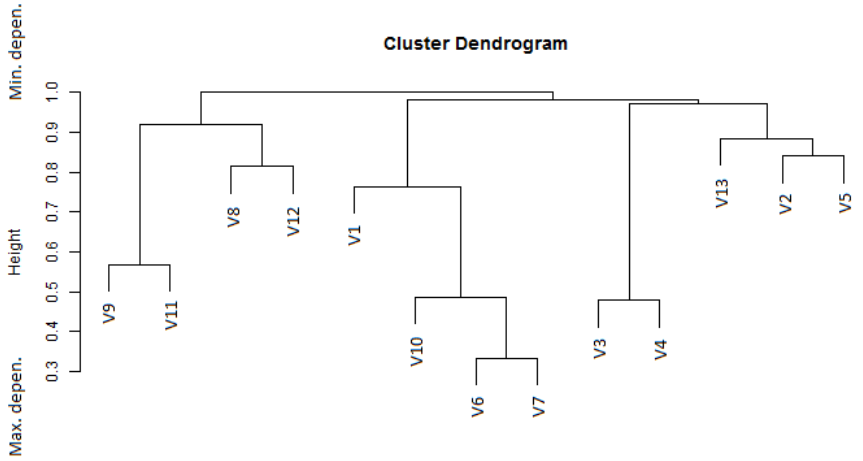
Table C.2: Sparsity results (10-fold CV) for `breast cancer`, `waveform` and `wine` datasets

		`breast cancer`			`waveform`			`wine`		
S	*Dependence measure*	$q = 0.4$	$q = 0.6$	$q = 0.8$	$q = 0.4$	$q = 0.6$	$q = 0.8$	$q = 0.4$	$q = 0.6$	$q = 0.8$
10	Pearson	6.00	5.96	5.68	37.28	27.4	30.24	7.48	6.24	6.60
10	Spearman	5.48	6.84	5.84	30.04	25.00	30.64	7.52	6.00	6.36
10	Hoeffding	6.52	6.20	6.08	33.08	26.20	30.08	6.64	5.80	6.68
10	MI	6.48	7.48	7.80	33.92	25.60	30.00	6.80	5.88	5.96
10	MIC	6.64	7.00	6.24	34.56	25.88	30.36	6.60	6.16	7.16
10	Distance Correlation	4.92	5.24	6.16	32.00	26.52	30.40	7.56	7.24	6.80
25	Pearson	4.96	5.76	5.32	33.08	26.32	30.88	7.32	6.08	6.00
25	Spearman	5.88	6.28	6.56	34.04	27.64	30.44	5.48	5.24	6.00
25	Hoeffding	5.52	5.40	5.36	32.72	25.28	29.40	6.28	6.36	6.20
25	MI	5.88	6.16	8.08	33.48	28.12	31.08	7.00	5.56	5.72
25	MIC	5.60	5.96	6.84	34.08	26.16	29.80	5.64	5.32	6.92
25	Distance Correlation	4.68	5.12	4.76	32.80	26.28	30.60	5.56	5.60	6.20
100	Pearson	4.60	4.60	4.84	32.84	24.04	30.80	6.08	5.04	4.88
100	Spearman	5.16	4.68	4.92	34.04	24.68	30.60	6.20	5.32	5.64
100	Hoeffding	5.40	5.40	4.84	34.76	27.56	31.40	6.04	5.40	5.28
100	MI	5.32	6.44	6.44	36.84	25.40	30.76	5.60	5.52	6.04
100	MIC	4.88	5.52	6.04	35.32	24.64	30.40	5.32	5.00	5.44
100	Distance Correlation	4.88	4.28	4.68	35.40	26.44	30.92	5.08	4.68	5.16
	Classic NB		**30.00**			**40.00**			**13.00**	

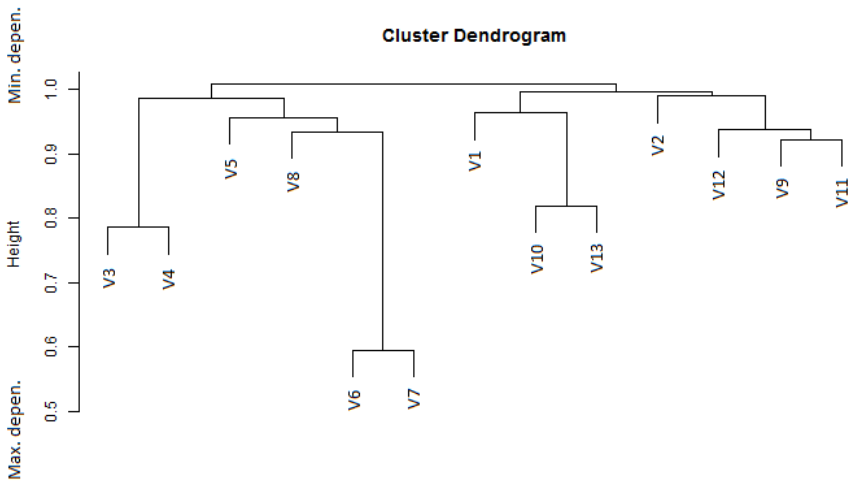
Dendrograms of the *Wine* database for the first fold in the cross validation procedure



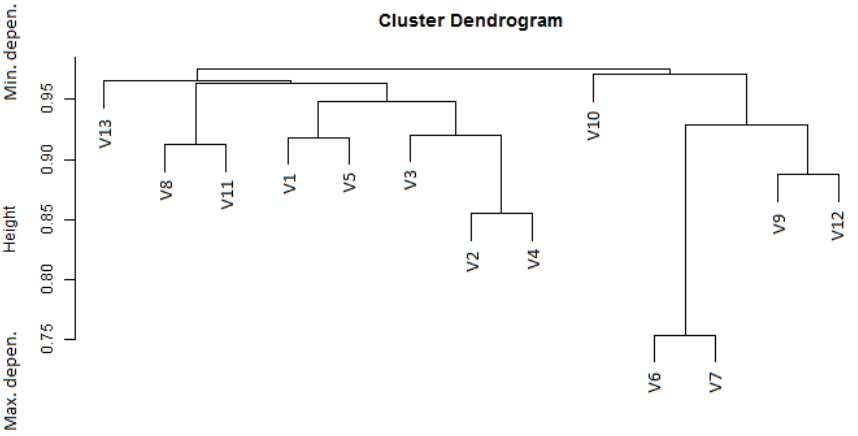
(a) Pearson



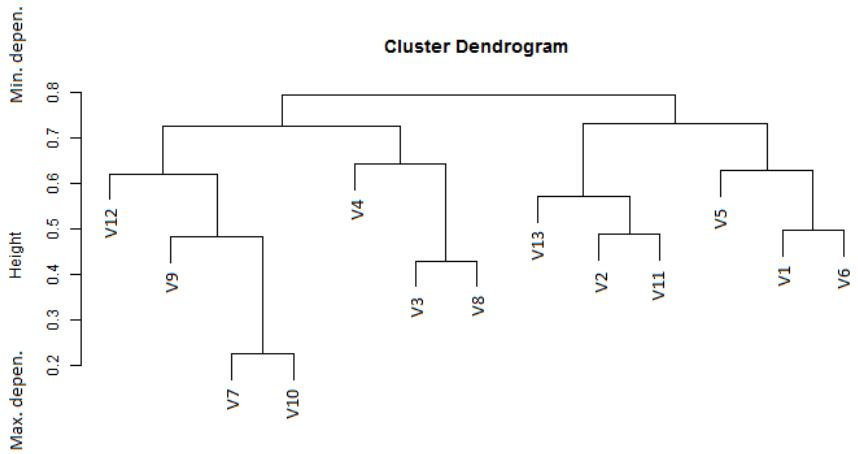
(b) Spearman



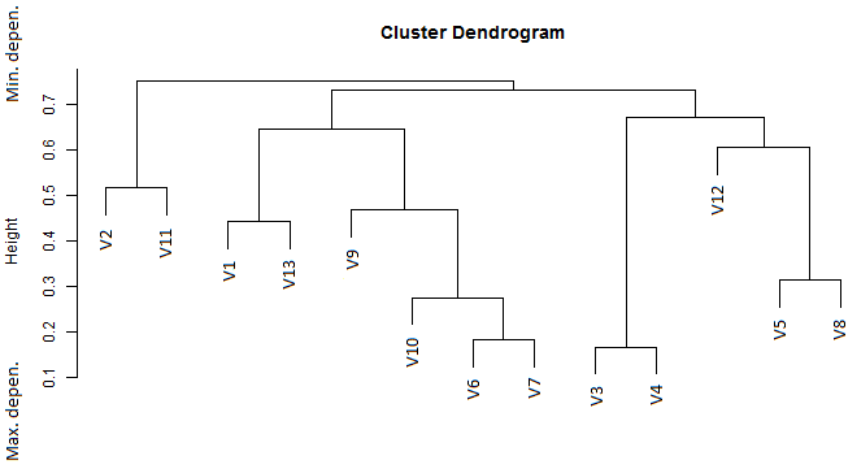
(c) Hoeffding



(d) MI



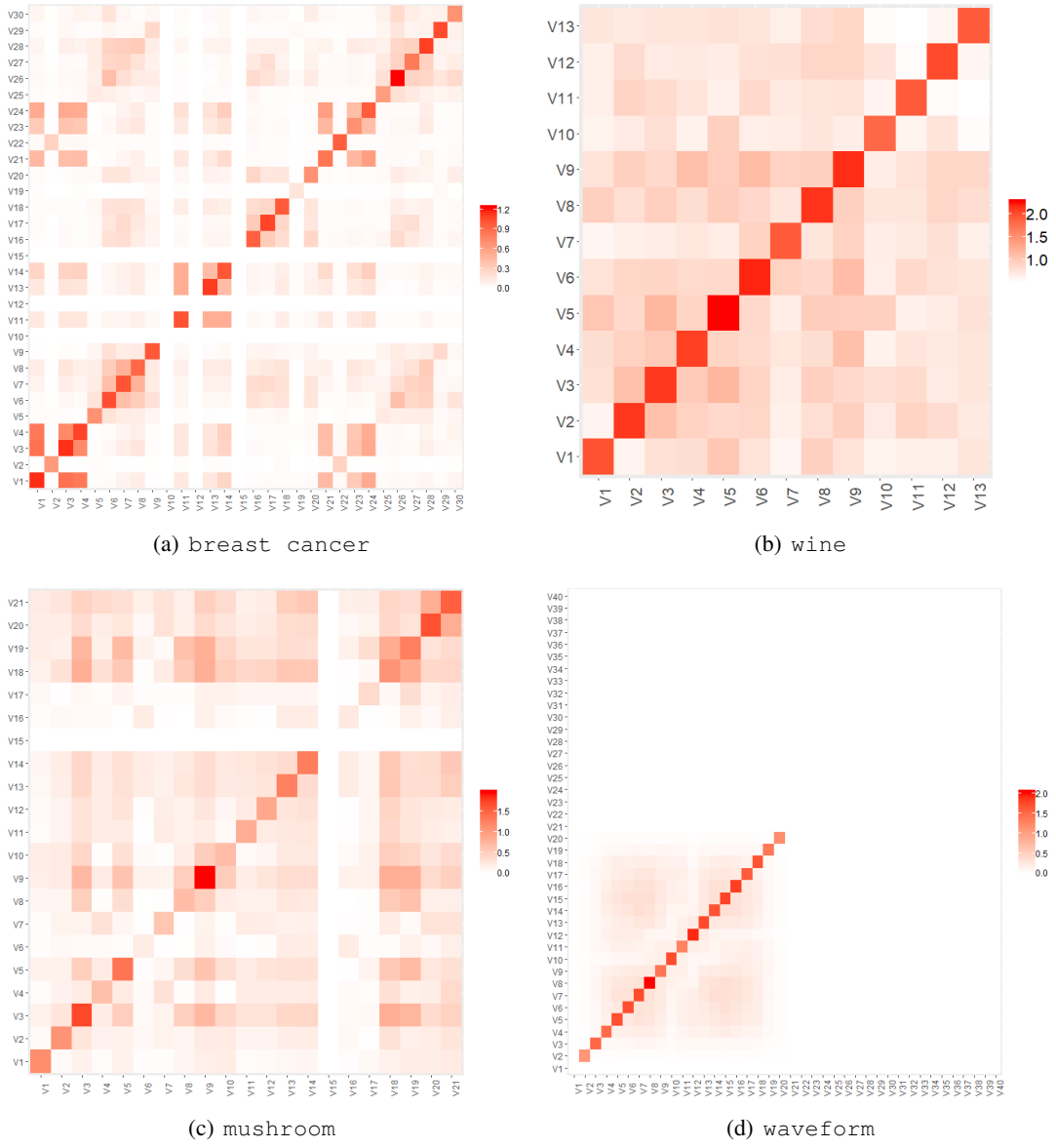
(e) MIC



(f) Distance Correlation

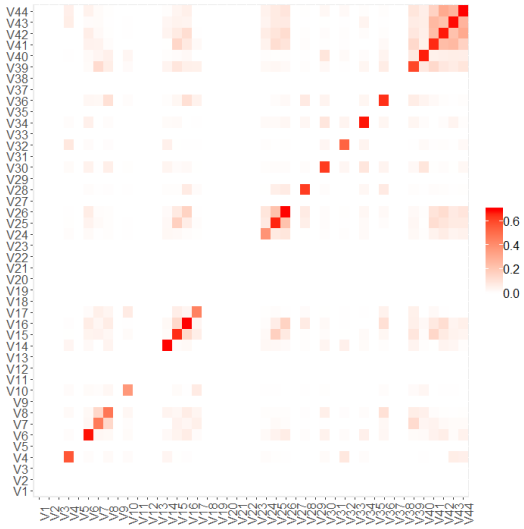
Heatmaps of the databases for the first fold in the cross validation procedure using MI dependence measure

Figure C.2: Heatmaps

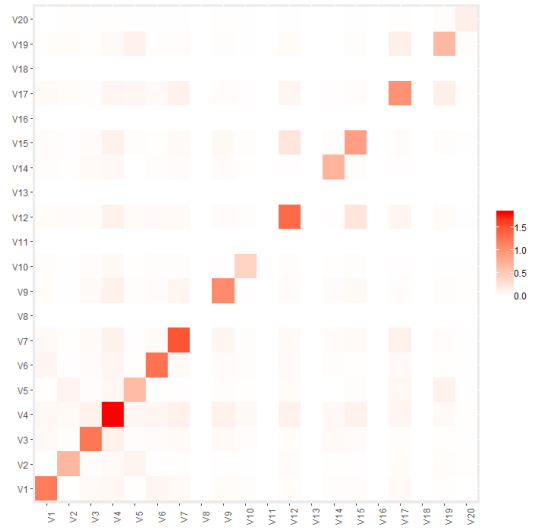


(a) `breast cancer` (b) `wine`

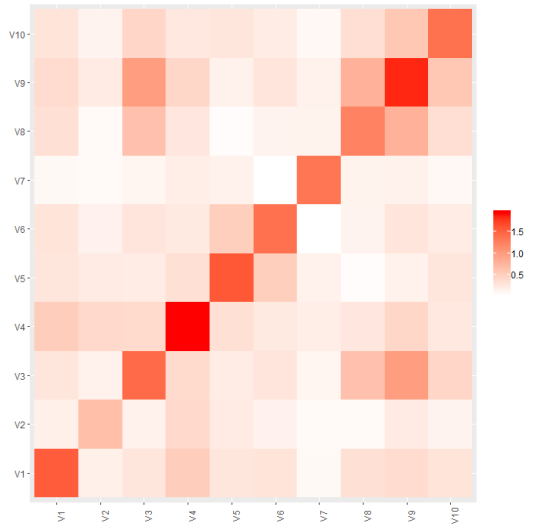
(c) `mushroom` (d) `waveform`



(e) SPECTF



(f) german



(g) page blocks

References

Abdous, B., Fougères, A., and Ghoudi, K. (2005). Extreme behaviour for bivariate elliptical distributions. *Canadian Journal of Statistics*, 33(3):317–334.

Akaike, H. (1998). *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY.

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011). KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17:255–287.

Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C., and Herrera, F. (2009). KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. *Soft Computing*, 13(3):307–318.

Anderson, T. (1992). Nonnormal multivariate distributions: Inference based on elliptically contoured distributions. Technical report, Technical report No. 28. Department of Statistics, Sanford University, California.

Anderson, T. (2003). *An introduction to Multivariate Statistical Analysis*. John Wiley & Sons.

Anderson, T. and Fang, K. (1982). Distributions of quadratic forms and Cochran's theorem for elliptically contoured distributions and their applications. Technical report, Technical report No. 53. Department of Statistics, Sanford University, California.

Anderson, T., Fang, K., and Hsu, H. (1986). Maximum-likelihood estimates and likelihood-ratio criteria for multivariate elliptically contoured distributions. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, pages 55–59.

Andrews, D. and Mallows, C. (1974). Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(1):99–102.

Andrews, R. W., Berger, J. O., and Smith, M. H. (1993). Bayesian Estimation of Fuel Economy Potential Due to Technology Improvements. In Gatsonis, C., Hodges, J. S., Kass, R. E., and

Singpurwalla, N. D., eds., *Case Studies in Bayesian Statistics*, pages 1–77, New York, NY. Springer New York. https://doi.org/10.1007/978-1-4612-2714-4_1.

"ann" Ratanamahatana, C. and Gunopulos, D. (2003). Feature selection for the naive bayesian classifier using decision trees. *Applied Artificial Intelligence*, 17(5-6):475–487.

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

Baena, D., Castro, J., and Frangioni, A. (2020). Stabilized Benders Methods for Large-Scale Combinatorial Optimization, with Application to Data Privacy. *Management Science*, 66(7):3051–3068.

Barbu, A. and Zhu, S.-C. (2020). *Monte Carlo Methods*. Springer.

Barnard, J., McCulloch, R., and Meng, X. (2000). Modeling covariance matrices in terms of standard deviations and correlations with applications to shrinkage. *Statistica Sinica*, 10:1281–1311.

Basu, S. and Chib, S. (2003). Marginal likelihood and bayes factors for dirichlet process mixture models. *Journal of the American Statistical Association*, 98:224–235.

Benítez-Peña, S., Blanquero, R., Carrizosa, E., and Ramírez-Cobo, P. (2019). Cost-sensitive Feature Selection for Support Vector Machines. *Computers & Operations Research*, 106:169 – 178.

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

Benítez-Peña, S., Carrizosa, E., Guerrero, V., Jiménez-Gamero, M. D., Martín-Barragán, B., Molero-Río, C., Ramírez-Cobo, P., Romero Morales, D., and Sillero-Denamiel, M. R. (2021). On Sparse Ensemble Methods: An Application to Short-Term Predictions of the Evolution of COVID-19. Forthcoming in *European Journal of Operational Research*, `https://doi.org/10.1016/j.ejor.2021.04.016`.

Berger, J. O. (2013). *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics, 2nd edition.

Berkane, M. and Bentler, P. (1987). Characterizing parameters of multivariate elliptical distributions. *Communications in Statistics-Simulation and Computation*, 16(1):193–198.

Berkane, M. and Bentler, P. (1990). Mardia's coefficient of kurtosis in elliptical populations. *Acta Mathematicae Applicatae Sinica, English Series*, 6(4):289–294.

Berkane, M., Kano, Y., and Bentler, P. (1994). Pseudo maximum likelihood estimation in elliptical theory: effects of misspecification. *Computational Statistics & Data Analysis*, 18(2):255–267.

Bermejo, P., Gámez, J. A., and Puerta, J. M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowledge-Based Systems*, 55:140–147.

Bertsimas, D., Pauphilet, J., and Parys, B. V. (2020). Sparse Regression: Scalable Algorithms and Empirical Performance. *Statistical Science*, 35(4):555 – 578.

Birgin, E. and Martínez, J. (2008). Improving ultimate convergence of an augmented lagrangian method. *Optimization Methods and Software*, 23(2):177–195.

Blanquero, R., Carrizosa, E., Jiménez-Cordero, A., and Martín-Barragán, B. (2019). Variable selection in classification for multivariate functional data. *Information Sciences*, 481:445 – 462.

Blanquero, R., Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2020). Sparsity in optimal randomized classification trees. *European Journal of Operational Research*, 284(1):255 – 272.

Blanquero, R., Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2021a). Optimal randomized classification trees. *Computers & Operations Research*, 132:105281.

Blanquero, R., Carrizosa, E., Ramírez-Cobo, P., and Sillero-Denamiel, M. R. (2021b). A cost-sensitive constrained lasso. *Advances in Data Analysis and Classification*, 15:121–158.

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

Boullé, M. (2004). Khiops: A Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55(1):53–69.

Boullé, M. (2006). MODL: A Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165.

Boullé, M. (2007). Compression-based Averaging of Selective Naive Bayes Classifiers. *Journal of Machine Learning Research*, 8:1659–1685.

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., and Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In Nédellec, C. and Rouveirol, C., eds., *Machine Learning: ECML-98*, pages 131–136, Berlin, Heidelberg. Springer Berlin Heidelberg.

Branco, M., Bolfarine, H., Iglesias, P., and Arellano-Valle, R. (2000). Bayesian analysis of the calibration problem under elliptical distributions. *Journal of Statistical Planning and Inference*, 90(1):69–85.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Bühlmann, P. and Van-De Geer, S. (2011). *Statistics for high-dimensional data*. Springer.

Cai, A., Tsay, R., and Chen, R. (2009). Variable selection in linear regression with many predictors. *Journal of Computational and Graphical Statistics*, 18(3):573–591.

Cambanis, S., Huang, S., and Simons, G. (1981). On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis*, 11(3):368–385.

Cao, P., Zhao, D., and Zaïane, O. R. (2013). A PSO-Based Cost-Sensitive Neural Network for Imbalanced Data Classification. In Li, J., Cao, L., Wang, C., Tan, K. C., Liu, B., Pei, J., and Tseng, V. S., eds., *Trends and Applications in Knowledge Discovery and Data Mining*, pages 452–463, Berlin, Heidelberg. Springer Berlin Heidelberg.

Carrizosa, E. and Guerrero, V. (2014). Biobjective sparse principal component analysis. *Journal of Multivariate Analysis*, 132:151–159.

Carrizosa, E., Martín-Barragán, B., and Romero Morales, D. (2008). Multi-group support vector machines with measurement costs: A biobjective approach. *Discrete Applied Mathematics*, 156:950–966.

Carrizosa, E., Mortensen, L. H., Romero Morales, D., and Sillero-Denamiel, M. R. (2020). On linear regression models with hierarchical categorical variables. Technical report, IMUS, Sevilla, Spain, `https://www.researchgate.net/publication/341042405_On_linear_regression_models_with_hierarchical_categorical_variables`.

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*, 18(2):244–259.

Carrizosa, E. and Romero Morales, D. (2001). Combining minsum and minmax: A goal programming approach. *Operations Research*, 49(1):169–174.

Carrizosa, E. and Romero Morales, D. (2013). Supervised classification and mathematical optimization. *Computers and Operations Research*, 40(1):150–165.

Cerda, P., Varoquaux, G., and Kégl, B. (2018). Similarity encoding for learning with dirty categorical variables. *Machine Learning*, 107(8):1477–1494.

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

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

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.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, 7:1–30.

Díaz-García, J.A .and Leiva-Sánchez, V. (2005). A new family of life distributions based on the elliptically contoured distributions. *Journal of Statistical Planning and Inference*, 128(2):445–457.

Dickey, J. and Chen, C. (1985). Direct subjective-probability modelling using ellipsoidal distributions. *Bayesian statistics*, 2:157–182.

Director, H., Wiel, S. V., and Gattiker, J. (2017). *SALTSampler: Efficient Sampling on the Simplex*. R package version 1.1.0.

Domingos, P. and Pazzani, M. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 105–112. Morgan Kaufmann.

Domingos, P. and Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995). Wavelet shrinkage: Asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):301–369.

Dougherty, J., Kohavi, R., and Sahami, M. (1995). Supervised and Unsupervised Discretization of Continuous Features. In Prieditis, A. and Russell, S., eds., *Machine Learning Proceedings 1995*, pages 194 – 202.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference of artificial intelligence*, pages 973–978. Seattle: Morgan Kaufmann.

Escobar, M. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588.

European Commission (2008). *NACE Rev. 2 - Statistical classification of economic activites in the European Community*. Luxembourg: Office for Official Publications of the European Communities. https://ec.europa.eu/eurostat/documents/3859598/5902521/KS-RA-07-015-EN.PDF.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360.

Fang, K. and Anderson, T. (1990). *Statistical inference in elliptically contoured and related distributions*. Allerton Press New York.

Fang, K., Kotz, S., and Ng, K. (1990). *Symmetric Multivariate and related distributions*. Monographs on Statistics and Applied Probability, Springer US.

Fang, K. and Li, R. (1999). Bayesian statistical inference on elliptical matrix distributions. *Journal of multivariate analysis*, 70(1):66–85.

Fayyad, U. M. and Irani, K. B. (1993). Multi-interval discretization of continuous valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1029. Morgan-Kaufmann.

Feng, G., Guo, J., Jing, B.-Y., and Sun, T. (2015). Feature subset selection using naive Bayes for text classification. *Pattern Recognition Letters*, 65:109–115.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230.

Ferguson, T. S. (1974). Prior distributions on spaces of probability measures. *Annals of Statistics*, 2(4):615–629.

Fortunati, S., Renaux, A., and Pascal, F. (2020). Robust Semiparametric Efficient Estimators in Complex Elliptically Symmetric Distributions. *IEEE Transactions on Signal Processing*, 68:5003–5015.

Frahm, G., Junker, M., and Szimayer, A. (2003). Elliptical copulas: applicability and limitations. *Statistics & Probability Letters*, 63(3):275–286.

Freitas, A., Costa-Pereira, A., and Brazdil, P. (2007). Cost-sensitive decision trees applied to medical data. In Song, I. Y., Eder, J., and Nguyen, T. M., eds., *Data Warehousing and Knowledge Discovery*, pages 303–312, Berlin, Heidelberg. Springer Berlin Heidelberg.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*. Springer-Verlag, Heidelberg.

Gaines, B. R., Kim, J., and Zhou, H. (2018). Algorithms for Fitting the Constrained Lasso. *Journal of Computational and Graphical Statistics*, 27(4):861–871.

Garside, M. J. (1965). The Best Sub-Set in Multiple Regression Analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 14(2-3):196–200.

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American Statistical Association*, 95(452):1300–1304.

Gelfand, A. E. and Smith, A. F. (1990). Sampling-based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, 85(410):398–409.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

George, E. and McCulloch, R. (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889.

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.

Gilks, W. R., Roberts, G. O., and George, E. I. (1994). Adaptive direction sampling. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 43(1):179–189.

Gómez, H. and Venegas, O. (2008). Erratum to: A new family of slash-distributions with elliptical contours? [Statistics & Probability Letters 77 (2007) 717–725]. *Statistics & Probability Letters*, 78(14):2273–2274.

Gómez, H. W., Quintana, F. A., and Torres, F. J. (2007). A new family of slash-distributions with elliptical contours. *Statistics & probability letters*, 77(7):717–725.

Gotway, C. A. and Young, L. J. (2002). Combining incompatible spatial data. *Journal of the American Statistical Association*, 97(458):632–648.

Griva, A., Bardaki, C., Pramatari, K., and Papakiriakopoulos, D. (2018). Retail business analytics: Customer visit segmentation using market basket data. *Expert Systems with Applications*, 100:1 – 16.

Guan, G., Guo, J., and Wang, H. (2014). Varying Naïve Bayes Models With Applications to Classification of Chinese Text Documents. *Journal of Business & Economic Statistics*, 32(3):445–456.

Gui, J., Sun, Z., Ji, S., Tao, D., and Tan, T. (2017). Feature selection based on structured sparsity: A comprehensive study. *IEEE Transactions on Neural Networks and Learning Systems*, 28(7):1490–1507.

Gupta, A., Varga, T., and Bodnar, T. (2013). *Elliptically contoured models in statistics and portfolio theory*. Springer.

Gurobi Optimization, L. (2018). Gurobi optimizer reference manual. `http://www.gurobi.com`.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature Extraction. Foundations and Applications. Studies in Fuzzines and Soft Computing*, volume 207. Springer.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 359–366, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hand, D. and Henley, W. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541.

Hand, D. J. and Yu, K. (2001). Idiot's Bayes - Not So Stupid After All? *International Statistical Review*, 69(3):385–398.

Hanson, T. E. (2006). Modeling censored lifetime data using a mixture of gammas baseline. *Bayesian Analysis*, 1(3):575–594.

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.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, NY.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity*. New York: Chapman and Hall/CRC.

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

Hoeffding, W. (1948). A Non-Parametric Test of Independence. *The Annals of Mathematical Statistics*, 19(4):546–557.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hogg, R. V., McKean, J., and Craig, A. T. (2005). *Introduction to Mathematical Statistics*. Pearson Education.

Hu, Q., Zeng, P., and Lin, L. (2015). The dual and degrees of freedom of linearly constrained generalized lasso. *Computational Statistics & Data Analysis*, 86:13 – 26.

Insua, D., Ruggeri, F., and Wiper, M. (2012). *Bayesian analysis of stochastic process models*. New York: Wiley.

James, G. M., Paulson, C., and Rusmevichientong, P. (2020). Penalized and Constrained Optimization: An Application to High-Dimensional Website Advertising. *Journal of the American Statistical Association*, 115(529):107–122.

Jara, A., Quintana, F., and San Martín, E. (2008). Linear mixed models with skew-elliptical distributions: A bayesian approach. *Computational statistics & data analysis*, 52(11):5033–5045.

Jiang, L., Cai, Z., Zhang, H., and Wang, D. (2012). Not so greedy: Randomly selected naive bayes. *Expert Systems with Applications*, 39(12):11022 – 11028.

Jiang, L., Zhang, H., Cai, Z., and Su, J. (2005). Evolutional naive Bayes. In *Proceedings of the 1st international symposium on intelligent computation and its applications*, ISICA 2005, pages 344–350. China University of Geosciences Press.

Jiang, L., Zhang, L., Li, C., and Wu, J. (2019). A correlation-based feature weighting filter for naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, 31(2):201–213.

Jiang, L., Zhang, L., Yu, L., and Wang, D. (2019). Class-specific attribute weighted naive Bayes. *Pattern Recognition*, 88:321 – 330.

Kano, Y., Berkane, M., and Bentler, P. M. (1993). Statistical inference based on pseudo-maximum likelihood estimators in elliptical populations. *Journal of the American Statistical Association*, 88(421):135–143.

Katz-Gerro, T. and López Sintas, J. (2019). Mapping circular economy activities in the European Union: Patterns of implementation and their correlates in small and medium-sized enterprises. *Business Strategy and the Environment*, 28(4):485–496.

Kelker, D. (1970). Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 419–430.

Kinney, J. B., Murugan, A., Callan, C. G., and Cox, E. C. (2010). Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proceedings of the National Academy of Sciences*, 107(20):9158–9163.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

Kouno, T., de Hoon, M., Mar, J. C., Tomaru, Y., Kawano, M., Carninci, P., Suzuki, H., Hayashizaki, Y., and Shin, J. W. (2013). Temporal dynamics and transcriptional control using single-cell gene expression analysis. *Genome Biology*, 14(10):R118.

Kuncheva, L. I. (2006). On the optimality of Naïve Bayes with dependent binary features. *Pattern Recognition Letters*, 27(7):830–837.

Kursa, M. B. and Rudnicki, W. R. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software*, 36(11):1–13.

Lange, K. and Sinsheimer, J. (1993). Normal/independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics*, 2(2):175–198.

Langley, P. and Sage, S. (1994). Induction of selective bayesian classifiers. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 399–406.

Lawless, J. F. (1982). *Statistical Models and Methods for Lifetime Data*. New York: Wiley.

LeBlanc, M. and Tibshirani, R. (1998). Monotone shrinkage of trees. *Journal of Computational and Graphical Statistics*, 7(4):417–433.

Lee, W., Jun, C.-H., and Lee, J.-S. (2017). Instance categorization by support vector machines to adjust weights in adaboost for imbalanced data classification. *Information Sciences*, 381(Supplement C):92 – 103.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(42).

Li, X., Wang, Y., and Ruiz, R. (2020). A survey on sparse learning models for feature selection. Forthcoming in *IEEE Transactions on Cybernetics. https://ieeexplore.ieee.org/document/9088292*.

Li, X.-B. and Sarkar, S. (2009). Against classification attacks: A decision tree pruning approach to privacy protection in data mining. *Operations Research*, 57(6):1496–1509.

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

Liebscher, E. (2005). A semiparametric density estimator based on elliptical distributions. *Journal of multivariate analysis*, 92(1):205–225.

Lin, D., Foster, D. P., and Ungar, L. H. (2011). VIF Regression: A Fast Regression Algorithm for Large Data. *Journal of the American Statistical Association*, 106(493):232–247.

Lindskog, F., Mcneil, A., and Schmock, U. (2003). Kendall's tau for elliptical distributions. In *Credit Risk*, pages 149–156. Springer.

Linfoot, E. (1957). An Informational Measure of Correlation. *Information and Control*, 1(1):85–89.

Ling, C. X., Yang, Q., Wang, J., and Zhang, S. (2004). Decision trees with minimal costs. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 69, New York, NY, USA.

Liu, H., Hussain, F., Tan, C. L., and Dash, M. (2002). Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, 6(4):393–423.

Lord, R. (1954). The use of the Hankel transform in statistics I. General theory and examples. *Biometrika*, 41(1/2):44–55.

Lu, R., Zhu, H., Liu, X., Liu, J. K., and Shao, J. (2014). Toward efficient and privacy-preserving computing in big data era. *IEEE Network*, 28(4):46–50.

Maldonado, S., Carrizosa, E., and Weber, R. (2015). Kernel penalized k-means: A feature selection method based on kernel k-means. *Information Sciences*, 322:150–160.

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.

Maruyama, Y. and Seo, T. (2003). Estimation of moment parameter in elliptical distributions. *Journal of the Japan Statistical Society*, 33(2):215–229.

McCallum, A. and Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI-98 Workshop on Learning for Text Categorization*, volume 752, pages 41–48.

Minnier, J., Yuan, M., Liu, J. S., and Cai, T. (2015). Risk Classification With an Adaptive Naive Bayes Kernel Machine Model. *Journal of the American Statistical Association*, 110(509):393–404.

Mukherjee, S. and Sharma, N. (2012). Intrusion Detection using Naive Bayes Classifier with Feature Reduction. *Procedia Technology*, 4:119 – 128.

Müller, P. (1991). A generic approach to posterior integration and Gibbs sampling. Technical report, Purdue University, Department of Statistics.

Müller, P. and Quintana, F. A. (2004). Nonparametric bayesian data analysis. *Statistical Science*, 19(1):95–110.

Niekerk, J. V., Bekker, A., Arashi, M., and Roux, J. (2015). Subjective Bayesian analysis of the elliptical model. *Communications in Statistics - Theory and Methods*, 44(17):3738–3753.

North, D. W. (2016). *Decision Analytic and Bayesian Uncertainty Quantification for Decision Support*, pages 1–39. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-11259-6_41-1.

Ollier, E. and Viallon, V. (2017). Regression modelling on stratified data with the lasso. *Biometrika*, 104(1):83–96.

Osiewalski, J. and Steel, M. (1993). Robust Bayesian inference in elliptical regression models. *Journal of Econometrics*, 57(1-3):345–363.

Owen, J. and Rabinovitch, R. (1983). On the class of elliptical distributions and their applications to the theory of portfolio choice. *The Journal of Finance*, 38(3):745–752.

Peng, L., Zhang, H., Yang, B., and Chen, Y. (2014). A new approach for imbalanced data classification based on data gravitation. *Information Sciences*, 288(Supplement C):347 – 373.

Prati, R. C., Batista, G. E. A. P. A., 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.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramírez-Cobo, P., Lillo, R. E., Wilson, S., and Wiper, M. P. (2010). Bayesian inference for double Pareto lognormal queues. *Annals of Applied Statistics*, 4(3):1533–1557.

Redmond, M. and Baveja, A. (2002). A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660 – 678.

Reshef, D. N., Reshef, Y. A., Finucane, H. K., Grossman, S. R., McVean, G., Turnbaugh, P. J., Lander, E. S., Mitzenmacher, M., and Sabeti, P. C. (2011). Detecting Novel Associations in Large Data Sets. *Science*, 334(6062):1518–1524.

Rezaei, M., Cribben, I., and Samorani, M. (2018). A clustering-based feature selection method for automatically generated relational attributes. *Annals of Operations Research. https://doi.org/10.1007/s10479-018-2830-2*.

Rippner, N. (2017). Cancer Trials. Retrieved from `http://data.world/exercises/linear-regression-exercise-1/workspace/file?filename=cancer_reg.csv`.

Robert, C. and Casella, G. (2013). *Monte Carlo Statistical Methods*. Springer, New York, NY.

Rockafellar, R. T. (1972). *Convex analysis*. Princeton University Press.

Rogers, W. and Tukey, J. (1972). Understanding some long-tailed distributions. *Statistica Neerlandia*, 26:211–226.

Romei, A. and Ruggieri, S. (2014). A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, 29(5):582–638.

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

Schoenberg, I. (1938). Metric spaces and completely monotone functions. *Annals of Mathematics*, 39(4):811–841.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). *Lectures on stochastic programming: modeling and theory*. SIAM.

Sharpee, T., Rust, N. C., and Bialek, W. (2004). Analyzing Neural Responses to Natural Signals: Maximally Informative Dimensions. *Neural Computation*, 16(2):223–250.

Sherali, H., Hobeika, A., and Jeenanunta, C. (2009). An optimal constrained pruning strategy for decision trees. *INFORMS Journal on Computing*, 21(1):49–61.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2011). Regularization paths for Cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427 – 437.

Spiegelhalter, D., Best, N., Carlin, B., and van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, 64:583–639.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A., and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate: II. Radical prostatectomy treated patients. *Journal of Urology*, 141(5):1076–1083.

Su, X., Wang, M., and Fan, J. (2004). Maximum likelihood regression trees. *Journal of Computational and Graphical Statistics*, 13(3):586–598.

Sun, Y., Kamel, M. S., Wong, A. K., and Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358 – 3378.

Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23:687–719.

Székely, G. J., Rizzo, M. L., and Bakirov, N. K. (2007). Measuring and Testing Dependence by Correlation of Distances. *The Annals of Statistics*, 35(6):2769–2794.

Tang, B., He, H., Baggenstoss, P. M., and Kay, S. (2016a). A Bayesian Classification Approach Using Class-Specific Features for Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1602–1606.

Tang, B., Kay, S., and He, H. (2016b). Toward Optimal Feature Selection in Naive Bayes for Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2508–2521.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tibshirani, R. J. and Taylor, J. (2011). The solution path of the generalized Lasso. *The Annals of Statistics*, 39(3):1335–1371.

Titi, G. and Marshall, D. (1996). The ARPA/NAVY Mountaintop Program: adaptive signal processing for airborne early warning radar. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 2, pages 1165–1168.

Torres-Barrán, A., Alaíz, C. M., and Dorronsoro, J. R. (2018). ν-SVM solutions of constrained Lasso and Elastic net. *Neurocomputing*, 275:1921 – 1931.

Turhan, B. and Bener, A. (2009). Analysis of Naive Bayes' assumptions on software fault data: An empirical study. *Data & Knowledge Engineering*, 68(2):278–290.

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409.

Turney, P. (2000). Types of cost in inductive concept learning. In *Proceedings of the workshop on cost-sensitive learning at the 17th international conference on machine learning, Stanford University, California*.

U.S. Department (1992a). U.S. Department of Commerce, Bureau of the Census, Census Of Population And Housing 1990 United States: Summary Tape File 1a & 3a (Computer Files), U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan.

U.S. Department (1992b). U.S. Department of Justice, Bureau of Justice Statistics, Law Enforcement Management And Administrative Statistics (Computer File) U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan.

U.S. Department (1995). U.S. Department of Justice, Federal Bureau of Investigation, Crime in the United States (Computer File).

Vincent, M. and Hansen, N. R. (2014). Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, 71:771 – 786.

Witten, D. M., Shojaie, A., and Zhang, F. (2014). The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping. *Technometrics*, 56(1):112–122.

Wolfson, J., Bandyopadhyay, S., Elidrisi, M., Vazquez-Benitez, G., Vock, D. M., Musgrove, D., Adomavicius, G., Johnson, P. E., and O'Connor, P. J. (2015). A Naive Bayes machine learning approach to risk prediction using censored, time-to-event data. *Statistics in Medicine*, 34(21):2941–2957.

Yu, G. and Liu, Y. (2016). Sparse regression incorporating graphical structure among predictors. *Journal of the American Statistical Association*, 111(514):707–720.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh international conference on Knowledge discovery and data mining*, pages 204–213.

Zhang, H. (2004). The optimality of Naive Bayes. In Barr, V. and Markov, Z., eds., *Proceedings of the Seventeenth International Florida Articial Intelligence Research Society Conference*, pages 562–567.

Zhang, H., Jiang, L., and Yu, L. (2020). Class-specific attribute value weighting for Naive Bayes. *Information Sciences*, 508:260–274.

Zhang, M., Peña, J., and Robles, V. (2009). Feature selection for multi-label naive Bayes classification. *Information Sciences*, 179(456):3218–3229.

Zhang, X., Boscardin, W., and Belin, T. (2006). Sampling correlation matrices in Bayesian models with correlated latent variables. *Journal of Computational and Graphical Statistics*, 15:880–896.

Zhi-Hua Zhou and Xu-Ying Liu (2006). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.