INTERNATIONAL DOCTORAL
SCHOOL OF THE USC

Laura
Freijeiro González

PhD Thesis

New covariates selection approaches in high dimensional or functional regression models

Santiago de Compostela, 2023

Doctoral Programme in Statistics and Operations Research

CENTRO INTERNACIONAL DE ESTUDOS
DE DOUTORAMENTO E AVANZADOS
DA USC (CIEDUS)

TESE DE DOUTORAMENTO

New covariates selection approaches in high dimensional or functional regression models

Laura Freijeiro González

ESCOLA DE DOUTORAMENTO INTERNACIONAL

PROGRAMA DE DOUTORAMENTO EN ESTATÍSTICA E INVESTIGACIÓN OPERATIVA

SANTIAGO DE COMPOSTELA

ANO 2023

DECLARACIÓN DO AUTOR DA TESE

New covariates selection approaches in high dimensional or functional regression models

Laura Freijeiro González

Presento a miña tese, seguindo o procedemento adecuado ao Regulamento, e declaro que:

1) A tese abarca os resultados da elaboración do meu traballo.
2) No seu caso, na tese faise referencia ás colaboracións que tivo este traballo.
3) A tese é a versión definitiva presentada para a súa defensa e coincide coa versión enviada en formato electrónico.
4) Confirmo que a tese non incorre en ningún tipo de plaxio doutros autores nin de traballos presentados por min para a obtención doutros títulos.

En Santiago de Compostela, 17 de marzo de 2023

Asdo. Laura Freijeiro González

DECLARACIÓN DO DIRECTOR DA TESE

New covariates selection approaches in high dimensional or functional regression models

Wenceslao González Manteiga

Manuel Febrero Bande

INFORMAN:

Que a presente tese se corresponde co traballo realizado por Laura Freijeiro González, baixo a nosa dirección, e autorizamos a súa presentación, considerando que reúne os requisitos esixidos no Regulamento de Estudos de Doutoramento da USC, e que como directores desta non incorremos nas causas de abstención establecidas na lei 40/2015.

En Santiago de Compostela, 17 de marzo de 2023

Asdo. Wenceslao González Manteiga

Asdo. Manuel Febrero Bande

"Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time." - Thomas A. Edison

Preface

Si hace unos años me hubiesen preguntado cómo me imaginaba mi futuro, desde luego, el hacer una tesis doctoral no entraría en mis planes. No sé muy bien como acabé subida a esta montaña rusa que es la investigación. Sin embargo, sí que tengo la certeza de que si no fuese por mucha gente que me ha acompañado durante todos estos años, me habría bajado en la primera curva. Por y para ellos van estos agradecimientos.

Me gustaría empezar agradeciendo a mis tutores, Wenceslao González Manteiga y Manuel Febrero Bande, por todo el camino andado. A Wences, por confiar en mí y por contagiarme su ilusión por las matemáticas y la estadística. A Manolo, por todos sus consejos, su ayuda y por prestarme su análisis crítico tan acertado. Gracias a los dos por marcarme el camino y por ayudarme a crecer, tanto a nivel profesional, como personal. También quiero extender mi agradecimiento a todos los miembros del tribunal de seguimiento: Andrés Antonio Vaamonde Liste, Germán Aneiros Pérez, Beatriz Pateiro López y Rosa María Crujeiras Casais. Gracias por vuestro tiempo y por vuestros valiosos comentarios e indicaciones durante todos estos años. I would also like to acknowledge Christophe Giraud for accepting our request and receiving me at Institut de Mathématiques d'Orsay, Université Paris-Saclay. Thanks for providing me with really interesting discussion and references. In addition, I would like to extend my gratitude to Rosa Elvira Lillo Rodríguez, Beatriz Pateiro López and Christophe Ley for accepting to be part of the thesis committee. In addition, I want to acknowledge Rahul Ghosal and Christophe Ley their work as international reviewers. Todos vosotros habéis ayudado a que este documento alcance su madurez a nivel académico. Muchas gracias por vuestro tiempo, de corazón.

Gracias también a todas aquellas personas con las que me he topado en la que ha sido mi segunda casa estos años, la Facultad de Matemáticas de la USC. Empezando por el Departamento de Estadística, Análisis Matemático y Optimización, por su buena acogida desde el principio. Quiero hacer una mención especial a Rosa, César, Wences y Mercedes, con los que he tenido el privilegio de compartir docencia. Gracias por hacer de la docencia una forma fácil y amena no solo de enseñar, sino también de gran aprendizaje y disfrute para mí. También quiero agradecerles a Rosa y Alberto por su implicación personal como coordinadores de doctorado. He aprendido mucho de todos vosotros. Además, no habría podido sobrellevar el día a día sin todos los compañeros con los que me he cruzado por los pasillos durante esta etapa. Gracias a los integrantes de la sala π, a los compañeros predocs, a los colegas de otros departamentos y a los CITMAgas. Gracias por los cafés, las risas, las comidas, las charlas, las coñas, las escapadas, los paseos, las cañas, las cenas, las fiestas y los partidos de tenis. Sin duda, me habéis ayudado a sobrellevar muchos malos momentos y habéis hecho que el camino sea más llano. También me gustaría agradecer a todos los compañeros con los que he tenido la suerte de coincidir en las clases de tenis. Como dije en una ocasión, hay tardes con vosotros que me devuelven años de vida.

No podría decir que siento la Facultad de Matemáticas como mi segunda casa sin dejar de mencionar a todas esas personas que me han acompañado desde el principio de esta

carrera de fondo: mis compañeros del Grado de Matemáticas. Con especial cariño, me gustaría mencionar a las integrantes del grupo Aplicadas y a los miembros del Grupo Abierto/Grupo haveliano. Gracias a todos aquellos que habéis pasado de compañeros a amigos. Gracias por vuestro tiempo y cariño. Me llevo mil anécdotas y muchos recuerdos bonitos de mi paso por Santiago a vuestro lado. Os admiro mucho como matemáticos, pero sobre todo, como personas.

También me gustaría agradecer a mis amigos "parisinos" del Colegio de España. Habéis hecho de mi estancia de tres meses en París una experiencia única. Sin duda, volvería a repetir esta aventura con todos vosotros.

No puedo cerrar estos agradecimientos sin dirigirme a la gente que me espera en la que considero mi verdadera casa, ya que una parte de mi corazón está, y estará siempre, situada entre Nanín, Sanxenxo y Portonovo. Me gustaría empezar agradeciendo a todos esos amigos que de una forma u otra están siempre, para arropar en los malos momentos y celebrar en los buenos. Sin duda, necesito agradecer de corazón a Leti y Sora, por todos los Findes de Chicas, por curar los problemas a base de cañas o vinos y por las fiestas para celebrar las buenas noticias. Gracias por multiemplearos por mí, por ser amigas, psicólogas, terapeutas o arquitectas, según toque. Gracias por todo el tiempo y cariño invertido. Gracias también a los Sommelieres Expertos por todos los buenos momentos y catas varias, así como a todas las personas que aportan su granito de alegría en mis fines de semana. También quiero agradecer a esos amigos que, aún estando más lejos, siempre os siento ahí. Gracias a todos por llenar mis días de luz.

Creo de verdad que no podría haber llegado a donde estoy sin los valores y el esfuerzo de mi familia. Gracias a mis abuelos, Manola y Quico, con los que he tenido la suerte de criarme, y gracias también a mis otros abuelos, Esmeralda y Severo, que ya no están, por vuestro esfuerzo y trabajo duro. Gracias por darnos, al resto, un futuro mejor con todo lo que vosotros no tuvisteis. Gracias a mis padres, Noli y Joaquín, por ser un ejemplo a seguir. Gracias por ser sinónimo de esfuerzo, responsabilidad y constancia. También gracias por todos vuestros esfuerzos para darnos todas las oportunidades y facilidades posibles que a vosotros os habría gustado tener en vuestra época. Gracias por el cariño, por aguantar mil peroratas y, aún así, estar siempre disponibles a una llamada. Gracias también a mi hermano, David, por estar ahí siempre dispuesto a ayudar. Gracias a mi familia por el apoyo continuo en esta etapa. Sois todos, para mí, una inspiración.

Ya para terminar me gustaría agradecerte a ti, Jose, todo el cariño incondicional. Gracias por ser mi pilar fundamental, por prestarme tus ojos cuando mi realidad se distorsiona en los días grises y por hacerme sentir acompañada allí donde esté. Gracias por hacer la vida tan bonita y liviana, por ponerle banda sonora a mis días y por echarme siempre la última copa de vino. Gracias por todos los pequeños detalles del día a día que no se ven. Creo que no podría haber escogido mejor compañero de viaje.

Laura Freijeiro González
Santiago de Compostela, 17 de marzo de 2023

Contents

Abstract

In a Big Data context, the number of covariates used to explain a variable of interest, p, is likely to be high, sometimes even higher than the available sample size ($p > n$). Ordinary procedures for fitting regression models start to perform wrongly in this situation. As a result, other approaches are needed. A first covariates selection step is of interest to consider only the relevant terms and to reduce the problem dimensionality. The purpose of this thesis is the study and development of covariates selection techniques for regression models in complex settings. In particular, we focus on recent high dimensional or functional data contexts of interest. Assuming some model structure, regularization techniques are widely employed alternatives for both: model estimation and covariates selection simultaneously. Specifically, an extensive and critical review of penalization techniques for covariates selection is carried out. This is developed in the context of the high dimensional linear model of the vectorial framework. Conversely, if no model structure wants to be assumed, state-of-the-art dependence measures based on distances are an attractive option for covariates selection. New specification tests using these ideas are proposed for the functional concurrent model. Both versions are considered separately: the synchronous and the asynchronous case. These approaches are based on novel dependence measures derived from the distance covariance coefficient.

KEYWORDS: High dimension; Covariates selection; Regularization techniques; Distance covariance; Functional concurrent model.

Introduction

Massive amounts of data have emerged in the last century in different fields due to enormous computational and technological progress. This phenomenon knows as the beginning of the "Big Data" age. Big Data was initially defined by three V's that characterize the new type of generated data: volume, velocity, and variety. Because of the high impact of this phenomenon on the population, other features were added, such as veracity, validity, and value. These characteristics question the quality of the information provided. Variability, volatility, and visualization have been new additions related to the complexity of managing these big data sets. These last three terms warn about the possible difficulties of extracting and interpreting results. Finally, other qualities, such as vulnerability, venue, or virality, have also been added to this list. These are related to computing management. All the cited features support that Big Data scenarios are challenging in several disciplines. In particular, these have a special interest in Statistics, where one wants to be able to manage the information provided by all new types of complex data.

As a result, Big Data sets have quite varied natures and characteristics. In Statistics, those derived from the volume feature are of great interest, giving place to new complex statistical objects. Some examples are the high dimensional framework or the ones known as high-frequency or functional data sets, among others. Therefore, it is necessary to develop suitable techniques for their management and analysis.

In the current Big Data context, it is of great interest the high dimensional case related to the concern for volume, where the number of covariates considered (p) is high, even higher than the available sample size (n). Examples of this framework are stock prices in Economics, genomic studies in Biology, or data from social media in Computing Science. Classical techniques start to perform poorly in this context. In the case of $p > n$, these are not available. Hence, new statistical procedures are required to deal with these drawbacks.

Another interesting type of data that has appeared in recent decades, related to the volume characteristic, is functional or high-frequency data. Here, variables are functions that depend on some t argument, resulting in objects with infinity dimensions. Thus, given a "time" instant t, the observed data are discretized values of the functions at that moment. Some examples of the use of functional data are meteorological modeling, statistical image or signal processing, monitored measures in medicine, functional magnetic resonances, and energy demand or consumption modeling. For these statistical objects, completely new procedures have been developed to deal with their functional nature. This has been another implication of the Big Data phenomenon.

The requirement of new procedures in the high dimensional and functional contexts also applies to regression models. In the first case, classical techniques for model estimation in the vectorial framework, which assumes that $n > p$, start to perform poorly for a large number p of covariates. This is due to the curse of dimensionality, where the local character is lost as p increases. This drawback also has an impact on model estimation in functional data. Specifically when considering a large number of covariates. Furthermore, these

approaches are not available when $p > n$. An additional drawback is that the increment of the p value, i.e. the increment of the considered covariates in the regression model, enhances the apparition of collinearity or concurvity effects. Therefore, not all covariates are relevant, but only a bunch are needed to explain the available information. As a result, the difficulty increases as this number grows. This fact is especially harmful in the functional context, where functions are estimated instead of vectors.

Given all these drawbacks, covariates selection techniques to reduce the problem dimensionality, i.e. to consider fewer than p covariates, and being able to include only the relevant terms, are desirable in the Big Data context. Specifically, covariates selection algorithms are an essential preliminary step to solve volume issues. Nevertheless, similar to prediction, ordinary procedures for covariates selection in regression models fail in these scenarios. Consequently, covariates selection, estimation, and prediction in these settings are a big challenge in contemporary Statistics.

Penalization techniques are a well-known and widely employed alternative for both: model estimation and covariates selection simultaneously. In particular, these add a regularization term to the estimation process. According to the penalization type, these could apply covariates selection, as in the L_1 case. An example is the noted LASSO regression of Tibshirani (1996). Furthermore, this approach works in the $p > n$ framework. These properties make the penalization procedures an outstanding option for the high dimensional context. However, it is necessary to assume some structure in the regressor function in advance (linear, additive, etc.), and this assumption could be incorrect or restrictive in practice. This last is especially tricky in the $p > n$ case, where there are no (or almost no) tools to test its adequacy.

An alternative idea to avoid assumptions about the model is the use of dependence coefficients for covariates selection. A novel measure of dependence is the distance covariance of Székely et al. (2007). This coefficient detects any dependence structure between two random vectors of arbitrary dimensions and, as a result, can be used as a covariates selector. Thus, it is enough to consider only the covariates with the highest distance covariance value with the response in the regression model. This idea can also be applied in the case of $p > n$ and has been shown that protects against the curse of dimensionality. Now assuming some structure of the regression model or estimating this is unnecessary. As a result, this is an attractive alternative to penalization techniques. Due to its good statistical properties, adaptations for conditional mean independence and tests of conditional independence have been proposed in the recent literature. Besides, some extensions of these coefficients for the functional framework have also been derived.

The main topic of this thesis project is the study and development of covariates selection techniques for regression models in complex settings. In particular, we focus on recent and high dimensional or functional data contexts of interest related to the high volume characteristic. Assuming some model structure, regularization techniques are widely employed alternatives for both: model estimation and covariates selection simultaneously. Specifically, an extensive and critical review of penalization techniques for covariates selection is carried out. This is developed in the context of the high dimensional

linear model of the vectorial framework. This study is followed by a critical analysis under dependence and considering covariates in different scales. Conversely, if no model structure wants to be assumed, state-of-the-art dependence measures based on distances are an attractive option for covariates selection. Concerning functional data, the functional concurrent model is considered. New specification tests focused on covariates selection are proposed for both: the synchronous and the asynchronous case. These approaches are based on novel dependence measures derived from the distance covariance coefficient.

The document is organized as follows. This manuscript begins with a review of the problems that regression models have to deal with in the high dimensional framework, motivating the need for covariates selection in Chapter 1. Next, a complete review of penalization techniques performs in Chapter 2. This study focuses on the linear model in the vectorial context with the LASSO regression as the core. Its advantages and disadvantages as a covariate selector are analyzed. Next, some modifications and alternatives are presented. In Chapter 3, a comparison of the performance of these regularization techniques is carried out. For this aim, complex scenarios are considered. In particular, data is simulated assuming distinct dependence structures and covariates in different scales. Some guidelines are given about what can be expected and what is the best procedure in terms of the data nature. Furthermore, these procedures are tested in real data sets. Later, in Chapter 4, an extensive review of the distance covariance properties, characteristics, and processes for the construction of estimators is collected. This analysis is also extended to its derivatives. Finally, some comments about their use for covariates selection in complex models arise. Then, the functional framework appears through the study of the functional concurrent model. Specifically, Chapter 5 is devoted to the development of new significance tests for its synchronous version using a mean conditional measure of dependence based on distances. Their theoretical properties are studied and examples of their good performance in practice are displayed. Besides, some applications in real data sets are illustrated. In Chapter 6 some ideas for the extension of the significance tests to the asynchronous version are displayed. In particular, new global tests are proposed using a conditional dependence measure derived from the distance covariance. A proper statistic is obtained and its good behavior is exemplified through a simulation study. Eventually, some conclusions and future work are discussed.

Chapter 1

Problems of regression models in the high dimensional framework: the need for dimensionality reduction

This chapter introduces problems that different regression models face in a high dimensional framework. These will motivate the content of the following chapters of this thesis document. First, in Section 1.1, some concepts about different structures of regression models employed throughout the manuscript are briefly reviewed. In the first place, the linear model, related to Chapters 2 and 3, is introduced in Section 1.1.1. Subsequently, the additive formulation is presented in Section 1.1.2 and employed later in Chapter 5. Finally, the local regression is analyzed in Section 1.1.3, connecting with the general formulation considered in Chapter 6. In all these cases, problems arising in a Big Data context are argued. Specifically, the difficulties arising in a high-dimensional framework for these formulations, considering a great value of p or even $p > n$, are discussed in detail in Section 1.2. These problems are classified into three groups: the curse of dimensionality (Section 1.2.1), model estimation inconsistencies (Section 1.2.2), and collinearity or concurvity effects (Section 1.2.3). Eventually, the need for dimensionality reduction in this framework is motivated in Section 1.3. This discussion gives rise to the necessity of employing covariate selection techniques as a preliminary step in high dimensional contexts, motivating the main topic of this thesis project.

1.1 Brief introduction to regression models

In a regression model, a variable of interest or response, Y, is explained in terms of $p \geq 1$ explanatory covariates $X = (X_1, \ldots, X_p)^\top$. For this purpose, one can assume that Y is related to X by a regression function $m(\cdot)$, which is typically unknown. As a result, the regression model is given by

$$Y = m(X) + \varepsilon, \tag{1.1}$$

where ε is the model error, not directly observed in practice. This error is commonly assumed to be conditional independent of X in terms of $m(\cdot)$.

It is usual to assume $m(X) = \mathbb{E}[Y|_X]$ in (1.1) in practice, which translates into the explanation of the conditional mean of Y, given X. Besides, it is usual to ask for condition $\mathbb{E}[\varepsilon|_X] = 0$ to guarantee independence between model error and covariates. However, there are other options available for the regressor function, such as considering certain quantile information, i.e. $m(X) = \mathbb{Q}[Y|_X]$. This last results in a quantile regression formulation. Following usual guidelines, $m(X) = \mathbb{E}[Y|_X]$ is assumed from now on as the information explained by the regressor function.

Regardless of the operator choice, once the type of information collected by $m(\cdot)$ is selected, it is needed to estimate its form. For this purpose, there are two possible options: to assume some structure in $m(\cdot)$, such as linear ($m(X) = X\beta$) or additive ($m(X) = \sum_{j=1}^{p} f_j(X_j)$), or directly resort to nonparametric techniques without any assumption about this. The first option is more restrictive but allows us to know how each covariate impacts the response. Examples of the linear and additive formulations are presented in Sections 1.1.1 and 1.1.2, respectively. The linear model is assumed in Chapters 2 and 3, whereas the additive formulation is considered in Chapter 5 in terms of covariates selection algorithms. In contrast, the nonparametric approach, considering the general function $m(X)$ where $X \in \mathbb{R}^p$, allows more flexibility but loses the interpretability of covariates effects. The local regression is introduced in Section 1.1.3 as an example of a nonparametric approach. This approach relates to the general regression formulation proposed in Chapter 6. It is in the case of having a high number of covariates, p, or even if $p > n$, when the first choice is more tractable. This result is due to dimensionality problems of nonparametric techniques. In particular, because of the phenomenon known as the curse of dimensionality and estimation inconsistencies. These problems are introduced and treated in more detail below, in Sections 1.2.1 and 1.2.2, respectively. However, assuming a preliminary structure also has some limitations in the $p > n$ case. These are related to the estimation of the model and collinearity or concurvity effects. These problems will be discussed later in Sections 1.2.2 and 1.2.3, respectively. Consequently, a dimensionality reduction to consider not p covariates, but just a bunch of them, is of special interest in the high dimensional framework. A safe passage is to appeal to covariates selection techniques. Nevertheless, classical approaches do not perform well in a high dimensional context. Instead, other procedures have to be considered. Similar to the estimation of the regressor function, there are now again two possible paths, depending on whether or not one assumes some structure on $m(X)$. A discussion about this topic takes place in Section 1.3.

It is remarkable to notice that the response, Y, and the explanatory covariates $X_1,\ldots,X_p$, can have quite different natures: scalar variables, vectorial variables, functional data, etc. Therefore, the estimation procedure of the regression model must be adapted on a case-by-case basis accordingly to these. As the high dimensional framework is quite tricky, we start working in the vectorial context because of simplicity. This framework is of great interest regarding the penalization techniques considered in Chapters 2 and 3. Later, a particular example of the functional model formulation is introduced and studied in Chapters 5 and 6. So, as mentioned above, three practical and widely employed fitting model techniques are displayed next, assuming $Y \in \mathbb{R}$ and $X = (X_1, \ldots, X_p)^\top \in \mathbb{R}^p$ where $p \geq 1$[1]. These are the linear model, additive formulation, and local regression.

1.1.1 Linear model

Assuming without loss of generality that Y and X are centered variables, the linear model corresponds with taking $m(X) = X^\top\beta$ for $\beta \in \mathbb{R}^p$ in (1.1) formulation. This

[1]Results could be extended for multivariate response $Y \in \mathbb{R}^q$ with $q \geq 1$, but we restrict ourselves to this framework just for simplicity.

formulation is a parametric model assuming a linear structure in $m(\cdot)$. Then, the model estimation translates into the proper estimation of the $\beta \in \mathbb{R}^p$ vector. One can estimate this vector by minimizing the mean squared error. Thus, assuming a fixed design and given $(\mathbf{X_n}, \mathbf{Y_n}) = \{(x_i, y_i), i = 1, \ldots, n\}$ an iid sample from the joint distribution function of $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, this results in minimizing the $\phi(\beta)$ function defined as

$$\min_{\beta} \phi(\beta) = \min_{\beta} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 = \min_{\beta} \left(\mathbf{Y_n} - \mathbf{X_n}\beta\right)^\top \left(\mathbf{Y_n} - \mathbf{X_n}\beta\right) = \min_{\beta} \|\mathbf{Y_n} - \mathbf{X_n}\beta\|_n^2, \tag{1.2}$$

being $\|\cdot\|_n$ the euclidean norm in $\mathbb{R}^n$.

Expanding the above expression (1.2), one gets

$$\begin{aligned}\phi(\beta) :=& \|\mathbf{Y_n} - \mathbf{X_n}\beta\|_n^2 = \left(\mathbf{Y_n} - \mathbf{X_n}\beta\right)^\top \left(\mathbf{Y_n} - \mathbf{X_n}\beta\right) = \left(\mathbf{Y_n}^\top - (\mathbf{X_n}\beta)^\top\right)\left(\mathbf{Y_n} - \mathbf{X_n}\beta\right) \\ &= \left(\mathbf{Y_n}^\top - \beta^\top \mathbf{X_n}^\top\right)\left(\mathbf{Y_n} - \mathbf{X_n}\beta\right) = \mathbf{Y_n}^\top \mathbf{Y_n} - \beta^\top \mathbf{X_n}^\top \mathbf{Y_n} - \mathbf{Y_n}^\top \mathbf{X_n}\beta + \beta^\top \mathbf{X_n}^\top \mathbf{X_n}\beta,\end{aligned}$$

next, differentiating and equating to zero

$$\frac{\partial \phi(\beta)}{\partial \beta} = -2\mathbf{Y_n}^\top \mathbf{X_n} + 2\beta^\top \mathbf{X_n}^\top \mathbf{X_n} = 0 \Rightarrow 2\beta^\top \mathbf{X_n}^\top \mathbf{X_n} = 2\mathbf{Y_n}^\top \mathbf{X_n} \Rightarrow \mathbf{X_n}^\top \mathbf{X_n}\beta = \mathbf{X_n}^\top \mathbf{Y_n}$$

and it is known that

$$\frac{\partial \phi^2(\beta)}{\partial \beta^2} = \mathbf{X_n}^\top \mathbf{X_n} \quad \text{and} \quad \det(\mathbf{X_n}^\top \mathbf{X_n}) \geq 0 \quad \text{because } \mathbf{X_n}^\top \mathbf{X_n} \text{ is positive semi-definite.}$$

This guarantees that the solution of the normal equations

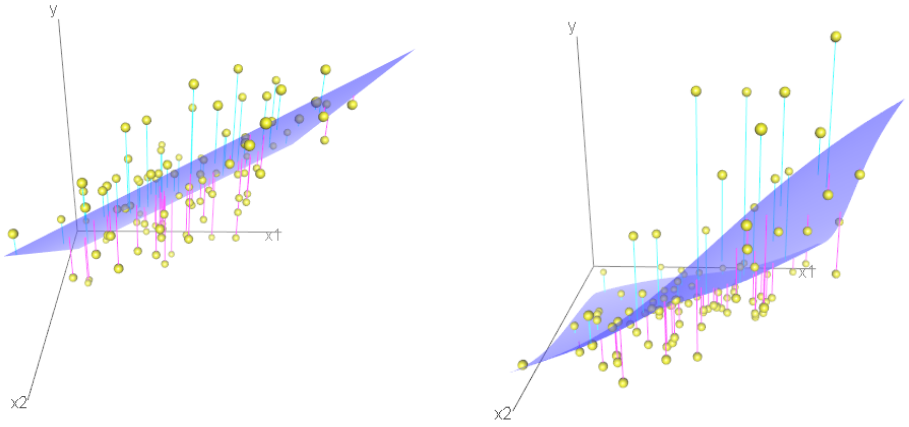
$$\mathbf{X_n}^\top \mathbf{X_n}\beta = \mathbf{X_n}^\top \mathbf{Y_n} \Rightarrow \hat{\beta} = \left(\mathbf{X_n}^\top \mathbf{X_n}\right)^{-1} \mathbf{X_n}^\top \mathbf{Y_n} \tag{1.3}$$

is a minimum and $\hat{\beta}$ is known as the ordinary least squares (OLS) estimator. An illustration of this estimator in three dimensions, jointly with a smooth version, is given in Figure 1.1. See Sections 1.1.2 and 1.1.3 for examples of this last.

The OLS estimator has good statistical properties. It is easy to see that this term is an unbiased estimator as

$$\begin{aligned}\mathbb{E}[\hat{\beta}] &= \mathbb{E}\left[(\mathbf{X_n}^\top \mathbf{X_n})^{-1} \mathbf{X_n}^\top \mathbf{Y_n}\right] = (\mathbf{X_n}^\top \mathbf{X_n})^{-1} \mathbf{X_n}^\top \mathbb{E}\left[\mathbf{Y_n}\right] \\ &= (\mathbf{X_n}^\top \mathbf{X_n})^{-1} \mathbf{X_n}^\top \mathbb{E}\left[\mathbf{X_n}\beta + \varepsilon\right] = (\mathbf{X_n}^\top \mathbf{X_n})^{-1} \mathbf{X_n}^\top \left(\mathbb{E}\left[\mathbf{X_n}\beta\right] + \mathbb{E}[\varepsilon]\right) \\ &= (\mathbf{X_n}^\top \mathbf{X_n})^{-1} \mathbf{X_n}^\top \mathbf{X_n}\beta = \beta\end{aligned}$$

and under the assumption of normality in the model error, the OLS estimator of (1.3) corresponds to the one that results from maximizing the likelihood. Besides, if the errors are uncorrelated with mean zero and homoskedastic with finite variance, the Gauss-Markov theorem (Theorem 1.1) guarantees that the OLS estimator has the lowest sampling variance within the class of linear unbiased estimators.



(a) Linear regression (b) Smooth regression

Figure 1.1: OLS method in three dimensions. This fits a surface (dark blue) by minimizing the sum of the positive (light blue) as well as negative (pink) squared residuals.

Theorem 1.1 (Gauss-Markov)**.** *Suppose* $m(X) = \mathbb{E}[Y|_X] = X\beta$ *and* $\mathbb{C}(X) = \sigma^2 I_p$*, being* $\mathbb{C}[\cdot]$ *the covariance operator and* $I_p \in \mathbb{R}^p$ *the identity matrix. Then, for all* $\tilde{\phi} = c^\top Y$ *linear unbiased estimators of* $\phi = z^\top \beta$*, where* z *is a random vector, it is verified that*

$$\mathbb{V}(\tilde{\phi}) \geq \mathbb{V}(\hat{\phi})$$

where $\mathbb{V}[\cdot]$ *is the variance operator and* $\hat{\phi} = z^\top \hat{\beta}$*, with* $\hat{\beta}$ *the OLS estimator.*

Some problems of the linear model displayed in (1.2) for the $p > n$ context are related to the estimation of the model and collinearity effects. Specifically, the $\hat{\beta}$ OLS estimator of (1.3) could not be obtained in this case, as is discussed in Section 1.2.2. Moreover, considering a large number of covariates makes collinearity effects more likely. This last topic is treated in Section 1.2.3. All these inconveniences bring the fact that, although the linear model is a fairly tractable formulation for the problem (1.1), this does not work in the high dimensional framework in the present form. As a result, covariates selection techniques are of interest in the $p > n$ case to reduce the problem dimensionality and solve these drawbacks. Next, more complicated structures are proposed and analyzed.

1.1.2 Additive model

In the additive model, the assumption that variables Y and X are centered remains, but one allows the regression formulation to have more flexibility than in the linear form. An illustrative example of linear and other smoothing effects is collected in Figure 1.1. This illustration gives a first intuition about the larger number of structures that a more flexible approach, like smoothing, could collect. In particular, each covariate effect is modeled by

an unknown function $f_j(\cdot)$, for $j = 1, \ldots, p$ to allow more flexibility in the model, and these effects are assumed to be additive. Hence, this results in plug-in the $m(X) = \sum_{j=1}^{p} f_j(X_j)$ structure in the general regression model (1.1). This expression is a nonparametric model, where it is needed to search for a proper estimation of each $f_j(\cdot)$ function without assuming any structure in these. There are two options to estimate these functional terms

- **Kernel smoothing techniques:** functions $f_j(\cdot)$, for $j = 1, \ldots, p$, are estimated as the weighted local average of observed data values. The weights are given by a kernel function that satisfies specific requirements, obtaining greater values for closer points. Examples of kernel functions are shown in Figure 1.2. The local character is ruled by a bandwidth parameter that needs estimation in practice.

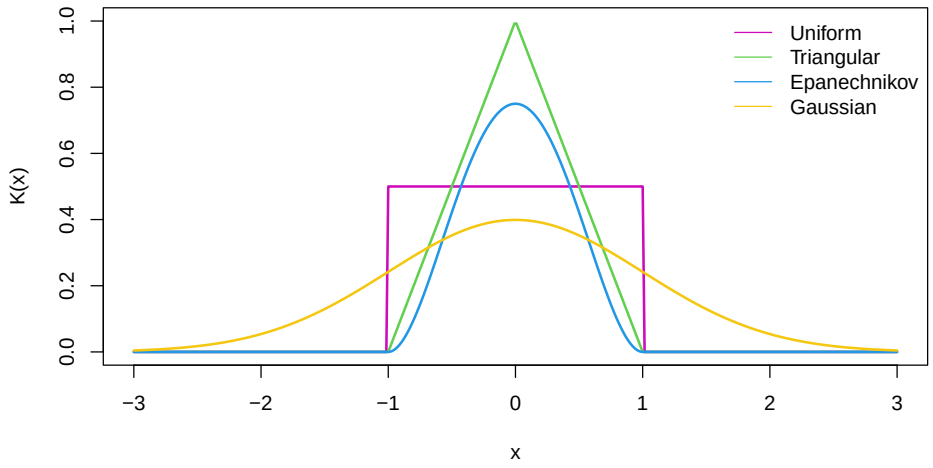


Figure 1.2: Example of univariate kernel functions.

- **Representation by means of functional basis:** each function $f_j(\cdot)$ is represented in terms of $Q_j \geq 1$ elements of a functional basis for all $j = 1, \ldots, p$. Then, given Q_j elements of a certain basis, partial effects are rewritten as a linear combination of these. As a result, the problem of estimating $f_j(\cdot)$ translates into estimating the coefficients of the linear combination of the functional basis terms that can be solved using similar techniques to the ones employed for linear regression (see Section 1.1.1). An example of some functional basis elements is displayed in Figure 1.3.

In both options mentioned above, given $(\mathbf{X_n}, \mathbf{Y_n}) = \{(x_i, y_i), i = 1, \ldots, n\}$ an iid sample from the joint distribution function of $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, one needs to resort to iterative algorithms to adjust the additive model. One of the most employed techniques is the backfitting algorithm (Algorithm 1.2).

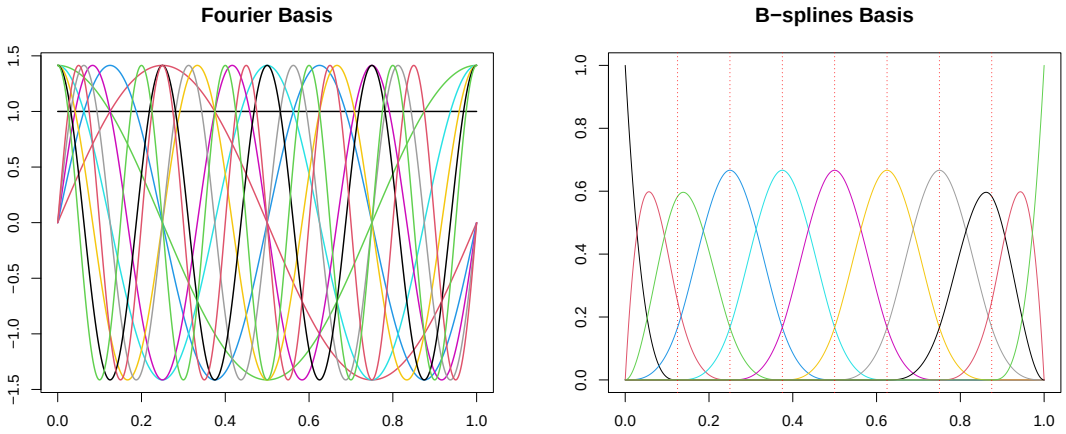


Figure 1.3: Example of the first 11^{th} elements of the Fourier (left) and B-splines (right) functional basis.

Algorithm 1.2 (Backfitting)**.**

1. Initialize $\hat{f}_j = 0$ for $j = 1, \ldots, p$.
2. For each $\hat{f}_j$, $j = 1, \ldots, p$, iteratively recalculate $\hat{f}_j = \mathcal{S}_j\left(\mathbf{Y_n} - \sum_{k \neq j} \hat{f}_k|_{\mathbf{X_j}}\right)$, being $\mathcal{S}_j(\cdot)$ a smoother of the response over X_j.
3. Repeat step 2 until convergence is achieved.

In this algorithm, for the $\mathcal{S}_j(\cdot)$ smoother function, it is necessary to apply kernel smoothing or functional basis representation techniques.

A problem with Algorithm 1.2 is its related high computational cost. In the additive model fitting, each sample requires kp steps, where $k \geq 1$ is the number of cycles of the fitting algorithm. Then, for n samples, a total of kpn operations are needed. Besides, the computational complexity of the smoother terms $\mathcal{S}_j(\cdot)$ has to be added as well. For example, in the case of using cubic smoothing splines, this number is $pn\log(n)$, resulting in a total of $pn\log(n) + kpn$ operations (see Section 9.7 of Hastie et al. (2009) for more details). This fact translates into complexity in the estimation procedure for large values of p, which implies more complexity in high dimensional contexts.

Next, the ideas of applying representation using functional bases to estimate the $f_j(\cdot)$ functions are detailed. We refer the reader to Section 1.1.3 for more details about kernel smoothing techniques implementation.

Functional basis representation

Once a functional basis is selected (Splines, Fourier, Wavelets, etc.) for each covariate effect, the $f_j(\cdot)$ functions are expressed in terms of $Q_j \in \mathbb{N}$ elements of the basis $\{\mathcal{B}_{jq}(\cdot), q =$

$1, \ldots, Q_j\}$. This representation results in

$$f_j(x) = \sum_{q=1}^{Q_j} \alpha_{jq} \mathcal{B}_{jq}(x),$$

being α_{jq} the unknown coefficients of the linear combination of basis functions that require estimation for $j = 1, \ldots, p$ and $q = 1, \ldots, Q_j$.

It is quite common to assume the same class of functional bases for all effects, when possible, and the same number of considered basis terms. Roughly speaking $Q_j = Q$ for all $j = 1, \ldots, p$. However, we work under the most general context, allowing different bases and numbers of components.

Considering the basis representation, one obtains that $y_i = \sum_{j=1}^{p} \sum_{q=1}^{Q_j} \alpha_{jq} \mathcal{B}_{jq}(x_i) + \varepsilon_i$. Then, the additive model can be rewritten as

$$\mathbf{Y_n} = \mathcal{B}_\mathbf{n} \alpha + \varepsilon \tag{1.4}$$

where $\mathcal{B}_\mathbf{n} = \left[\{\mathcal{B}_{1q}(X_1)\}_{q=1}^{Q_1}; \ldots; \{\mathcal{B}_{pq}(X_p)\}_{q=1}^{Q_p}\right]$ is a matrix of dimension $n \times \sum_{j=1}^{p} Q_j$ and $\alpha = [\{\alpha_{1q}\}_{q=1}^{Q_1}, \ldots, \{\alpha_{pq}\}_{q=1}^{Q_p}]$ is a vector of dimension $\sum_{j=1}^{p} Q_j$.

The resulting linear expression given in (1.4) allows us to estimate the α_{jq} coefficients using the OLS procedure introduced above for the linear model. Nevertheless, we need to choose a proper value of Q_j first. As this is a difficult task, it is common to consider a great enough value and penalize the excess of possible curvature. As a result, for each $f_j(\cdot)$ term it is added a $\lambda_j \int_{\mathcal{D}_j} (f_j''(x))^2 dx$ one to the model, being $\lambda_j > 0$ a penalization parameter and $\mathcal{D}_j$ the domain of X_j for $j = 1, \ldots, p$. In practice, the regularization values of $\lambda_j > 0$ are usually obtained using cross-validation techniques over a grid of values. It is verified that $\int_{\mathcal{D}_j} (f_j''(x))^2 dx = \alpha^\top \mathbb{B}_\boldsymbol{j} \alpha$, where $\mathbb{B}_\boldsymbol{j} = \int_{\mathcal{D}_j} (\mathcal{B}''_\mathbf{nj}(x))^\top (\mathcal{B}''_\mathbf{nj}(x)) dx$ is a known $\sum_{j=1}^{p} Q_j \times \sum_{j=1}^{p} Q_j$ dimensional matrix since $\mathcal{B}''_\mathbf{nj}(x) = [\mathcal{B}''_{j1}(x), \ldots, \mathcal{B}''_{jQ_j}(x)]$ is the $n \times Q_j$ matrix which only depends on the known basis functions. Here, each $\mathbb{B}_\boldsymbol{j}$ term is a matrix of zeros except for elements $\mathbb{B}_{j\,(k+Q_{j-1}, l+Q_{j-1})} = (\mathcal{B}''_{jk}(x) \cdot \mathcal{B}''_{jl}(x))$ with $k, l = 1, \ldots, Q_j$ and being $Q_0 = 0$ for $j = 1, \ldots, p$.

Then, the cited problem is rewritten as a penalized linear regression model given by the minimization problem

$$\min_{\alpha} \|\mathbf{Y_n} - \mathcal{B}_\mathbf{n} \alpha\|_n^2 + \alpha^\top \mathbb{B}\, \alpha,$$

being $\mathbb{B} = \sum_{j=1}^{p} \alpha_j \mathbb{B}_\boldsymbol{j}$ a matrix of dimension $\sum_{j=1}^{p} Q_j \times \sum_{j=1}^{p} Q_j$ such that

$$\mathbb{B} = \begin{pmatrix} \overbrace{\lambda_1 \tilde{\mathbb{B}}_\mathbf{1}}^{Q_1} & \cdots & \overbrace{0}^{Q_p} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_p \tilde{\mathbb{B}}_\boldsymbol{p} \end{pmatrix} \begin{matrix} \} \ Q_1 \\ \\ \} \ Q_p \end{matrix}$$

where $\tilde{\mathbb{B}}_\boldsymbol{j}$ represents the submatrices of $\mathbb{B}_\boldsymbol{j}$ which elements are not null.

Thus, obtaining the penalized OLS estimator of α is possible once values for λ_j are

defined. This estimation translates into solving the minimization problem

$$\begin{aligned}\min_{\alpha}\|\mathbf{Y_n}-\mathcal{B}_\mathbf{n}\alpha\|_n^2+\alpha^\top\mathbb{B}\ \alpha &= \min_{\alpha}\ (\mathbf{Y_n}-\mathcal{B}_\mathbf{n}\alpha)^\top(\mathbf{Y_n}-\mathcal{B}_\mathbf{n}\alpha)+\alpha^\top\mathbb{B}\ \alpha\\ &= \min_{\alpha}\ \mathbf{Y_n}^\top\mathbf{Y_n}-\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}\alpha-(\mathcal{B}_\mathbf{n}\alpha)^\top\mathbf{Y_n}+(\mathcal{B}_\mathbf{n}\alpha)^\top(\mathcal{B}_\mathbf{n}\alpha)\\ &\qquad +\alpha^\top\mathbb{B}\ \alpha\\ &= \min_{\alpha}\phi(\alpha).\end{aligned}$$

Deriving, one gets to

$$\begin{aligned}\frac{\partial\phi(\alpha)}{\partial\alpha}=0 &\Rightarrow -2\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}+2(\mathcal{B}_\mathbf{n}\alpha)^\top\mathcal{B}_\mathbf{n}+2\alpha^\top\mathbb{B}=0\\ &\Rightarrow -2\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}+2\alpha^\top\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+2\alpha^\top\mathbb{B}=0\\ &\Rightarrow -2\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}+2\alpha^\top(\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+\mathbb{B})=0\\ &\Rightarrow 2\alpha^\top(\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+\mathbb{B})=2\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}\\ &\Rightarrow \alpha^\top=\mathbf{Y_n}^\top\mathcal{B}_\mathbf{n}(\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+\mathbb{B})^{-1}\\ &\Rightarrow \hat{\alpha}=(\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+\mathbb{B})^{-1}\mathcal{B}_\mathbf{n}^\top\mathbf{Y_n},\end{aligned}\tag{1.5}$$

where $\hat{\alpha}$ is guaranteed to be the OLS estimator because $\frac{\partial\phi^2(\alpha)}{\partial\alpha^2}=2\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+2\mathbb{B}$ and $\mathcal{B}_\mathbf{n}^\top\mathcal{B}_\mathbf{n}+\mathbb{B}$ is a positive semi-definite matrix.

In the $p>n$ context, other inconvenience arises, apart from the high computational cost required as a trade-off for more flexibility due to additive effects consideration. Similar to collinearity effects in the linear model, this formulation suffers from possible concurvity effects. See Section 1.2.3 for more details. Again, a preliminary covariates selection step is desirable to consider a small number of covariates entering the model and avoid the high dimensional drawbacks in this setting.

Eventually, we allow for more flexibility in the regressor function considering the general formulation in the local regression adjustment.

1.1.3 Local regression

The local regression is a nonparametric technique designed to estimate the general form of the $m(\cdot)$ function of (1.1). This procedure allows complete flexibility and does not require model assumptions as a preliminary step. As its name suggests, for a given value x of the covariates $X\in\mathbb{R}^p$, the idea is to adjust a local model using the values of close observations. For this purpose, one can resort to kernel functions to weight these quantities as necessary.

This results in estimating the regressor function as a locally weighted average of the response values, given by

$$\hat{m}(x)=\frac{1}{n}\sum_{i=1}^{n}\Phi_H(x)y_i\quad\text{with}\quad\Phi_H(x)=\frac{K_H(x_i-x)}{\frac{1}{n}\sum_{i=1}^{n}K_H(x_i-x)},\tag{1.6}$$

where H is a $p\times p$ bandwidth matrix, being symmetric and positive definite, and $K_H(x_i-$

$x) = |H|^{-1}K\left(H^{-1}(x_i - x)\right)$ is the rescaled kernel with $K(\cdot)$ a p dimensional kernel function. Here $|\cdot|$ applies for the matrix determinant. Some examples of kernel functions for the $p = 1$ case are displayed in Figure 1.2. Multivariate kernels can be obtained as the product of these univariate versions.

The bandwidth matrix H controls the shape and size of the local neighborhood and must be estimated. When this takes small values, which translates into few data in the neighborhood, undersmoothing will occur in each direction. Conversely, taking large values will include too many observations in the adjustment. This scenario will produce oversmoothing. Thus, obtaining a proper estimate of the H matrix values is a difficult task, and its complexity increases with the p dimension.

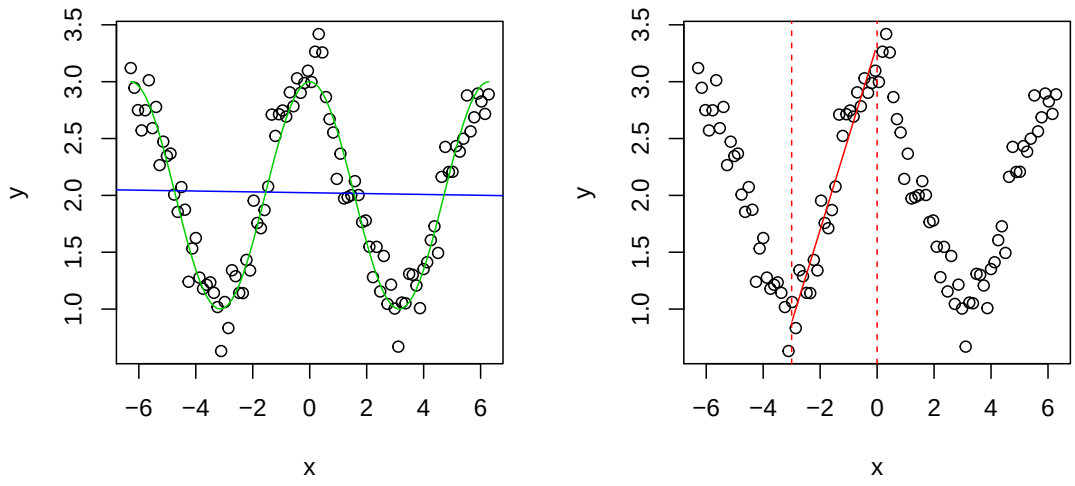


Figure 1.4: Example of modeling a nonlinear relation in two dimensions. Left: real model in green and linear fit in blue. Right: local linear regression at the point $x = -1.5$ in red.

Another local regression alternative is the local polynomial estimators. These, for each value $x \in \mathbb{R}$, adjust a polynomial of degree $d \geq 1$ using the neighborhood information. An example of the local linear case ($d = 1$) in two dimensions is displayed in Figure 1.4. Thus, for a point x and a fixed degree d, the value of $m(x)$ is estimated by means of a $b(x)^\top \beta(x)$ term, where the $\beta(x)$ vector is obtained solving the problem

$$\min_{\beta(x)} \sum_{i=1}^{n} \left(y_i - b(x_i)^\top \beta(x)\right)^2 K_H(x_i - x) = \min_{\beta(x)} \left(\mathbf{Y_n} - \mathbf{B_n}\beta(x)\right)^\top \mathbf{K_n}(\mathbf{x}) \left(\mathbf{Y_n} - \mathbf{B_n}\beta(x)\right) \tag{1.7}$$

where $b(u)$ is a vector $\sum_{j=1}^{p} \binom{p}{j} d^j$ dimensional of polynomial terms in the point u of maximum degree d, $\mathbf{B_n} = [b(x_1)^\top; \ldots; b(x_n)^\top]^\top$ is a $n \times \sum_{j=1}^{p} \binom{p}{j} d^j$ matrix and $\mathbf{K_n}(\mathbf{x}) = \text{diag}\{K_H(x_1 - x), \ldots, K_H(x_n - x)\}$ is a $n \times n$ matrix.

As the problem (1.7) is just a weighted version of the linear formulation displayed in

equation (1.2), the same steps and argumentation of Section 1.1.1 can be employed to obtain a proper estimator. In particular, this results in

$$\hat{m}(x) = b(x)^\top \hat{\beta}(x) = b(x)^\top \left(\mathbf{B_n}^\top \mathbf{K_n}(\mathbf{x})\mathbf{B_n}\right)^{-1} \mathbf{B_n}^\top \mathbf{K_n}(\mathbf{x})\mathbf{Y_n}. \tag{1.8}$$

It is easy to see that the local estimator displayed in (1.8) is just a particular case of the Nadaraya-Watson estimator for local regression considering the weights $\Phi_H(x) = b(x)^\top \left(n\mathbf{B_n}^\top \mathbf{K_n}(\mathbf{x})\mathbf{B_n}\right)^{-1} \mathbf{B_n}^\top \mathbf{K_n}(\mathbf{x})$ in (1.6). Additional details about local regression in $\mathbb{R}^p$ can be found in Section 6.3 of Hastie et al. (2009).

The addition of flexibility in the model estimation process has some additional problems as a trade-off. The first drawback relates to the difficult interpretation of the model effects for values of $p \geq 3$, even if $n > p$. Concerning the increment of covariates appears the curse of dimensionality. As was mentioned above and will be discussed in more detail in Section 1.2.1, great dimensionality spoils the local character. As a result, a proper selection of the H values is still more difficult in a high dimensional framework. Similar procedures to the linear model ones were employed to obtain the local estimator of the equation (1.8). Then, this estimator also inherits its inconsistency problems when $p > n$. A more detailed explanation is given in Section 1.2.2. Eventually, as the local regression is a broad approach, this can also suffer from collinearity and concurvity effects. These will be introduced next in Section 1.2.3. Summing up, it is also desirable to be able to reduce the problem dimensionality in the local regression framework. For this purpose, one can resort to covariates selection techniques in the high dimensional context.

1.2 High dimensional problems

In a Big Data context, it is quite common to face high dimensional situations. These translate into considering a large number of covariates, p. This amount could be even higher than the available sample size ($p > n$). It is in these situations that new drawbacks arise, and classical techniques start to perform poorly. In this section, we introduce the main problems of regression models in the framework of high dimensions. These inconveniences are related to the phenomenon known as the curse of dimensionality (Section 1.2.1), the appearance of inconsistencies in the model estimation procedures (Section 1.2.2), and the increased probability of collinearity and concurvity effects (Section 1.2.3).

1.2.1 The curse of dimensionality

In a high dimensional framework, the local nature is lost as the dimension of p increases. This phenomenon is known as the curse of dimensionality. As a result, the neighborhood concept is missed, and larger areas are needed. This effect is a drawback that affects several statistical techniques in high dimensions. Some comments about its cause and implications can be found in Hastie et al. (2009), Giraud (2014), or Hastie et al. (2015).

This phenomenon can be understood through a simple example. For a fixed number of points uniformly simulated in the unit hypercube of the p-dimensional space, we can

consider a hypercube inside the first one with the origin as an edge and containing, on average, a span of the total number of points. Therefore, a side length is set for the small cube of span$^{1/p}$ to verify this last condition. Then, for a span=0.1, one needs a side length of 0.1 for $p = 1$ and a side length of 0.8 for $p = 10$. This example states the loss of local character when the dimension of the covariates increases. See Figure 1.5 for an illustration.

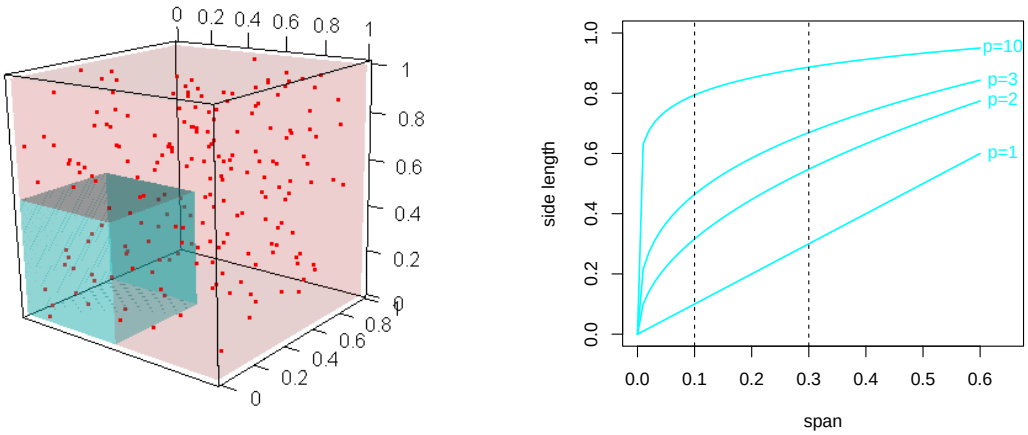


Figure 1.5: Left: the red cube is a unit cube containing points uniformly distributed in three dimensions, whereas the blue and small one is a cube with a volume equal to a value of span. Right: relation between the span and side length of the blue cube for different dimensions in p.

In conclusion, the curse of dimensionality is a hard inconvenience for local procedures. Specifically, this is a problem for nonparametric techniques, which rely on the local nature, employing a proper selection of the bandwidth parameters. Then, even for a moderate dimension of p, it is expected to require larger values of the bandwidths associated with $X_1,\ldots,X_p$, or the bandwidth matrix, respectively, to guarantee that a given span of information is collected. This drawback also applies to other procedures which make use of local ideas as well, for example, the K-nearest neighbors. Hence, techniques such as the local regression introduced in Section 1.1.3 to estimate the $m(\cdot)$ general function of (1.1) perform poorly for large values of p. This fact states the usefulness of considering some structure on the regressor function to avoid the curse of dimensionality in a high dimensional context.

Another problem related to the curse of dimensionality is the loss of interpretability for models considering $p > 3$ covariates. If it is assumed the general formulation displayed in equation (1.1), the hypersurface resulting from the $m(X_1, \ldots, X_p)$ function estimation is quite difficult to be interpreted in practice. Then, procedures to reduce the number of considered covariates are of interest as a preliminary step. Covariates selection techniques for general formulations of the regression model are available using the distance covariance ideas introduced by Székely et al. (2007). These are displayed and treated later in

Chapter 4 and employed in Chapters 5 and 6. However, these do not apply to model estimation, and details about the model structure are not provided in practice. To solve the problem concerning interpretability is quite common to resort to simpler and more specific formulations of $m(\cdot)$ in the high dimensional context. Two popular options are the linear model structure, taking $Y = X\beta + \varepsilon$, or the additive formulation given by $Y = f_1(X_1) + \cdots + f_p(X_p) + \varepsilon$. Both have been introduced previously in Section 1.1. However, some inconsistency problems appear when $p > n$. Next, we analyze the proper adjustment of these models in the high dimensional framework.

1.2.2 Model estimation inconsistencies

As seen above, because of the curse of dimensionality, taking simple structures in the (1.1) formulation could be of interest when p is large. Nevertheless, in a high dimensional context where p is equal or larger than n, denoting as $p > n$ henceforth, some usual techniques for regression model estimation do not perform correctly. They suffer from inconsistencies in the estimation process. Next, we explain these problems for the linear and additive formulations, jointly with the local regression drawbacks. All these formulations are introduced above in Section 1.1.

Linear regression

In the situation of $p > n$, it is not possible to obtain the OLS estimator, $\hat{\beta}$, displayed in expression (1.3). This is because $\mathbf{X_n}$ is a matrix of $n \times p$ dimension, $\mathbf{X_n}^\top \mathbf{X_n}$ is $p \times p$ dimensional matrix, and Corollary 1.3 guarantees that $\text{rank}(\mathbf{X_n}^\top \mathbf{X_n}) \leq n < p$. As we know that

$$\exists (\mathbf{X_n}^\top \mathbf{X_n})^{-1} \Leftrightarrow \det(\mathbf{X_n}^\top \mathbf{X_n}) \neq 0 \Leftrightarrow \text{rank}(\mathbf{X_n}^\top \mathbf{X_n}) = p,$$

where $\det(\cdot)$ denoting the determinant operator, the inverse $(\mathbf{X_n}^\top \mathbf{X_n})^{-1}$ cannot be guaranteed to exist in a unique way. Then, there is not uniqueness of the OLS estimator.

Corollary 1.3. *Being A a matrix of $p \times n$ dimension and B a matrix of $n \times p$ dimension where $p > n$, then*

$$\left.\begin{array}{c} \text{rank}(A \cdot B) \leq \text{rank}(A) \\ \textit{and} \\ \text{rank}(A \cdot B) \leq \text{rank}(B) \end{array}\right\} \Rightarrow \text{rank}(A \cdot B) \leq n$$

given that $\text{rank}(A) \leq n$ *and* $\text{rank}(B) \leq n$ *because* $p > n$.

A widely employed way of solving this drawback is imposing a penalization in the OLS problems displayed in (1.2) over the β parameters. Some examples of these approaches are the LASSO regression (Tibshirani (1996)), the SCAD penalization (Fan (1997)) or the Dantzig selector (Candes and Tao (2007)), among others. These are reviewed in subsequent Chapters 2 and 3.

Local regression

In the case of the local regression treated in Section 1.1.3, as the estimator obtained in (1.8) is a result of solving a weighted linear regression problem, this inherits its inconsistency in the estimation procedure. In particular, estimator $b(x)^\top \hat{\beta}(x)$ will exists if there is a unique inverse of $\mathbf{B_n}^\top \mathbf{K_n(x)} \mathbf{B_n}$. For this aim, similar to linear regression, it is necessary to guarantee that $\text{rank}(\mathbf{B_n}^\top \mathbf{K_n(x)} \mathbf{B_n}) = \sum_{j=1}^{p} \binom{p}{j} d^j \geq p$, otherwise, the determinant of the resulting matrix will be null.

Applying Corollary 1.3, it can be easily seen that, when $n > p$, it is verified that $\text{rank}(\mathbf{B_n}^\top \mathbf{K_n(x)} \mathbf{B_n}) \leq n < p \leq \sum_{j=1}^{p} \binom{p}{j} d^j$. As a result, there is not a unique estimator resulting in solving equation (1.8). Following the guidelines of linear regression to protect against estimation inconsistency, one could think about the imposition of penalties. Nevertheless, this is not straightforward for local regression techniques. It is unclear how to penalize the model coefficients for high degrees d of the $b(x)$ polynomial function. Also, the penalty would have to depend on the bandwidth matrix H. Due to these drawbacks, penalization procedures are not a good safe passage in this context. An example of using regularization techniques for the case of $d = 1$ is the work of Vidaurre et al. (2012) using L_1 penalty ideas.

Additive regression

The additive regression allows more flexibility than the linear model but less than the local regression approach. In particular, the linear model is a specific case of additive regression, where each covariate function takes the $f_j(X_j) = \beta_j X_j$ value for all $j = 1, \ldots, p$. In consequence, assuming that there are, at least, $k < p$ covariates that have a linear effect, it is needed for the submatrix $X_K = (X_1, \ldots, X_k)^\top$ formed by these terms to have rank greater or equal to k. Otherwise, the matrix X_K is ill-conditioned, and the estimation procedure of the additive model would be inconsistent for these covariates, similar to the linear regression case. This implication is, again, a risky situation for the high-dimensional framework where high values of p are expected. However, we will see that this does not apply to nonlinear additive effects.

In the case of estimating an additive model, the limitations of kernel smoothing techniques when p is large have already been justified due to the curse of dimensionality problem exposed in Section 1.2.1. One expects many covariates in the $p > n$ context, so previous limitations could be an issue.

In terms of the representation of the basis, this problem is avoided. After applying basis representation and penalizing the possible excess of curvature, we get to the OLS estimator of the corresponding linear formulation, $\hat{\alpha}$, displayed in (1.5). This is given by $\hat{\alpha} = (\mathcal{B_n}^\top \mathcal{B_n} + \mathbb{B})^{-1} \mathcal{B_n}^\top \mathbf{Y_n}$. Similar to the linear regression problems, using Corollary 1.3, it can be seen that do not exist a unique $(\mathcal{B_n}^\top \mathcal{B_n})^{-1}$ inverse. Now, this problem is solved considering $(\mathcal{B_n}^\top \mathcal{B_n} + \mathbb{B})^{-1}$, after adding the curvature penalization terms collected in $\mathbb{B}$. If these penalizations were not added, two limitations would arise: the consideration of too many elements of the basis could result in an exceed of curvature, and it is not possible to

get the OLS estimator displayed in (1.5) for the $p > n$ case.

Another restrictive limitation of the additive model is the threat of possible collinearity or concurvity effects when p is large. This inconvenience is treated in the next section.

1.2.3 Collinearity and concurvity

The collinearity effect appears when some of the $1, \ldots, p$ considered covariates can be explained in terms of the remaining ones employing a linear relationship. This consequence translates into ill-conditioned of the matrix X in linear regression or the submatrix X_K considering only the covariates with linear effects for the additive model and the local regression. In contrast, the concurvity effect is analogous to collinearity, but this also collects nonlinear effects. This condition is based on the fact that a smooth model effect $f_j(\cdot)$ can be completely explained as a linear combination of the remaining $f_k(\cdot)$ terms, where $j \neq k = 1, \ldots, p$. Then, linear regression avoids this phenomenon, but additive or local regression can suffer from this. The concurvity effect results in inconsistencies in the estimation procedures too.

Thus, consideration of a large number of covariates, p, increases the probability of the linear model suffering from collinearity and the ones of the additive and local formulations suffering from concurvity effects. Besides, this is more likely and much more difficult to detect in the case of concurvity because of its functional nature. As a result, procedures for dimensionality reduction are desirable to avoid these effects, with a notable emphasis on nonlinear formulations.

1.3 Need for dimensionality reduction: covariates selection approaches

The high dimensional framework in regression models is quite challenging in practice, especially for the $p > n$ context. Its most remarkable limitations have been displayed in Section 1.2. These refer to the curse of dimensionality (Section 1.2.1), possible inconsistencies in the model estimation process (Section 1.2.2), and collinearity or concurvity effects (Section 1.2.3). In particular, it has been shown that estimating the regression function under the general formulation displayed in (1.1) is a rough problem. Local regression techniques (Section 1.1.3) suffer from the curse of dimensionality in terms of the bandwidth parameters selection when p is high. Besides, these have inconsistency problems in the estimation procedure for the $p > n$ case and are more prone to collinearity or concurvity effects when the covariates dimension increases. All these drawbacks state the necessity to resort to simpler formulations when $p > n$, such as the linear model (Section 1.1.1) or the additive regression (Section 1.1.2). However, both alternatives present some problems in the high dimensional framework. In the case of linear regression, this also suffers from model estimation inconsistencies when $p > n$ and collinearity effects. Nevertheless, the inconsistency problem is solved through the imposition of penalties in the estimation process can solve the inconsistency problem. This solution is treated in Chapters 2 and 3. By its part, the additive formulation inherits the estimation problems of the linear case, resulting in estimation inconsistency only in the linear effects, and can suffer from

collinearity and concurvity effects. Again, estimation drawbacks of the additive model, related to its linear part, can be solved using penalization techniques. However, the curse of dimensionality, introduced in Section 1.2.1, or the collinearity/concurvity effects of Section 1.2.3, can only be avoided just by reducing the problem dimensionality.

Given all these inconveniences, it is clear that simple models are more tractable in the high dimensional framework. Furthermore, a first step of dimensionality reduction is always desirable to avoid some of the cited problems, such as possible collinearity or concurvity effects. Regarding regression models, one way to apply dimensionality reduction is to resort to covariates selection techniques. These select which of the p covariates considered are relevant and exclude the rest from the adjustment. As a result, they seek to ensure that all provided information is valuable and that the noise is excluded from the model. In addition, this selection results in a dimensionality reduction, considering fewer than p terms in the posterior estimation process. Nevertheless, for the $p > n$ case, classical techniques for covariates selection do not perform well. As a result, specific approaches are needed in this context. The development of these procedures is the main topic of the remaining document. A review of traditional, as well as novel covariates selection techniques, specially designed for the $p > n$ context, is performed in subsequent chapters.

Similar to regression model estimation argued in Section 1.1, there are two possible ways to apply covariates selection techniques: assuming some model structure or selecting terms without any assumption about its form. This classification also applies to the $p > n$ context. In the case of the first option, using penalization techniques in the high dimensional framework performs both simultaneously: covariates selection and model estimation. These procedures are studied and analyzed in more detail in Chapters 2 and 3 under the linear assumption in the vectorial framework. Conversely, completely different approaches are necessary to apply covariates selection without any model assumption when $p > n$. A possibility is the employment of novel dependence coefficients based on distances. These coefficients are introduced in detail in Chapter 4. These procedures propose to implement independence tests to detect the relevant terms. Thus, one can select covariates without any assumption, but no model estimation is performed as a trade-off. We employ these techniques in Chapters 5 and 6 for a particular case of the functional model.

It is important to remark that developments carried out in this chapter assume that response and explanatory covariates are vectorial. This choice is so because, although the vectorial case is one of the easiest contexts, worrying limitations already appear here when working in a high dimensional framework.

Chapter 2

The least absolute shrinkage and selection operator (LASSO)

Once the need for dimensionality reduction has been motivated in Chapter 1, especially for the $p > n$ context, some covariates selection techniques are proposed. For this aim, we start giving solutions for the most naive framework: assuming linearity in the vectorial regression model. This results in the (1.2) formulation. In this context, a great effort has been made in the literature by means of the study and implementation of regularization techniques. The most well-known and still widely employed approach is the Least Absolute Shrinkage and Selection Operator (LASSO) introduced by Tibshirani (1996). In this chapter, the LASSO procedure is motivated and presented in Section 2.1, followed by an analysis of its requirements and inconveniences as a variable selector (Section 2.2). Then, a brief review of the evolution of the LASSO is carried out in Section 2.3, resulting in new procedures which try to solve some of its limitations. In Section 2.4, a list of quite employed or novel competitors of the LASSO is proposed, analyzing its advantages and drawbacks. Eventually, the study is motivated by some examples of real data problems where the necessity of dimension reduction arises. These are presented in Section 2.5. Part of the content of this chapter is collected in Freijeiro-González et al. (2022a).

2.1 Introduction to the LASSO regression

As it has been proved in Chapter 1 when $p > n$, the classic OLS estimation procedure for linear regression fails. This implication is because there are infinite solutions for the resulting system of equations displayed in (1.2). Then, it is necessary to impose modifications or to propose new estimation methods capable of recovering the β values.

There are many situations where not all $p$explanatory covariates are relevant, but several are unnecessary. In these scenarios, we can assume that the β vector is sparse and then search for the relevant covariates, avoiding noisy ones. The idea is, somehow, to obtain a methodology able to compare the covariates and select only those most important, avoiding irrelevant information and keeping the prediction error as small as possible. As there are 2^p possible sub-models, it is unattainable to compare all of them using techniques such as forward selection or backward elimination.

One of the most typical solutions is restricting the number of included covariates, selecting only the relevant ones. This option can translate into adding some constraints to the OLS problem (1.2). This way of proceeding brings up the idea of a model selection criterion, which expresses a trade-off between the goodness of fit and the complexity of the model, such as the AIC (Akaike (1998)) or BIC (Schwarz (1978)) criteria. Nevertheless,

these approaches are computationally intensive, hard to derive sampling properties, and unstable. As a result, they are not suitable for scenarios where the dimension of p is large.

Hence, having a scenario where the number of covariates is greater than the number of available samples ($p > n$) and verifying that the true β has (or can be approximated by) a sparse structure, we could think in penalizing the irrelevant information employing the number of coefficients included in the model. As a result, we could resort to imposing a penalization on the β coefficients in the OLS problem using a penalty factor $p_\lambda(\beta)$ as

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + p_\lambda(\beta) \right\}. \tag{2.1}$$

For this purpose, following the ideas of goodness-of-fit measures, a L_0 regularization, $\lambda\|\beta\|_0 = \lambda \sum_{j=1}^{p} \mathbb{I}_{\beta_j \neq 0}$, could be applied. Here $\mathbb{I}_a$ denotes the indicator function, taking the unit value if condition a is verified and zero otherwise, and $\lambda > 0$ is the penalty parameter. This criterion penalizes models that include more covariates but do not improve so much the performance results. This translates into a model with the best trade-off between interpretability and accuracy, as the AIC or BIC criterion philosophy does, obtaining

$$\hat{\beta}^{L_0} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \mathbb{I}_{\beta_j \neq 0} \right\}. \tag{2.2}$$

The problem of the L_0 penalization is known as the best subset selection (Beale et al. (1967), Hocking and Leslie (1967)). This problem is non-smooth and non-convex, which hinders achieving an optimal solution. As a result, the estimator $\hat{\beta}^{L_0}$ is infeasible to compute when p is of medium or large size, as (2.2) becomes an NP-hard problem with exponential complexity (Natarajan (1995)). However, when p is small, this estimator can still be used in practice. Moreover, it is known that this estimator is optimal in the minimax sense (Bunea et al. (2007)), even when the assumptions required for the LASSO are not satisfied. These assumptions are treated in Section 2.2. See Hastie et al. (2017) for a comparison of this procedure with more current methodologies.

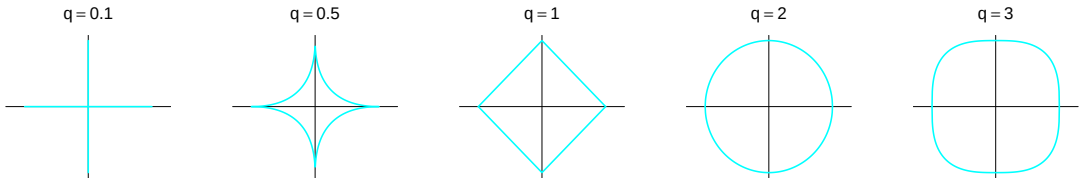


Figure 2.1: Contours of the constant restrictions $\|\beta\|_q = 1$ for some values of q.

To avoid this drawback, one can replace $\lambda\|\beta\|_0$ with other classes of penalizations. Taking into account that this belongs to the family $p_\lambda(\beta_j) = \lambda\|\beta\|_q := \lambda \left(\sum_{j=1}^{p} |\beta_j|^q \right)^{1/q}$, with $|\cdot|$ the absolute value operator and $q \geq 0$, we can commute this for a more appropriate

one. See Figure 2.1 for a comparison between $\|\beta\|_q$ possible structures. The problem (2.1) with this type of penalization is known as the bridge regression (Fu (1998)). The caveat of this family is that this only selects covariates for the $0 < q \leq 1$ values. Moreover, the problem (2.1) is only convex for the $q = 1$ case (see Figures 2.1 and 2.2). Then, it seems reasonable to work with the norm $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$, which is convex, allows covariates selection and leads to the extensively studied Least Absolute Shrinkage and Selection Operator (LASSO) regression, see Tibshirani (1996) and Tibshirani (2011). See Figure 2.2 for a comparison between L_1 penalization form and other well-known as well as widely employed penalties in the literature. These are introduced later in Section 2.4.

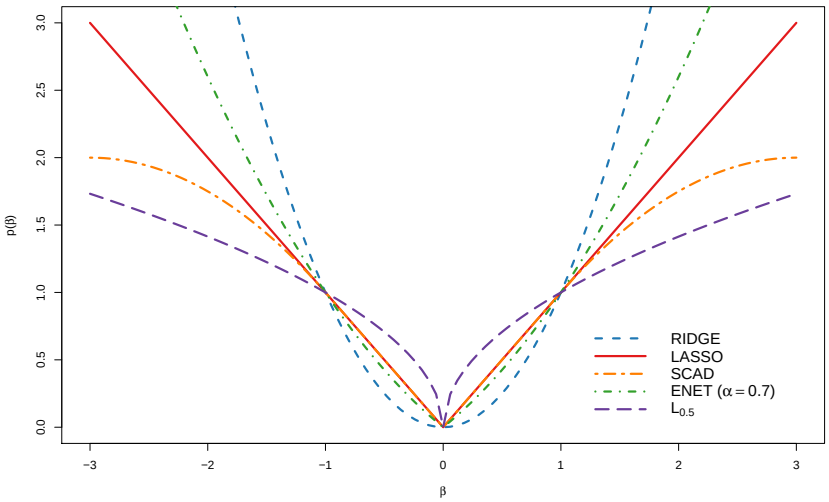


Figure 2.2: Comparison of different penalization methods: L_2 or RIDGE penalization ($RIDGE$), L_1 or LASSO penalization ($LASSO$), SCAD regularization ($SCAD$), Elastic Net penalization method for $\alpha = 0.7$ ($ENET(\alpha = 0.7)$) and $L_{0.5}$ regularization ($L_{0.5}$).

The LASSO regression, also known as basis pursuit in image processing (Chen et al. (2001)), was introduced by Tibshirani (1996). This approach proposes the imposition of a L_1 penalization in (1.2) to perform covariates selection and overcome the drawback of the β estimation in high dimensional frameworks where $p > n$. In this way, one needs to solve the problem given by

$$\hat{\beta}^{LASSO} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.3}$$

which can be rewritten in an analogous way like the optimization problem

$$\hat{\beta}^{LASSO} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2,$$
$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq \theta.$$

In these problems, the term $\theta > 0$, or $\lambda > 0$ equivalently, is the shrinkage parameter in charge of regulating the coefficients penalization. For large values of λ (small values of θ), the coefficients of β are more penalized, which results in a higher number of elements that are shrinkaged to zero. Nevertheless, the estimator $\hat{\beta}^{LASSO}$ introduced in (2.3) has not got an explicit expression.

These formulations translate into convex optimization problems, which guarantee that there is always at least a solution, although if $p > n$, there may be multiple minima (see Tibshirani (2013) for more details). This problem is illustrated for the two-dimensional case in Figure 2.3. Besides, if we think of the noise term ε of (1.2) as being Gaussian, $\hat{\beta}^{LASSO}$ (2.3) can be interpreted as a penalized maximum likelihood estimate, in which the fitted coefficients are penalized in a L_1 sense, thereby encouraging sparsity.

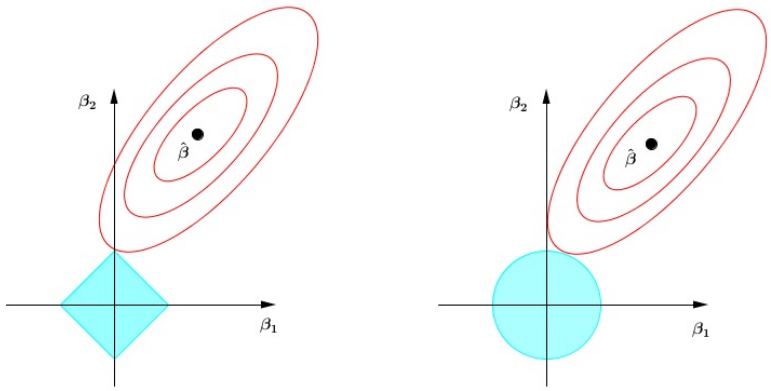


Figure 2.3: Graphics of the LASSO regression estimation (left) and RIDGE regression (right). The blue areas are the restrictions $|\beta_1| + |\beta_2| \leq t$ and ${\beta_1}^2 + {\beta_2}^2 \leq t^2$ respectively, while the red ellipses are the contours of the mean square error function.

The LASSO procedure, defined in (2.3), can be viewed as a convex relaxation of the optimization problem with the L_0 analog of a norm in (2.2). Then, the requirement of computational feasibility and statistical accuracy can be met by the LASSO estimator.

For a properly chosen value of $\lambda > 0$, it is needed to solve the convex optimization problem (2.3). The computational complexity of the ordinary LASSO is $O(np\min\{n,p\})$, as there are $m = O(\min\{n,p\})$ steps, each of complexity $O(np)$. Its complexity corresponds with $O(n^2p)$ in the $p > n$ framework. For this purpose, there are several efficient algorithms

like the typical coordinate descent method, the FISTA algorithm, or the famous LARS procedure (Efron et al. (2004)). See, for example, Giraud (2014) for more details. A proper selection of the λ parameter is treated in Section 2.2.4.

As commented above, it is noticeable that the problem (2.3) is equivalent to the basis pursuit denoising (Chen et al. (2001), Candes et al. (2006), Donoho et al. (2005)). This last is a problem well studied in mathematical signal processing given by formulation

$$\begin{aligned} &\min_{\beta} \|\beta\|_1, \\ &\text{subject to } \|y - X\beta\|_2 \leq \theta. \end{aligned} \tag{2.4}$$

The LASSO problem has been widely studied over the last years owing to its good statistical properties. See, for example, the review of Tibshirani (2011). It has been shown that this procedure is consistent in terms of prediction (see Van De Geer and Bühlmann (2009) for an extensive analysis), and this guarantees consistency of the parameter estimates at least in a L_2 sense (Van De Geer and Bühlmann (2009), Meinshausen and Yu (2009), Candes and Tao (2007)); besides, this is a consistent variable selector (Meinshausen and Bühlmann (2006), Wainwright (2009), Zhao and Yu (2006)).

In spite of all these good qualities, the LASSO regression has some important limitations in practice (see for example Zou and Hastie (2005) or Su et al. (2017)). These limitations are analyzed in the next section.

2.2 Analysis of the LASSO regression requirements and inconveniences

In this section, we introduce the requirements and problems of the LASSO regression. These are related to the inherent bias of the LASSO estimator (Section 2.2.1), the imposition of necessary conditions over the design matrix as well as the vector of parameters to guarantee consistency (Section 2.2.2), a large number of false discoveries (Section 2.2.3) and the proper selection of the λ value (Section 2.2.4). These limitations are analyzed in the subsequent sections, collecting some theoretical properties recently developed and displaying how far it is possible to ensure its good behavior.

2.2.1 Biased estimator

In the context of having more covariates p than a number of samples n, it depends on the class of optimization algorithm employed to solve (2.3) that the LASSO regression can identify more than n relevant covariates. For example, resorting to algorithms as the coordinate descent (see, for example, Section 4.2.4 of Giraud (2014)) allows this to select until p covariates. However, using the LARS algorithm of Efron et al. (2004), the LASSO procedure can pick at most n variables before this saturates (see Zou and Hastie (2005)). This restriction is usual for almost all regression adjustment methods that appeal to penalizations in this framework, especially for those based on L_1 ideas.

Another caveat of penalization processes is their bias, which produces higher prediction errors. In the LASSO adjustment, the imposition of the L_1 penalization in the OLS

problem (1.2) as a safe passage to estimate β has a cost. This payment translates into bias (see Chapter 3 of Hastie et al. (2009), Chapter 4 of Giraud (2014) or Chapter 2 of Hastie et al. (2015)). This can be easily illustrated under orthogonal design, where the L_1 penalization results in a perturbation of the unbiased OLS estimator $\hat{\beta}^{OLS}$ given by

$$\hat{\beta}_j^{LASSO} = \text{sign}(\hat{\beta}_j^{OLS})(|\hat{\beta}_j^{OLS}| - \lambda)_+, \tag{2.5}$$

where $\text{sign}(\cdot)$ denotes the sign of the coefficients and $(\cdot)_+$ equals to zero all quantities which are not positive. This results in a soft threshold of the ordinary mean square estimator ruled by the $\lambda > 0$ parameter, where the coefficients $|\hat{\beta}_j^{OLS}| \leq \lambda$ are adjusted to zero.

In order to correct the bias, it is usual to apply a two-step LASSO-OLS procedure: first, we employ a LASSO regression to select variables, and then, we obtain a least squared estimator using the selected variables. The properties of this procedure are studied in Belloni and Chernozhukov (2013).

Other options are weighted versions of the LASSO method based on iterative schemes. An example is the popular adaptive LASSO (Zou (2006), Huang et al. (2008), Van de Geer et al. (2011)). This procedure gives different weights to each covariate in the penalization part, readjusting these in every step of the iterative process until convergence. More details on these procedures are given in Section 2.3.1.

2.2.2 Consistency of the LASSO: neighborhood stability condition

Despite the fact that the LASSO is a broadly employed procedure, it is not always possible to guarantee its proper performance as a variable selector in practice (Bunea (2008), Lounici (2008)). As we can see in Bühlmann and Van De Geer (2011), certain conditions are required to guarantee an efficient screening property for variable selection. However, this presents some important limitations as a variable selector when these do not hold.

For example, when the model has several highly correlated covariates with the response, LASSO tends to pick only one or a few of them randomly and shrinks the rest to 0 (see Zou and Hastie (2005)). This fact results in a confusion phenomenon if there are high correlations between relevant and unimportant covariates, and in a loss of information when the subset of important covariates have a strong dependence structure. Some algorithms that result in non-sparse estimators try to relieve this effect, like the RIDGE regression (Hoerl and Kennard (1970)) or the Elastic Net (Zou and Hastie (2005)). An interpretation of their penalties is displayed in Figure 2.2.

Besides, denoting $S = \{j : \beta_j \neq 0\}$ the set of non-zero real values, for consistent variable selection using $\hat{S}^{L_1} = \{j : \hat{\beta}_j^{LASSO} \neq 0\}$, the design matrix of the model, $\mathbf{X_n}$, needs to satisfy some assumptions. The strongest of which is arguably the so-called "neighborhood stability condition" (Meinshausen and Bühlmann (2006)). This condition is equivalent to the irrepresentable condition (Zhao and Yu (2006); Zou (2006); Yuan and Lin (2007)):

$$\max_{j \in S^c} |\text{sign}(\beta_S)^\top (\mathbf{X_{nS}}^\top \mathbf{X_{nS}})^{-1} \mathbf{X_{nS}}^\top X_j| \leq \theta \quad \text{for some } 0 < \theta < 1, \tag{2.6}$$

being β_S the subvector of β, S^c the complementary of S and X_S the submatrix of X considering the elements of S.

If this condition is violated, all that we can hope for is a recovery of the regression vector β in an L_2-sense of convergence by achieving $\| \hat{\beta}^{LASSO} - \beta \|_2 \underset{n\to\infty}{\longrightarrow_p} 0$ (see Meinshausen and Bühlmann (2010) for more details). Moreover, under some assumptions in the design, the irrepresentable condition can be expressed as the called "necessary condition" (Zou (2006)). It is not an easy task to verify these conditions in practice, especially in contexts where p can be huge.

Quoting Bühlmann and Van De Geer (2011): roughly speaking, the neighborhood stability or irrepresentable condition (2.6) fails to hold if the design matrix X is too much "ill-posed" and exhibits a too strong degree of linear dependence within "smaller" sub-matrices of X.

In addition, we need to ensure that enough information and suitable characteristics are available for "signal recovery" of the sparse β vector. These conditions require coefficients of relevant covariates to be large enough to distinguish them from the zero ones. Then, the non-zero regression coefficients need to satisfy

$$\inf_{j\in S}|\beta_j| > \sqrt{s\log(p)/n}, \tag{2.7}$$

where $s = \#S$ is the cardinal of S, in order to guarantee the consistency of the $\hat{\beta}^{LASSO}$ estimator of problem (2.3). This is called a beta-min condition. Nevertheless, this requirement may be unrealistic in practice, and small non-zero coefficients may not be detected (in a consistent way). See Bühlmann and Van De Geer (2011) for more information.

Eventually, related to all these requirements, it is important to remind that for all covariates selection procedures, an estimator $\hat{S}$ trying to recover S would be consistent if this verifies

$$\mathbb{P}(\hat{S} = S) \underset{n\to\infty}{\longrightarrow} 1. \tag{2.8}$$

The condition (2.8) places a restriction on the growth of the number p of variables and sparsity $s = \#S$, typically of the form $s\log(p) = o(n)$ (see Meinshausen and Bühlmann (2006)). Then, this forces the necessity of $n > s\log(p)$ in order to achieve consistency.

Bunea (2008) explains that, under mild assumptions, the LASSO verifies condition (2.8) and, in consequence, this is capable of selecting the relevant variables. However, one needs more assumptions, as the irrepresentable condition of (2.6), to verify the suitable recovering of S. This may explain why the LASSO overestimates the support of β.

Owing to these difficulties, different methodologies based on ideas derived from subsampling and bootstrap have been developed. Examples are the random LASSO (Wang et al. (2011)), an algorithm based on subsampling, or the stability selection method mixed with randomized LASSO of Meinshausen and Bühlmann (2010). This last searches for consistency, although the irrepresentable condition introduced in (2.6) would be violated. These are introduced in Section 2.3.2.

2.2.3 False discoveries of the LASSO

As explained in Su et al. (2017): In regression settings where explanatory variables have very low correlations and relatively few effects, each of great magnitude, we expect the LASSO to find the relevant variables with few errors, if any. Nevertheless, in a regime of linear sparsity, there exists a trade-off between false and true positive rates along the LASSO path, even when the design variables are stochastically independent. Besides, this phenomenon occurs no matter how strong the effect sizes are. By linear sparsity, one understands that the fraction of variables with a non-vanishing effect, i.e. $s = \#S$, tends to a constant, however small.

This existing trade-off between the false discovery proportion (FDP) and the true positive proportion (TPP) translates into one of the major disadvantages of using LASSO as a variable selector. These quantities are defined as

$$FDP(\lambda) = \frac{F(\lambda)}{\#\{j : \hat{\beta}_j(\lambda) \neq 0\} \vee 1} \quad \text{and} \quad TPP(\lambda) = \frac{T(\lambda)}{s \vee 1}, \tag{2.9}$$

where $F(\lambda) = \#\{j \in S^c : \hat{\beta}_j(\lambda) \neq 0\}$ denotes the number of false discoveries, $T(\lambda) = \#\{j \in S : \hat{\beta}_j(\lambda) \neq 0\}$ is the number of positive discoveries and $a \vee b = \max\{a, b\}$.

Then, it is unlikely to simultaneously achieve high power and a low false positive rate. Being FDP a natural measure of type I error, and $1 - TPP$ the fraction of missed signals (a natural notion of type II error), the results say that nowhere on the LASSO path can both types of error rates be simultaneously low. This also happens even for noiseless situations with stochastically independent regressors. Hence, there exists only a possible reason: it is because of the L_1 shrinkage that results in pseudo-noise. Furthermore, this does not occur with other types of penalizations, like the L_0 penalty. See Su et al. (2017) for more details.

In fact, it can be proved in a quite global context, that the LASSO is not capable of selecting the correct subset of important covariates without adding some noise to the model in the best case (see Wasserman and Roeder (2009) or Su et al. (2017)).

Then, modifications of the traditional LASSO procedure are needed to control the FDP. Some alternatives, such as the boLASSO procedure (see Bach (2008)), which uses bootstrap to calibrate the FDP, the thresholded LASSO (Lounici (2008), Zhou (2010)), based on the use of a threshold to avoid noisy covariates or more recent ones, like the stability selection method (see Meinshausen and Bühlmann (2010)) or the use of knockoffs (see Hofner et al. (2015), Weinstein et al. (2017), Candes et al. (2018) and Barber and Candès (2019)), were proposed to solve this drawback. To the best of our knowledge, there is no version of this last for the $p > n$ framework yet. These modifications and alternatives are presented along Sections 2.3.2, 2.3.3, and 2.4.

2.2.4 Correct selection of the penalization parameter λ

One of the most important parts of a LASSO adjustment is the suitable selection of the penalization parameter $\lambda \geq 0$. Its size controls both: the number of selected variables and

the degree to which their estimated coefficients are shrunk to zero, ruling the bias as well. A too-large value of λ forces all coefficients of $\hat{\beta}^{LASSO}$ to be null, while a quantity next to zero includes too many noisy covariates. Then, a good choice of λ is needed in order to achieve a balance between simplicity and selection accuracy. See the work of Lahiri (2021) for a current analysis of the required λ conditions.

It is essential to highlight that it is possible to understand variable selection in two different ways: trying to identify the right set of relevant covariates or applying dimension reduction in order to improve predictions without guaranteeing the recovery of the true model. It is well-known that both are not compatible. See, for example, Yang (2005)). Thus, we need to pick one of them. In the LASSO case, for example, the optimal value of the penalization parameter λ may not be the same for both objectives (Leng et al. (2006), Bühlmann and Van De Geer (2011)). Besides, some drawbacks for the real recovery of the not null elements of the β vector are less harmful to the prediction accuracy target. See, for example, Dalalyan et al. (2017).

The problem of the proper choice of the λ parameter depends on the unknown error variance σ^2. We can see in Bühlmann and Van De Geer (2011) that the oracle inequality states to select λ of order $\sigma\sqrt{\log(p)/n}$ to keep the mean squared prediction error of LASSO as the same order as if we knew the active set S in advance. In practice, the σ value is unknown, and its estimation with $p > n$ is quite complex. To give some guidance on this topic, we refer to Fan et al. (2012) or Reid et al. (2016). However, σ estimation for $p > n$ is still a growing study field.

Thus, other methods to estimate λ are proposed in the literature. Following the classification of Homrighausen and McDonald (2018), we can distinguish three categories: minimization of a generalized information criterion (like AIC or BIC), using resampling procedures (such as cross-validation or bootstrap) or reformulating the LASSO optimization problem. Due to computational cost, the most used criteria to fit a LASSO adjustment are cross-validation techniques. Nevertheless, it can be shown that this criterion achieves a suitable λ value for prediction risk, but this leads to inconsistent model selection for sparse methods (see Meinshausen and Bühlmann (2006)). Then, for recovering the set S, a larger penalty parameter would be needed (Bühlmann and Van De Geer (2011)).

Su et al. (2017) argue that the LASSO estimator is seriously biased downwards when the regularization parameter λ is needed to be large for a proper variable selection. The residuals still contain much of the effects associated with the selected variables, and this phenomenon is called shrinkage noise. As many strong variables get picked up, this gets inflated, and its projection along the directions of some of the null variables may actually dwarf the signals coming from the strong regression coefficients, selecting null variables.

Nevertheless, to the best of our knowledge, there is no mutual agreement about how to choose the λ value. Hence, cross-validation techniques are widely used to adjust the LASSO regression. See Homrighausen and McDonald (2018) for more details.

Modifications of the LASSO algorithm as the square-root LASSO (Belloni et al. (2011)), which does not need to know σ to obtain an optimal λ, the work of Städler et al. (2010) or the scaled LASSO (Sun and Zhang (2012)), which simultaneously estimate σ and β, have

been proposed to relieve these inconveniences. A complete survey on this topic is carried out in Giraud et al. (2012). The square-root LASSO and the scaled LASSO are introduced in Section 2.4.

2.3 Evolution of the LASSO in the last years

Once all the inconveniences of the standard LASSO have been introduced in Section 2.2, the necessity of finding modifications or alternatives arises. For this purpose, a review of the existing literature is carried out in this section, where different methodologies that solve these issues are introduced and analyzed. Nevertheless, it is impossible to include all the existing algorithms here. Instead, we attempted to collect the most relevant ones. As a result, a summary of the most innovative and used methodologies nowadays is provided.

Methods proposed to alleviate the limitations of the LASSO algorithm employ a wide range of different philosophies. Some of them opt to add a second selection step after solving the LASSO problem, such as the relaxed LASSO (Meinshausen (2007)) or thresholded LASSO (Lounici (2008), Zhou (2010), Van de Geer et al. (2011)). Other alternatives focus on giving different weights to the covariates proportional to their importance, such as the adaptive LASSO (Zou (2006), Huang et al. (2008), Van de Geer et al. (2011)), and some techniques pay attention to the group structure of the sparse vector β when this exists, like the grouped LASSO procedure (Yuan and Lin (2006)) or the fused LASSO (Tibshirani et al. (2005)) to say a few.

The resampling or iterative procedures are other approaches that make use of subsampling or computational power, algorithms like boLASSO (Bach (2008)), stability selection with randomized LASSO (Meinshausen and Bühlmann (2010)), the random LASSO (Wang et al. (2011)), the scaled LASSO (Sun and Zhang (2012)) or the combination of traditional estimators with variable selection diagnostics measures (Nan and Yang (2014)), among others, are based on this idea. Furthermore, more recent techniques, like the Knockoff filter (Barber and Candès (2015), Candes et al. (2018)) or SLOPE (Bogdan et al. (2015)), have been introduced to control some measures of the type I error. However, as far as we know, the Knockoff filter is not yet available for the $p > n$ case.

Other alternatives modify the constraints of the LASSO problem (2.3) in order to achieve better estimators of β, like the Elastic Net (Zou and Hastie (2005)) or the Dantzig selector (Candes and Tao (2007)). Other different options have been developed recently, such as the Elem-OLS Estimator (Yang et al. (2014)), the LASSO-Zero (Descloux and Sardy (2021)), the spike-and-slab LASSO (Ročková and George (2018)) or some Bayesian approaches (see for example Castillo et al. (2015) or Bhadra et al. (2019)).

Quoted Descloux and Sardy (2021): although differing in their purposes and performance, the general idea underlying these procedures remains the same. Roughly speaking, to avoid overfitting by finding a trade-off between the fit $y - X\beta$ and some measure of the model complexity.

Along the many papers, we have found that a modest classification of the different proposals can be done, although, in this classification, some of the procedures do not only

fit in a single class. These categories are

- **Weighted LASSO:** weighted versions of the LASSO that attach the particular importance of each covariate for a suitable selection of the weights. Joint with iteration, this modification allows a reduction of the bias.
- **Resampling LASSO procedures:** mix of the LASSO adjustment with resampling procedures for randomizing the covariates selection process to reduce unavoidable random noise.
- **Thresholded versions of the LASSO:** a second thresholding step in the covariates selection is implemented in order to reduce the irrelevant ones.
- **Alternatives to the LASSO:** procedures with different nature and aims designed to solve the LASSO drawbacks.

This extensive list of procedures makes noticeable the impact the LASSO has nowadays. A summary is displayed in Table 2.1 at the end of Section 2.4.

Next, we provide details about the weighted LASSO, LASSO versions using resampling ideas, a threshold version and special structures of the LASSO. Alternatives to the LASSO philosophy are analyzed in Section 2.4.

2.3.1 Weighted LASSO

In the LASSO procedure (2.3), all the covariates are penalized by the same quantity, $\lambda > 0$. So, assuming that all the explanatory variables are on the same scale, every covariate has the same prior importance as the rest. Then, the algorithm determines which covariates enter the model based on their estimated values $\hat{\beta}_j$.

Nevertheless, we may want to make a difference between the covariates in terms of their importance or scale. For this purpose, we would need to define a different penalization term for each covariate. This regularization could be expressed as $\lambda_j = \lambda w_j$, where $\lambda > 0$ is a common term and $w_j > 0$ are specific weights of each covariate. Following this idea, the penalized mean square problem of (2.3) is reformulated as

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j| \right\}. \tag{2.10}$$

Accordingly, we should search for $j = 1, \ldots, p$ penalization parameters. It is possible to use preliminary information to define the weights w_j, like in the adaptive LASSO algorithm (Zou (2006), Huang et al. (2008)) or an external information criterion, see for example Bergersen et al. (2011). Then, the problem becomes again a LASSO regression adjustment for which we know how to obtain a solution.

Next, we introduce some examples of these algorithms and explain how to implement them for the $p > n$ context of interest.

Adaptive LASSO algorithm

The adaptive LASSO procedure was proposed by Zou (2006) and Huang et al. (2008). This procedure is based on the idea of using adaptive weights for penalizing different coefficients in the L_1 penalty of (2.3). So, for an initial vector $\hat{\beta}$, the problem (2.10) is solved several times, recomputing the weights w_j of the $\hat{\beta}$ vector until convergence. As a result, this iterative procedure helps to reduce the bias suffered by the LASSO algorithm.

If we take a $\hat{\beta}$ $\sqrt{n}$-consistent estimator of β, for example that obtained through the OLS estimator of (1.2), choose a value $\gamma > 0$ and define the weights vector $w = 1/|\hat{\beta}|^{\gamma}$, the adaptive LASSO estimator is given by the expression

$$\hat{\beta}^{AdapL} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_j|^{\gamma}} \right\}. \tag{2.11}$$

In this context, we need to estimate an optimal value for two parameters, $\lambda > 0$ and $\gamma > 0$, which control the penalization and the weighting, respectively.

Another advantage of the adaptive LASSO is that its good characteristics guarantee that, for a proper selection of the $\hat{\beta}$ estimator to define the weights w_j, this procedure enjoys the oracle properties (see Zou (2006) for more information).

In order to estimate the weights vector $w = 1/|\hat{\beta}|^{\gamma}$, Zou (2006) suggests using the estimator $\hat{\beta}^{OLS}$ of (1.2) unless collinearity is a concern, in which case we can try the RIDGE estimator, $\hat{\beta}^{RIDGE}$ (Hoerl and Kennard (1970)). This is due to the fact that, in practice, adaptive LASSO with $\hat{\beta}^{OLS}$ suffers from the multicollinearity caused by strong correlations among covariates. Using the RIDGE coefficients as initial weights helps to keep the stability of the process.

In a high dimensional problem with $p > n$, it is nontrivial to find an initial consistent estimator for constructing the weights of (2.11). A practical solution is to use the RIDGE estimator to guarantee that the adaptive LASSO is well-defined. Note that, in this case, an extra tuning parameter is included in the procedure. This parameter is devoted to correctly estimating the penalization term of the RIDGE regression. Huang et al. (2008) study other initial estimators, and the consistency of the procedure is proved.

Again, due to the intrinsic constraint of the L_1-norm penalty, the number of variables selected by the adaptive LASSO cannot exceed n when $p > n$. This procedure can be implemented using schemes like the one proposed in the Algorithm 2.1.

Algorithm 2.1 (Adaptive LASSO)**.**

1. Compute the RIDGE regression estimator $\hat{\beta}^{RIDGE}$ and the weights $w_j = 1/|\hat{\beta}_j^{RIDGE}|^{\gamma}$ for a given $\gamma > 0$, $\forall j = 1, \ldots, p$.

2. Define $x_j^* = w_j x_j$, $\forall j = 1, \ldots, p$.

3. Apply iteratively the procedure

i) Solve the LASSO problem for $\lambda > 0$:

$$\hat{\beta}^* = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}^* \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

ii) Update $w_j = 1/|\hat{\beta}_j^*|^\gamma$ and $x_j^* = w_j x_j$, $\forall j = 1, \ldots, p$.

4. Output $\hat{\beta}_j^{\text{AdapL}} = w_j \hat{\beta}_j^*$, $\forall j = 1, \ldots, p$.

Other variations of the adaptive LASSO procedure considering different weight functions have been proposed over the last few years. Some of these modify the penalization weights w_j using the information of the data, like the correlation between the variable of interest, Y, and the explanatory covariates (Jiang et al. (2014)). In contrast, other authors include external information that they consider relevant to pick covariates under a certain criterion (Bergersen et al. (2011)). However, as far as we know, the consistency properties of this type of adaptive LASSO procedure have not yet been proved in the $p > n$ context. As a result, we recommend using these approaches "carefully".

2.3.2 Resampling LASSO procedures

One may think of creating an indicator to decide when a covariate is included just because of randomness or if this is really important. This implementation would help to reduce the number of irrelevant covariates randomly selected by the LASSO. For this purpose, it is possible to resort to resampling procedures, such as bootstrap (Tibshirani and Efron (1993)). Then, subsamples of size $m < n$ are selected, and a LASSO model is adjusted with these. This procedure is repeated a total of B times. Next, we count the number of times that each covariate has been selected and decide if the quantity is large enough to include this in the final model.

Thus, we can define a criterion based on the results of the B repetitions to determine the importance of every covariate. Some examples are selecting only those which, in mean, have an associated coefficient $\hat{\beta}_j$ big enough (Wang et al. (2011)) or defined an adequate "cut point" (Meinshausen and Bühlmann (2010)) to discriminate between the relevant covariates and the noise.

As a trade-off, these procedures required a significant increment in computational complexity and time. Furthermore, a proper selection criterion is needed.

Next, some examples of these types of algorithms are displayed. These are the BoLASSO (Bach (2008)), the random LASSO (Wang et al. (2011)) and the Stability selection technique with randomized LASSO (Meinshausen and Bühlmann (2010)).

BoLASSO algorithm

From an asymptotic analysis of model consistency point of view, the LASSO selects all the variables that should enter the model with probability tending to one exponentially

fast. Instead, the rest of the covariates are only guaranteed to be selected with strictly positive probability. Therefore, if several data sets generated from the same distribution were available, this last property would suggest considering the intersection of all supports of the LASSO estimates. Thus, all relevant variables would always be selected for all data sets, while the irrelevant ones would enter the model randomly. As a result, the intersection would eliminate the noise. However, it is common to have a single data set in practice. Resampling methods, such as the bootstrap, are dedicated to mimicking the availability of these data sets. The boLASSO (bootstrap-enhanced least absolute shrinkage operator) of Bach (2008) carries out this idea.

It can be seen in Bach (2008) that the use of this procedure gets a consistent model estimate, without the consistency condition required by the regular LASSO given in (2.6). As explained in Bach (2008), the boLASSO procedure is consistent for a proper selection of the penalization parameter. Indeed, in this situation, the correct signs of the relevant variables (those in S) are recovered with probability tending to one. Nevertheless, all possible sign patterns consistent with the true configuration are also recovered, i.e. all other variables (those not in S) may be non-zero with asymptotically strictly positive probability. See Bach (2008) for more information.

Therefore, for a given $(\mathbf{X_n}, \mathbf{Y_n}) = \{(x_i, y_i), i = 1, \ldots, n\}$ iid sample from the joint distribution function of $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, a total of B bootstrap replications of the n data points are considered (Tibshirani and Efron (1993)). That is, for $b = 1, \ldots, B$, it is generated a ghost sample $(\mathbf{X_n^{(b)}}, \mathbf{Y_n^{(b)}}) = \left\{\left(x_i^{(b)}, y_i^{(b)}\right), i = 1, \ldots, n\right\}$. The n pairs $\left(x_i^{(b)}, y_i^{(b)}\right)$ result from randomly sampling with the original data points. Then, for each of the $b = 1 \ldots, B$ resamples, the support $S^{(b)} = \{j : \hat{\beta}^{(b)} \neq 0\}$ is obtained and the true support is estimate by $S = \bigcap_{b=1}^{B} S^{(b)}$. Once S is selected, we estimate β by the OLS procedure restricted to variables in S. This procedure is summarized in Algorithm 2.2.

Algorithm 2.2 (BoLASSO)**.** For a given data $(\mathbf{X_n}, \mathbf{Y_n}) \in \mathbb{R}^{n \times p}$, number of bootstrap replicates B and regularization parameter λ:

1. For $b = 1, \ldots, B$:
   i) Generate bootstrap samples $(\mathbf{X_n^{(b)}}, \mathbf{Y_n^{(b)}}) \in \mathbb{R}^{n \times p}$.
   ii) Compute the LASSO estimate $\hat{\beta}$ from $(\mathbf{X_n^{(b)}}, \mathbf{Y_n^{(b)}})$.
   iii) Compute support $S^{(b)} = \{j : \hat{\beta}^{(b)} \neq 0\}$.
2. Compute $S = \bigcap_{b=1}^{B} S^{(b)}$.
3. Compute $\hat{\beta}_S$ from $(\mathbf{X_{nS}}, \mathbf{Y_{nS}})$.

One drawback of this methodology is that this has been developed under the assumption that there are more observations than variables, i.e. in the $n > p$ framework. Next, we propose alternatives based on resampling and apt to the high dimensional framework.

Random LASSO algorithm

Another approach in this line is the random LASSO algorithm proposed by Wang et al. (2011). This method is a modification of the LASSO procedure based on subsampling. As said by the authors, this procedure can handle highly correlated variables more flexibly than RIDGE regression (Zou and Hastie (2005)), especially when their effects have different magnitudes and signs. This can also select more variables than the sample size n in the $p > n$ framework.

As we saw in Section 2.2, one of the limitations of LASSO is that this procedure selects only a set of essential variables when all of these are highly correlated. Thus, if we generated several independent data sets from the same distribution, we would expect LASSO to select nonidentical subsets of those relevant variables strongly correlated among them. Then, taking the union of the selected covariates in every data set, the final collection may be most, or even all, of the relevant variables. Such a process may yield more than n covariates when $p > n$, overcoming other limitations of the LASSO.

Since only a single data set is available in practice, bootstrap techniques are needed. Then, we can randomly select q candidate variables, with $q \leq p$, for each bootstrap sample. This process becomes the basic idea of the proposed random LASSO approach of Wang et al. (2011).

In Wang et al. (2011), the Algorithm 2.3 is proposed to implement this methodology.

Algorithm 2.3 (Random LASSO)**.**

1. Generating importance measures for all coefficients:

   i) Draw B bootstrap samples with size n by sampling with replacement from the original training data set.

   ii) For each b_1^{th} bootstrap sample, $b_1 \in \{1, \ldots, B\}$, randomly select $q^{(b_1)} \leq p$ candidate covariates, and apply LASSO to obtain estimators $\hat{\beta}_j^{(b_1)}$ for $j = 1, \ldots, p$. Estimators are zero for coefficients of the not selected covariates $(p - q^{(b_1)})$ or the ones excluded by LASSO.

   iii) Compute the importance measure of every X_j by $w_j = |B^{-1} \sum_{b_1=1}^{B} \hat{\beta}_j^{(b_1)}|$.

2. Selecting variables:

   i) Draw another set of B bootstrap samples with size n by sampling with replacement from the original training data set.

   ii) For each b_2^{th} term of the new bootstrap sample, $b_2 \in \{1, \ldots, B\}$, select again $q^{(b_2)} \leq p$ candidate covariates with a selection probability proportional to its importance w_j. Next, apply LASSO (or Adaptive LASSO) to obtain estimators $\hat{\beta}_j^{(b_2)}$. Estimators are zero for coefficients associated with covariates outside the subset of $q^{(b_2)}$ elements or excluded by LASSO.

   iii) Compute the final estimators as $\hat{\beta}_j = B^{-1} \sum_{b_2=1}^{B} \hat{\beta}_j^{(b_2)}$.

Now, there is the added difficulty of choosing suitable values for $q^{(b_1)}$, $q^{(b_2)}$, and B. The procedure may return incorrect results if these parameters are not well defined.

For variable selection with random LASSO, since the final estimator is the average of all bootstrap samples, it is very easy for a covariate to have a nonzero coefficient. To solve this problem, Wang et al. (2011) introduce a threshold δ_n, and consider a variable X_j to be selected, only if the corresponding coefficient verifies $|\hat{\beta}_j| > \delta_n$. In this article, they choose $\delta_n = 1/n$. With the addition of this threshold value, one needs to estimate another parameter, increasing the problem's complexity.

Stability selection with randomized LASSO algorithm

The stability selection procedure is proposed by Meinshausen and Bühlmann (2010). This technique is introduced as a procedure based on subsampling in combination with high dimensional selection algorithms. This approach provides finite sample control for some error rates of false discoveries. Hence, this is a transparent principle to choose a proper amount of regularization for a suitable structure estimation or relevant covariates recovery.

Specifically, the stability selection is introduced using the LASSO as a selector. Thus, for every value λ in a positive path $\Lambda \in \mathbb{R}^+$, we obtain what is denoted as a structure estimate $\hat{S}^\lambda \subseteq \{1, \ldots, p\}$. Then, it is interesting to determine whether there is a $\lambda \in \Lambda$ value such that $\hat{S}^\lambda$ is identical to S with high probability and how to achieve that right amount of regularization.

Stability paths are derived from the concept of regularization paths. A regularization path is given by the coefficient values of each variable over all regularization parameters: $\{\hat{\beta}_j^\lambda; \lambda \in \Lambda, j = 1, \ldots, p\}$. Stability paths are, in contrast, the probability for each variable to be selected when randomly resampling from the data. For any given regularization parameter $\lambda \in \Lambda$, the selected set $\hat{S}^\lambda$ is implicitly a function of the samples $L = \{1, \ldots, n\}$. We write $\hat{S}^\lambda = \hat{S}^\lambda(L)$ where necessary to express this dependence.

Definition 2.1. (Selection probabilities) Let L be a random subsample of $\{1, \ldots, n\}$ of size $\lfloor n/2 \rfloor$, drawn without replacement. For every set $M \subseteq \{1, \ldots, p\}$, the probability of being in the selected set $\hat{S}^\lambda(L)$ is

$$\hat{\Pi}_J^\lambda = \mathbb{P}^*\{M \subseteq \hat{S}^\lambda(L)\}.$$

Thus, for every variable $j = 1, \ldots, p$, the stability path is given by the selection probabilities $\hat{\Pi}_j^\lambda$. We remind that variable selection would be equivalent to choosing an element of the set of models $\{\hat{S}^\lambda; \lambda \in \Lambda\}$ in a traditional setting. Therefore, there are typically two problems. First, the correct model S might not be a member of this set. Second, even if this model is contained, it is typically challenging for high dimensional data to determine the right amount of regularization λ to select exactly S, or, at least, a close approximation. Nevertheless, the stability selection approach proposes perturbing the data many times and then choosing all structures or variables that occur in a large fraction of the resulting selection sets to alleviate this drawback. For this purpose, one selects what they denote as stable variables, keeping only variables with a high selection probability.

Definition 2.2. (Stable variables) For a cut-off δ, with $0 < \delta < 1$, and a set of regularization parameters Λ, the set of stable variables is defined as

$$\hat{S}^{stable} = \{j : \max_{\lambda \in \Lambda}(\hat{\Pi}_j^{\lambda}) \geq \delta\}.$$

The exact cut-off δ, with $0 < \delta < 1$, is a new tuning parameter to determine. In practice, quoting Meinshausen and Bühlmann (2010), results tend to be quite similar for sensible values in the range $\delta \in (0.6, 0.9)$.

Furthermore, when one tries to recover the set S, a natural goal is to include as few noisy variables as possible. Hence, the choice of the regularization parameter is crucial. An advantage of this method is that the selection of the initial set of regularization parameters Λ has not got a pretty strong influence on the results typically, as long as Λ varies within reason. Another advantage is its ability to choose this set of regularization parameters in a way that guarantees, under stronger assumptions, some bound on the expected number of false selections. Specifically, this algorithm enables us to control the per-family error rate ($PFER$), defined as $\mathbb{E}(V)$ where V is the expected number of falsely selected variables, employing an upper bound (see the *Additional file 1* of Hofner et al. (2015)).

Then, controlling the $PFER$ is a (very) conservative approach for controlling errors in multiple testing situations. Hence, a procedure that controls the $PFER$ at a certain level α also controls other error rates such as the per-comparison error rate ($PCER$), the family-wise error rate ($FWER$) or the false discovery rate (FDR). We refer to Meinshausen and Bühlmann (2010) for more details.

As a result, if we understand the selection of relevant variables like a hypothesis test, where we test if $H_0 : \beta_j = 0$, $\forall j = 1, \ldots, p$, we can fix a value $\alpha \in [0, 1]$ as the proportion of misclassification allowed in the adjustment. Meinshausen and Bühlmann (2010) show that choosing $PFER = \alpha p$ is a suitable option to control this error rate and verify that the model includes, at most, a percentage α of noisy covariates.

Additionally, Meinshausen and Bühlmann (2010) propose to use the stability selection criterion mixed with what they call randomized LASSO. As the name suggests, this algorithm is a modification of the LASSO, adding randomness to the selection procedure. In particular, the randomized LASSO changes the penalty λ to a randomly chosen value in a range $[\lambda, \lambda/\alpha]$. Therefore, this new selection adds more complexity to the estimation procedure because one has to determine a suitable value for α. Meinshausen and Bühlmann (2010) propose to choose this in the range $(0.2, 0.8)$. As a result, what Meinshausen and Bühlmann (2010) propose in their article is to apply a stability selection procedure over a randomized LASSO adjustment. This process is claimed to be consistent for variable selection even though the "irrepresentable condition" of (2.6) is violated. An example of the implementation scheme of this procedure is introduced in the Algorithm 2.4.

Algorithm 2.4 (Randomized LASSO)**.**

1. Compute the LASSO regression estimator $\hat{\beta}^{LASSO}$ and the weights $w_j = 1/\alpha$, for α, with probability $p_w \in (0,1)$ and $w_j = 1$ otherwise, $\forall j = 1, \ldots, p$.
2. Define $x_j^* = w_j x_j$, $\forall j = 1, \ldots, p$.
3. Apply iteratively the procedure:
   i) Solve the LASSO problem for $\lambda > 0$:
   $$\hat{\beta}^* = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}^* \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$
   ii) Update $w_j = 1/\alpha$, with probability p_w or $w_j = 1$ otherwise, and $x_j^* = w_j x_j$, $\forall j = 1, \ldots, p$.
4. Output $\hat{\beta}_j^{\text{Rand}} = w_j \hat{\beta}_j^*$, $\forall j = 1, \ldots, p$.

A proposal for the distribution of the weights w_j is to take $w_j = \alpha$ with a given probability $p_w \in (0,1)$ and $w_j = 1$ otherwise.

As a result, a combination of the randomized LASSO and the stability selection procedure is capable of overcoming the problem of the LASSO trade-off between the FDP and the TPP. However, this methodology has several disadvantages. Apart from the computational cost, the estimation of several tuning parameters that hinders its correct implementation is needed.

2.3.3 Thresholded LASSO

Due to the huge amount of false positives, the LASSO tends to include in the model, a two-step procedure is proposed to select only those covariates that provide important information. Firstly, a screening procedure that reduces the possible subset of relevant covariates is performed and the β vector is calculated using this information. Secondly, to guarantee that all selected covariates are important, a threshold is employed in a second step, establishing a cut point over the absolute value of each component of $\hat{\beta}$. Eventually, the β vector is estimated considering only the selected covariates. As in the $p > n$ framework, a sparse estimator is needed for the first step, so one can resort to the LASSO estimator $\hat{\beta}^{LASSO}$ as a screening method. This scheme gives place to the thresholded LASSO algorithm (Lounici (2008), Zhou (2010), Van de Geer et al. (2011)).

Thresholded LASSO algorithm

The thresholded LASSO algorithm (Lounici (2008), Zhou (2010), Van de Geer et al. (2011)) proposes to implement a thresholded version of the LASSO estimator. As a result, one

would obtain a vector $\hat{\beta}^{Thr}$, computing the OLS estimator over the parameters which verify $\{j : |\hat{\beta}^{LASSO}| > \delta\}$ for a given value $\delta > 0$. Now, the challenge is how to define the threshold value and guarantee good statistical properties for this methodology, as well as the correct recovery of S. For this last, we expect the thresholded version to require weaker conditions than the ones the classical LASSO requires. See, for example, Tardivel and Bogdan (2022) and references therein.

Additionally, the thresholded version of the LASSO guarantees the recovery of the signs under milder assumptions than those of the classical LASSO regression. A review of this issue, comparing LASSO and thresholded LASSO requirements and collecting a complete list of references about this topic, can be found in Tardivel and Bogdan (2022).

Concerning the estimation of the threshold value, different approaches can be employed. A first option is to select δ as a given quantile of the $\{\hat{\beta}_j^{LASSO}\}_{j=1}^p$ terms, similar to Descloux and Sardy (2021) ideas. However, no theoretical guarantees are given for proper selection following these guidelines. A more conservative way to proceed is to define the cut-off in terms of the model error variance. An example is the work of Zhou (2010), extending theoretical results regarding the adequate selection of the LASSO penalty for this purpose. In Zhou (2010), authors propose to obtain the thresholded LASSO estimator following the iterative procedure displayed in Algorithm 2.5.

Algorithm 2.5 (Thresholded LASSO)**.**

1. Obtain an initial estimator using the usual LASSO procedure $\hat{\beta}^{LASSO}$. Let $\hat{S}_0 = \{j : |\hat{\beta}_j^{LASSO}| > \lambda_n\}$ and $\hat{\beta}^{(0)} := \hat{\beta}^{LASSO}$, being $\lambda_n = 0.69\lambda\sigma$ and $\lambda = \sqrt{2\log(p)/n}$.

2. Iterate through the following steps twice, for $i = 0, 1$:

   i) Set $L := \hat{S}_i$, compute $\hat{\beta}_L^{(i)} = (\mathbf{X_{nL}}^\top \mathbf{X_{nL}})^{-1}\mathbf{X_{nL}}^\top \mathbf{Y_n}$ and $\delta_i = \sigma\lambda$.

   ii) Threshold $\hat{\beta}_L^{(i)}$ with δ_i to obtain $L := \hat{S}_{i+1}$ where

   $$\hat{S}_{i+1} = \left\{ j \in \hat{S}_i : |\hat{\beta}_{jL}^{(i)}| \geq \delta_i \right\}.$$

3. Compute $\hat{\beta}_L^{(2)} = (\mathbf{X_{nL}}^\top \mathbf{X_{nL}})^{-1}\mathbf{X_{nL}}^\top \mathbf{Y_n}$ and output $\hat{\beta}^{Thr} = \hat{\beta}_L^{(2)}$.

   Return the final set of variables in $\hat{S}_2$ and output $\hat{\beta}$ such that $\hat{\beta}_{\hat{S}_2} = \hat{\beta}_{\hat{S}_2}^{(2)}$ and $\hat{\beta}_j = 0$, $\forall j \in \hat{S}_2^c$, being $\hat{S}_2^c$ the complementary of S.

One important caveat of this implementation is the necessity of knowing the σ parameter in advance to define the threshold. However, this is not possible in practice. Thus, we notice that the complexity of finding a correct threshold is similar to obtaining the optimal value of λ for the LASSO adjustment. In both cases, we would need to know the dispersion of the error σ^2 in advance.

Next, related to the LASSO family, modifications that take advantage of special structures in the data sets are introduced and analyzed.

2.3.4 Special structures of the LASSO

There are contexts where the sparsity of the β vector of the problem (1.2) can have a special structure. Examples are when we can assume some order in the covariates $X_1, \ldots, X_p$ or when the sparsity is in terms of groups of variables. As a result, the LASSO approach fails to recover the correct sparsity structure, and new estimators are needed.

In those contexts spring up methods such as the fused LASSO of Tibshirani et al. (2005) or the group LASSO introduced by Yuan and Lin (2006). The fused LASSO technique uses the L_1 penalization philosophy of the LASSO, but this also adds a new constraint to the optimization problem incorporating the order of the considered covariates. In the case of the group LASSO algorithm, this penalizes the covariates by splitting them into groups. In this way, this achieves a group sparse estimator of β. Next, we briefly introduced three approaches that make use of these ideas: the fused LASSO (Tibshirani et al. (2005)), the group LASSO (Yuan and Lin (2006)), and the sparse-group LASSO (Simon et al. (2013)).

Fused LASSO

The fused LASSO was introduced by Tibshirani et al. (2005) for frameworks where quoting the authors: features can be ordered in some meaningful way. We can think of contexts where our data has a spatial or time series structure as simple examples. Besides, we can see that this approach can be useful when we want to establish an "artificial" order in our covariates. For this purpose, we can employ some measures of importance, like correlation.

Therefore, if we denote $X_{(1)}, \ldots, X_{(p)}$ as the ordered covariates set and $\beta_{(1)}, \ldots, \beta_{(p)}$ their associated coefficients to estimate, we can define the fused LASSO problem as

$$\hat{\beta}^{\mathrm{FL}} = \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{i(j)} \beta_{(j)} \right)^2,$$
$$\text{subject to} \sum_{j=1}^{p} |\beta_{(j)}| \leq \theta_1 \quad \text{and} \quad \sum_{j=2}^{p} |\beta_{(j)} - \beta_{(j)-1}| \leq \theta_2,$$

which can be rewritten in the compact form

$$\hat{\beta}^{\mathrm{FL}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{i(j)} \beta_{(j)} \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_{(j)}| + \lambda_2 \sum_{j=2}^{p} |\beta_{(j)} - \beta_{(j)-1}| \right\}, \quad (2.12)$$

being λ_1 and λ_2 the shrinkage parameters to estimate, which are inversely proportional to θ_1 and θ_2 terms.

The problem (2.12) is convex and tends to shrink the value of consecutive covariates to equal them up to a constant. A two-dimension interpretation is displayed in Figure 2.4. Now, one needs to estimate two regularization parameters: λ_1 and λ_2.

More discussion about asymptotic properties, computational approach, or degrees of freedom can be found in Tibshirani et al. (2005).

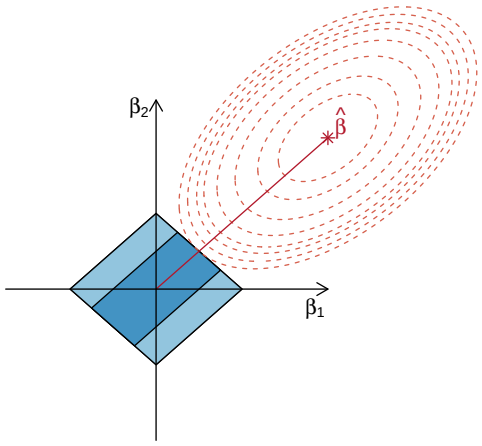


Figure 2.4: Interpretation of the fused LASSO problem in two dimensions. The light blue area denotes the LASSO restriction $|\beta_1| + |\beta_2| \leq \theta_1$, whereas the dark one corresponds with the penalization term $|\beta_2 - \beta_1| \leq \theta_2$. The orange ellipses are the contours of the mean square errors function.

Group LASSO

Sometimes, the high dimensional vector β carries a group structure partitioned into disjoint pieces. The group LASSO (Yuan and Lin (2006)) is designed to select grouped variables, which can be designated as factors instead of individual variables. The most well-known example is the multiple-factor analysis of variance or ANOVA. Another ordinary example is basis expansions in additive models. There, the selection of relevant variables corresponds to the choice of groups of basis functions.

Thus, the penalization term is applied over the grouped covariates. As a result, assuming a number K of groups, the group LASSO problem has the form

$$\hat{\beta}^{GL} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} X_k \beta_k \right)^2 + \lambda \sum_{k=1}^{K} \|\beta_k\|_{Z_k} \right\}, \tag{2.13}$$

where $\|w\|_{Z_k} = (w^\top Z_k w)^{1/2}$ for a vector $w \in \mathbb{R}^d$, $d \geq 1$, and Z_k is a symmetric $d \times d$ positive definite matrix Z_k, for $k = 1, \ldots, K$. To simplify, it is assumed that the X_k are orthonormalized, i.e. $X_k^\top X_k = I_{p_k}$, for $k = 1, \ldots, K$ and being p_k the number of covariates of the k factor, $\sum_{k=1}^{K} p_k = p$.

One can choose the Z_k matrices displayed in (2.13) as reproducing kernels of the functional space induced by the kth factor. See Yuan and Lin (2006) for more information. Besides, it is clear that, for the special case of $p_1 = \cdots = p_K = 1$, this problem corresponds with the ordinary LASSO adjustment of (2.3).

Furthermore, we can extend this idea of penalizing the group dependence to other approaches, such as the LARS procedure or the non-negative garrote.

More information about algorithms for solving these problems, similarities and differences, or even the estimation of the tuning parameter can be found in Yuan and Lin (2006) or Bühlmann and Van De Geer (2011).

Sparse-group LASSO

One drawback of the formulation (2.13) is that all the covariates of the non-used groups are forced to be zero. However, there are situations where we want not only sparsity by groups but within each group too. Examples are some genomics studies, where one is interested not only in which groups are important in explaining a particular disease, but also in the importance of particular genes in each group. With this objective, Simon et al. (2013) introduced the sparse-group LASSO. This procedure is based on solving the problem

$$\hat{\beta}^{SGL} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} X_k \beta_k \right)^2 + (1-\alpha)\lambda \sum_{k=1}^{K} \|\beta_k\|_{Z_k} + \alpha\lambda\|\beta\|_1 \right\}, \qquad (2.14)$$

where $\alpha \in (0,1)$. For $\alpha = 0$ we obtain the group LASSO fit (2.13) and for $\alpha = 1$ the classic LASSO problem (2.3). The rest of the parameters are the ones defined above.

This criterion resembles the RIDGE regression approach of Zou and Hastie (2005) with the L_2 penalty. Nevertheless, they differ in that the norm $\|.\|_{Z_k}$ is not differentiable at zero, and therefore, some groups are zeroed out completely.

More details about this methodology are given in Simon et al. (2013). They discuss its properties, present algorithms for solving the problem (2.14), study its extension to other contexts, and compare its performance with the LASSO and the group LASSO.

Once the possible modifications of the LASSO have been studied, we want to exploit other alternatives to improve the LASSO results. Thus, in the next section, we present different approaches.

2.4 Alternatives to the LASSO

In Section 2.2, the problems the LASSO selector has to deal with in practice were introduced, motivating the need for modifications or alternatives. Subsequently, in Section 2.3, modifications of the LASSO to solve some of these inconveniences were introduced and analyzed, although none can solve all drawbacks at once. In parallel with the LASSO adjustment, alternatives to this approach have been developed. Some of these still have a high impact on literature nowadays. As a result, it is of interest to review and consider some of these procedures as possible alternatives to the LASSO.

There are a lot of different approaches designed to select relevant information and adjust a regression model able to explain the behavior of the response variable. Especially, covariates selection in the linear regression model for the β sparse framework is the focus of the study. For this purpose, a few of the most popular methodologies concerning the comparison, or improvement of the LASSO, have been selected among the existing literature. In addition, we add some of the most recent covariate selection algorithms that

have been proven efficient and provide novel approaches over the last years, apart from the classic LASSO competitors.

Next, we briefly describe these procedures, arguing their good qualities, explaining their methodology, and analyzing some drawbacks. Besides, references for more detailed information are given. We introduce the methods in timeline order. Table 2.1 at the end of this section displays a summary of the approaches that can be directly compared with the LASSO structure.

2.4.1 SCAD penalization

The smoothly clipped absolute deviation (SCAD) penalty was proposed by Fan (1997). This approach is non-convex and non-differentiable at zero penalization, able to simultaneously select variables and estimate their regression coefficients (Fan et al. (2004)).

To select a good penalty function, Fan and Li (2001) proposed three principles that this should satisfy: unbiasedness, in which there is no over-penalization of high parameters to avoid unnecessary bias; sparsity, automatically setting the insignificant parameters to 0 to reduce model complexity; and continuity to avoid instability in model prediction. For this last, the penalty function should be chosen such that its corresponding optimization problem produces continuous estimators.

Then, a penalty function in terms of Fan and Li (2001) is wanted to plug this in (2.1). A first option could be the L_1 regularization, leading to the LASSO regression displayed in (2.3). The LASSO problem results in sparse solutions, but this procedure can not keep the resulting estimators unbiased for large values of the penalty parameter. The higher the penalty value, the greater the bias. See Section 2.2.1 for more details. Thus, other functions with better properties are needed. Another type of penalty function is the hard thresholding penalty, given by $p_\lambda(|\beta|) = \lambda^2 - (|\beta| - \lambda)^2 \mathbb{I}_{(|\beta|<\lambda)}$, which results in $\hat{\beta} = \beta \cdot \mathbb{I}_{(|\beta|>\lambda)}$, where $\mathbb{I}_{(\cdot)}$ is the indicator operator. However, this estimator is not continuous in terms of β.

As the penalty functions introduced above can not simultaneously satisfy the three principles mentioned above, motivated by wavelet analysis, Fan (1997) proposed the SCAD continuous differentiable penalty function, which derivative function is defined by

$$p'_\lambda(\beta) = \lambda \left\{ \mathbb{I}_{(\beta \leq \lambda)} + \frac{(a\lambda - \beta)_+}{(a-1)\lambda} \mathbb{I}_{(\beta > \lambda)} \right\} \quad \text{for some } a > 2.$$

This penalization corresponds to a quadratic spline function with knots at λ and $a\lambda$. Explicitly, the penalty is

$$p_\lambda(\beta) = \begin{cases} \lambda|\beta|, & \text{if } |\beta| \leq \lambda, \\ \dfrac{2a\lambda|\beta| - \beta^2 - \lambda^2}{2(a-1)}, & \text{if } \lambda < |\beta| \leq a\lambda, \\ \dfrac{\lambda^2(a+1)}{2}, & \text{otherwise.} \end{cases}$$

The SCAD penalty retains the penalization rate (and bias) of the LASSO for small coefficients. Conversely, this procedure continuously relaxes the penalty rate as the absolute value of the coefficient increases. This new penalization satisfies the three properties proposed by Fan and Li (2001).

Under orthogonal design, taking $z_j = X_j^\top y$, we get the SCAD solution

$$(\hat{\beta}_\lambda)_j = \begin{cases} \text{sign}(z_j)(z_j - \lambda)_+, & \text{if } |z_j| \leq 2\lambda, \\ \dfrac{(a-1)z_j - \text{sign}(z_j)a\lambda}{a-2}, & \text{if } 2\lambda < |z_j| \leq a\lambda, \\ z_j, & \text{otherwise.} \end{cases}$$

Figure 2.5 shows how the SCAD estimate looks like ($\lambda = 1, a = 3$). The dotted line is the $y = x$ line. We see as the SCAD estimates are the same as soft-thresholding for $|x| \leq 2\lambda$ and are equal to hard-thresholding for $|x| > a\lambda$. The estimates in the remaining regions are linear interpolations of these two regimes. This penalization keeps the bias of the LASSO estimator for little shrinkaged values of β, which absolute values are close to λ, and applies a hard threshold for distant ones removing the bias.

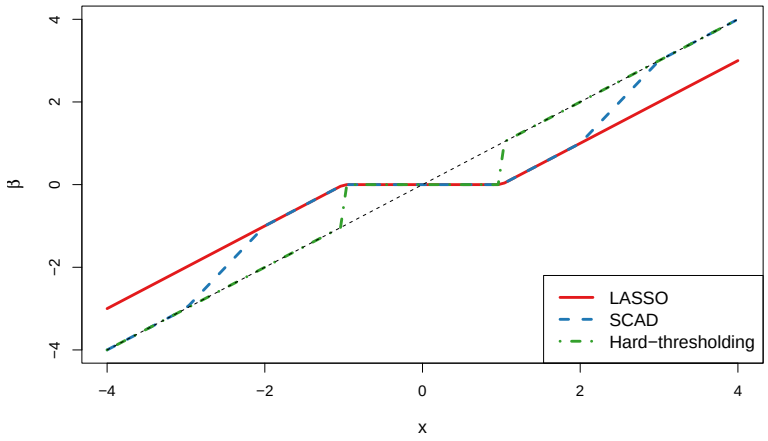


Figure 2.5: Shrinkage of the β parameter with the LASSO (soft-thresholding), SCAD and hard-thresholding penalizations taking $\lambda = 1$ and $a = 3$ under orthogonal design.

As a result, one appreciates that the SCAD penalization reduces the estimator bias and keeps the continuity. Nevertheless, two regularization parameters, λ and a, have to be selected here. This fact notably increases the computational cost if one wants to search for an optimal combination of both.

2.4.2 Elastic-Net algorithm

The Elastic Net (ENET) algorithm of Zou and Hastie (2005) was one of the first proposed methods to deal with the LASSO drawbacks. This method was born to protect against

the "correlation confusion phenomenon" of the LASSO estimator mentioned in Section 2.2.2: the LASSO procedure tends to select only a predictor among a bunch of them that are highly correlated.

Taking into account the quality of the L_1 regularization as a covariates selector, jointly with the gain in prediction accuracy of the L_2 penalty, it seems reasonable to establish an appropriate combination of both to avoid spurious correlations. So, this methodology imposes a combination of the L_1 and L_2 penalizations, from the LASSO (Tibshirani (1996)) and RIDGE regression (Hoerl and Kennard (1970)) respectively, in the linear regression problem (1.2) resulting in

$$\hat{\beta}^{\text{ENET}} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)\beta_j^2 \right) \right\} \quad \text{with } \alpha \in (0,1).$$

Due to the inclusion of the L_2 penalization, this model has the advantage that the number of covariates selected by the sample size n in the $p > n$ case is no longer limited. Besides, the quadratic penalty corrects the confusion effect of the L_1 penalization caused by highly correlated covariates. Nevertheless, the L_2 penalty forces the estimated coefficients of highly correlated predictors to be close to each other, which adds bias to the model. One can appreciate this phenomenon graphically through the two-dimensional framework displayed in Figure 2.3. Furthermore, we need to estimate an extra parameter, $\alpha \in (0,1)$, for example employing cross-validation techniques, although this last implementation increases the computational cost. Also, the ENET algorithm will only select covariates for values of α close to zero. Otherwise, if the L_2 penalty has a large enough load, the ENET procedure does not perform covariates screening. As a result, a suitable choice of α is so tricky in practice.

2.4.3 Dantzig selector

The Dantzig selector (Dant) is a method for covariates selection based on linear programming ideas. This was introduced by Candes and Tao (2007), and its name pays tribute to the father of linear programming: George Bernard Dantzig, developer of the simplex algorithm.

The Dantzig estimator is the result of solving the convex optimization problem

$$\begin{aligned} &\min_{\beta} \|\beta\|_1, \\ &\text{subject to } \|X^\top r\|_\infty \leq \lambda_p \cdot \sigma, \end{aligned} \tag{2.15}$$

for some $\lambda_p > 0$, where $\|X^\top r\|_\infty := \sup_{1 \leq j \leq p} |(X^\top r)_j|$, $r = y - X\beta$ is the vector of residuals and σ is the standard deviation of the model errors.

The reason to constrain the size of the correlated residual vector $X^\top r$, rather than the size of the residual vector r, is to guarantee invariance to orthonormal transformations. Suppose an orthonormal transformation is applied to the data, giving $\tilde{y} = Uy$, where $U^\top U$ is the identity. It is clear that a good estimation procedure for estimating β

should not depend upon U (after all, one could apply $U^\top$ to return to the original problem). It turns out that the estimation procedure (2.15) is, actually, invariant to orthonormal transformations applied to the data vector since the feasible region is invariant: $(UX)^\top(UX\tilde{\beta} - Uy) = X^\top(X\tilde{\beta} - y)$.

It can be shown that, taking $\lambda_p = (1 + \delta^{-1})\sqrt{2\log(p)}$ where δ is a positive scalar, if X obeys a uniform uncertainty principle (with unitnormed columns), and if the true parameter vector β is sufficiently sparse (which here roughly guarantees that the model is identifiable), then it is verified with a very high probability that

$$\|\hat{\beta} - \beta\|_2^2 \leq C^2 \cdot 2\log(p) \cdot \left(\sigma^2 + \sum_j \min(\beta_j^2, \sigma^2)\right).$$

We refer to Candes and Tao (2007) or Bickel et al. (2009) for more information about the consistency of the Dantzig selector.

It is important to notice that the Dantzig selector (2.15) uses the unknown noise parameter σ^2. The estimation of this value in the $p > n$ case is not simple, as we have discussed early for the LASSO in Section 2.2.4. A solution in practice is to change the $\lambda_p \cdot \sigma$ value for a generic $\lambda > 0$, although one loses information about the scale of the penalization. Besides, due to the non-convex nature of the problem (2.15), it is highly costly to find an approximate solution for a high dimensional case in general. This last is due to the presence of local minima in the objective function.

2.4.4 Relaxed LASSO

As seen in Section 2.2.2, if no consistency condition is verified, the LASSO adjustment has an L_2-loss rate of convergence when the number of predictor variables grows fast with the number of observations. See Meinshausen (2007)). However, this can be quite slow for a sparse high dimensional context. Moreover, many noisy variables are prone to be selected if the estimator is chosen using cross-validation techniques. The relaxed LASSO (RelaxL) is a two-stage procedure introduced in Meinshausen (2007) to achieve a faster convergence rate. Besides, this new approach produces sparser models with equal or lower prediction loss than the regular LASSO estimator in high dimensions.

Keeping a similar philosophy to the SCAD procedure displayed in Section 2.4.1, the relaxed LASSO is a generalization of both soft and hard thresholding. See Figure 2.5. This method controls model selection and shrinkage estimation by two separate parameters, $\lambda \in [0, \infty]$ and $\theta \in (0, 1]$, through the optimization problem

$$\hat{\beta}^{\text{RelaxL}} = \arg\min_{\beta} \left\{ n^{-1} \sum_{i=1}^{n} \left(y_i - x_i^\top \{\beta \cdot \mathbb{I}_{\mathcal{M}_\lambda}\}\right)^2 + \theta\lambda\|\beta\|_1 \right\}, \tag{2.16}$$

where $\mathbb{I}_{\mathcal{M}_\lambda}$ is the indicator function on the set of variables $\mathcal{M}_\lambda \subseteq \{1, \ldots, p\}$, which cardinal

is m. So that, for all $j \in \{1, \ldots, p\}$,

$$\{\beta \cdot \mathbb{I}_{\mathcal{M}_\lambda}\}_j = \begin{cases} 0, & j \notin \mathcal{M}_\lambda, \\ \beta_j, & j \in \mathcal{M}_\lambda. \end{cases}$$

Notably, the LASSO and the relaxed LASSO estimators are identical for the $\theta = 1$ value in (2.16). In contrast, for $\theta < 1$, the shrinkage of the coefficients in the new model is reduced compared to ordinary LASSO estimation.

This method has a low computational cost. Often, this is identical to that of an ordinary LASSO solution, and unlike the LASSO, convergence rates are fast, irrespective of the growth rate of the number of predictor variables. However, for high dimensional problems with $p > n$, the computational cost of the relaxed LASSO $O(m^2np)$ translates into $O(n^3p)$ and hence slightly more expensive than the $O(n^2p)$ computations of the standard LASSO.

Moreover, relaxed LASSO leads to consistent variable selection under a prediction-optimal choice of the penalty parameters, which does not hold for traditional LASSO solutions in a high dimensional setting. In addition, the rate of convergence of the relaxed LASSO estimator is not influenced by the presence of many noise variables. See Meinshausen (2007) for more information.

The main advantages of RelaxL over the classical LASSO in the high dimensional setting are two. On the one hand, this estimator achieves sparser estimates: it selects fewer coefficients without compromising the accuracy, producing more harmonious models. On the other hand, its predictions are more accurate: the accuracy of relaxed and ordinary LASSO is comparable in a low signal-to-noise ratio; however, for a high signal-to-noise ratio, the RelaxL often achieves more accurate predictions. This phenomenon explains that the relaxed LASSO is adaptive to the signal-to-noise ratio.

Nevertheless, the inclusion of an extra tuning parameter, θ, increases the model complexity. Furthermore, the computational cost of the RelaxL is greater than the one of the standard LASSO. These all translate into more complexity in the estimation procedure.

2.4.5 Square root LASSO

The Square root LASSO (SqrtL), see Belloni et al. (2011), is an alternative that allows one to solve a least squares problem without assumptions on the error distribution. In particular, we do not need to know the variance σ to obtain an optimal penalization value, unlike the LASSO procedure. In contrast, this method only requires assumptions about the moments of the error distribution and suitable design conditions.

This estimator is the result of solving the problem

$$\hat{\beta}^{\text{SqrtL}} = \arg\min_{\beta} \left\{ \left[\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right]^{1/2} + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{2.17}$$

which is a convex conic programming problem.

The resulting estimate of (2.17) has a computational cost similar to that of the LASSO problem displayed in (2.3).

As mentioned above, the main advantage of this procedure is that the optimal value of the penalization parameter λ does not depend on the knowledge of the error variance σ^2. This fact contrasts with the LASSO requirements. See Section 2.2.4 for more details. Then, it is possible to match the near oracle performance of the LASSO when the noise level is unknown.

In particular, to guarantee with approaching probability $1-\alpha$ that

$$\|\hat{\beta}^{\mathrm{SqrtL}} - \beta\|_2 \leq \sigma\{s\log(2p/\alpha)/n\}^{1/2},$$

the optimal value of λ needed is given by the expression

$$\lambda = c \cdot n^{-1/2}\Phi^{-1}(1-\alpha/2p) \quad \text{for some } c > 1.$$

See Belloni et al. (2011) for more information.

A posterior analysis of its characteristics and behavior in comparison with more recent methods can be found in papers such as Giraud et al. (2012) or Wang (2013), among others. Besides, a generalization of this methodology to the nonparametric regression model is also possible. See Belloni et al. (2014) for more details about this topic.

2.4.6 Scaled LASSO

Another possibility is the use of the scaled LASSO algorithm of Sun and Zhang (2012) (ScalL). The ScalL is constructed to avoid the LASSO regression drawback that the penalty term, λ, should be proportional to the model error variance to achieve consistency. See Section 2.2.4 for more details.

For this purpose, the regression approach estimates the noise level, σ^2, and the regression coefficients vector, β, at the same time. This procedure iteratively estimates the noise level using the mean residual square to achieve a consistent estimator of β. Subsequently, this approach scales the penalty using this obtained value. This process is performed by following the ideas of the iterative algorithm introduced in Städler et al. (2010). As a result, knowing the value of σ^2 in advance is unnecessary. This results in a sparse estimator $\hat{\beta}$ and an error variance estimator $\hat{\sigma}^2$.

In order to implement this procedure, it is taken into account that an estimation vector $\hat{\beta}$ is a critical point of the LASSO penalized loss function

$$L_\lambda(\beta) = \frac{\|y - X\beta\|_2^2}{2n} + \lambda\sum_{j=1}^{p}\beta_j \tag{2.18}$$

if and only if this verifies

$$\begin{cases} x_j^\top(y - X^\top\hat{\beta})/n = \lambda\mathrm{sign}(\hat{\beta}_j), & \hat{\beta}_j \neq 0, \\ x_j^\top(y - X^\top\hat{\beta})/n \in \lambda[-1,1], & \hat{\beta}_j = 0. \end{cases} \tag{2.19}$$

Due to the convexity of the loss function (2.18), this last condition (2.19) corresponds to the Karush–Kuhn–Tucker condition for its minimization.

How a penalty term λ is still needed to characterize the solutions of (2.19), an iterative algorithm minimizing the scaled penalized least-squares estimator is proposed. Its scheme is collected in Algorithm 2.6.

Algorithm 2.6 (Scaled LASSO)**.**

- $\hat{\sigma} \leftarrow \|y - X^\top \hat{\beta}^{old}\|_2 / n^{1/2}$,
- $\lambda \leftarrow \hat{\sigma}\lambda_0$,
- $\hat{\beta} \leftarrow \hat{\beta}^{new},\ L_\lambda(\hat{\beta}^{new}) \leq L_\lambda(\hat{\beta}^{old})$.

where λ_0 is a prefixed-penalty level, not depending on σ, $\hat{\sigma}$ estimates the noise level, L_λ is given in (2.18) and $\hat{\beta}^{new}$ is a solution of (2.19) for the given λ.

As is explained in Sun and Zhang (2012), the first step of the implementation is the computation of a solution path $\hat{\beta}(\lambda)$ of (2.19) beginning from $\hat{\beta}(\lambda) = 0$ for $\lambda = \|X^\top y/n\|_\infty$. The second step of the implementation is the iteration of Algorithm 2.6 along the solution path $\beta(\lambda)$ computed in the first step. Thus, the previously computed $\hat{\beta}^{new} = \hat{\beta}(\lambda)$ is employed in Algorithm (2.6). Furthermore, for large data sets, to compute $\hat{\beta}^{new}$ from $\hat{\beta}^{old}$, one may use a few steps of a gradient descent algorithm.

An analysis of the theoretical properties of this procedure, as well as numerical results, is displayed in Sun and Zhang (2012) and Sun and Zhang (2013).

2.4.7 SLOPE

The Sorted L-One Penalized Estimation (SLOPE) algorithm is another alternative proposed by Bogdan et al. (2015). The resulting new estimator $\hat{\beta}^{SLOPE}$ is the solution of the convex optimization problem given by

$$\hat{\beta}^{\text{SLOPE}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \sum_{j=1}^{p} \lambda_j |\beta|_{(j)} \right\}, \tag{2.20}$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_p \geq 0$ and $|\beta|_{(1)} \geq |\beta|_{(2)} \geq \cdots \geq |\beta|_{(p)}$ are the absolute values of the entries of β in decreasing order. Here, the penalty term is a sorted L_1 norm. This term penalizes the regression coefficients according to their rank. Then, the higher the rank, the stronger the signal and the larger the penalty value, respectively.

This procedure is motivated by the need to select relevant variables to avoid noise. For this purpose, this approach seeks to control the expected proportion of irrelevant variables among the selected ones. This process translates into controlling the False Discovery Rate (FDR). In particular, Bogdan et al. (2015) propose some results of FDR expression under orthogonality and Gaussian assumptions.

The problem of covariates selection in the $p > n$ framework can be seen as a multiple-testing problem where p simultaneous contrasts are needed. Then, a covariate j, for

$j = 1, \ldots, p$, would be included in the model if the null hypothesis $H_{0j} : \beta_j = 0$ is rejected. However, one needs to adjust the significance levels α_j of each test to guarantee, in the end, a prefixed significance level α. Then, a correction for multiple testing implementation is needed. Strategies such as the well-known Bonferroni's method (Dunn (1958)) or the Benjamini and Hochberg (1995) procedure (BH) are some of the possible options proposed.

Assuming orthogonal design and Gaussian errors, i.e. $\mathbf{X_n}^\top \mathbf{X_n} = I_p$ and $\varepsilon \in N\left(0, \sigma^2 I_n\right)$, it is possible to obtain an expression for FDR using BH ideas. This method begins by sorting the entries of $\tilde{y} = \mathbf{X_n}^\top y$ in decreasing order of magnitude, $|\tilde{y}|_{(1)} \geq \cdots \geq |\tilde{y}|_{(p)}$, which yields corresponding ordered hypotheses $H_{(01)}, \ldots, H_{(0p)}$. Then, to control the FDR at level $q \in [0, 1]$, BH rejects all hypothesis $H_{(0i)}$ for which $i \leq i_{BH}$, where i_{BH} is

$$i_{BH} = \max\{i : |\tilde{y}|_{(i)}/\sigma \geq \Phi^{-1}(1 - q_i)\}, \quad q_i = i \cdot q/2p.$$

Letting V, respectively R, be the total number of false rejections, respectively the total number of rejections, Benjamini and Hochberg (1995) shown that for BH

$$FDR = \mathbb{E}\left[\frac{V}{R \vee 1}\right] = q\frac{s_0}{p},$$

where s_0 is the number of true null hypotheses, $s_0 := \#\{i : \beta_i = 0\} = p - \|\beta\|_0$.

The employed threshold of the BH procedure, $|y|_{(i_{BH})}$, is data-dependent in the sense that this is sensitive to the sparsity and magnitude of the true signal. Quoting Bogdan et al. (2015): in a setting where there are many large β_j's, the last selected variable needs to pass a far less stringent threshold than it would in a situation where no β_j is truly different from 0. Hence, this behavior allows BH to adapt to the unknown signal sparsity.

Given the BH methodology, this idea applies to the SLOPE algorithm. For this purpose, a sorted L_1 norm that penalizes the coefficients attended to their magnitude as a regularization of the problem (1.2) is introduced. Then, letting $\lambda \neq 0$ be a nonincreasing sequence of nonnegative scalars $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, it is defined the sorted L_1 norm of a vector $\beta \in \mathbb{R}^p$ as $p_\lambda(\beta) = \lambda_1|\beta|_{(1)} + \lambda_2|\beta|_{(2)} + \cdots + \lambda_p|\beta|_{(p)}$. This new regularization gives place to the SLOPE problem (2.20).

The convexity of (2.20) guarantees that the SLOPE problem is tractable. In fact, its computational cost is similar to the LASSO one because this is a sorted extension of the L_1 norm. However, the general formulation of the SLOPE allows one to achieve more adaptivity. See Bogdan et al. (2015) for more information about optimization techniques for this problem.

It is interesting to highlight that the idea behind SLOPE contrasts with the one associated with the adaptive LASSO procedure introduced in Section 2.3.1. In the AdapL, the penalty tends to decrease as the magnitude of coefficients increases. Conversely, for the SLOPE approach, the opposite happens. However, the control of the FDR can only be theoretically guaranteed when the orthogonal and Gaussian assumptions are verified. These assumptions can be quite restrictive in practice.

2.4.8 Knockoffs filter

Another procedure devoted to controlling the inclusion of unnecessary noise in the model, i.e. to control the False Discovery Rate (FDR), is the knockoff algorithm introduced by Barber and Candès (2015). This approach is named the knockoffs filter. Again, the covariates selection problem is rewritten as a multiple hypothesis testing with null hypothesis $H_{0j} : \beta_j = 0$ for each of the $j = 1, \ldots, p$ covariates. This procedure needs the Gaussian assumption but allows all types of fixed designs, not only restricted to the orthogonal framework as in some previous techniques.

For this aim, given a level $q \in [0, 1]$, it is said that a selection rule controls the FDR at level q if $FDR \leq q$ regardless of the value of the β coefficients. This corresponds to controlling for type I error in testing problems.

Next, three steps to carry out this methodology are given.

Step 1: Construct knockoffs. For each feature X_j in the model, a new "knockoff" feature $\tilde{X}_j$ is built, for $j = 1, \ldots, p$. These knockoff variables mimic the correlation structure of the original features in a particular way that allows for FDR control.

After normalizing each variable, the Gram matrix $\Sigma = \mathbf{X_n}^\top \mathbf{X_n}$ is calculated for all $j = 1, \ldots, p$. Then, knockoffs features $\widetilde{\mathbf{X}}_\mathbf{n}$ are generated verifying

$$\widetilde{\mathbf{X}}_\mathbf{n}^\top \widetilde{\mathbf{X}}_\mathbf{n} = \Sigma, \quad \mathbf{X_n}^\top \widetilde{\mathbf{X}}_\mathbf{n} = \Sigma - \text{diag}\{\eta\},$$

where $\eta \in \mathbb{R}^n$ is a non-negative vector adequately chosen. Thus, $\widetilde{\mathbf{X}}_\mathbf{n}$ has the same covariance structure as $\mathbf{X_n}$, and correlations between original and knockoff variables are the same as the ones of the original design.

Step 2: Calculate statistics for each pair of original and knockoff variables. Now, W_j statistics are introduced for each β_j to carry out partial tests for $j = 1, \ldots, p$. These W_j's are constructed to ensure that large enough positive values are evidence against the null hypothesis of $H_{0j} : \beta_j = 0$. So, defining λ_j as the penalization value λ on the LASSO path at which covariate X_j first enters the model:

$$\lambda_j = \sup\{\lambda : \hat{\beta}_{j\lambda} \neq 0\},$$

then, W_j can be defined by means of

$$W_j = \lambda_j \vee \tilde{\lambda}_j \cdot \begin{cases} +1, & \lambda_j > \tilde{\lambda}_j, \\ -1, & \lambda_j < \tilde{\lambda}_j \end{cases}$$

where $\lambda_j \vee \tilde{\lambda}_j = \max\{\lambda_j, \tilde{\lambda}_j\}$. When $\lambda_j = \tilde{\lambda}_j$, set $W_j = 0$.

A large enough positive value of W_j means that variable X_j enters the LASSO model early, and before its knockoff copy $\tilde{X}_j$ ($\lambda_j > \tilde{\lambda}_j$). This last happens because a decreasing sequence of λ values is considered. Therefore, this fact indicates that the associated variable has a genuine signal and belongs to the relevant terms of the model.

Step 3: Calculate a data-dependent threshold for the statistics. As we want to select covariates with large and positive W_j values, one has to find a correct value $\delta > 0$ able to

verify that, taking $\hat{S} = \{j : W_j \geq \delta\}$, the $FDR \leq q$ for a fixed q. For this purpose, it is enough to take $\delta = T$, being

$$T = \min\left\{\delta \in \mathcal{W} : \frac{|j : W_j \leq -\delta|}{|j : W_j \geq \delta| \vee 1} \leq q\right\} \quad (\text{or } T = +\infty \text{ if this set is empty}),$$

where $\mathcal{W} = \{|W_j| : j = 1, \ldots, p\} \setminus \{0\}$. Hofner et al. (2015) can be consulted for extra information about the procedure implementation.

The problem of this methodology is the necessity of assuming that $n \geq p$, opposite to the high dimensional framework of interest. Barber and Candès (2019) propose a modification of this algorithm that allows one to work with fewer samples than covariates, guaranteeing the control of FDR too. However, the idea of this last modification is to split the process into two parts: in the first one, a subset of $\{1, \ldots, p\}$ covariates is selected, of size r, imposing that $|r| \leq n$ and then, the usual knockoff filter is implemented. As a result, this methodology needs to apply a preliminary covariates selection algorithm suitable for high dimensions to select the initial r covariates efficiently. As far as we know, there is still no other extension in the literature of this procedure to the high dimensional context.

2.4.9 Debiased LASSO

The debiased LASSO of Javanmard and Montanari (2018) (DebL) is an alternative approach concerning multiple hypothesis testing to select covariates in the $p > n$ regime. This methodology transfers the debiased ideas proposed in Bühlmann et al. (2013) and Van de Geer et al. (2014), among others, to the LASSO context. As a result, a sparse estimator of β for the high dimensional context with a Gaussian known distribution for each component is achieved. It is the first time that one of the procedures presented in Section 2.4 gives a distribution for the β components, which allows us to make inferences about these.

Therefore, to be able to characterize the distribution of a $\hat{\beta}$ estimator for the $p > n$ framework, a debiased (or de-sparsified) estimator is needed. Javanmard and Montanari (2018) propose to consider

$$\hat{\beta}^{DebL} = \hat{\beta}^{LASSO} + \frac{1}{n}\mathbf{M_n}\mathbf{X_n}^\top(y - \mathbf{X_n}^\top\hat{\beta}^{LASSO}), \tag{2.21}$$

where $\mathbf{M_n} \in \mathbb{R}^{p\times p}$ is a function of $\mathbf{X_n}$, but not of y.

Quoted Javanmard and Montanari (2018): the intuition is that $\mathbf{M_n}$ should be a good estimator of the precision matrix Σ^{-1}. Then, using the LASSO vector and if $s < \sqrt{n}/\log(p)$ $(n > (s\log(p))^2)$, it is possible to guarantee that

$$\hat{\beta}_j^{DebL} \in N(\beta_j, \sigma^2/n) \quad \forall j \in \{1, \ldots, p\}. \tag{2.22}$$

In Javanmard et al. (2019) it is proposed to use the estimation of $\mathbf{M_n}$ of Javanmard and Montanari (2014) to control the FDR. Then, the decorrelating matrix $\mathbf{M_n}$ is constructed via a convex optimization problem focused on reducing bias and variance of the coordinates of the $\hat{\beta}^{DebL}$ vector at the same time.

Thus, $\mathbf{M_n} = (m_1, \ldots, m_p)^\top \in \mathbb{R}^{p\times p}$, where each $m_j \in \mathbb{R}^p$ is the solution of the convex problem given by

$$\begin{aligned} &\min_m \ m^\top \hat{\mathbf{\Sigma}}_\mathbf{n} m, \\ &\text{subject to } \|\hat{\mathbf{\Sigma}}_\mathbf{n} m - e_i\|_\infty \leq \theta, \end{aligned}$$

where $e_i \in \mathbb{R}^p$ is the ith standard unit vector, $\hat{\mathbf{\Sigma}}_\mathbf{n} = (\mathbf{X_n}^\top \mathbf{X_n})/n$ and θ is a constraint properly chosen (see Javanmard et al. (2019) for more information).

Once we can assure the limit normal distribution of (2.21), we need to estimate the error variance σ^2 in order to characterize this. Javanmard et al. (2019) propose to make use of procedures like the scaled LASSO (Section 2.4.6) to estimate this quantity. Eventually, one can perform partial tests of the form $H_{0j}\colon \hat{\beta}_j^{DebL} = 0$ to verify if some of the selected covariates by $\hat{\beta}^{DebL}$ can be omitted in the regression model, reducing noise.

Although its good properties, this procedure has some drawbacks in practice. The first ones are related to the Gaussian assumption of the model errors or the sparsity condition of $s < \sqrt{n}/\log(p)$. Both might not be possible to verify in practice. Another inconvenience is to estimate the $\mathbf{M_n}$ matrix when Σ is unknown, which requires the solution of an additional optimization problem.

2.4.10 Distance covariance algorithm

Next, a completely different approach is proposed employing the novel distance covariance coefficient (DC) of Székely et al. (2007). This measure of dependence is introduced in more detail in Section 4.2.1. In particular, here, a covariates selection procedure for regression models using the DC philosophy is introduced: the distance covariance algorithm for variable selection (DC.VS) of Febrero-Bande et al. (2019).

The main advantage of the DC coefficient is that this allows measuring the grade of dependence between two random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ for all possible dependence patterns. As a result, this coefficient characterizes the independence between X and Y. In Particular, the DC.VS algorithm employs the scale-invariant version of the DC coefficient: the distance correlation coefficient (DCor). As its name suggests, this verifies that $0 \leq DCor(X, Y) \leq 1$; besides, X and Y are independent if and only if $DCor(X, Y) = 0$. A more detailed explanation can be found in Section 4.2.1.

The DC.VS algorithm has a similar philosophy to the LARS one (Efron et al. (2004)) but uses the DCor coefficient as a measure of dependence. Namely, this is based on a sequential process that tests, at each step and among the remaining covariates, if the most correlated term with the model residuals contributes to improving the explanation of the response. For this last, the DCor coefficient is used. Then, this covariate could be included in the present model or ignored according to the test result. Once this decision is taken, if a new covariate is incorporated, the model is updated, and the residuals recalculated. The complete process repeats until all the p covariates are tested or if the correlation distances of the remaining ones are negligible. We refer the reader to Febrero-Bande et al. (2019) for a more detailed scheme of its implementation.

This algorithm has shown good performance in terms of selecting relevant covariates and avoiding noise (see Febrero-Bande et al. (2019) for more information). However, due to its implementation, a high computational time is required. As a result, in a high dimensional context where $p > n$, this could result in an expensive process, which may be an important caveat, especially when the number of covariates, p, is high.

2.4.11 LASSO-Zero

The LASSO-Zero (Descloux and Sardy (2021)) is a new L_1-based estimator whose novelty lies in an "overfit then threshold" paradigm and the use of noise dictionaries concatenated to X to overfit the response.

This procedure relies on ideas of the basis pursuit denoising approach (see Chen et al. (2001)). A naive interpretation of the LASSO-Zero estimator is that this solves an adaptation of the basis pursuit problem in a first step, and then, this thresholds the obtained solution appropriately to retain only the largest coefficients. The principal novelty of LASSO-Zero resides in the use of several random noise dictionaries in the overfitting step, followed by the aggregation of the corresponding estimates.

Then, this method uses a noise dictionary consisting of a random matrix $G \in \mathbb{R}^{n\times q}$. The purpose of the random dictionary G is to provide new columns in $\mathbf{X_n}$ that can be selected to fit the noise term ε. Thus, columns of $\mathbf{X_n}$ can be mostly used to fit the true signal ${\mathbf{X_n}}^\top\beta$. Indeed, if $\text{rank}(G) = n$, there exists a vector $\gamma_{\varepsilon,G} \in \mathbb{R}^q$ such that $\varepsilon = G\gamma_{\varepsilon,G}$ and the model (1.2) can be rewritten as $\mathbf{Y_n} = {\mathbf{X_n}}^\top\beta + G\gamma_{\varepsilon,G}$. The estimates for β and $\gamma \equiv \gamma_{\varepsilon,G}$ are attained solving the basis pursuit problem (2.4) for the extended matrix $\tilde{\mathbf{X}}_\mathbf{n} = (\mathbf{X_n}; G) \in \mathbb{R}^{n\times(p+q)}$. The resulting problem is displayed in equation (2.23).

$$\begin{aligned} &\min_{\beta} \|\beta\|_1 + \|\gamma\|_1 \\ &\text{subject to } y = \tilde{\mathbf{X}}_\mathbf{n}^\top\beta + G\gamma. \end{aligned} \tag{2.23}$$

This procedure repeats several times to take the median for each component of the $\hat{\beta}$ vector. Then, these medians are thresholded, and only the covariates with an associated value large enough are selected. The scheme to obtain this estimator is collected in Algorithm 2.7.

Algorithm 2.7 (LASSO-Zero)**.**

Given $q \in \mathbb{N}$, $M \in \mathbb{N}$ and a threshold $\theta \geq 0$:

1. For $k = 1, \ldots, M$: generate $G^{(k)} \in \mathbb{R}^{n\times q}$ with entries $G_{ij}^{(k)} \overset{iid}{\sim} N(0,1)$ and compute the solution $(\hat{\beta}^{(k)}, \hat{\gamma}^{(k)})$ to (2.23) with $G = G^{(k)}$.
2. Define $\hat{\beta}_j^{L_1} = \text{median}\{\hat{\beta}_j^{(k)}, k = 1, \ldots, M\}$ for $j \in \{1, \ldots, p\}$.
3. Threshold the coefficients at level θ to obtain $\hat{\beta}_{\theta,j}^{LASSO-Z} := \eta_\theta(\hat{\beta}_j^{L_1})$ for $j \in \{1, \ldots, p\}$, where η_θ denotes any thresholding function satisfying $\eta_\theta(x) = 0$ if $|x| \leq \theta$ and $\text{sign}(\eta_\theta(x)) = \text{sign}(x)$ otherwise; typically soft or hard thresholding are used.

One can appreciate as the LASSO-Zero can be considered an extension of thresholded basis pursuit (Saligrama and Zhao (2011)). Concerning an appropriate selection of the threshold $\theta \geq 0$, the quantile universal thresholding (QUT) is employed (Giacobino et al. (2017)). More information about the implementation of this thresholding function, jointly with how to choose proper values for the parameters q or M, is detailed in Descloux and Sardy (2021). Furthermore, the theoretical properties of this new method are also analyzed.

An important drawback of this methodology is the computational time required because of the use of the noise dictionaries G. As a result, for medium or great values of n and p, the problem is intractable. In addition, it is necessary to select, apart from the threshold, two additional tuning parameters: q and M. Although some recommendations arise in Descloux and Sardy (2021) for these values selection, all these characteristics of the LASSO-Zero adjustment translate into a costly procedure.

PROBLEM FORMULATION		PROS
▲ **Best subset selection** – Beale et al. (1967), Hocking and Leslie (1967) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \mathbf{1}_{\beta_j \neq 0} \right\}$	×	*Better selection*
■ **LASSO** – Tibshirani (1996) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \lvert\beta_j\rvert \right\}$	✓	–
▲ **SCAD** – Fan (1997) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + p_\lambda(\beta) \right\}$ with $p_\lambda(\beta) = \begin{cases} \lambda\lvert\beta\rvert, & \text{if } \lvert\beta\rvert \leq \lambda, \\ \dfrac{2a\lambda\lvert\beta\rvert - \beta^2 - \lambda^2}{2(a-1)}, & \text{if } \lambda < \lvert\beta\rvert \leq a\lambda \quad (a > 2) \\ \dfrac{\lambda^2(a+1)}{2}, & \text{otherwise.} \end{cases}$	×	*Better selection* *Bias reduction*
■ **Basis Pursuit Denoising** – Chen et al. (2001) $\min_{\beta} \lVert\beta\rVert_1 \quad \text{subject to } \lVert y - X\beta\rVert_2 \leq \theta$	×	–
▲ **Elastic Net** – Zou and Hastie (2005) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \left(\alpha\lvert\beta_j\rvert + (1-\alpha)\beta_j^2 \right) \right\}$ with $\alpha \in (0,1)$	✓	*Better prediction* *Possible selection of more than n covariates ($p > n$)*
▲ **Fused LASSO** – Tibshirani et al. (2005) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{p} \lvert\beta_j\rvert + \lambda_2 \sum_{j=2}^{p} \lvert\beta_j - \beta_{j-1}\rvert \right\}$	✓	*Ordered structure*
● **Adaptive LASSO** – Zou (2006) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} w_j \lvert\beta_j\rvert \right\}$ (taking $w_j = 1/\lvert\hat{\beta}_j^{RIDGE}\rvert^q$ where $\hat{\beta}^{RIDGE}$ is the ridge estimator (Hoerl and Kennard (1970)) and $q \geq 1$)	✓	*Better selection* *Bias reduction*
▲ **Group LASSO** – Yuan and Lin (2006) $\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{K} X_k\beta_k \right)^2 + \lambda \sum_{k=1}^{K} \lVert\beta_k\rVert_{Z_k} \right\}$ with $\lVert w\rVert_{Z_k} = (w^\top Z_k w)^{1/2}$ (Z_k are kernel matrices of the functional space induced by the kth factor)	✓	*Group structure*

PROBLEM FORMULATION		PROS
▲ **Dantzig selector** – Candes and Tao (2007) $\min_{\beta} \|\beta\|_1 \quad \text{subject to } \|X^\top r\|_\infty \leq \lambda_p \cdot \sigma$ $\left(\text{with } \|X^\top r\|_\infty := \sup_{1\leq j\leq p} |(X^\top r)_j| \text{ and } r = y - X\beta\right)$	×	*Consistent to orthogonal transformations*
▲ **Relaxed LASSO** – Meinshausen (2007) $\min_{\beta} \left\{ n^{-1} \sum_{i=1}^{n} \left(y_i - x_i^\top \{\beta \cdot \mathbf{1}_{\mathcal{M}_\lambda}\}\right)^2 + \theta\lambda\|\beta\|_1 \right\} \text{ with } \theta \in (0,1]$	✓	*Faster convergence rates* *More accurate predictions*
▲ **Square-root LASSO** – Belloni et al. (2011) $\min_{\beta} \left\{ \left[\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \right]^{1/2} + \lambda \sum_{j=1}^{p} \|\beta_j\| \right\}$	✓	*It is not needed to known σ to obtain an optimal λ*
▲ **Scaled LASSO** – Sun and Zhang (2012) $\hat{\sigma} \leftarrow \|y - X^\top \hat{\beta}^{old}\|_2 / n^{1/2}, \quad \lambda \leftarrow \hat{\sigma}\lambda_0$ $\hat{\beta}^{new} = \arg\min_{\beta} \begin{cases} x_j^\top (y - X^\top \hat{\beta})/n = \lambda \text{sign}(\hat{\beta}_j), & \hat{\beta}_j \neq 0, \\ x_j^\top (y - X^\top \hat{\beta})/n \in \lambda[-1,1], & \hat{\beta}_j = 0. \end{cases}$ $\hat{\beta} \leftarrow \hat{\beta}^{new}, \; L_\lambda(\hat{\beta}^{new}) \leq L_\lambda(\hat{\beta}^{old})$ $\left(\text{where } L_\lambda(\beta) = \frac{\|y - X^\top\beta\|_2^2}{2n} + \lambda \sum_{j=1}^{p} \|\beta_j\|\right)$	✓	*Simultaneous estimation of σ and β*
● **SLOPE** – Bogdan et al. (2015) $\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \sum_{j=1}^{p} \lambda_j \|\beta\|_{(j)} \right\}$	✓	*Control of the False Discovery Rate (FDR)*
▲ **Debiased LASSO** – Javanmard and Montanari (2018) $\hat{\beta}^{debiased} = \hat{\beta}^{LASSO} + \frac{1}{n} M X^\top (y - X\hat{\beta}^{LASSO}) \sim N(\beta, \sigma^2/n)$ with $M = (m_1, \ldots, m_p)^\top \in \mathbb{R}^{p\times p}$, where each $m_i \in \mathbb{R}^p$ is the solution of $\min_{m} \; m^\top \hat{\Sigma} m \quad \text{subject to } \|\hat{\Sigma}m - e_i\|_\infty \leq \mu$ ($e_i \in \mathbb{R}^p$ is a standard unit vector, $\hat{\Sigma} = (X^\top X)/n$ and μ a constraint)	✓	*Characterization of the probability distribution for $\hat{\beta}$ ($\hat{\beta}^{debiased}$)*
▲ **LASSO-Zero** – Descloux and Sardy (2021) $\min_{\beta} \|\beta\|_1 + \|\gamma\|_1$ subject to $y = \tilde{X}\beta + G\gamma$ ($G \in \mathbb{R}^{n\times q}$ a noise dictionary and $\tilde{X} = (X\|G)$)	×	*Excellent trade-off between high TPR and low false discovery rate FDR*

Table 2.1: Formulated problems to estimate the β vector for linear regression in a high dimensional framework ($p > n$). It is indicated if the optimization problems are convex (✓) or not (×). Their main advantages in comparison with the LASSO are displayed in column (**PROS**) and it is shown if they shared LASSO properties (■), are a weighted version of the LASSO (●) or alternatives to this procedure (▲).

2.5 Examples of real data problems

This section collects some examples of real problems where covariates selection arises naturally. These data sets contain different relations between p and n, distinct dependence patterns, and covariates in different scales. These will be employed later in Chapter 3 to test the performance of LASSO and its derivatives for covariates selection. In particular, four data sets of different characteristics and natures are considered for this purpose.

The first one is a genomic study. In this example, the production rate of riboflavin (vitamin B2) of the bacterium Bacillus subtilis is modeled employing different gene expressions. A total of $p = 4088$ expression levels of genes have been measured in $n = 71$ experiments. This data set is a high dimensional example where the number of covariates is higher than the number of available samples ($p > n$). Moreover, there are different strengths and types of dependence between these genes. This fact displays in Section 2.5.1. In terms of scales, all covariates are of a similar magnitude. This results in a high dimensional example where some dependence patterns and covariates on a similar scale arise.

The second example is a well-known prostate cancer clinical study of male patients subjected to radical prostatectomy. This was introduced in Stamey et al. (1989) for the first time. This is presented and analyzed in more detail in Section 2.5.2. In this study, one wants to determine what factors affect the level of prostate-specific antigen before surgery. For this aim, eight clinical measures are taken in $n = 97$ patients. Different covariates selection techniques based on penalties or distance covariance ideas (see Section 4.2.1 of Chapter 4) have been employed in literature to detect which features are the most involved with this antigen. This example has covariates in a different range of values. Besides, this mainly contains medium and strong positive dependence relations between explanatory terms. The previous analysis will allow us to compare our results with the preceding ones. In particular, suitable penalization techniques, introduced in Chapter 2 and detailed in Section 3.1.3 of Chapter 3, will be tested. This comparison is carried out throughout Section 3.4 of Chapter 3.

Next, in Section 2.5.3, a third data set is studied. This is related to body fat prediction in men using body measures (see Siri (1956)). A total of $p = 14$ variables are measured in $n = 174$ men. This study aims to determine which are the important covariates in terms of body fat explanation. In this case, there are again covariates with different scales, and there exist a lot of dependence structures with varied strengths between them.

Eventually, a wine data set from Portuguese regions studied in Cortez et al. (2009) is employed for covariates selection. This is introduced in Section 2.5.4. In this case, we aim to explain the percentage of alcohol based on eight chemical measures. This database collects information about $n = 1599$ samples, resulting in a case where $n >> p$. Moreover, this is an example where only two feature scales are quite different, and only some strong dependence structures exist between them.

Summing up, all these examples have some dependence structure between covariates. Moreover, measured covariates tend to be in different scales. This fact motivates the critical analysis under dependence and different scales on covariates performed in Chapter

3. Next, these databases are detailed throughout this section. These will be employed and analyzed later in Section 3.4 of Chapter 3 to illustrate the dependence as well as the different scale effects for LASSO and derivatives in real problems.

2.5.1 Riboflavin

The first example is the high dimensional genomic data set `riboflavin` available in library `hdi` (Dezeure et al. (2015)) of R Core Team (2019). In this data set, the logarithm of the riboflavin (vitamin B_2) production rate of the *Bacillus subtilis* bacterium is measured. This vitamin is necessary for the cellular respiration of the body. This vitamin is found in foods such as eggs, green vegetables, or milk, among others.

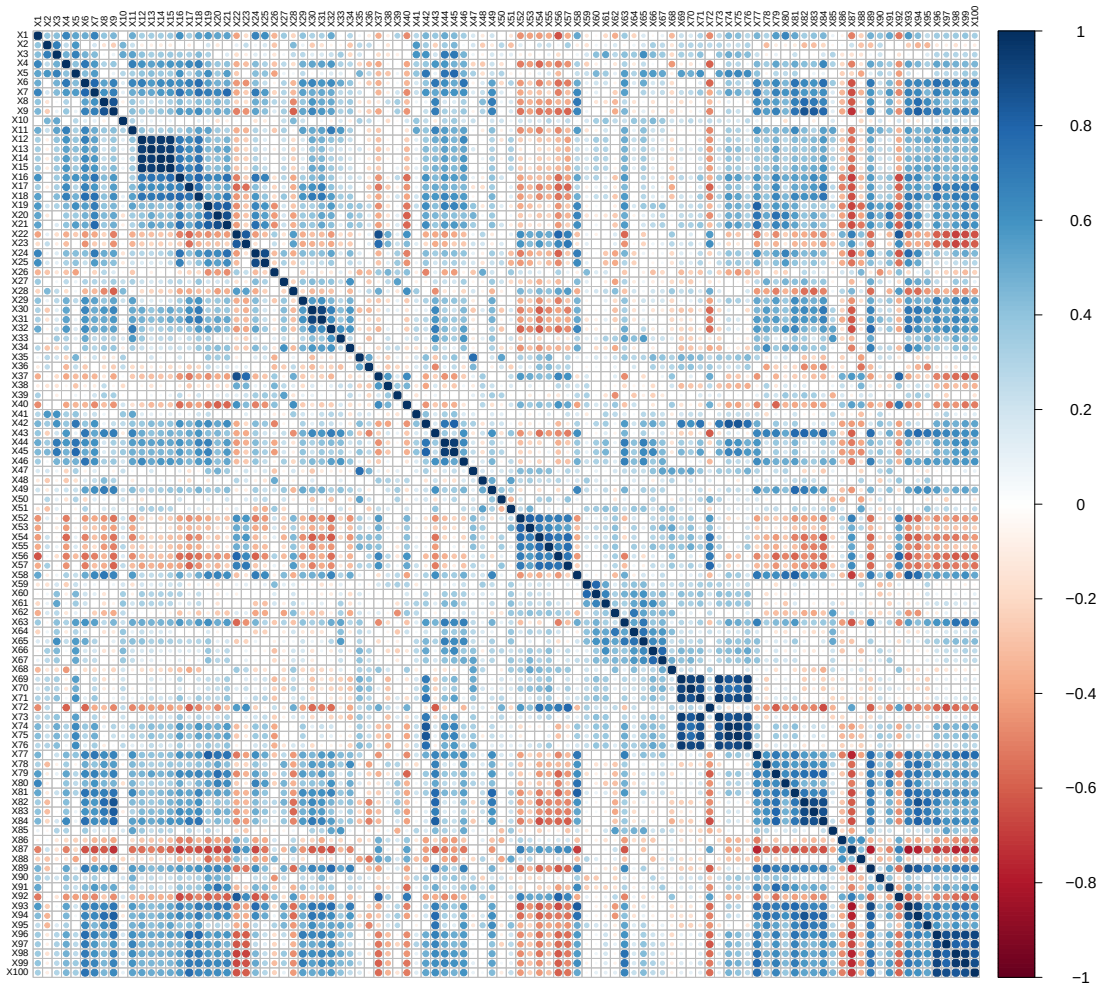


Figure 2.6: Correlation matrix of the first 100 covariates of the riboflavin data set.

Then, in order to identify which genes are crucial to explaining the riboflavin production of this bacterium, the logarithm of the expression level of $p = 4088$ genes has been measured in $n = 71$ experiments. See Bühlmann et al. (2014) for more details. Therefore, covariates selection algorithms in the $p > n$ framework are required to identify relevant genes. This data set has already been studied in works as the one of Bühlmann et al. (2014).

Figure displays 2.6 a summary of the data structure. For this purpose, we calculate the correlation matrix of the first 100 genes. As can be seen, there are many dependence structures of quite different magnitudes between some subsets of genes. As a result, not all genes, but a bunch of them may be enough to explain correctly the riboflavin production.

Furthermore, concerning covariates scales, Figure 2.7 shows their values varying between the $[0.1, 1.84]$ rank. Thus, there are slight differences in covariates scales, especially between low and high values. Then, we can assume that genes are on a similar scale.

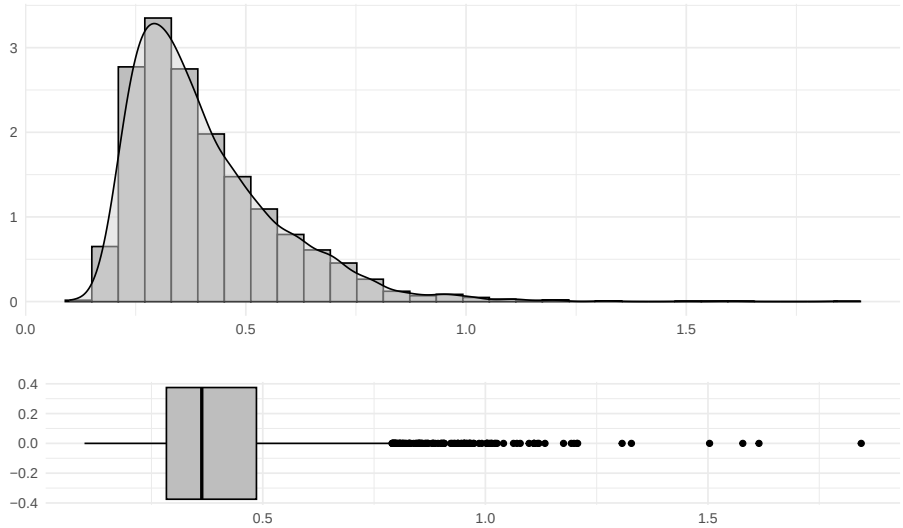


Figure 2.7: Histogram (top) and boxplot (bottom) of the standard deviation of the genes from the riboflavin data set.

As a result, this is a high dimensional data set where $p > n$ with several dependence structures between covariates and similar scale effects.

2.5.2 Prostate cancer

Next, we consider an additional example of medical data. This data set collects information about men suffering from prostate cancer[1]. In particular, these patients were about to have a radical prostatectomy to remove their prostate. This data was introduced in Stamey et al. (1989) and has been previously studied in the covariates selection field in works as Hastie et al. (2009) or Székely and Rizzo (2014). In the first work, they apply classical techniques as principal component analysis (PCA) and partial least squares (PLS) analysis,

[1]This is available in `https://hastie.su.domains/ElemStatLearn/`

jointly with ordinary LASSO, Ridge regression (Hoerl and Kennard (1970)) or Best subset selection (Miller (2002)). In contrast, Székely and Rizzo (2014) focus on partial covariates selection, assuming nonlinear effects of the covariates over the response. For this aim, they propose a new adaptation of the distance correlation (Székely et al. (2007), Szekely and Rizzo (2017)) defining the partial distance correlation coefficient.

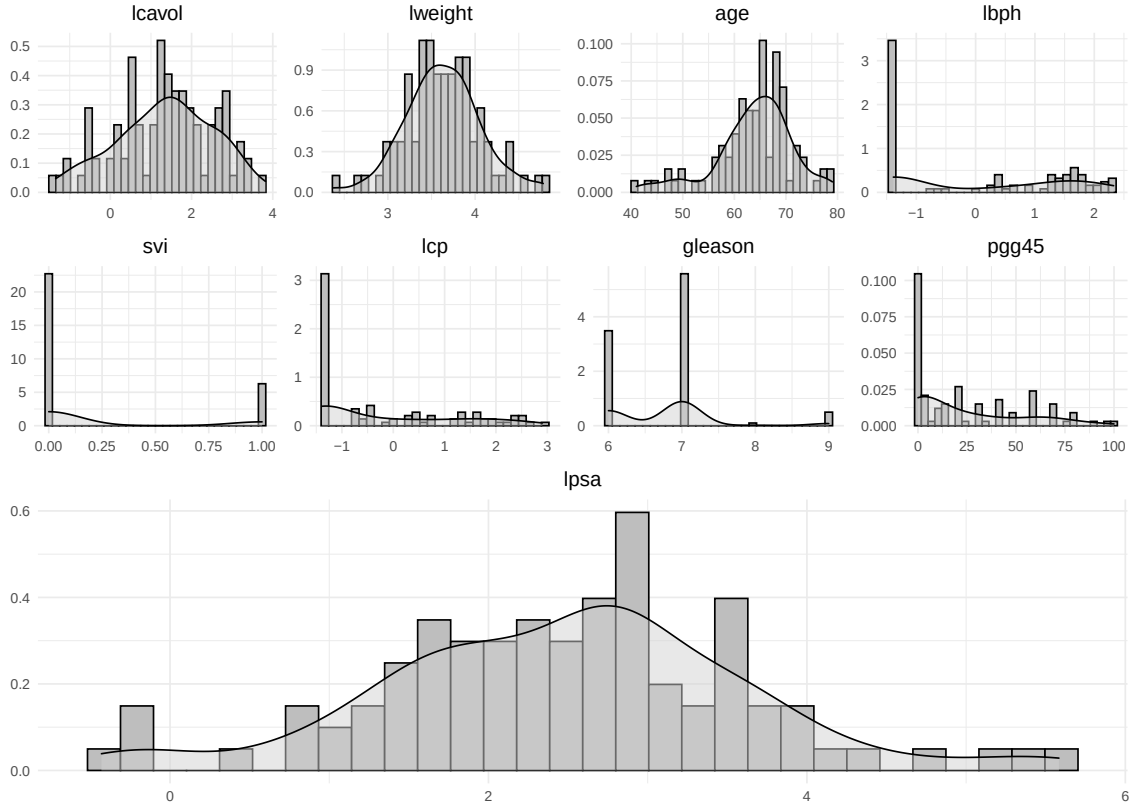


Figure 2.8: Histograms of explanatory covariates (the first two rows) and response (the last row) of the prostate cancer data set.

In this clinical study, doctors want to model the logarithm of the level of prostate-specific antigen before surgery, lpsa henceforth, in terms of eight clinical measures: log cancer volume (lcavol), log prostate weight (lweight), age, log of benign prostatic hyperplasia amount (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason) and percentage of Gleason scores 4 or 5 (pgg45). We refer the reader to Stamey et al. (1989) for more details about data. A summary of these variables is displayed in Figure 2.8. For the modeling purpose, a linear relation is assumed. There were a total of 97 men in the study, and this sample was randomly divided into a training set of size 67 and a test set of size 30 (see Hastie et al. (2009)). Following Hastie et al. (2009) and Székely and Rizzo (2014) guidelines, the training sample of $n = 67$ individuals is employed to select covariates and then, the remaining 30 samples are used to perform prediction.

In this case, there are covariates in quite different scales. The lowest value is 0.41, and this scale corresponds with the chlorides variable. In contrast, the variable with the highest

scale value is pgg45, taking a quantity of 28.20. The associated standard deviations of all covariates are displayed in Table 2.2.

	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$	$\mathbf{X_6}$	$\mathbf{X_7}$	$\mathbf{X_8}$	$\mathbf{Y}$
sd	1.18	0.43	7.45	1.45	0.41	1.40	0.72	28.20	1.15

Table 2.2: Standard deviations of X_1: lcavol, X_2: lweight, X_3: age, X_4: lbph, X_5: svi, X_6: lcp, X_7: gleason, X_8: pgg45 and Y: lpsa.

In terms of dependence structures, one can pay attention to the correlation matrix. There, we can appreciate as these covariates are related to each other. Specifically, there are high positive correlations. The correlation matrix is displayed in Figure 2.9.

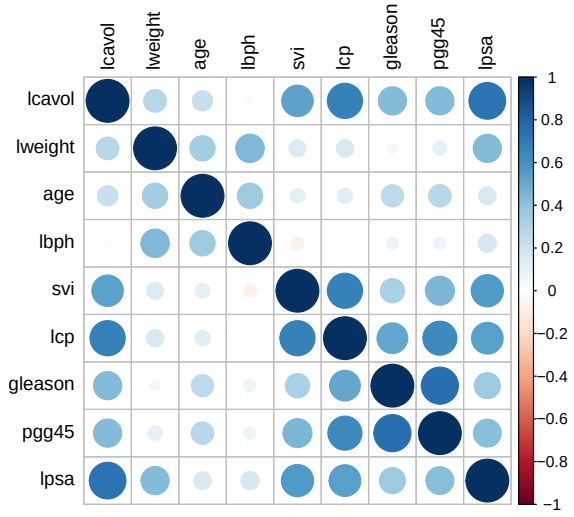


Figure 2.9: Correlation matrix of the prostate cancer data set.

In conclusion, this is a well-known example in the covariates selection field that exhibits different covariates scales and strong dependence structures between covariates. Comments about covariates selected in existing literature arise in Section 3.4 of Chapter 3.

2.5.3 Body fat

The body fat data set[2] consists in measures of 252 men to determine their percentage of body fat. This quantity is obtained using Siri's equation (Siri (1956)). For this aim, several measures about their body are taken. These are density determined from underwater weighing, age (years), weight (lbs), height (inches), and neck, chest, abdomen, hip, thigh, knee, ankle, biceps (extended), forearm as well as wrist circumference (cm). As a result, the aim is to determine which covariates are the most relevant ones in terms of proper body fat explanation.

[2]See `https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset`

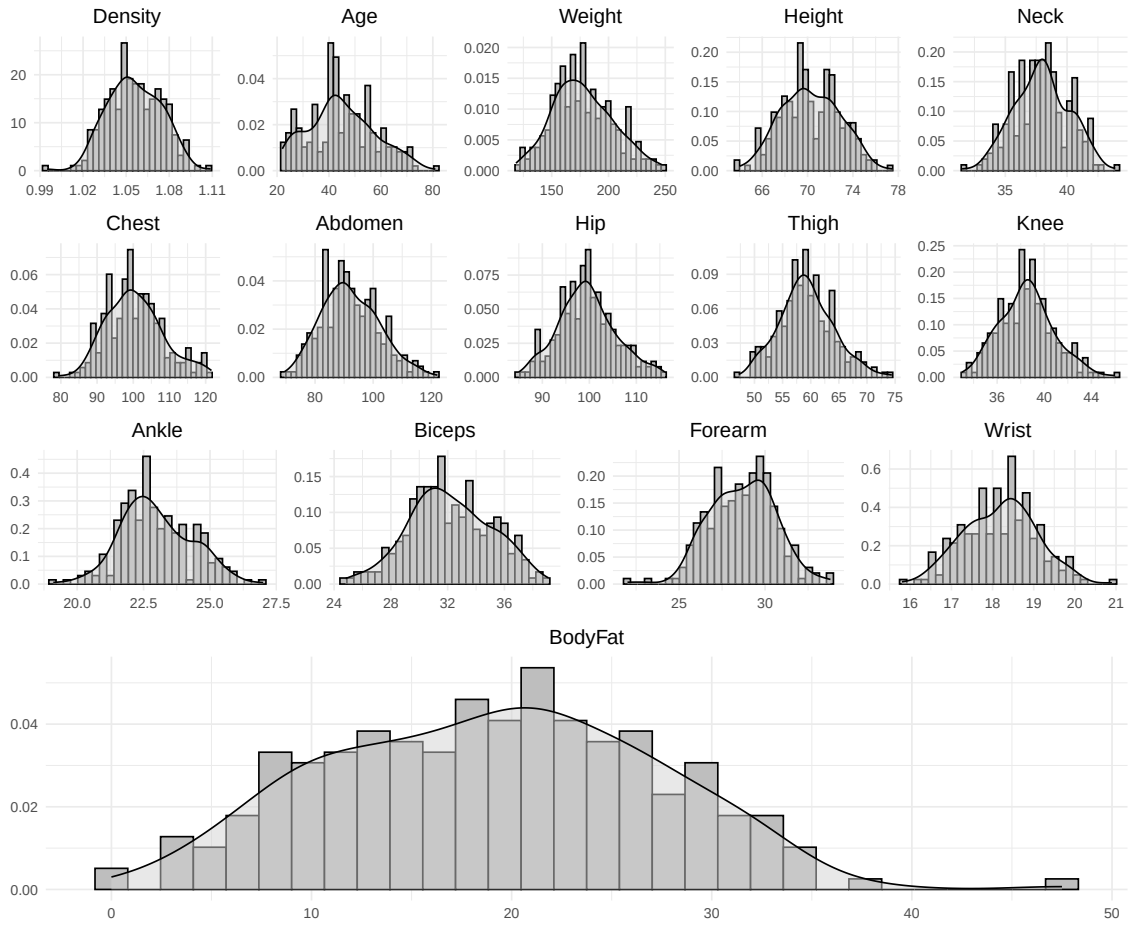


Figure 2.10: Histograms of explanatory covariates (the first three rows) and response (the last row) of the body fat data set.

In the first place, Box-Cox transformations are applied in $X+0.01$ and $Y+0.01$ variables to avoid skewness[3]. Next, we remove some outliers. For this aim, the Mahalanobis distance is employed over the data matrix to detect the 5% of less depth samples[4]. As a result, all samples out of this condition are removed. The preprocessed data has a sample size of length $n = 239$ and $p = 14$ covariates. The resulting variables are displayed in Figure 2.10.

The scales of the transformed covariates vary between them. These quantities are displayed in Table 2.3. These standard deviations range between values of $[0.02, 26.38]$. As a result, we can not assume that these terms are on a similar scale.

[3]The `boxcox` function of the `MASS` library of R (R Core Team (2019)) is used to perform the Box-Cox transformations of the covariates.

[4]The `mdepth.MhD` function of the library `fda.usc` (Febrero-Bande and Oviedo de la Fuente (2012)) is employed to detect outliers in a multivariate way.

	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$	$\mathbf{X_6}$	$\mathbf{X_7}$	$\mathbf{X_8}$
sd	0.02	12.76	26.38	2.57	2.21	7.86	9.92	6.13
	$\mathbf{X_9}$	$\mathbf{X_{10}}$	$\mathbf{X_{11}}$	$\mathbf{X_{12}}$	$\mathbf{X_{13}}$	$\mathbf{X_{14}}$	$\mathbf{Y}$	
sd	4.80	2.32	1.33	2.89	1.91	0.88	8.17	

Table 2.3: Standard deviations of X_1: Density, X_2: Age, X_3: Weight, X_4: Height, X_5: Neck, X_6: Chest, X_7: Abdomen, X_8: Hip, X_9 : Thigh, X_{10} : Knee, X_{11} : Ankle, X_{12} : Biceps, X_{13} : Forearm, X_{14} : Wrist and Y: BodyFat.

Next, it is of interest to inquire about possible dependence structures. Similar to previous examples, one can resort to the sample correlation matrix to have an idea. The resulting correlation matrix of the clean data is displayed in Figure 2.11. In this, strong relationships, positive and negative ones, are appreciated between most of the covariates.

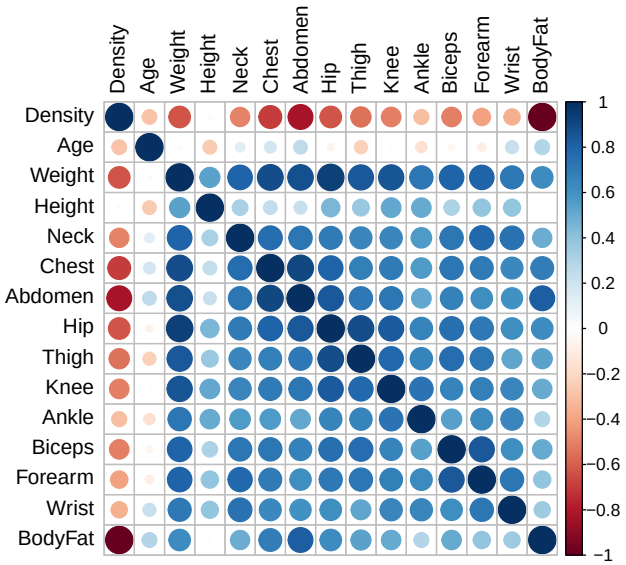


Figure 2.11: Correlation matrix of the body fat data set.

Thus, this is an example of a real data set with great discrepancies in covariates scales and with a lot of strong dependence patterns between the explanatory covariates.

2.5.4 Portuguese wine

In the last example, a selection process is carried out for a totally different framework. This is the Portuguese red wine data set studied in Cortez et al. (2009) [5]. Several physicochemical parameters are measured about the red vinho verde, which is a typical wine type from the northwest regions of Portugal. These parameters are fixed acidity (X_1), volatile acidity

[5]This is available in http://www3.dsi.uminho.pt/pcortez/wine/

(X_2), citric acid (X_3), residual sugar (X_4), chlorides (X_5), free sulfur dioxide (X_6), total sulfur dioxide (X_7), density (X_8), pH (X_9), sulphates (X_{10}) and alcohol (Y). All of them are continuous variables, and a total of $n = 1599$ samples, taken from May/2004 to February/2007, are available. Our objective is to model the alcohol by volume content using the rest of the $p = 10$ covariates.

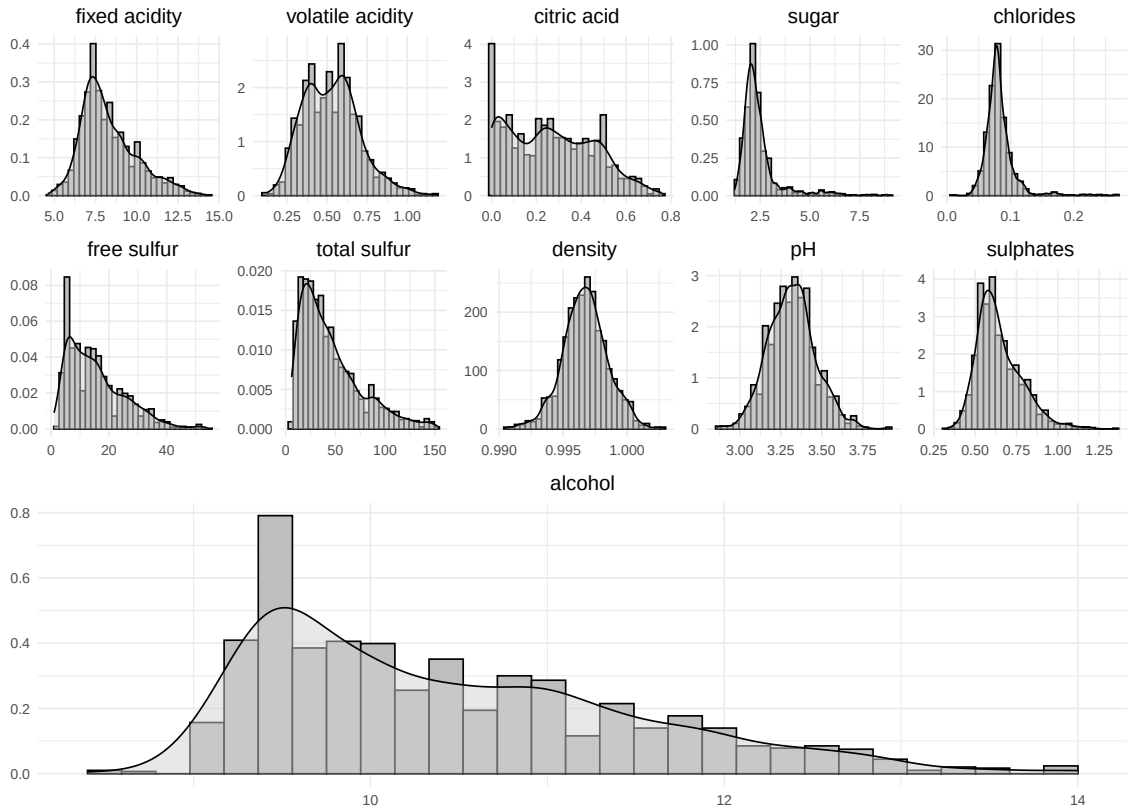


Figure 2.12: Histograms of explanatory covariates (the first two rows) and response (the last row) of the Portuguese wine data set.

Following the guidelines introduced above in Section 2.5.3, we first apply Box-Cox transformations over the $X+0.01$ and $Y+0.01$ variables to correct possible skewness. Next, data is cleaned from outliers, removing the 5% of extreme values. The less-depth samples are detected in the same way as in Section 2.5.3 and these are removed from the data set. This procedure results in a total of $n = 1519$ samples, removing 80 ones. A summary of the resulting variables is displayed in Figure 2.12.

After applying Box-Cox transformations and cleaning outliers, we can check for scales between covariates. These are estimated utilizing their standard deviations and are displayed in Table 2.4. We can see that all scales are between $(0, 30.39]$. Thus, this is an example where covariates scales differ among them. In particular, all covariates have similar scales except for X_6: free sulfur and X_7: total sulfur.

	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$	$\mathbf{X_6}$	$\mathbf{X_7}$	$\mathbf{X_8}$	$\mathbf{X_9}$	$\mathbf{X_{10}}$	$\mathbf{Y}$
sd	1.67	0.17	0.19	1	0.02	9.88	30.39	$< 10^{-2}$	0.14	0.14	1.05

Table 2.4: Standard deviations of X_1: fixed acidity, X_2: volatile acidity, X_3: citric acidity, X_4: sugar, X_5: chlorides, X_6: free sulfur, X_7: total sulfur, X_8: density, X_9: pH, X_{10}: sulphates and Y: alcohol.

A study of correlation values after removing outliers is displayed in Figure 2.13 computing the sample correlation matrix. In this case, we appreciate that there exists some strong dependence relation between some covariates, but most of them are weak.

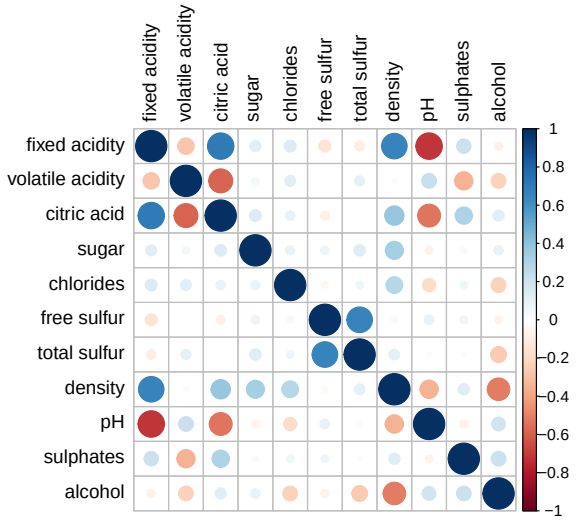


Figure 2.13: Correlation matrix of the Portuguese wine data set.

In this last case, there are only two considered covariates that have a quite different range of values. Besides, there are some, but not too many, strong correlations.

2.6 Analysis of the LASSO evolution and alternatives

Motivated by the increasing number of data sets in the high dimensional framework, the LASSO regression has gained vast popularity in the last few years. Some examples are the ones introduced in Section 2.5. Consequently, the LASSO procedure results in a tool widely employed due to its desirable qualities, such as convexity or getting a sparse estimator $\hat{\beta}$ in the linear regression model formulation given in (1.2). This last property allows one to implement covariates selection, even when $p > n$. We refer to Section 2.1 for more details. However, the LASSO approach has some important limitations in practice. These are related to bias, strict theoretical properties to ensure consistency, the occurrence of false discoveries in the LASSO selection, and the difficulty in selecting a suitable value for the penalty term λ. All these inconveniences have been explained in detail in Section 2.2.

As a result of these limitations, new adaptations of the usual LASSO were proposed. See Section 2.3 for an overview. We do a humble classification by distinguishing between weighted versions, resampling techniques mixed with LASSO, or thresholded versions. Whereas the weighted versions are devoted to correcting the bias, the remaining two groups focus on reducing false discoveries. Moreover, different formulations of the LASSO approach were also developed to consider the information provided by special structures of the data in the selection process. Nevertheless, these options keep some of the limitations of the usual LASSO. For example, some conditions in the matrix design, β vector, and sample size are still necessary to guarantee the proper recovery of the relevant terms. In addition, a suitable selection of the penalty parameter is necessary as well. In addition, new tuning parameters appear for weighted and thresholded versions, which translates into greater complexity. In contrast, the resampling LASSO procedures "solve" the problem of false discoveries, but requires high computational cost and suffer from the previously mentioned problems as well.

To deal with the drawbacks mentioned above, alternatives to LASSO have also been proposed in the literature. A summary of some of the most employed or more innovative approaches is collected in Table 2.1. Besides, these are treated in more detail throughout Section 2.4. These procedures have diverse forms and natures. Some examples of these ideas are based on the use of concave penalties to reduce bias, as the SCAD penalization (Section 2.4.1), or the mix of L_1 and L_2 penalizations via the ENet penalty (Section 2.4.2) to protect against the confusion phenomenon. An additional alternative is a non-convex penalty proposed by the Dantzig selector (Section 2.4.3), resulting in an estimator invariant to orthonormal transformations. The relaxed LASSO (Section 2.4.4) is another option that achieves a faster converge rate than the LASSO one and obtains sparser estimators. Related to the optimal choice of the λ term, which depends on the unknown error variance σ^2 (see Section 2.2.4), arises the square root LASSO (Section 2.4.5) and the scaled LASSO (Section 2.4.6). The first approach guarantees that a proper selection of the penalty term does not depend now on the error variance, whereas the second one provides a two-step procedure capable of estimating this variance. Alternative procedures devoted to correcting the false discoveries produced by the LASSO algorithm are the SLOPE penalization (Section 2.4.7) or the innovative knockoffs filter (Section 2.4.8). However, to the best of our knowledge, this last procedure is only available for the $n > p$ framework. Other novel procedures which perform covariates selection by means of hypothesis testing are the debiased LASSO (Section 2.4.9) and the distance covariance algorithm (Section 2.4.10). The debiased LASSO obtains a sparse estimator with a Gaussian known distribution and allows one to perform multiple hypothesis testing to select covariates. By its part, the distance covariance algorithm employs the distance correlation coefficient of Székely et al. (2007), which will be treated in more detail in Chapter 4, to perform independence tests. This last translates into a covariates selection procedure. The last proposed procedure is the LASSO-zero (Section 2.4.11). This technique, using noise dictionaries concatenated to X, is consistent for sign recovery of the true β vector and obtains good results in terms of avoiding false recoveries. In spite of these improvements, all alternative procedures also

suffer from some of the LASSO drawbacks. So these approaches are expected to improve the LASSO performance in some sense, but there is no one-size-fits-all solution. As a result, it will depend on the desired goal that one algorithm is more suitable than others.

Although varying in their form and characteristics, all proposed algorithms have been developed to answer the covariates selection problem. In particular, these procedures have been built to solve some of the LASSO drawbacks introduced in Section 2.2. Nevertheless, it is not clear which of these proposals is the best option in terms of recovery of the relevant covariates. This topic is discussed below in Chapter 3. Specifically, their behavior is compared under tricky contexts, as assuming distinct dependence structures between variables (Section 3.1) or when there are covariates in different scales (Section 3.2). These frameworks are motivated by the usual properties of real data sets. In Section 2.5, four examples are given in which these phenomena arise. The performance of the considered algorithms is also tested in these examples in Section 3.4 of Chapter 3.

Apart from the linear formulation of (1.2), studied in Section 1.1.1, penalization techniques can be extended to more complex models. An example is their use in the additive regression model introduced in Section 1.1.2. The work of Ravikumar et al. (2009), with the SpAM (Sparse Additive Models) procedure, is an example where a L_1 penalty is imposed to force covariates selection and avoid the concurvity effect discussed in Section 1.2.3. The Lazy LASSO is an alternative proposed by Vidaurre et al. (2012) to penalize the local regression introduced in Section 1.1.3. In particular, this imposes a type of L_1 penalty to avoid overfitting and apply covariates selection in local regression settings. Another possibility is their implementation in generalized linear models (see McCullagh and Nelder (2019)), including the well-known logistic regression. A survey in the use of the L_1 penalty for these models can be found in Vidaurre et al. (2013). Even for the generalized additive model (see Hastie and Tibshirani (1990)), penalization techniques like the generalized SpAM algorithm of Haris et al. (2022) can be employed for covariates selection.

Furthermore, the L_1 penalty philosophy of the LASSO has been extended to trickier contexts. An example is the work of Lee et al. (2016) for the varying coefficient models of Hastie and Tibshirani (1993). The form of these models is introduced in equation (4.28). The authors propose an adaptation of the group LASSO procedure (see Section 2.3.4) to the varying coefficient models in order to obtain a sparse version in a high dimensional context. Besides, in the case of functional regression models, approaches applying penalizations for covariates selection have been proposed as well. A review of this topic can be seen in Aneiros et al. (2022). There, several covariates selection procedures related to LASSO ideas are collected. A particular case of a functional model is the functional concurrent model (FCM). In this, the relation between covariates and response is concurrent or point-by-point. A more detailed introduction is given in Section 5.1. For this model, Ghosal and Maity (2022b) propose a covariates selection procedure using ideas of group penalizations. In particular, the group LASSO approach of Section 2.3.4 is considered. Below, in Chapter 4, novel selection techniques are introduced, and a discussion about their implementation in the commented contexts arises in Section 4.3. In particular, new covariates selection approaches for the FCM are introduced in Chapters 5 and 6 using these methodologies.

Chapter 3

LASSO regression as a variable selector. Performance under dependence structures and different scales on covariates

In this chapter, we analyze the LASSO performance as variables selector under different dependence frameworks where all covariates are in a unit scale. For this aim, an extensive simulation study is performed. Its behavior is also compared with that of some adequate derivatives and competitors. This analysis is carried out in Section 3.1. Complete Section 3.1 is collected in Freijeiro-González et al. (2022a). Next, in Section 3.2, there are considered not only dependence structures but covariates with different scales as well. In this case, we test how LASSO and its competitors perform in these contexts. Besides, we compare the use of without or univariate standardizations for all contexts. Next, in Section 3.3, the possibility of applying a first screening step for dimensionality reduction is analyzed, considering different dependence coefficients. Eventually, we analyze the four real data sets introduced in Section 2.5 of Chapter 2, considering all the proposed guidelines. This analysis is developed in Section 3.4.

3.1 Problems of the LASSO regression under dependence structures

As commented in Section 2.2, the LASSO suffers from some limitations as a variable selector. These are related to its biased nature, the great number of false discoveries, and its difficulty in estimating a proper value of the regularization parameter λ in practice. Furthermore, an additional limitation is the requirement of strict conditions in the model design. These can not be always verified in practice, specifically when there exist strong dependence patterns between covariates, resulting in possible collinearity effects. Related to this last, Zou and Hastie (2005) proved that, when high dependence structures exit, i.e. some covariates highly correlated, the LASSO algorithm tends to pick some of them randomly and avoid the remaining ones. This selection could result in a loss of information if these related terms are relevant or in a confusion phenomenon when there are strong relations between important and noisy covariates. In practice, one expects to have some class of dependence structure in the study data and face this issue. Some examples are the real data sets introduced in Section 2.5. Consequently, it is of great interest to determine how LASSO works under different dependence structures, whether any covariates selection technique can prevent these drawbacks, and to what extent.

Thus, is LASSO the best option or at least a good starting point to identify the relevant covariates? Although some studies like the one of Su et al. (2017) discuss this topic, one can not find a totally convincing answer to this question for dependence scenarios. In

order to shed light on this topic, the LASSO performance is tested under some different and controlled dependence structures. Both are analyzed: the classic situation considering $n \geq p$ and the high dimensional framework where $p > n$.

Furthermore, in view of the LASSO limitations, a global comparison with suitable modifications and alternatives is developed. These competitors are introduced throughout Sections 2.3 and 2.4. Hence, we test what procedures are capable of overcoming the LASSO drawbacks in the proposed dependence contexts. For this comparison, we select different approaches that have proved their efficiency in practice. Finally, some conclusions are drawn based on the simulation results about what is the best possible option in terms of the dependence nature of the data.

3.1.1 Simulation scenarios

In this section, we introduce different simulation scenarios to test the performance of LASSO as a covariates selector under different dependence structures. Scenarios verifying and not the consistency conditions mentioned in Section 2.2.2 are simulated, and their results are compared with those of other procedures. For this purpose, we carry out a Monte Carlo study taking $M = 500$ simulations. Three dependence scenarios are introduced, simulating them under the linear regression model structure given by (1.2). Being S the index set of relevant covariates with $s = \#S$, β is considered as a sparse vector of length p with only $s < p$ values not equal zero, $\mathbf{X_n} \in \mathbb{R}_{n\times p}$ where n is the sample size and $\varepsilon \in N_n(0, \sigma^2 I_n)$. We fix $p = 100$ and chose σ^2 by verifying that the percentage of explained deviance is explicitly the 90%. Calculation of this parameter is collected in Section A.1 of the Appendix A. To guarantee an optimal LASSO performance, it is needed $n > 4.61s$ as seen in (2.8), $\inf |\beta_j| > 2.15\sqrt{s/n}$ for $j \in S$ as in (2.7) and to take λ of order $2.15\sigma\sqrt{1/n}$. We test its behavior considering different combinations of parameter values, taking $n = 25, 50, 100, 200, 400$ and $s = 10, 15, 20$. A study of when these conditions hold is shown in Section A.2 of the Appendix A. In every simulation, the number of covariates correctly selected ($|\hat{S} \cap S|$) and the noisy ones ($|\hat{S} \setminus S|$) are counted. Besides, the prediction power of the algorithm is measured using the mean squared error (MSE) and the percentage of explained deviance $\%\text{Dev} = (RSS - RSS_0)/(RSS_0)$, being $RSS = \sum_{i=1}^{n}(y_i - \hat{\beta}X_i)^2$ the residual sum of squares of the model and $RSS_0 = \sum_{i=1}^{n} y_i^2$. The MSE gives one an idea about the bias produced by the LASSO (see Section 2.2.1).

- **Scenario 1** (*Orthogonal design*). Only the first s values are not equal zero for β_j with $j = 1, \ldots, s$ and $p > s > 0$, $\beta_1 = \cdots = \beta_s = 1.25$, while $\beta_j = 0$ for all $j = s+1, \ldots, p$. X is simulated as a $N_n(0, I_p)$.

- **Scenario 2** (*Dependence by blocks*). The vector β has the first $s < p$ components not null, of the form $\beta_1 = \cdots = \beta_s = 1$ and $\beta_j = 0$ for the rest. X is simulated as a $N_n(0, \Sigma)$, where $\sigma_{jj} = 1$ and $\sigma_{jk} = cov(X_j, X_k) = 0$ for all pairs (j, k) except if $mod_{10}(j) = mod_{10}(k)$, in that case $\sigma_{jk} = \rho$, taking $\rho = 0.5, 0.9$.

- **Scenario 3** (*Toeplitz covariance*). Again, only s ($p > s > 0$) covariates are important,

simulating X as a $N_n(0, \Sigma)$ and assuming $\beta_j = 0.5$ in the places where $\beta \neq 0$. In this case, $\sigma_{jk} = \rho^{|j-k|}$ for $j, k = 1, \ldots, p$ and $\rho = 0.5, 0.9$. Now, we analyze two different dependence structures varying the location of the s relevant covariates:

- **Scenario 3.a**: we assume that the relevant covariates are the first $s = 15$.
- **Scenario 3.b**: consider $s = 10$ relevant variables placed every 10 sites, which means that only the $\beta_1, \beta_{11}, \beta_{21}, \ldots, \beta_{91}$ terms of β are not null.

The first choice, the orthogonal design of Scenario 1, is selected as the best possible framework. This scenario verifies the consistency conditions for values of n large enough and avoids the confusion phenomenon, given that there are no correlated covariates.

In contrast, to assess how the LASSO behaves in the case of different dependence structures, Scenario 2 and Scenario 3 are proposed. In the dependence by blocks context (Scenario 2), the design is forced to have a dependence structure where the covariates are correlated ten by ten. As a result, a more challenging scenario for the LASSO is induced, in which the algorithm has to overcome a fuzzy signal produced by irrelevant covariates. Different magnitudes of dependence are considered in Scenario 2 with $\rho = 0.5$ and $\rho = 0.9$ to test the effect of the confusion phenomenon. As a result, different sizes of n are needed in terms of s to guarantee the proper behavior of the LASSO. This scenario has already been studied in other works, like in Meinshausen and Bühlmann (2010).

Eventually, we test the LASSO performance in a scenario where all the covariates are correlated: the Toeplitz covariance structure (Scenario 3). This scenario mimics a type of functional dependence pattern. This setting is an example where the irrepresentable condition holds (see Bühlmann and Van De Geer (2011)), but the algorithm suffers from highly correlated relations between the actual set of covariates and unimportant ones. The LASSO has been employed previously to study this framework. See, for example, Meinshausen and Bühlmann (2010) or Bühlmann and Van De Geer (2011). Because the distance between covariates is relevant to establish their dependence, two different frameworks are studied. In the first scenario (Scenario 3.a), the important covariates are highly correlated among them and little with the rest. Particularly, there are only notable confusing correlations in the case of the last variables of $S = \{1, \ldots, 15\}$ with their noisy neighbors. Here, the LASSO can only recover S in the $n = 400$ case. In contrast, in Scenario 3.b, the important covariates are markedly correlated with unimportant ones, so this location magnifies the spurious correlations phenomenon. For this scenario, a sample size of $n = 200, 400$ is necessary.

3.1.2 Performance of the LASSO in practice under dependence

In order to test the inconveniences of the LASSO when there exists dependence among covariates, a complete simulation study is carried out. For this purpose, the simulation scenarios introduced above in Section 3.1.1 are employed. Here, a summary of the results is shown, and the behavior of LASSO in the three proposed frameworks is analyzed. Complete results are collected in Section A.6 of the Appendix A.

The performance of the standard LASSO is tested using the library `glmnet` (Friedman et al. (2010)) implemented in R (R Core Team (2019)). This algorithm uses K-fold cross-validation (CV) to select the λ parameter that minimizes the MSE, $\lambda^{\min}$. This procedure is denoted by LASSO.min. See Friedman et al. (2010) for more details. As explained in Section 2.2.4, this is one of the most popular ways of estimating λ. To be capable of comparing different models and following recommendations of the existing literature, we fix $K = 10$ for all simulations. Besides, the response $\mathbf{Y_n}$ is centered, and the matrix $\mathbf{X_n}$ is standardized by columns. This last would not be really necessary for these frameworks because the covariates are all on the same scale. However, this last is done to keep the usual implementation of LASSO-type algorithms in practice. A two-step LASSO-OLS version to adjust the model is applied (see Belloni and Chernozhukov (2013)). This scheme is also followed for the rest of the procedures. The grid of values for the tuning parameter is taken of length 100 and is calculated based on the sample data and methodology employed, following the author's recommendation. More details are given in Section A.3 of the Appendix A.

There are other faster algorithms available in R, such as the famous LARS procedure (Efron et al. (2004), Hastie and Efron (2013)). However, the decision of making use of the `glmnet` library is due to its easy implementation and interpretation, jointly with its simple adaptation to other derivatives of the LASSO that are tested in this study later.

In the orthogonal design of Scenario 1, one would expect the LASSO to recover the whole set of important covariates and not add too much noise into the model for a large enough value of n. However, different results have been observed.

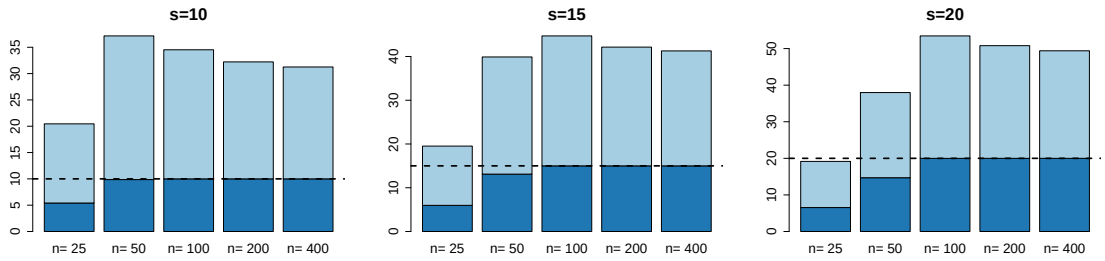


Figure 3.1: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.min in Scenario 1. The dashed line marks the s value.

Firstly, one can appreciate that it does not really matter the number of relevant covariates considered ($s = 10, 15, 20$) in relation to the capability of recovering this set. It is because the algorithm only includes the complete set under the $n \geq p$ framework except for the $s = 10$ scenario taking $n = 50$. See this fact in Figure 3.1. This can be easily explained in terms of the consistency requirements given in (2.8). Besides, although we are under orthogonal design assumption, this includes a lot of noisy variables in the model. What is shocking is the fact that the number of irrelevant covariates selected is always larger than the important ones. This exemplifies the existing trade-off between FDP and TPP introduced in (2.9) as well as that both quantities can not be simultaneously low.

	s = 10		**s = 15**		**s = 20**	
	MSE (1.736)	**% Dev**	**MSE** (2.604)	**% Dev**	**MSE** (3.472)	**% Dev**
$n = 50$	0.169	0.990	0.579	0.973	1.747	0.944
$n = 100$	0.702	0.959	0.841	0.967	0.914	0.973
$n = 200$	1.164	0.932	1.628	0.936	2.060	0.940

Table 3.1: Summary of the LASSO.min results for Scenario 1. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

In the second place, one notices that this procedure clearly overestimates its results. This obtains values for the MSE and percentage of explained deviance fewer and greater, respectively, of the oracle ones (see values in brackets in Table 3.1). In conclusion, this toy example, it is illustrated how the LASSO.min procedure performs very poorly and presents important limitations even in an independence framework.

The overestimation of the set S is likely because a larger value of λ is necessary for proper covariates selection. In Section A.3.2 of the Appendix A, some values greater than $\lambda^{\min}$ are chosen and tested. These outperform the LASSO.min performance regarding the recovery of S and avoid irrelevant information. Nevertheless, some guidance criterion is needed to select a penalization value in practice. Friedman et al. (2010) proposed the alternative of estimating the mean cross-validated error for every value of the λ grid and taking λ^{1se}. This value is the largest value of λ verifying that its error is within 1 standard error of the minimum ($\lambda^{\min}$). Complete results of LASSO using λ^{1se} (LASSO.1se) are collected in Section A.6 of the Appendix A. Given the results, one can appreciate that this selection of the penalization term makes sense and improves the LASSO.min performance.

The inclusion of too many noisy covariates could also be due to the selection criterion. Cross-validation searches for the λ value minimizing the mean squared error, which is helpful for estimation of $X\beta$ but can fail for covariates selection. As mentioned in Section 2.2.4, the optimal value of λ changes according to one or the other objective. Thus, different techniques as a criterion based on information theory may achieve a better performance recovering S. To fill this gap, we carry out a comparison with the Bayesian information criterion (BIC). This methodology is denoted by LASSO.BIC. A summary of its results is displayed in Figure A.6 and Table A.2. See the complete results and details bout its implementation in Section A.3.3 of the Appendix A.

It is possible to appreciate as the BIC criterion helps to reduce the inclusion of noisy covariates for $n > p$. Besides, this approach corrects a bit of the overestimation in this framework, although this is not removed. In contrast, its performance is pretty bad for $p \geq n$. The BIC criterion strongly overfits the results in the last cases: this adds more noise to the model and produces more overestimation than the LASSO.min. See an analysis of this topic in Giraud et al. (2012) for more details.

Next, the results of the dependence by blocks context are analyzed. In the case of dependence, one expects a "smart" algorithm to be capable of selecting a portion of relevant covariates and explaining the remaining ones using the existing correlation structure. The

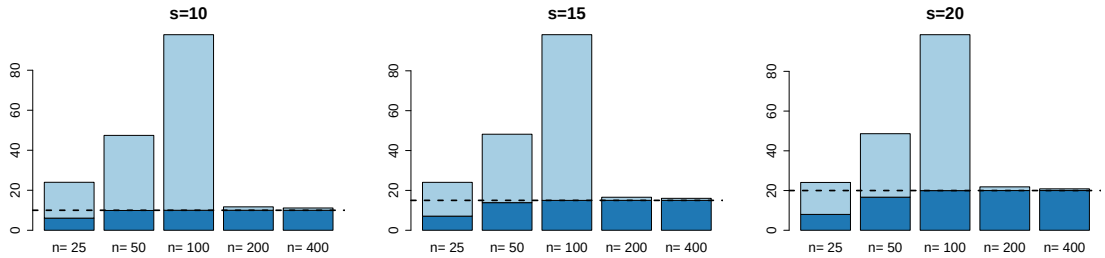


Figure 3.2: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 1. The dashed line marks the s value.

	s = 10		**s = 15**		**s = 20**	
	MSE (1.736)	**% Dev**	**MSE** (2.604)	**% Dev**	**MSE** (3.472)	**% Dev**
$n = 50$	0.001	1	0.001	1	0.001	1
$n = 100$	0.014	0.999	0.011	1	0.005	1
$n = 200$	1.544	0.910	2.272	0.911	2.950	0.913

Table 3.2: Summary of the LASSO.BIC results for Scenario 1. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

subset of S that is really necessary to explain this type of model is denoted as "effective covariates" and is unknown in practice. An idea to calculate this is to measure how many terms are necessary to explain a certain percentage of Σ_S variability, being Σ_S the submatrix of Σ considering the elements of S. This number is inversely proportional to the dependence strength. For example, to explain the $90 - 95\%$ of variability in Scenario 2 with $\rho = 0.5$, we need about $12 - 14$ covariates taking $s = 15$ and about $16 - 18$ for the case of $s = 20$. Conversely, only 10 terms are necessary for Scenario 2 with $\rho = 0.9$. The complete calculation for the different combinations of parameters in the simulated scenarios is displayed in Section A.5 of the Appendix A. Again, the LASSO.min presents some difficulties for an efficient recovery in the dependence by blocks context.

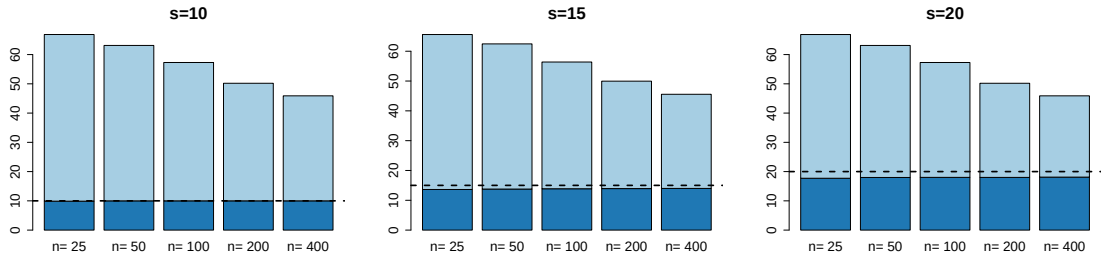


Figure 3.3: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.min in Scenario 2 with $\rho = 0.5$. The dashed line marks the s value.

A summary of the results for the Scenario 2 with $\rho = 0.5$ is displayed in Table 3.3 and Figure 3.3, while for the Scenario 2 with $\rho = 0.9$ is shown in Table 3.4 and Figure 3.4. If

	s = 10		s = 15		s = 20	
	MSE (0.556)	**% Dev**	**MSE** (1.389)	**% Dev**	**MSE** (2.222)	**% Dev**
$n = 50$	0.438	0.956	1.095	0.956	1.752	0.956
$n = 100$	0.495	0.951	1.238	0.951	1.981	0.951
$n = 200$	0.523	0.951	1.307	0.950	2.091	0.951

Table 3.3: Summary of the LASSO.min results for Scenario 2 with $\rho = 0.5$. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

only $s = 10$ relevant explanatory variables are considered, its behavior is quite similar to the one in Scenario 1. Besides, in both scenarios with $s = 10$, LASSO.min almost recovers the complete set S, even for $n = 25$, although its proper recovery is guaranteed from $n = 50$. However, more noise is included in this scenario as expected.

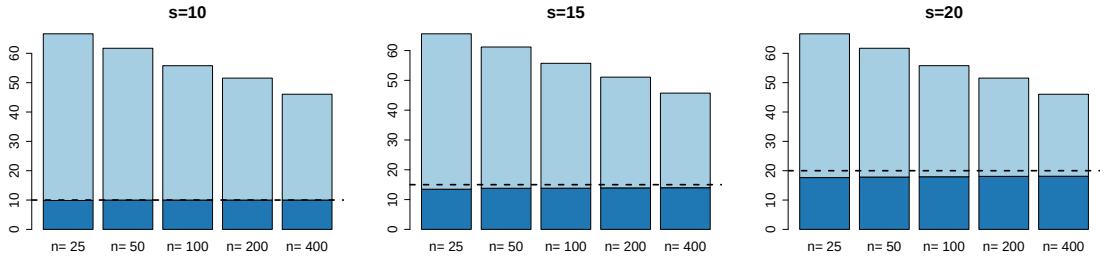


Figure 3.4: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.min in Scenario 2 with $\rho = 0.9$. The dashed line marks the s value.

	s = 10		s = 15		s = 20	
	MSE (1)	**% Dev**	**MSE** (2.5)	**% Dev**	**MSE** (4)	**% Dev**
$n = 50$	0.784	0.926	1.96	0.925	3.137	0.926
$n = 100$	0.888	0.918	2.22	0.918	3.551	0.918
$n = 200$	0.939	0.913	2.347	0.913	3.756	0.913

Table 3.4: Summary of the LASSO.min results for Scenario 2 with $\rho = 0.9$. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

In contrast, the situation is different if we simulate with $s = 15$ or $s = 20$ relevant covariates. Then, the LASSO.min does not tend to recover all the covariates of S, not even for values of n verifying $n \geq p$ as well as conditions (2.8) and (2.7). See Section A.2 of the Appendix A for more information. However, this selects more than the effective number of covariates. It seems the LASSO.min tries to recover the set S but, due to the presence of spurious correlations, this randomly chooses between two highly correlated and important covariates. We can appreciate in Tables A.35 and A.36 in Section A.7.1 of the Appendix A that the 10 first covariates are selected with high probability, near 1. However, due to the confusion phenomenon, some of these are interchanged by a different

representative term. The following $s - 10$ relevant variables have a lower selection rate, and some irrelevant ones are selected a higher number of times, adding quite a noise to the model. This inconvenience seems not to be overcome by increasing the number of samples n. Again, we can see by the percentage of explained deviance and the MSE as the LASSO.min keeps overestimating its results.

We observe a similar behavior selecting the λ value greater than $\lambda^{\min}$ (see LASSO.1se) and using the BIC criterion, although this last tends to add more noise. Furthermore, for $p \geq n$, the LASSO.BIC can not deal with the dependence structure, selecting fewer than s covariates in some cases and overestimating the prediction results. In contrast, the LASSO.1se does not improve the LASSO.min performance in this scenario. We observe the same phenomenon for greater values than λ^{1se}. See Section A.3.2 of the Appendix A. Results for the LASSO.BIC and LASSO.1se algorithms are collected in Sections A.3.3 and Section A.6 of the Appendix A, respectively.

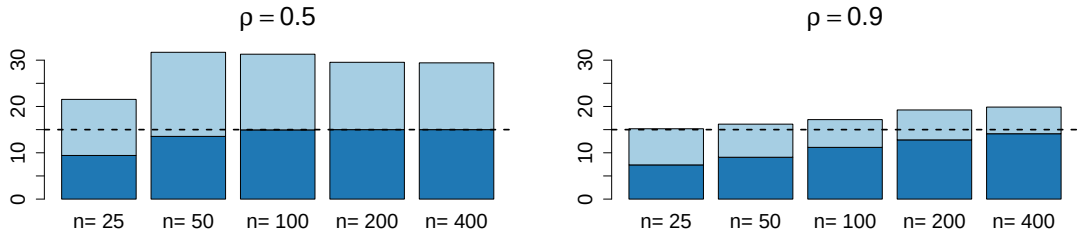


Figure 3.5: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.min in Scenario 3.a. The dashed line marks the $s = 15$ value.

	Scenario 3.a				**Scenario 3.b**			
	$\boldsymbol{\rho = 0.5}$		$\boldsymbol{\rho = 0.9}$		$\boldsymbol{\rho = 0.5}$		$\boldsymbol{\rho = 0.9}$	
	MSE (1.139)	**% Dev**	**MSE** (3.807)	**% Dev**	**MSE** (0.278)	**% Dev**	**MSE** (0.53)	**% Dev**
$n = 50$	0.19	0.983	1.894	0.950	0.034	0.987	0.147	0.971
$n = 100$	0.546	0.951	2.815	0.928	0.123	0.955	0.309	0.94
$n = 200$	0.825	0.927	3.302	0.916	0.195	0.929	0.417	0.920

Table 3.5: Summary of the LASSO.min results for Scenario 3.a and Scenario 3.b. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

Finally, the results of LASSO in the Toeplitz covariance structure framework are studied. For this aim, there are considered Scenario 3.a, where the relevant covariates are the first $s = 15$ (Table 3.5 and Figure 3.5), and the Scenario 3.b, where there are only $s = 10$ important variables placed every 10 sites (Table 3.5 and Figure 3.6).

Interpreting its results, one sees that the LASSO.min procedure recovers the important set of covariates for $\rho = 0.5$, taking a value of $n = 100, 200, 400$. Nevertheless, the LASSO.min exceeds the number of efficient covariates selected in Scenario 3.a for $\rho = 0.9$,

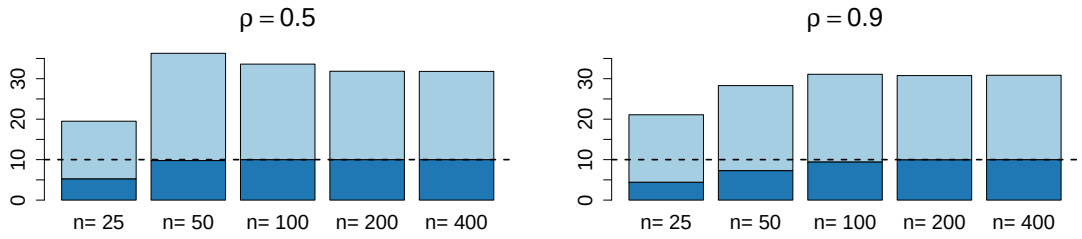


Figure 3.6: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.min in Scenario 3.b. The dashed line marks the $s = 10$ value.

because with just 10 covariates it is explained the 98% of variability. Moreover, this algorithm returns to include many pointless covariates in the model and overestimates the prediction accuracy.

The LASSO.1se behavior is quite similar to the LASSO.min, although the LASSO.1se reduces the noise and corrects the overestimation a bit. Taking larger values than λ^{1se}, the performance of the LASSO is even better. However, it seems quite difficult to establish a common rule for the optimal λ selection in the different frameworks of Scenarios 3.a and 3.b. Specifically, Scenario 3.b with $\rho = 0.9$ seems to need a larger value than the remaining ones. See Section A.3.2 of the Appendix A for a graphical comparison.

In contrast, the BIC adjustment selects fewer covariates, tending $\#\hat{S} = s$ as n increases, except in Scenario 3.b with $\rho = 0.9$. For Scenario 3.a, this selection procedure tends to recover S for a large enough value of n, avoiding irrelevant information. Nevertheless, something different happens for Scenario 3.b. In this last, the algorithm interchanges relevant covariates with irrelevant ones quite correlated with the ones of S. Despite this, the algorithm includes with high probability representatives of the $s = 10$ relevant covariates, especially for $\rho = 0.9$, capturing the essential information. This can be appreciated in Section A.7.2 of the Appendix A. Moreover, this procedure corrects a bit of the overestimation of the CV technique. Results are collected in the Appendix A: in figures in Sections A.3.3 and A.7.2, and Table A.5 in Section A.3.3.

3.1.3 Comparison with competitors

In Sections 2.3 and 2.4 of Chapter 2, an extensive list of adaptations and competitors of the LASSO is given. Nevertheless, not all these procedures are considered for the study. A selection of the most relevant methodologies in terms of good qualities and reasonable computational time has been made. Because of the nature of the simulation scenarios of Section 3.1.1, some procedures have been discarded due to their unsuitable characteristics.

Owing to the computational cost required by the resampling LASSO procedures, such as the boLASSO of Bach (2008) or the random LASSO algorithm (Wang et al. (2011)), these procedures are too slow. Even for small values of p, computational time was high. For this reason, these are excluded from the comparative analysis studio. The LASSO-Zero

technique of Descloux and Sardy (2021) suffers from the same issue, so this is excluded too.

Another problem springs up for the thresholded versions of the LASSO. In this case, the complexity of finding a correct threshold is similar to that of obtaining the optimal value of λ for the LASSO adjustment. In both cases, it would be necessary to know the dispersion of the error σ in advance, which is unknown in practice and can be difficult to estimate, especially when $p > n$. Then, procedures like the thresholded LASSO algorithm of Lounici (2008) are excluded to avoid more complications in the adjustment.

A method in the middle of both groups is the stability selection procedure proposed by Meinshausen and Bühlmann (2010). This methodology pays attention to the probability of each covariate being selected. Only the covariates with probability greater than a fixed threshold $q \in (0, 1)$ are added to the final model. Although the authors recommend taking $q \in (0.6, 0.9)$, we have observed in practice that a proper choice of the threshold value seems to depend on the sample size considered, n, as well as the sparsity of the vector β. Besides, an extra tuning parameter is needed: the bound for the expected number of false positives. See Dezeure et al. (2015) for more practical details. For all these reasons, this approach is not included in the comparison either.

In addition, methodologies with available code in R (R Core Team (2019)) are chosen, so everyone can make use of them. Thus, there have been chosen libraries that provide one with enough resources to fit the models, selecting those created by the author's methodology or the most recently updated option in case of doubt. This selection is:

LASSO: `glmnet` of Friedman et al. (2010), last update November 27, 2022.

SCAD: `ncvreg` of Breheny and Huang (2011), last update October 13, 2022.

AdapL: `glmnet` of Friedman et al. (2010), last update November 27, 2022.

Dant: `flare` of Li et al. (2019), last update October 13, 2022.

RelaxL: `relaxo` of Meinshausen (2012), last update May 23, 2022.

SqrtL: `flare` of Li et al. (2019), last update October 13, 2022.

ScalL: `scalreg` of Sun (2019), last update October 14, 2022.

Distance correlation algorithm for variable selection (DC.VS): `fda.usc` of Febrero-Bande and Oviedo de la Fuente (2012), last update October 17, 2022.

Next, we display the results of the simulation study, comparing the performance of different procedures that have shown suitable properties.

The first framework to be studied is the easiest one: the orthogonal design (Scenario 1). In the case of simulating under independence between covariates, one can see that any of the studied algorithms performs better than the LASSO.min. These obtain good results searching for the s relevant covariates when $p > n$, and they seem to be able to recover the set S for a large enough value of n (see Figure 3.7). Besides, all of them add less noise to the model and do not overestimate the prediction results too much, as the LASSO.min

does. See Table 3.6 for a brief comparison. Nevertheless, only the AdapL.1se algorithm recovers the complete set S without including any noise in the model for a large enough value of n. This last procedure performs incredibly well in this setting. The performance of the LASSO.BIC and RelaxL are also remarkable, although the first one only outperforms the LASSO.min for values of $n > p$. The Dant achieves good results in terms of avoiding noise too, however, its convergence to the set S seems slower.

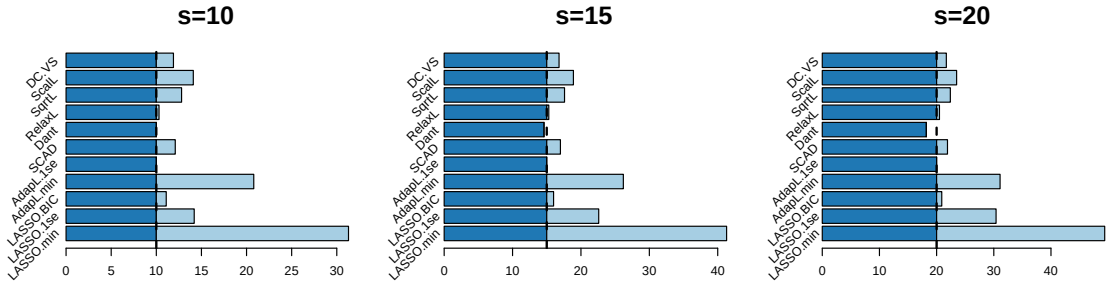


Figure 3.7: Comparison of the important covariates number (dark area) and noisy ones (soft area) for $n = 400$ in Scenario 1. The dashed line marks the s value.

Scenario	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (2.604)	% Dev (0.9)
LASSO.min	15	26.3	41.3	2.091	0.919
LASSO.1se	15	7.6	22.6	2.300	0.911
LASSO.BIC	15	1	16	2.443	0.905
AdapL.min	15	11.2	26.2	2.235	0.913
AdapL.1se	15	0	15	2.491	0.903
SCAD	15	2	17	2.425	0.906
Dant	14.6	0	14.6	2.976	0.885
RelaxL	15	0.3	15.3	2.478	0.904
SqrtL	15	2.6	17.6	2.403	0.907
ScalL	15	3.9	18.9	2.371	0.908
DC.VS	15	1.8	16.8	2.421	0.906

Table 3.6: Comparison of all proposed algorithms for Scenario 1 taking $n = 400$ and $s = 15$. The oracle values are in brackets.

Once we have seen that the proposed alternatives to the LASSO.min improve the results when there is no correlation structure between covariates, it is interesting to test their performance under dependence. The first considered model is the dependence by blocks context (Scenario 2), simulating a correlation structure of value ρ every ten places. In Section 3.1.2, we saw that the LASSO.min does not select a representative subset of S formed by a bunch of efficient covariates as expected. Instead, this procedure always tries to recover the complete set, adding many noisy ones in the process, which translates into overestimation. A comparative example of all algorithms performance in this scenario, for $s = 15$ and $n = 400$, is displayed in Table 3.7 taking $\rho = 0.5$ (Scenario 2 with $\rho = 0.5$) and

in Table 3.7 simulating with $\rho = 0.9$ (Scenario 2 with $\rho = 0.9$). Visual examples are shown in Figure 3.8 and Figure 3.9, respectively.

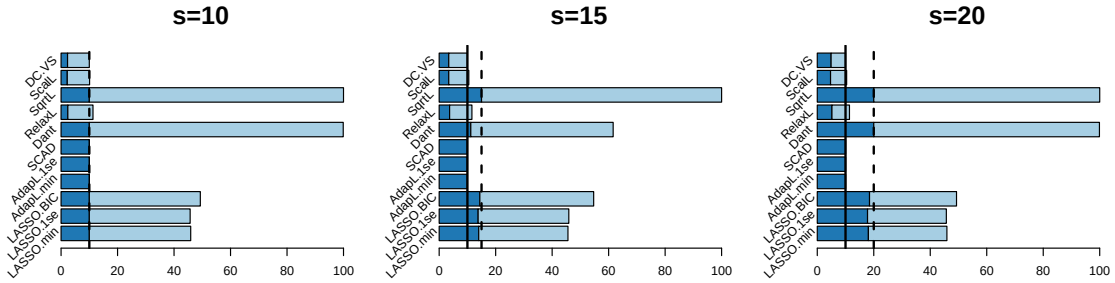


Figure 3.8: Comparison of the important covariates number (dark area) and noisy ones (soft area) for $n = 400$ in Scenario 2 with $\rho = 0.5$. The dashed line marks the considered s value while the continuous line where $s = 10$.

	$\rho = 0.5$					$\rho = 0.9$				
	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
LASSO.min	14	31.6	45.6	1.346	0.949	14	31.7	45.7	2.42	0.912
LASSO.1se	13.8	32.1	45.9	1.346	0.949	13.7	31.6	45.2	2.420	0.912
LASSO.BIC	14.4	40.3	54.7	1.346	0.949	14.4	41.7	56.1	2.420	0.912
AdapL.min	10	0	10	1.346	0.949	10	0	10	2.423	0.912
AdapL.1se	10	0	10	1.346	0.949	9.9	0.1	10	2.423	0.912
SCAD	10.1	0	10.1	1.346	0.949	10.1	0	10.1	2.423	0.912
Dant	11.2	50.4	61.6	4.968	0.811	11.1	50.2	61.3	6.013	0.781
RelaxL	3.7	8	11.7	1.377	0.947	4.5	7.2	11.7	2.438	0.911
SqrtL	15	85	100	1.346	0.949	15	84.9	100	2.42	0.912
ScalL	3.4	7.1	10.4	1.374	0.948	4	6.5	10.4	2.567	0.906
DC.VS	3.4	6.6	10	1.346	0.949	3.8	6.2	10	2.423	0.912

Table 3.7: Comparison of all proposed algorithms for Scenario 2 with $\rho = 0.5$ and with $\rho = 0.9$ taking $n = 400$ and $s = 15$. The oracle values are in brackets.

The LASSO.1se, LASSO.BIC, Dant algorithm, and SqrtL suffer from the same issue. These algorithms are not capable of interpreting the data structure. As a result, these tend to select almost the p covariates in some cases. Here, Dant mimics the performance of the LASSO.min when there is an equal correlation between the relevant covariates and noisy ones. Examples are the $s = 15$ and $s = 20$ frameworks. In these situations, this algorithm recovers 10 out of the s relevant variables, but then, this is unable to distinguish between the rest of the relevant covariates and noise. It is due to the dependence by blocks structure. Relevant covariates already selected by the model have an equal correlation with the rest of the relevant ones, as with noisy covariates placed every ten locations.

In contrast, the rest of the alternatives seem to perform better, trying to select a representative subset of length 10 approximately. However, not all the remaining procedures

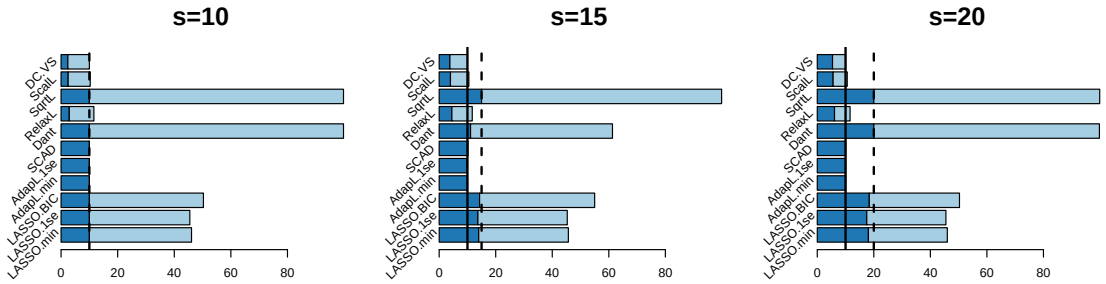


Figure 3.9: Comparison of the important covariates number (dark area) and noisy ones (soft area) for $n = 400$ in Scenario 2 with $\rho = 0.9$. The dashed line marks the considered s value while the continuous line where $s = 10$.

select a representative subset between the s relevant variables. Instead, the majority change relevant covariates for noisy ones strongly correlated with the previous ones, covering the complete set S. In a word, if a procedure chooses a noise covariate, it is expected this last to be a representative of some relevant one not included in the model yet to achieve a good explanation of the data. We can see proof of this phenomenon for RelaxL, ScalL, and DC.VS in Section A.6.1 of the Appendix A. Only the AdapL.min, AdapL.1se, and the SCAD algorithms seem to behave properly in this sense, recovering 10 elements of the set S. All these methodologies correct a bit of the overestimation produced by the LASSO.min algorithm.

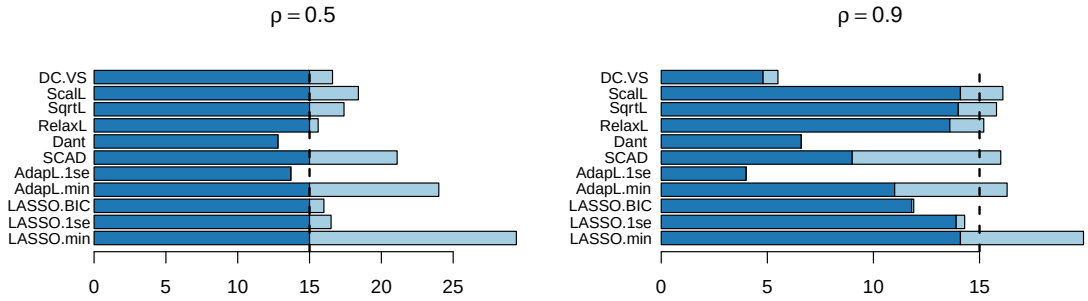


Figure 3.10: Comparison of the important covariates number (dark area) and noisy ones (soft area) for $n = 400$ in Scenario 3.a. The dashed line marks the s value.

Finally, the Toeplitz covariance structure of Scenario 3 is analyzed. It is considered a first scenario, where the relevant covariates are located in the first $s = 15$ placements (Scenario 3.a), and a second one, where there are simulated only $s = 10$ important variables and they are placed every ten sites (Scenario 3.b). Hence, one expects Scenario 3.a to obtain a representative subset of the set S, with cardinal less than s as it was explained in Section 3.1.2. Especially, when the correlation between covariates is strong, as for $\rho = 0.9$.

It is because one has, in this scenario, several relevant covariates with a representative correlation between them. Roughly speaking, because of the Toeplitz covariance structure, one variable could be "easily" explained by others in its neighborhood. This translates into the possibility of interchanging the last variables of S with nearby ones. Then, for $\rho = 0.5$, because $0.5^5 \leq 0.05$, one considers as good representatives those covariates whose distance is less than 4 to some position of the terms in S. When $\rho = 0.9$, this distance enlarges, and there are many more possibilities. In contrast, simulating Scenario 3.b, one would expect the algorithm to select all the 10 relevant covariates in the best case or a representative subset following this criteria.

	$\rho = 0.5$					$\rho = 0.9$				
	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
LASSO.min	15	14.4	29.4	0.972	0.914	14.1	5.8	19.9	3.620	0.908
LASSO.1se	15	1.5	16.5	1.063	0.906	13.9	0.4	14.3	3.762	0.905
LASSO.BIC	15	1	16	1.066	0.906	11.8	0.1	11.9	3.824	0.903
AdapL.min	15	9	24	0.995	0.912	11	5.3	16.3	3.593	0.909
AdapL.1se	13.7	0	13.7	1.195	0.894	4	0	4	4.543	0.885
SCAD	15	6.1	21.1	1.016	0.910	9.1	7	16	3.658	0.907
Dant	12.8	0	12.8	1.443	0.873	6.6	0	6.6	4.92	0.875
RelaxL	15	0.6	15.6	1.078	0.905	13.6	1.6	15.2	3.728	0.906
SqrtL	15	2.4	17.4	1.053	0.907	14.1	1.8	15.9	3.717	0.906
ScalL	15	3.4	18.4	1.039	0.908	14.1	2	16.1	3.701	0.906
DC.VS	15	1.6	16.6	1.061	0.906	4.8	0.8	5.6	4.285	0.891

Table 3.8: Comparison of all proposed algorithms for Scenario 3.a taking $n = 400$ with $\rho = 0.5$ and $\rho = 0.9$. The oracle values are in brackets.

For Scenario 3.a, it is appreciated in Figure 3.10 a similar phenomenon as the one observed in Scenario 2. This translates into the existence of algorithms that try to recover the complete set S, like the LASSO.min, the AdapL.min, the SCAD, the RelaxL, the SqrtL, or the ScalL. One could also include the LASSO.1se, the LASSO.BIC and the DC.VS to this list for the $\rho = 0.5$ case. The rest of the algorithms, the AdapL.1se, and the Dant algorithm, always search for a representative subset without including noise. A summary of their performance is displayed In Table 3.8. Taking $\rho = 0.5$, one appreciates that the AdapL.1se and the Dant are the only procedures that select the number of efficient covariates needed to explain, at least, the 90% of the covariance. A similar behavior could be considered for the DC.VS, but this adds more noise and selects more than $s = 15$ covariates for $\rho = 0.5$. Section A.6.3 of the Appendix A displays the percentage of times the relevant covariates are selected for these algorithms.

Studying the provided results for $\rho = 0.9$ (Table 3.8) one can claim that DC.VS achieves the best results in terms of prediction when the correlation is large, but this pays the price of including more irrelevant information than the AdapL.1se or the Dant. In contrast, when $\rho = 0.5$, the selection of covariates made by the DC.VS results in an overestimation

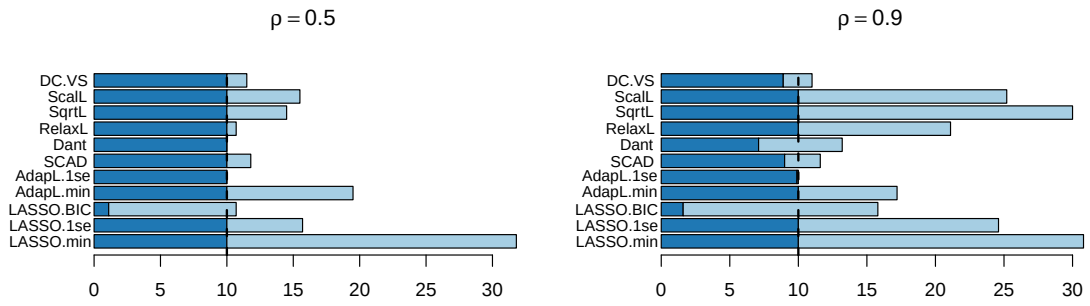


Figure 3.11: Comparison of the important covariates number (dark area) and noisy ones (soft area) for $n = 400$ in Scenario 3.b. The dashed line marks the s value.

of the model. Finally, if the results of AdapL.1se and the Dant algorithms are compared, it seems like the first one obtains a better trade-off between the selection of covariates and estimation. For $\rho = 0.9$, the AdapL.1se selects fewer covariates of S but achieves a better performance in terms of the explanation of the data. These three approaches select fewer than $s = 15$ covariates for $\rho = 0.9$, but a number large enough to guarantee a good explanation of the covariance. See Section A.5 of the Appendix A for more details.

At this point, it is interesting to notice that the Dant performs correctly in this dependence context in contrast to the Scenario 2 framework. Now, this procedure can recover a representative subset of S without adding noise to the model. This phenomenon could be explained by considering that in Scenario 3.a there are not too many noisy covariates highly correlated with the ones of S, especially for $\rho = 0.5$. Only those in the neighborhood of the 15^{th} could be a threat. However, in Scenario 2, there are covariates correlated ten by ten, and every relevant covariate is correlated with 8 irrelevant ones at least. Conversely, the SqrtL keeps misbehaving, and the SCAD algorithm starts to perform poorly. This last result brings out the fact that the SCAD procedure suffers when all the covariates are correlated among them. This happens when the important covariates are close in location, as in the case of Scenario 3.a. However, the algorithm performs better when these covariates are more scattered, like in Scenario 3.b.

Next, we compare results obtained for Scenario 3.b. An example is displayed in Figure 3.11, and the rest of the results are provided in Appendix A. Simulating for $\rho = 0.5$, we observe that all the proposed algorithms outperform the LASSO.min results. At first sight, it may seem that the LASSO.BIC does not perform properly, however, this selects a representative subset of S but changes relevant covariates for correlated ones. The remained procedures try to recover the complete set S as expected, considering that the number of efficient covariates is 10 now. Nevertheless, when $\rho = 0.9$, some drawbacks come up. Some of them interchange relevant covariates with irrelevant ones quite correlated with these. This results from the strong correlation structure of the Toeplitz covariance. These are the LASSO.BIC, the SCAD, and the DC.VS algorithms. Maybe, we can include

in this last group the Dant, although this is doubtful. Section A.6.2 of the Appendix A collects the percentage of times a representative of the 10 relevant covariates enters the model. Other procedures, like the RelaxL, the SqrtL, or the ScalL add unnecessary noise, overestimating the model. Even the LASSO.BIC could be included in this list. Only one algorithm is almost capable of recovering the s variables without adding more noise to the model, this is the AdapL.1se algorithm. All the alternatives correct the overestimation in the prediction made by the LASSO.min though.

Eventually, it is important to highlight that computational time varies from one methodology to another. This is because of the way these are implemented and their nature. Some of them have the cross-validation scheme integrated into the employed R library considered. For example, the famous library `glmnet` (Friedman et al. (2010)) has implemented an optimal cross-validation algorithm in Fortran code, which improves the computational cost of the R language. This results in a quite competitive computational time for the LASSO and AdapL adjustments. The library `ncvreg` (Breheny and Huang (2011)) for the SCAD has utilities for carrying out cross-validation also. In contrast, other methods have not got implemented this scheme as for the `flare` library used for the Dant and the SqrtL. In this case, it is needed to program the cross-validation scheme, resulting in higher computational costs. Besides, the SqrtL pays the price of higher computational time required to be able to select an optimal λ without knowing σ. As a result, SqrtL is the slowest algorithm of the study. Another distinct procedure, the ScalL, is fitted by employing an iterative algorithm rather than a cross-validation process. Then, its computational time depends on convergence criteria. Last, the DC.VS of Febrero-Bande et al. (2019) has a different nature to the previous ones. This applies a special forward selection scheme recalculating the distance matrices between samples on every step to obtain the correlation distance coefficients (Székely et al. (2007), Szekely and Rizzo (2017)), increasing its computational time in terms of n. A merely illustrative comparison of the computational time of our implementations is collected in Section A.4 of Appendix A, as this comparison is not totally fair for the given reasons.

3.1.4 Discussion: some guidance about LASSO under dependence

Currently, the LASSO regression keeps being a broadly employed covariates selection technique. Despite its several advantages, some strict requirements could make difficult a correct performance of this methodology, as was explained in Section 2.2. As we argued at the beginning of this chapter, there are no global recommendations about the LASSO use in terms of the nature of the data under dependence or when some of these conditions do not hold. The LASSO drawbacks have been analyzed (Section 2.2). In addition, modifications (Section 2.3) as well as alternatives (Section 2.4) able to overcome these have been studied to shed light on this topic. Besides, we implement an extensive simulation study throughout this chapter. This study illustrates the behavior of the LASSO in the best possible scenario and trickier ones carefully chosen (Sections 3.1.1 and 3.1.2). Besides, we compare its behavior with recent modifications and alternatives (Section 3.1.3). In view of the results, some guidance on how to choose a proper covariates selector according to the nature of the

data is given. Results are summarized in Table 3.9.

Orthogonal design	**Dependence by blocks**	**Toeplitz covariance**	
		3.a	**3.b**
AdapL.1se Dant RelaxL DC.VS	AdapL.min AdapL.1se SCAD	AdapL.1se Dant DC.VS	AdapL.1se
LASSO.1se, LASSO.BIC, AdapL.min, SCAD, SqrtL, ScalL	*RelaxL, ScalL, DC.VS*	*LASSO.1se, LASSO.BIC*	*SCAD, Dant, DC.VS*

Table 3.9: Most competent procedures in terms of the considered simulation scenarios. Under the dashed line other studied techniques that improve the LASSO.min performance in practice are shown.

One sees that, even in scenarios with no dependence, the LASSO procedure performs poorly regarding recovery of the relevant covariates and avoiding noisy ones. This procedure adds more noise than relevant covariates to the model when λ is selected by cross-validation techniques minimizing a prediction criterion, like the LASSO.min or LASSO.1se (see Section 3.1.2). Nevertheless, this recovers the complete set S, paying the price of noise addition. As a result, this selection of covariates overestimates the prediction errors. This phenomenon is also appreciated for the BIC version (LASSO.BIC) when $p > n$, although this improves the results for $n \geq p$ performing a good covariates selection for a large enough value of n. These drawbacks can be overcome easily using other penalization techniques, keeping the ideas of the L_1 regularization, as the ones proposed in Section 3.1.3. All these procedures improve the LASSO results in this independence context, decreasing the number of selected noisy covariates and correcting the overestimation. We can highlight the adaptive LASSO of Zou (2006) (AdapL.1se), the relaxed LASSO of Meinshausen (2007) (RelaxL), the Dantzig selector of Candes and Tao (2007) (Dant) and the distance correlation algorithm of Febrero-Bande et al. (2019) (DC.VS) as the best of the proposed algorithms for this framework. They can recover the complete set S, adding little noise for a great enough value of n. Besides, they correct the prediction errors.

These disadvantages of the LASSO are also transferred to dependence structures. The confusion phenomenon appears in these situations involving an increment of false discoveries and overestimation. Here, not all the proposed methods of Section 3.1.3 perform properly. The selection of an efficient methodology depends on the nature of the correlation. To test their adequacy, we consider different scenarios: simulating under a dependence by blocks structure and a time series style structure. We found that a version of the adaptive LASSO of Zou (2006) (AdapL.1se) and the distance correlation algorithm of Febrero-Bande et al. (2019) (DC.VS) are the only procedures reasonably competent in all these scenarios concerning different types of dependence.

The quality of some procedures' performance varies accordingly to the type of correlation structure of the data. Examples of this fact are the SCAD penalization of Fan (1997) (SCAD) and the Dantzig selector of Candes and Tao (2007) (Dant). The first one achieves

a good performance except for the case when there exist strong correlations between all the relevant covariates. In contrast, the Dantzig selector performs properly in these scenarios. However, this procedure is not capable of recovering the important covariates and avoiding noise under a dependence structure by blocks.

The rest of the analyzed methods: relaxed LASSO (RelaxL), square-root LASSO (SqrtL), and scaled LASSO (ScalL), present a deficient behavior when there exists some class of dependence structure between the covariates. In the case of the dependence by blocks, as in Scenario 2, the relaxed LASSO and the scaled LASSO mix relevant covariates with unimportant ones even for $\rho = 0.5$, whereas the square-root LASSO does not take advantage of the correlation structure. For the Toeplitz covariance scenario, all of them mimic the LASSO behavior trying to recover the complete set S instead of making use of the structure of the data to adjust the regression model correctly.

As mentioned in Section 3.1.1, in the different considered dependence structures, all covariates are in the same scale. Analysis of the effect of different scales on the covariates, combined with dependence structures, would be of interest. Consequently, the effect of the covariates scales under dependence scenarios is analyzed next.

3.2 Problems of the LASSO regression facing covariates with different scales under dependence

Along Section 3.1, some problems of the LASSO regression under different dependence structures have been displayed. Nevertheless, the covariates were on a unit scale in all the considered frameworks. In practice, it is usual to have dependence scenarios with covariates in quite different ranges of values. Some real data examples verifying these conditions are displayed in Section 2.5 of Chapter 2. As a result, it is interesting to know what one can expect about the LASSO as a variable selector in this context. In particular, it is interesting to determine if the scale effect of the covariates has a role in LASSO selection, translating this, for example, in an increment of confusion phenomenon when there are noisy covariates with higher scales than relevant ones.

To the best of our knowledge, no existing literature studies this topic in the LASSO framework. As a result, we extend the analysis developed in Section 3.1, adding the study of the scale effect of the covariates. For this purpose, we consider two cases: results selecting covariates using the raw data (without standardization case) and employing the classical LASSO approach standardizing these first in a univariate manner (univariate standardization case). This last procedure is the usual way to proceed when applying the LASSO, as mentioned in Section 3.1.2. Thus, for the univariate standardization case, it is observed that the algorithm can draw the important terms better than the raw data when there are covariates with different scales.

Hence, the LASSO performance is tested under controlled simulation scenarios with different dependence structures and covariates scales. Besides, similar to Section 3.1, these results are compared with suitable modifications and alternatives to the LASSO in new simulation scenarios. In particular, the same procedures selected in Section 3.1.3 are

employed. Eventually, some discussion arises about the obtained results.

3.2.1 Simulation scenarios

Here, we consider new versions of simulation scenarios for testing the standardization effect in covariates selection. For this purpose, we simulate under distinct dependence structures considering variables with varied scales. This study starts analyzing the performance of without and univariate standardizations under independence between covariates with different scales (Scenario 1). There are considered three different scenarios in this orthogonal framework: all covariates are standardized (Scenario 1.a), only some relevant covariates are not standardized (Scenario 1.b), and when some nonstandardized noisy ones are added as well to the previous model (Scenario 1.c). Next, we move to dependence structures. In the first place, we assume that all covariates have unit variance and that all of them are related through a Toeplitz covariance matrix (Scenario 2). As the relevant covariates' location plays an important role, we consider two configurations: relevant covariates in the first $s = 15$ locations (Scenario 2.a) and $s = 10$ important covariates spread every three places (Scenario 2.b). Eventually, Scenario 2 is mixed with different scales. In particular, we modify the structure of Scenario 2.b, considering the values of Scenario 1 for the covariates scales. This procedure gives place to two new adaptations of Scenario 2.b: only important covariates with different scales (Scenario 3.a) and the previous scenario changing the variance of some noisy terms related to the important ones (Scenario 3.b). There have been considered a total of $p = 100$ covariates and sample sizes of $n = 25, 50, 100, 150, 300$. All these sample sizes verify the consistent condition of $n > \log(p)s \approx 4.61s$ displayed in (2.8) except for $n = 25$, being S the set of true relevant covariates and $s = \#S$ its cardinal. Furthermore, β is generated guaranteeing that the signal recovery property of $\inf_{j\in S}|\beta_j| > \sqrt{s\log(p)/n}$ displayed in (2.7) is always guaranteed for Scenario 1, but for $n = 25$, and only taking $n = 300$ in Scenarios 2 and 3. The variance of the error term of the model, σ^2, is calculated to verify that the 90% of deviance can be explained at most (see Section B.1 of the Appendix B). For this study, we carry out a total of $M = 500$ Monte Carlo replicates. Similar to Section 3.1.1, the selection capability is tested by counting the number of covariates corrected selected ($|\hat{S} \cap S|$) and the noisy ones picked ($|\hat{S} \setminus S|$) over the total ($\hat{S}$). Moreover, the prediction accuracy is measured by computing the mean square error (MSE) and the percentage of explained deviance as $\%Dev = (RSS - RSS_0)/RSS$.

- **Scenario 1** (*Independence*). Only the first $s = 10$ values are not equal zero for β_j with $j = 1, \ldots, s$, $\beta_1 = \cdots = \beta_s = 1.25$, while $\beta_j = 0$ for all $j = s+1, \ldots, p$. X is simulated as a $N_n(0, \Sigma_p)$, where the covariance matrix Σ has different diagonal structures:
  - **Scenario 1.a**: all covariates in the same scale taking $\Sigma = I_p$.
  - **Scenario 1.b**: only some relevant covariates are not standardized. Covariance matrix is given by the structure $\text{diag}(\Sigma^{1.b}) = (0.5, 0.5, 1, 1, 3, 3, 10, 10, 25, 25;\ 1, \overset{p-s}{\ldots}, 1)$

  - **Scenario 1.c**: keep the structure of Scenario 1.b and add different scales for the next 12 noisy covariates ($j = 11, \ldots, 22$). This translates into the diagonal covariance matrix $\text{diag}(\Sigma^{1.c}) = \Big(\Big(\text{diag}(\Sigma^{1.b})_j \Big)_{j=1}^{s}$, 0.5,0.5,1.5,1.5,3,3,10,10,25,25, 50,50; $1, \overset{p-s-12}{\ldots}, 1\Big)$.

- **Scenario 2** (*Toeplitz covariance with unit scales*). Again, only s ($p > s > 0$) covariates are important. X is simulated as a $N_n(0, \Sigma)$, and $\beta_j = 0.5$ are assumed in the places where $\beta \neq 0$. In this case, $\sigma_{jk} = \rho^{|j-k|}$ for $j, k = 1, \ldots, p$, and $\rho = 0.5, 0.9$. Now, two different dependence structures varying the location of the s relevant covariates are analyzed:

  - **Scenario 2.a**: the relevant covariates are the first $s = 15$.
  - **Scenario 2.b**: consider $s = 10$ relevant variables placed every 3 sites, which means that only the $\beta_3, \beta_6, \beta_9, \ldots, \beta_{30}$ terms of β are not null.

- **Scenario 3** (*Toeplitz covariance with different scales*). Similar structure as Scenario 2.b but adding different covariates scales. It is taken $\Sigma = D\Sigma^{2.b}D^t$ with $\rho = 0.5, 0.9$ and D a diagonal matrix given by $\text{diag}(D) = (\sigma_1, \sigma_2, \ldots, \sigma_p)^t$:

  - **Scenario 3.a**: relevant covariates have variance equal to $\text{diag}(\Sigma^{1.b})$. This means $\sigma_3^2 = 0.5, \sigma_6^2 = 0.5, \sigma_9^2 = 1 \ldots, \sigma_{30}^2 = 25$ and the rest equal one.
  - **Scenario 3.b**: same as Scenario 3.a but adding noisy covariates with different scales. In particular, it is defined $\sigma_2^2 = 0.5, \sigma_5^2 = 0.5, \sigma_8^2 = 1.5, \sigma_{11}^2 = 1.5, \sigma_{14}^2 = 3, \sigma_{17}^2 = 3, \sigma_{20}^2 = 10, \sigma_{23}^2 = 10, \sigma_{26}^2 = 25, \sigma_{29}^2 = 25, \sigma_{32}^2 = 50$ and $\sigma_{35}^2 = 50$.

Again, Scenario 1 is the easiest context concerning dependence structure and possible confusion phenomena. Thus, one would expect an efficient algorithm to be able to detect relevant covariates without adding too much noise, especially in Scenario 1.a. Moreover, for Scenario 1.b and Scenario 1.c, a suitable algorithm is expected not to be influenced by different scales, particularly in Scenario 1.c, where there are noisy covariates with larger scales than important ones. In all these three scenarios, without and univariate standardizations are applied, and their results are compared.

Next, a Topelitz covariance structure is simulated. In Scenario 2 (2.a and 2.b), all covariates are assumed to have a unit scale. Two different configurations for relevant covariates disposition are considered, as in Section 3.1.1, to have distinct confounding effects. More details can be found in Section 3.1.1. Scenario 2.a has been introduced above in Section 3.1.1, and Scenario 2.b is similar to Scenario 3.b of that section. Nevertheless, we compare now if differences exist between the without and the univariate standardization techniques when some class of dependence pattern arises.

Finally, more complexity is added to Scenario 2, resulting in a Toeplitz covariance structure with different scales on covariates (Scenario 3). In this context, we consider Scenario 2.b, but there are changes in the scale values of the relevant covariates (Scenario 3.a), as well as in some irrelevant terms pretty related to the first ones (Scenario 3.b).

Thus, this is a dependence structure with covariates in different scales, mimicking a real data problem. It is expected for an adequate procedure to be able to use the dependence structure, especially for strong correlations ($\rho = 0.9$), and avoid irrelevant covariates even though these are highly correlated with the important ones and have a larger scale.

Note that when assuming different scales for some covariates, we take amounts less than and greater than unity for the scales associated with these terms. Also, we consider important as well as irrelevant covariates in different scales. Furthermore, when some of the scales of unimportant covariates are assumed to be different from the unit, some greater values than those considered for the relevant ones are taken.

3.2.2 Performance of the LASSO in practice considering covariates with different scales

In this section, we analyze the performance of LASSO methods for the different simulation scenarios introduced above in Section 3.2.1. In this way, we test the effect of the covariates scales under different dependence structures. Complete results are collected in Section B.3 in Appendix B.

The LASSO is implemented following the guidelines previously given in Section 3.1.2, using the `glmnet` library of Friedman et al. (2010) and considering the LASSO.min, LASSO.1se and LASSO.BIC variants. See Section 3.1.2 for more details.

Therefore, we begin testing the effect of standardization in the covariate selection procedure when covariates are independent, considering a different configuration of the scale values. Next, we move on to dependence scenarios.

First, we start studying the LASSO performance in Scenario 1.a, simulating under the assumption of $\Sigma = I_p$. A summary of its results is collected in Figure 3.12 and Table 3.10. Complete results display in Tables B.4 and B.5 of Section B.3.1 of the Appendix B. Due to the identity covariance structure, similar performance is expected for the two types of considered standardization techniques (without and univariate). This fact is verified since both standardizations select the same number of variables in all cases, as appreciated in Figure 3.12. Nevertheless, this number exceeds the optimal value $s = 10$ always for all LASSO techniques, as was expected due to the obtained results of the dependence study (Section 3.1). The three considered procedures (LASSO.min, LASSO.1se, and LASSO.BIC) completely recover all relevant covariates for $n \geq 50$, i.e. when consistent conditions are verified. However, all add noisy covariates to the model in the process. In contrast, for $n = 25$, the LASSO variations can not recover S and keep adding noise to the model. LASSO.BIC is the methodology that obtains the best results about avoiding noisy covariates for $n > p$, followed by the LASSO.1se and LASSO.min. Opposite, in scenarios where $p \geq n$, the LASSO.1se and LASSO.min outperform the LASSO.BIC results.

In terms of prediction, the three variants tend to overestimate the results. Table 3.10 displays an example. Here the value of the mean squared error (MSE) and the percentage of explained deviance (% Dev) are smaller and higher than the oracle values, respectively. In particular, the LASSO.min is the approach that overestimates most of the prediction accuracy results when $n > p$. This phenomenon also occurs in the above study of Section

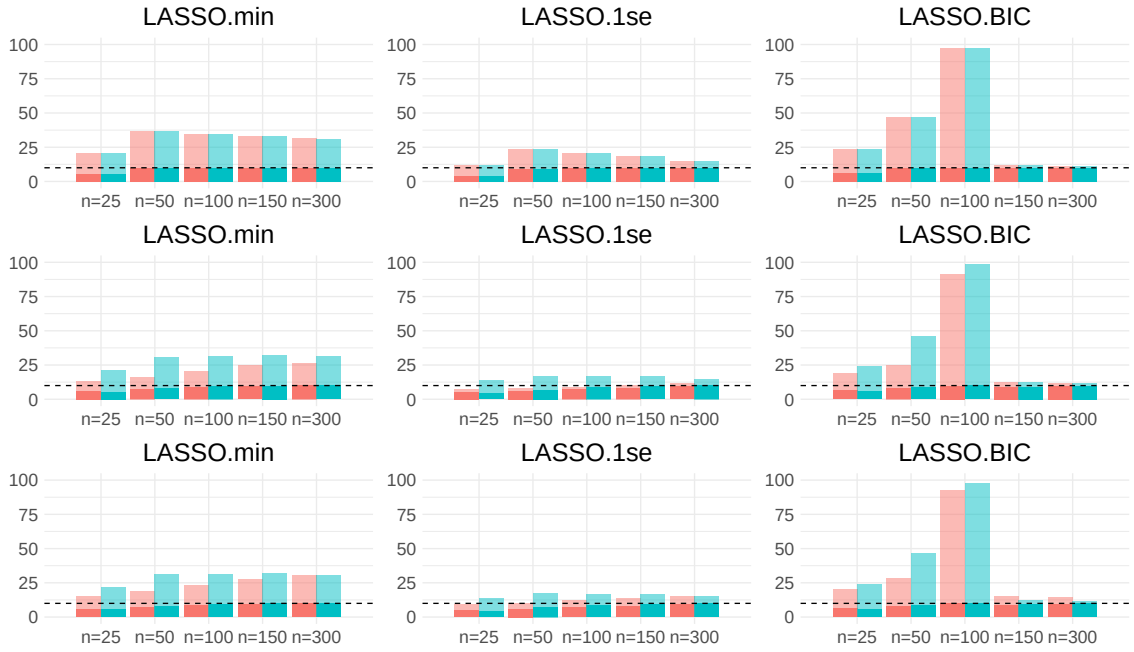


Figure 3.12: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for $p = 100$ selected in terms of the without/univariate standardization in Scenarios 1.a (the first row), 1.b (the second row) and 1.c (the third row). The dashed line marks the $s = 10$ value.

3.1. In contrast, the worst results concerning overestimation in the $p \geq n$ framework are the ones of the LASSO.BIC. The reader can see Tables B.4 and B.5 of Section B.3.1 of the Appendix B for complete results. Given these results, under orthogonal design and when all covariates are in the same scale, it seems similar to work under the without or univariate standardization frameworks.

	Scenario 1.a				Scenario 1.b				Scenario 1.c			
	WITHOUT		**UNIV.**		**WITHOUT**		**UNIV.**		**WITHOUT**		**UNIV.**	
METHOD	**MSE** (1.736)	**% Dev** (0.9)	**MSE** (1.736)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)
LASSO.min	1.345	0.922	1.346	0.922	10.972	0.919	10.638	0.922	10.866	0.920	10.677	0.921
LASSO.1se	1.538	0.910	1.539	0.910	12.972	0.904	12.174	0.910	12.813	0.891	12.180	0.891
LASSO.BIC	1.616	0.906	1.616	0.906	12.671	0.907	12.739	0.906	12.729	0.906	12.722	0.906

Table 3.10: Results of LASSO.min, LASSO.1se and LASSO.BIC for $p = 100$ and $n = 300$ using different standardization techniques in Scenario 1. Oracle values are in brackets.

Secondly, we consider an independence framework where all covariates have a unit scale but for $s - 2$ of the relevant ones (Scenario 1.b). A summary of the results displays in Figure 3.12 and Table 3.10. Tables B.4 and B.5 of Section B.3.1 of the Appendix B collect the complete results. Related to covariates selection in Scenario 1.b, it can be seen in Figure 3.12 as both standardizations recover the same amount of relevant covariates for

each n although the univariate version adds more noise in the process. This difference of noise addition tends to vanish as the sample size increases and $n > p$ is verified. The rate of recovering the S set is similar to that of Scenario 1.a for each algorithm. Besides, the change of scales in the relevant covariates helps the LASSO to select fewer noisy covariates, especially in the without standardization case. This result can be explained considering that the newly considered scales are higher than 1 for 8 out of 10 terms. Compared to the noisy covariates, all in unitary scale, this fact endows the relevant covariates with more importance. This fact reveals that the LASSO seems to be influenced by covariates scale effects. Furthermore, if one studies the percentage of times each of the $j = 1, \ldots, p$ relevant terms are included in the model for the without/univariate standardization taking $n = 300$ (see Figure B.1 in Section B.3.1 of the Appendix B), some small differences are appreciated between LASSO.min, LASSO.1se and LASSO.BIC. One can see as LASSO.min selects a greater percentage of times the covariates with the lowest scales ($j = 1, 2, 3, 4, 5, 6$), followed by the LASSO.BIC and LASSO.1se. In all cases, we appreciate as covariates with scales greater than the noisy terms are always selected, whereas those with smaller values ($j = 1, 2, 3, 4$) are chosen a lower percentage of times. This brings out the fact that LASSO techniques suffer from scale effects.

For prediction, similar to Scenario 1.a, one appreciates that the LASSO.min is the procedure that overestimates the results most when $n > p$. This procedure always obtains values for MSE and %Dev under and above, respectively, of the oracle quantities. This fact happens no matter the type of standardization employed. In contrast, LASSO.1se and LASSO.BIC correct this overestimation a bit, although they also obtain smaller and greater values than oracle ones for MSE and %Dev. However, for the $p \geq n$ case, LASSO.BIC again produces the largest overestimation.

Next, more complexity is added. In Scenario 1.c, the scales of relevant covariates introduced in Scenario 1.b are held, and there are set 12 extra noisy terms with different scales as well. One can check the details in Section 3.2.1. Besides, the diagonal covariance matrix structure is maintained, which results in independence between covariates. Results for Scenario 1.c related to estimation are displayed in Table 3.10, taking $n = 300$, and those collecting the number of selected covariates in Figure 3.12. Full results are displayed in Tables B.4 and B.5 in Section B.3.1 of the Appendix B.

It is appreciated in Figure 3.12 as results for covariates selection are quite similar between Scenario 1.b and Scenario 1.c. However, this last scenario adds a bit more noise using without standardization. Thus, univariate standardization seems to protect against false discoveries when there are noisy covariates with greater scales than those associated with important terms. One also observes this consequence by paying attention to the percentage of times that the first 22 covariates are selected. An example of these percentages taking $n = 300$ is displayed in Figure B.2 in Section B.3.1 of the Appendix B.

In this framework, prediction results are similar to the ones of Scenario 1.b. Then, similar conclusions are derived.

Summing up all the information, different scale effects are appreciated in the LASSO performance. First, it seems that the LASSO is influenced by the presence of covariates

in different scales, even in an orthogonal context. For example, the LASSO.BIC tends to select those with the highest values, no matter if they are relevant or not. Secondly, it depends on the type of effect, whether one or another standardization technique is more suitable. The without standardization approach makes use of the covariates scales, reducing the number of false negatives when only relevant terms have high scales. Nevertheless, the without version adds more noise when relevant and noisy covariates have different scales. In contrast, univariate standardization protects against this phenomenon.

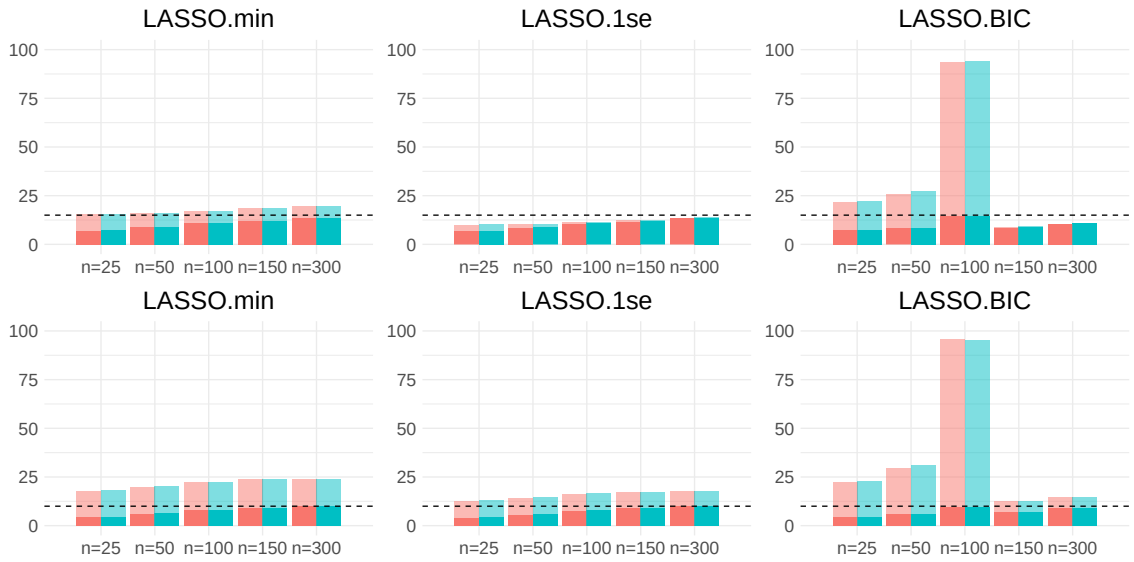


Figure 3.13: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for $p = 100$ and $\rho = 0.9$ selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row). The dashed line marks the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

Next, we test the scale effect in a dependence framework. For this purpose, we employ the Toeplitz covariance dependence framework given in Scenario 2. Firstly, we start considering that all covariates are in the same unitary scale to analyze how these standardization procedures perform under this class dependence pattern. Two different cases of dependence structures are considered: when relevant covariates are the first $s = 15$ (Scenario 2.a) and when there are only $s = 10$ and these are placed every three locations in $j = 3, \ldots, 30$ (Scenario 2.b). LASSO results are summarized in Figure 3.13 and Table 3.11 for scenarios 2.a and 2.b taking $\rho = 0.9$. Complete results are collected in Section B.3.2 of the Appendix B. As expected, we obtain similar results using without or univariate standardization techniques in all cases, indistinctly. Thus, conclusions are similar to the ones previously exposed in Section 3.1.2 for selection capability and prediction accuracy.

Eventually, we move to a challenging framework considering dependence and covariates with different scales. For this purpose, Scenario 3 introduced above in Section 3.2.1 is employed. This configuration is an extension of Scenario 2.b where relevant covariates (Scenario 3.a) and relevant jointly with related unimportant ones (Scenario 3.b) have

METHOD	Scenario 2.a				Scenario 2.b			
	WITHOUT		**UNIV.**		**WITHOUT**		**UNIV.**	
	MSE (3.807)	**% Dev** (0.9)	**MSE** (3.807)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)
LASSO.min	3.525	0.910	3.526	0.910	1.096	0.911	1.098	0.911
LASSO.1se	3.715	0.905	3.715	0.905	1.154	0.906	1.154	0.906
LASSO.BIC	3.822	0.902	3.800	0.903	1.183	0.904	1.180	0.904

Table 3.11: Results of LASSO.min, LASSO.1se and LASSO.BIC for $p = 100$, $n = 300$ and $\rho = 0.9$ using different standardization techniques in Scenario 2. Oracle values are in brackets.

different scales. In particular, these quantities are the scale values considered for Scenarios 1.b and 1.c, respectively. A summary of the obtained results for these scenarios is displayed in Figure 3.14 for covariates selection and in Table 3.12 in terms of prediction. The remaining results are displayed in Tables B.10-B.11 in Section B.3.3 of the Appendix B.

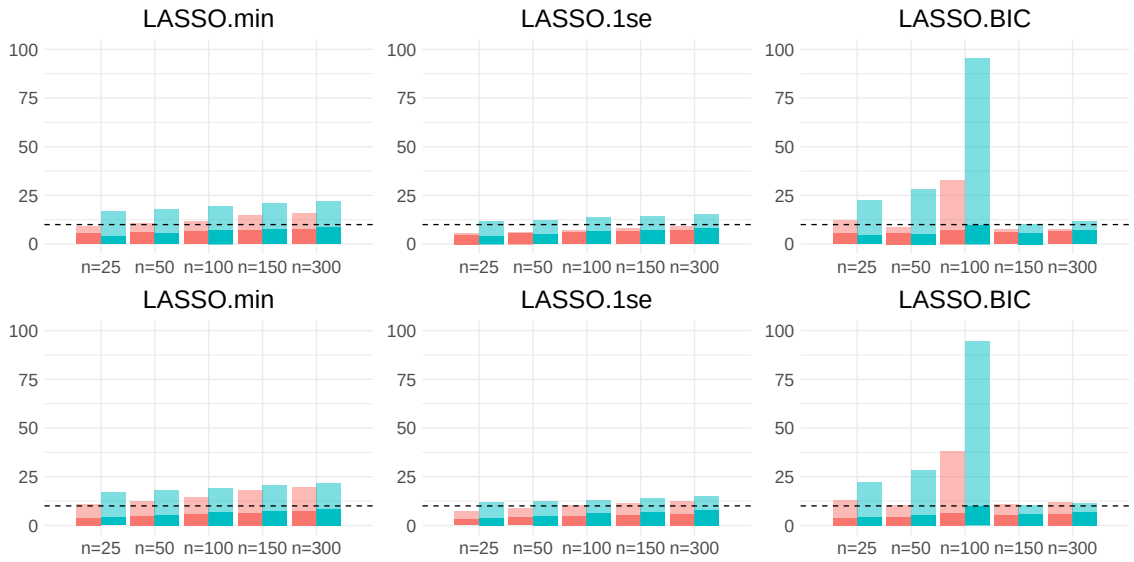


Figure 3.14: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for $p = 100$ and $\rho = 0.9$ selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row). The dashed line marks the $s = 10$ value.

Comparing Scenario 3.a and Scenario 3.b, displayed in Figure 3.14, it is appreciated that the rate of recovery of the s relevant covariates is equal in all cases for $\rho = 0.9$. Nevertheless, we observe more noise addition in the second scenario because of unimportant covariates with high scales, especially for the without standardization approach. Although the univariate standardization seems more "consistent" to scale effects, this adds more noise

than the without approach. Furthermore, both techniques recover a quite similar number of covariates of S. An explanation is that, as mentioned above, the without standardization framework tends to select covariates with the highest scales. As a result, this selects the terms that better explain the variability of the data and needs fewer covariates than the univariate approach, which searches for efficiency. Besides, $s-2$ of the relevant covariates have scales greater than the unit, and hence, these are possible candidates to be selected by this methodology. In contrast, if the large-scale variables were only the unimportant ones, we would expect the without standardization methodology to not recover the S set. Opposite, one expects the univariate standardization to correct this drawback, achieving a better recovery. Similar behavior is observed, respectively, for scenarios 3.a and 3.b taking $\rho = 0.5$, although, in this last case, the algorithms early recover the complete set S. However, these add more noise to the model. See Figure B.15 in Section B.3.3 of the Appendix B. In addition, when the dependence structure is not too strong, taking $\rho = 0.5$, the LASSO variations can completely recover S. In contrast, when the dependence is strong, as for the $\rho = 0.9$ case, these procedures can not detect the s relevant terms. Instead, these algorithms interchange some of the s variables for noisy ones. The noisy terms tend to be representative variables of the missing relevant covariates. Figure B.17 collected in Section B.3.3 of the Appendix B displays an example of their results. It is important to notice that the effects of the covariates are more notable for the without standardization case. This can be seen by paying attention to the percentage of times each of the relevant covariates is selected. See Figure B.16 for Scenario 3.a and Figure B.17 for Scenario 3.b in Section B.3.3 of the Appendix B. The considered procedures always select the important covariates associated with high scales ($j = 21, 24, 27, 30$) with probability one. In contrast, the algorithms select the covariates with the lowest scales a smaller percentage of times, especially when dependence is strong, as in the $\rho = 0.9$ case.

	Scenario 3.a				Scenario 3.b			
	WITHOUT		**UNIV.**		**WITHOUT**		**UNIV.**	
METHOD	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)
LASSO.min	7.100	0.908	6.920	0.910	7.022	0.909	6.926	0.911
LASSO.1se	7.561	0.902	7.302	0.906	7.563	0.902	7.302	0.906
LASSO.BIC	7.643	0.901	7.527	0.903	7.546	0.903	7.526	0.903

Table 3.12: Results of LASSO.min, LASSO.1se and LASSO.BIC for $p = 100$, $n = 300$ and $\rho = 0.9$ using different standardization techniques in Scenario 3. Oracle values are in brackets.

For prediction, the considered procedures keep overestimating the prediction results in all cases and both considered scenarios (Table 3.12). This result is appreciated through values of the MSE and %Dev fewer and higher, respectively, than oracle values. Again, LASSO.BIC is the procedure that overestimates less for $n > p$, followed by LASSO.1se and

LASSO.min. In these scenarios, there are not too many differences between the prediction accuracy of the three procedures, no matter the type of standardization employed. In contrast, for $p \geq n$, things turn around, being the LASSO.BIC the worst option, and LASSO.1se the best one.

Given the results for Scenarios 2 and 3, it is possible to conclude that the without standardization approach could help to reduce the noise without sacrificing relevant covariates. Nevertheless, this consideration must be taken cautiously, as the without framework suffers from scale effects. This means that if the covariates with the highest scales correspond to irrelevant ones, this could result in a confusion phenomenon. This result is because the without standardization tend to select covariates with large scales a percentage of times greater than the univariate case. On its own, the univariate standardization is less sensitive to the scale values. In conclusion, it seems more suitable to apply the univariate scheme when quite different scales are observed in practice.

3.2.3 Comparison with competitors

Next, we compare LASSO results with adaptations and competitors of this procedure. For this purpose, simulation scenarios introduced in Section 3.2.1 are employed. Specifically, a distinction between results for the $p > n$ framework and those for the $n > p$ context is made. We simulate by taking $n = 50$, guaranteeing that $p > n$ for the first case. Opposite, $n = 300$ scenarios are considered for the $n > p$ framework. These values satisfy the required consistency conditions, as was argued in Section 3.2.1.

Context of $p > n$

Following the guidelines of Section 3.1.2, the performance of these algorithms under the independence assumption, considering different scales for covariates, is tested first. Next, scenarios with a different dependence structure are analyzed.

In particular, we consider Scenarios 1.b and 1.c in the first place. There are relevant, jointly with noisy covariates, respectively, with different scales. Results taking $n = 50$ are displayed in Figure 3.15 and Table 3.13. Similar behavior of the studied procedures is appreciated between both scenarios, although some more noise is added for LASSO or ScalL algorithms for Scenario 1.c.

In terms of covariates selection, similar to the conclusions of Section 3.1, it is possible to distinguish between two different types of algorithms based on their selection strategy: the first group tries to recover the set S completely, whereas the second one tends to select a representative subset of covariates making use of the data structure. The AdapL.min, AdapL.1se, Dant, and DC.VS approaches belong to this second group, whereas the rest of the procedures belong to the first class. The LASSO.min, and the LASSO.BIC are the noisiest algorithms that attempt to recover all covariates of S in the model. These select more irrelevant covariates than important ones. Only the AdapL.1se and Dant methodologies avoid including noisy covariates. However, none of the procedures succeeds in recovering S. Concerning the standardization effect, SCAD, Dant, RelaxL, SqrtL, and

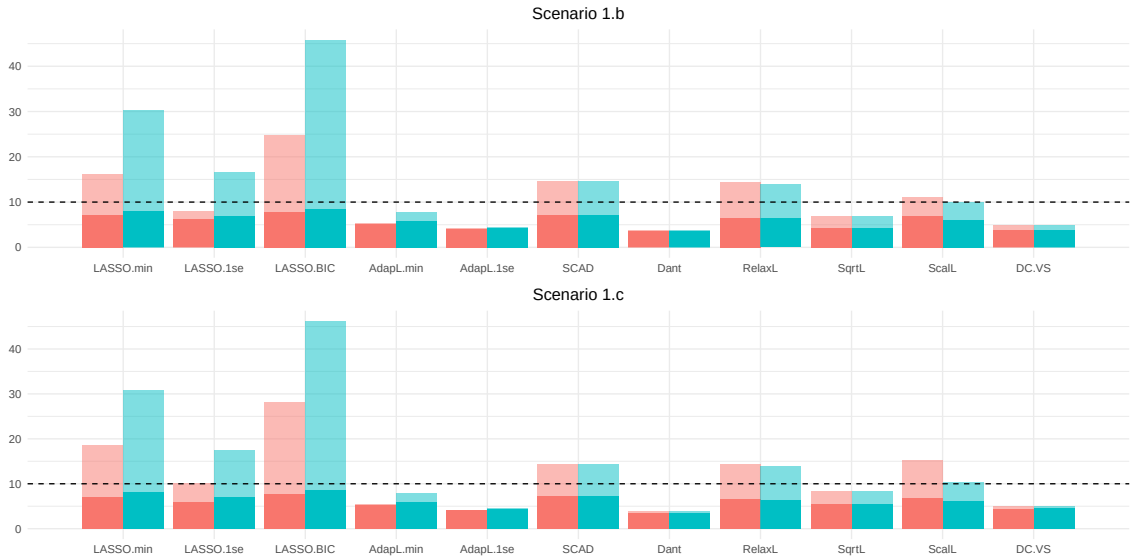


Figure 3.15: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 1.b (the first row) and 1.c (the second row) for $n = 50$. The dashed line marks the $s = 10$ value.

DC.VS do not seem to be affected by the type of standardization employed, obtaining the same or quite similar results for both frameworks. This result holds when we move on to dependence scenarios as well. In contrast, for the remaining procedures, using univariate standardization seems to help recover a few more elements of S but pays the price of adding more noise. Only for the ScalL happens the opposite. This procedure recovers better the relevant covariates but adds more irrelevant terms using the raw data.

Moreover, as mentioned above, AdapL.1se, Dant, and DC.VS are the procedures that select the less amount of covariates but the ones that avoid more noise. In particular, these choose the relevant covariates with the highest scales a higher percentage of times than the ones with a value for the standard deviation less or equal to one. This fact is illustrated in Figure B.3, collected in Section B.3.1 of the Appendix B. This phenomenon also happens for the $n > p$ case (see Figure B.4 in Section B.3.1 of the Appendix B). This consequence contrasts with observed results for the orthogonal scenario considering covariates with a unitary scale (see results for Scenario 1 in Section 3.1.3). AdapL.1se, Dant, and DC.VS perform well when all covariates are in the same scale, recovering the s relevant terms and avoiding noise. However, things change when there are different scales between covariates. These three procedures, AdapL.1se, Dant, and DC.VS, select fewer than $s = 10$ terms now. Paying attention to the eigenvalues of covariance matrices for Scenarios 1.b and 1.c (see Table B.1 in Section B.2 of the Appendix B), it is proved that this amount of elements is not enough to explain the variability of the data correctly.

Besides, for Scenarios 1.b and 1.c, one can appreciate that all considered algorithms overestimate the results obtaining less MSE and greater %Dev values than the oracle ones,

METHOD	Scenario 1.b				Scenario 1.c			
	WITHOUT		**UNIV.**		**WITHOUT**		**UNIV.**	
	MSE (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)
LASSO.min	6.257	0.953	2.476	0.982	6.147	0.953	2.394	0.982
LASSO.1se	12.624	0.905	7.092	0.947	12.366	0.905	6.806	0.948
LASSO.BIC	2.468	0.983	0.024	1	1.860	0.987	0.022	1
AdapL.min	17.46	0.87	11.604	0.914	17.275	0.869	11.709	0.912
AdapL.1se	23.520	0.823	21.975	0.834	24.107	0.816	22.031	0.832
SCAD	5.892	0.956	5.892	0.956	6.181	0.952	6.181	0.952
Dant	30.202	0.775	30.202	0.775	30.584	0.769	30.584	0.769
RelaxL	9.298	0.929	9.789	0.925	9.250	0.929	9.849	0.924
SqrtL	9.789	0.925	9.789	0.925	13.856	0.891	13.856	0.891
ScalL	8.186	0.938	11.054	0.914	6.950	0.947	11.038	0.914
DC.VS	73.652	0.901	73.652	0.901	21.029	0.838	22.618	0.818

Table 3.13: Comparison of all proposed algorithms for $p = 100$ and $n = 50$ using different standardization techniques in Scenarios 1.b and 1.c. Oracle values are in brackets.

except for AdapL.1se, Dant, and DC.VS. Results do not change too much for prediction in terms of the employed standardization technique either.

Next, results for competitors in Scenario 2 are analyzed. In this case, a Toeplitz dependence structure is assumed for a different disposition of the relevant terms: the first 15^{th} locations (Scenario 2.a) or every 3 places, from 3 to 30 (Scenario 2.b). Results taking $\rho = 0.9$ are summarized in Figure 3.16 and Table 3.14. Those for the $\rho = 0.5$ case are collected in Figure B.10 and Table B.8 in Section B.3.2 of the Appendix B.

Again, as already noted in Section 3.1 for the Toeplitz scenario, some procedures attempt to recover the whole set S, whereas others use the dependence structure and select a representative subset for the $p > n$ framework. This last is especially remarkable in the $\rho = 0.9$ case of Figure 3.16. There, one can appreciate that the algorithms select fewer covariates in comparison with the $\rho = 0.5$ scenario (Figure B.10 in Section B.3.2 of the Appendix B). Some examples of this second group are the AdapL.1se, Dant, and DC.VS algorithms, which seem to employ the data dependence structure selecting a subset of S. In Table B.2 of Section B.2 in the Appendix B, it can be seen that with a bunch of covariates smaller than s, it is possible to explain a large amount of variability. Besides, as expected because of data structure and results of Section 3.1.3, more noise is added for Scenario 2.b than for Scenario 2.a. Furthermore, results are pretty similar between both types of standardization techniques for each procedure.

Seeing the percentage of times each of the first 20 covariates enters the model for AdapL.1se, Dant, and DC.VS taking $n = 50$ in Scenario 2.a, one appreciates as the

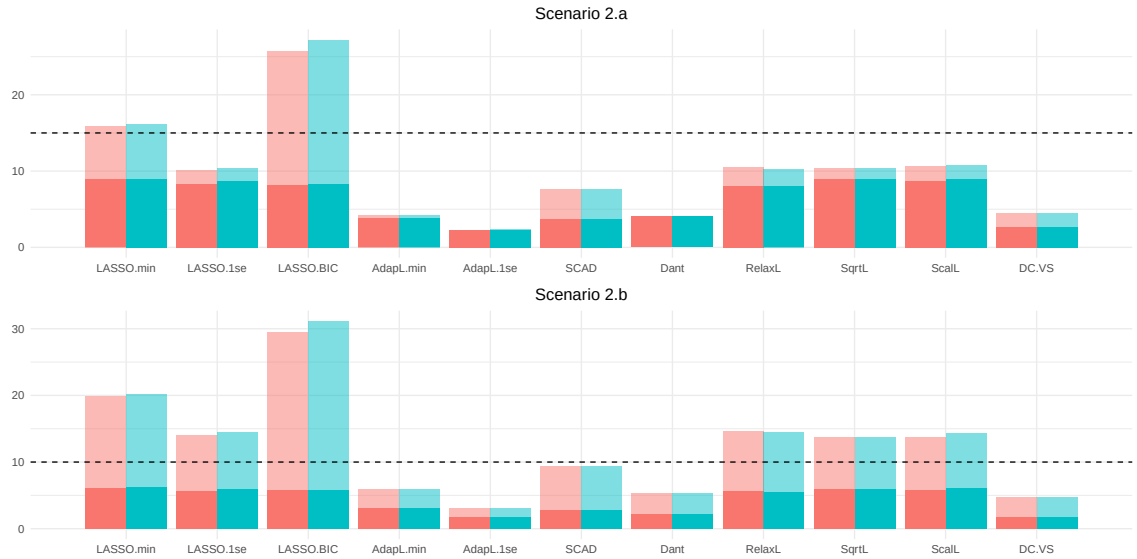


Figure 3.16: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row) for $\rho = 0.9$ and $n = 50$. The dashed lines mark the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

AdapL.1se and Dant selector protect against false discoveries, especially for the $\rho = 0.9$ framework. In contrast, the DC.VS selects some covariates in the neighborhood of the 15^{th} term because of their correlation structure. See Figure B.8 in Section B.3.2 of the Appendix B. In the case of Scenario 2.b with $n = 50$ (Figure B.10 in Section B.3.2 of the Appendix B), one can appreciate as the three approaches perform well for the $\rho = 0.5$ case, selecting important variables the highest percentage of times (Figure B.9 in Section B.3.2 of the Appendix B). However, these approaches are not able to fully recover S. In particular, a greater value of n may be required for its proper recovery. This hypothesis is proved next for the $n > p$ case (see Figure B.12 in Section B.3.2 of the Appendix B). Nevertheless, things change moving on to a more complicated context, Scenario 2.b taking $\rho = 0.9$. One sees in Figure B.9 in Section B.3.2 of the Appendix B as the confusion phenomenon appears. Noisy covariates, related to relevant ones, are selected a greater percentage of times for these procedures. Specifically, it is quite difficult to distinguish the true signal based on the covariates selection percentages in the Dant and DC.VS case.

Concerning prediction, we note that all algorithms overestimate the results, but for the AdapL.1se, Dant, and DC.VS. These results coincide with those that search for a representative subset of S. Besides, no relevant distinctions are appreciated between without or univariate standardization versions having all covariates on a unit scale. A summary of these results is displayed in Table 3.14 for $\rho = 0.9$, and those for $\rho = 0.5$ are collected in Table B.8 in Section B.3.2 of the Appendix B.

Eventually, some complexity is added to Scenario 2.b, simulating relevant covariates

	Scenario 2.a				Scenario 2.b			
	WITHOUT		**UNIVARIATE**		**WITHOUT**		**UNIVARIATE**	
METHOD	**MSE** (3.807)	**% Dev** (0.9)	**MSE** (3.807)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)
LASSO.min	1.914	0.948	1.898	0.948	0.514	0.955	0.513	0.955
LASSO.1se	2.643	0.928	0.728	0.937	0.728	0.937	0.713	0.938
LASSO.BIC	0.956	0.976	0.847	0.979	0.224	0.982	0.188	0.985
AdapL.min	3.183	0.915	3.171	0.915	0.897	0.923	0.895	0.923
AdapL.1se	4.818	0.870	4.713	0.872	1.580	0.864	1.554	0.866
SCAD	2.757	0.926	2.757	0.926	1.554	0.866	1.554	0.866
Dant	4.82	0.87	4.82	0.87	1.554	0.866	1.554	0.866
RelaxL	2.671	0.928	2.714	0.926	0.729	0.937	0.734	0.936
SqrtL	2.623	0.929	2.623	0.929	0.734	0.936	0.749	0.935
ScalL	2.562	0.930	2.534	0.931	0.743	0.935	0.726	0.937
DC.VS	3.958	0.894	3.958	0.894	1.402	0.880	1.402	0.880

Table 3.14: Comparison of all proposed algorithms for $p = 100$, $n = 50$ and $\rho = 0.9$ using different standardization techniques in Scenario 2. Oracle values are in brackets.

with different scales (Scenario 3.a) and adding noisy ones with different scales to this last case (Scenario 3.b) too. Results for covariates selection in the $p > n$ case taking $n = 50$ are summarized in Figure 3.17 for $\rho = 0.9$. Results for the $\rho = 0.5$ framework are displayed in Figure B.18 of Section B.3.3 in the Appendix B.

Taking $\rho = 0.9$, we see in Figure 3.17 as results are pretty similar to the ones of Scenario 2.b for both, Scenario 3.a and 3.b. Here, some differences arise between the type of employed standardization for the considered LASSO versions. The univariate option selects a similar number of important terms as the without version but adds more noise during the process. Besides, a slight increment of noise is appreciated for some algorithms in Scenario 3.b, with the addition of noisy covariates in different scales than the unit. Similar behavior for each algorithm is observed for the $\rho = 0.5$ case adding a little less noise. This can be seen in Figure B.18 of Section B.3.3 in the Appendix B.

For $\rho = 0.5$, the AdapL.1se, Dant, and DC.VS procedures select the covariates with the greatest scales a higher percentage of times for both scenarios (see Figure B.19 in Section B.3.3 of the Appendix B). This selection also happens for the $\rho = 0.9$ case. See Figure B.20 in Section B.3.3 of the Appendix B. Particularly, relevant covariates with the largest scales (j=15,18,21,24,27,30) are selected a higher number of times than the remaining ones. This behavior is especially remarkable for AdapL.1se considering the without standardization approach. Dant and DC.VS also select these last covariates, jointly with some noisy ones related to these, a significant percentage of times. While the AdapL.1se seems to be affected by the type of standardization employed, this phenomenon does not occur with the Dant and DC.VS algorithms. Similar results are appreciated for $n = 300$ in Figures B.22 and

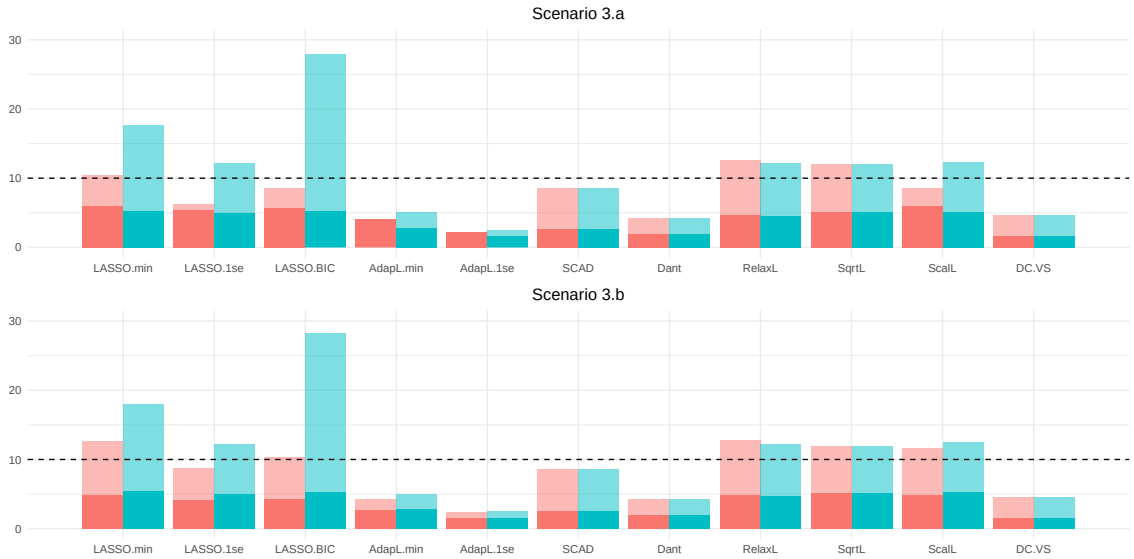


Figure 3.17: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row) for $\rho = 0.9$ and $n = 50$. The dashed line marks the $s = 10$ value.

B.23, collected in Section B.3.3 of the Appendix B.

For prediction, it can be seen in Table 3.15 as the same pattern is repeated. Again, only AdapL.1se, Dant, and DC.VS can correct the overestimation, obtaining MSE and %Dev values greater and fewer than the oracle ones, respectively.

Context of $n > p$

Again, similar to the $p > n$ study, the orthogonal framework of Scenario 1 is analyzed first and then dependent ones using Scenarios 2 and 3. In this case, we take a sample size $n = 300$, which verifies that $n > p$.

Results for Scenarios 1.b and 1.c are collected in Figure 3.18 and in Table 3.16 taking $n = 300$. Similar behavior to the $p > n$ case for the recovery of S applies for all studied algorithms. Now, in this $n > p$ scenario, all methodologies, even the DC.VS algorithm searches for a complete recovery of S, except for AdapL.1se and Dant. In contrast, these select fewer than $s = 10$ terms. In particular, these two procedures recover the relevant covariates with the largest scales. An example of this fact is displayed in Figure B.4, collected in Section B.3.1 of the Appendix B. However, the amounts of selected variables by these two procedures are not enough to correctly explain the variability of the data (see Table B.1 in Section B.2 of the Appendix B). Despite this, both procedures are the only ones able to guarantee the absence of added noise in the selection process. In this framework, most algorithms detect all the relevant covariates and reduce the noisy ones selection. Nevertheless, the LASSO.min, AdapL.min, and SCAD get the worst results, selecting a higher percentage of unimportant covariates. One can note that the LASSO.BIC

	Scenario 3.a				Scenario 3.b			
	WITHOUT		**UNIVARIATE**		**WITHOUT**		**UNIVARIATE**	
METHOD	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)
LASSO.min	5.060	0.932	3.577	0.952	4.828	0.935	3.570	0.952
LASSO.1se	6.927	0.907	4.871	0.934	6.500	0.913	4.856	0.935
LASSO.BIC	5.258	0.929	1.596	0.980	5.265	0.930	1.505	0.981
AdapL.min	8.358	0.888	6.077	0.919	7.683	0.898	6.157	0.918
AdapL.1se	12.461	0.83	10.416	0.860	11.898	0.841	10.349	0.862
SCAD	5.348	0.928	5.348	0.928	10.349	0.862	10.349	0.862
Dant	10.899	0.853	10.899	0.853	10.905	0.855	10.905	0.855
RelaxL	4.925	0.933	5.051	0.932	4.899	0.934	5.083	0.932
SqrtL	5.051	0.932	5.051	0.932	4.971	0.933	4.971	0.933
ScalL	5.562	0.925	4.843	0.934	5.114	0.931	4.838	0.935
DC.VS	8.724	0.883	8.724	0.883	8.649	0.885	8.649	0.885

Table 3.15: Comparison of all proposed algorithms for $p = 100$, $n = 50$ and $\rho = 0.9$ using different standardization techniques in Scenario 3. Oracle values are in brackets.

improves its performance a lot when $n > p$, as expected and as already mentioned in Section 3.2.2. In fact, the LASSO.BIC is the approach with the best selection results in the $n > p$ framework.

Once again, similar to the $p > n$ framework, all algorithms continue to overestimate the prediction results. Only the AdapL.1se and Dant correct a bit this overestimation. This fact could be motivated due to the selection of fewer than $s = 10$ terms in total. Results for prediction are displayed in Table 3.16.

Next, the number of important and noisy covariates selected in Scenario 2 is displayed in Figure 3.19 for $n = 300$ and taking $\rho = 0.9$. Results for $\rho = 0.5$ are collected in Table B.9 and Figure 3.19 of Section B.3.2 in the Appendix B. In all cases, performance between without or univariate standardization frameworks is quite similar because all covariates are in unit scales. As a result, we do not notice a significant distinction. Then, Scenario 2.a, where the relevant covariates are placed together, is analyzed first. Scenario 2.b, where important terms are scattered every three places, follows this.

Concerning covariates selection for Scenario 2.a with $n > p$, one observes a different performance of the algorithms for $\rho = 0.5$ (Figure B.10 in Section B.3.2 of the Appendix B) and $\rho = 0.9$ (Figure 3.19). In the $\rho = 0.5$ case, all algorithms try to recover S, except for AdapL.1se and Dant, whereas for $\rho = 0.9$, more procedures as LASSO.1se, LASSO.BIC and DC.VS join to select a representative subset of relevant covariates. The DC.VS algorithm adds some noise to the model, as can be appreciated in Figure B.11 in Section B.3.2 of the Appendix B. For $\rho = 0.5$, the best results focused on the total recovery of S, adding little or no noise, are obtained for RelaxL, LASSO.BIC and LASSO.1se. On the other hand,

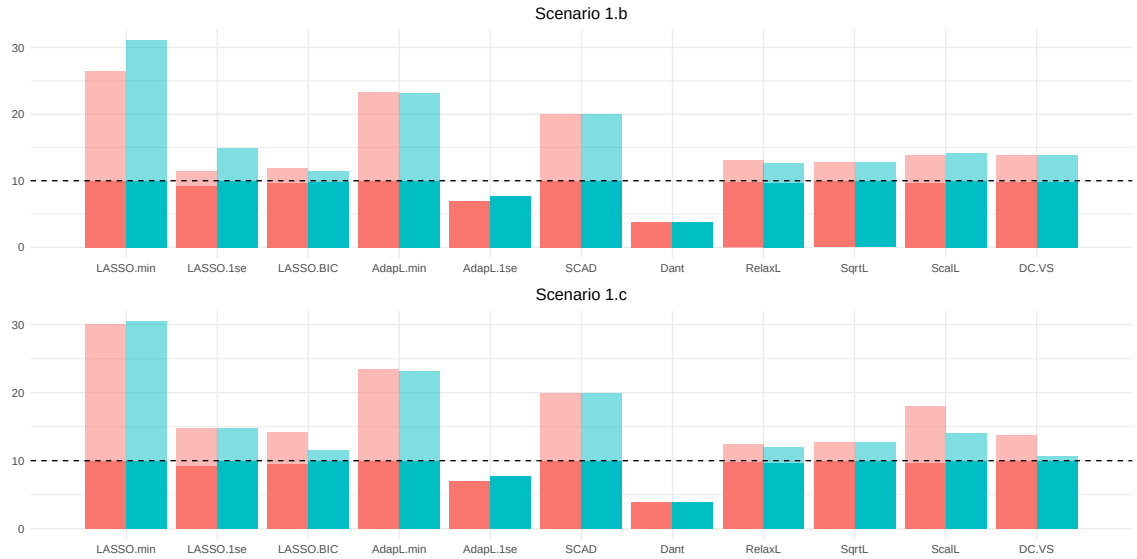


Figure 3.18: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 1.b (the first row) and 1.c (the second row) for $n = 300$. The dashed line marks the $s = 10$ value.

AdapL.1se and Dant select fewer than $s = 10$ covariates. These two procedures use the dependence structure, guaranteeing that all covariates entering the model are relevant. For $\rho = 0.9$, no procedure is able to achieve a complete recovery of S. Instead, some important covariates exchange for irrelevant ones pretty correlated. Procedures that recover a great amount of the $s = 10$ important terms without adding too much noise are LASSO.1se, RelaxL, SqrtL, and ScalL. The AdapL.min and the SCAD approaches tend to select a total of $s = 10$ covariates, but so many are noise. The rest of the algorithms, as LASSO.BIC, AdapL.1se, Dant, or DC.VS add fewer covariates to the model, but they guarantee that all are relevant terms. As a result, there are always procedures that search for the full recovery of S, exchanging some relevant variables for noisy ones in some cases, and approaches that make use of the dependence structure and select a small number of terms guaranteeing the relevance of all of them as a trade-off.

Next, we consider Scenario 2.b. For $\rho = 0.5$ (Figure B.10 in Section B.3.2 of the Appendix B), all algorithms try to completely recover the whole bunch of relevant covariates for both employed standardizations. However, the LASSO versions (LASSO.min, LASSO.1se, and LASSO.BIC) can not recover S but rather select spurious covariates. The rest of the approaches recover the $s = 10$ terms successfully, but procedures as the AdapL.min, SqrtL, or ScalL add too much noise compared to the remaining algorithms. Conversely, one can appreciate different behaviors regarding $\rho = 0.9$ (Figure 3.19). Again, there are two groups: the first group trying to fully recover S (LASSO.min, LASSO.1se, LASSO.BIC, AdapL.min, SCAD, RelaxL, SqrtL, and ScalL), and the second one selecting a representative bunch of them (AdapL.1se, Dant and DC.VS). It is interesting to note

METHOD	Scenario 1.b				Scenario 1.c			
	WITHOUT		**UNIV.**		**WITHOUT**		**UNIV.**	
	MSE (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)	**MSE** (13.715)	**% Dev** (0.9)
LASSO.min	10.972	0.919	10.638	0.922	10.866	0.920	10.677	0.921
LASSO.1se	12.972	0.904	12.174	0.910	12.813	0.891	12.180	0.891
LASSO.BIC	12.671	0.907	12.739	0.906	12.729	0.906	12.722	0.906
AdapL.min	11.115	0.918	11.138	0.918	11.117	0.918	11.136	0.918
AdapL.1se	15.921	0.883	15.024	0.889	15.901	0.883	14.977	0.890
SCAD	11.441	0.916	11.441	0.916	11.453	0.915	11.453	0.915
Dant	29.407	0.784	29.407	0.784	29.549	0.782	29.549	0.782
RelaxL	12.630	0.907	12.754	0.906	12.687	0.906	12.813	0.905
SqrtL	12.521	0.908	12.521	0.908	12.508	0.908	12.508	0.908
ScalL	12.324	0.909	12.270	0.910	12.096	0.911	12.286	0.909
DC.VS	12.353	0.909	12.353	0.909	12.371	0.909	13.595	0.894

Table 3.16: Comparison of all proposed algorithms for $p = 100$ and $n = 300$ using different standardization techniques in Scenarios 1.b and 1.c. Oracle values are in brackets.

here that for Scenario 2.b, procedures such as RelaxL, SqrtL, and ScalL can select the covariates of S but add a vast amount of noise to the model, especially for the $\rho = 0.9$ case. This fact contrasts with their performance in Scenario 2.a. Moreover, the AdapL.1se seems to be the most reliable option for the second group because this pretty much only selects relevant covariates regardless of the dependence strength. Moreover, in Table B.2, collected in Section B.2 of the Appendix B, one appreciates as with a small number of covariates is possible to explain a large enough percentage of variability correctly. As a result, this fact justifies the proper performance of the algorithms which select a representative subset of S for Scenarios 2.a and 2.b.

A summary of prediction results in terms of MSE and %Dev are collected in Table 3.17. We can see in Tables 3.17 and B.9 concerning Scenario 2.a that most studied algorithms overestimate prediction results, but for AdapL.1se and Dant. It is also possible to add the DC.VS procedure to this list, although only in the $\rho = 0.9$ case. In contrast, in Scenario 2.b, only the Dant selector corrects the overestimation for all contexts jointly with the AdapL.1se and the DC.VS in the $\rho = 0.9$ framework.

Eventually, we analyze results for scenarios where not only a dependence structure plays a relevant role but also the scales of covariates. For this purpose, we extend Scenario 2.b: considering that relevant covariates can have different scales from irrelevant ones (Scenario 3.a) and the case where both important and unimportant variables have different scales (Scenario 3.b). These scenarios are introduced above in Section 3.2.1 in detail. Again, we consider different magnitudes of dependence, taking $\rho = 0.5$ and $\rho = 0.9$. Results taking

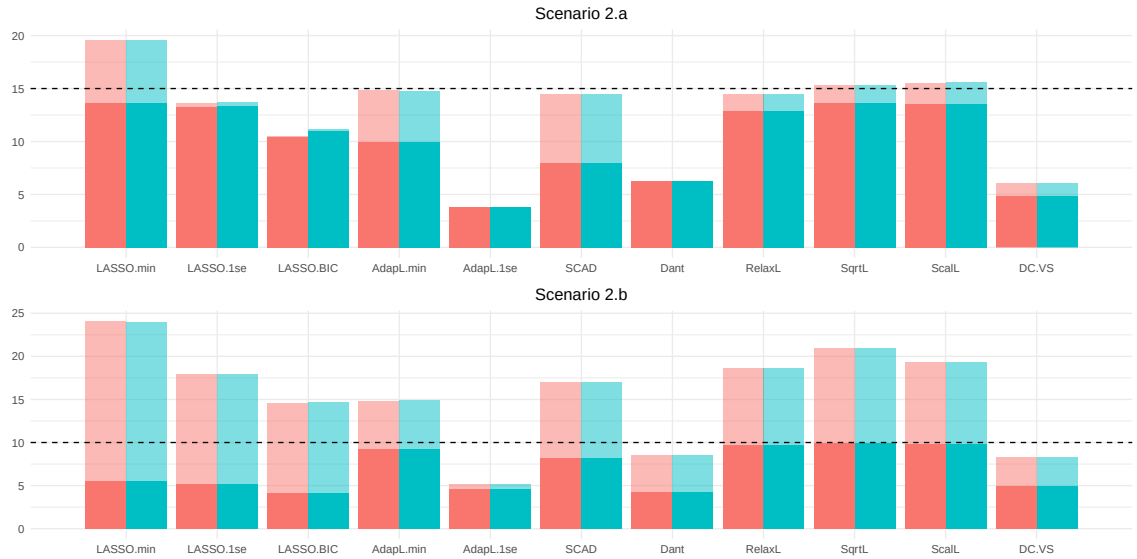


Figure 3.19: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row) for $\rho = 0.9$ and $n = 300$. The dashed lines mark the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

$\rho = 0.5$ are collected in Figure B.21 and Table B.13 in Section B.3.3 of the Appendix B and results for $\rho = 0.9$ are displayed in Figure 3.20 and Table 3.18.

For covariates selection in Scenario 3.a we appreciate that the univariate standardization selects more noisy covariates in the case of LASSO.min, LASSO.1se, LASSO.BIC for $\rho = 0.9$, AdapL.min, and ScalL. Only the RelaxL algorithm in the univariate case selects fewer covariates than the without standardization approach. In contrast, both standardizations play a similar role for the rest of the techniques (AdapL.1se, SCAD, Dant, SqrtL, and DC.VS) . Again, as it happened for Scenario 2.b, some procedures try to completely recover S (LASSO.min, AdapL.min, SCAD, RelaxL, SqrtL, ScalL), and others make use of the dependence structure selecting fewer covariates (AdapL.1se, Dant and DC.VS), especially for the $\rho = 0.9$ value. Besides, we must add a new category for procedures that change their objective based on the employed standardization when the correlation is high ($\rho = 0.9$). These are the LASSO.1se and the LASSO.BIC. Both try to recover S for the univariate standardization case, whereas both select a representative subset for the without standardization context. For $\rho = 0.5$ (Figure B.21 in Section B.3.3 of the Appendix B) the best procedures that guarantee a proper recovery of S with small noise addition are the LASSO.1se, LASSO.BIC and DC.VS. In contrast, AdapL.1se and Dant are the best options to guarantee that all selected covariates are important, and that overestimation is corrected. Conversely, for $\rho = 0.9$ (Figure 3.20), any of the proposed approaches is capable of selecting all variables of S. However, spurious correlations appear and some noisy variables are selected instead of important ones. We can highlight the without LASSO.1se, without

	Scenario 2.a				Scenario 2.b			
	WITHOUT		**UNIVARIATE**		**WITHOUT**		**UNIVARIATE**	
METHOD	**MSE** (3.807)	**% Dev** (0.9)	**MSE** (3.807)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)	**MSE** (1.244)	**% Dev** (0.9)
LASSO.min	3.525	0.910	3.526	0.910	1.096	0.911	1.098	0.911
LASSO.1se	3.715	0.905	3.715	0.905	1.154	0.906	1.154	0.906
LASSO.BIC	3.822	0.902	3.800	0.903	1.183	0.904	1.180	0.904
AdapL.min	3.513	0.910	3.518	0.910	1.117	0.909	1.116	0.909
AdapL.1se	4.540	0.884	4.533	0.884	1.501	0.878	1.509	0.877
SCAD	3.602	0.908	3.602	0.908	1.116	0.909	1.116	0.909
Dant	4.886	0.875	4.886	0.875	1.700	0.862	1.700	0.862
RelaxL	3.683	0.906	3.685	0.906	1.143	0.907	1.144	0.907
SqrtL	3.654	0.906	3.654	0.906	1.139	0.908	1.139	0.908
ScalL	3.636	0.907	3.634	0.907	1.137	0.908	1.137	0.908
DC.VS	4.194	0.892	4.194	0.892	1.347	0.891	1.347	0.891

Table 3.17: Comparison of all proposed algorithms for $p = 100$, $n = 300$ and $\rho = 0.9$ using different standardization techniques in Scenario 2. Oracle values are in brackets.

LASSO.BIC, AdapL.1se, Dant, and DC.VS like options that select a bunch of S terms without adding too much noise during the process.

Finally, we pay attention to covariates selection in Scenario 3.b taking $\rho = 0.5$ (Figure B.21 in Section B.3.3 of the Appendix B) and for $\rho = 0.9$ (Figure 3.20). In this case, complexity is added to the selection process, including irrelevant terms with scales greater than important variables. For the $\rho = 0.5$ case, there are not quite relevant differences for selection between Scenarios 3.a and 3.b. Specifically, the noise increases for the without LASSO procedures (LASSO.min, LASSO.1se, and LASSO.BIC) as well as for the without ScalL. See Figure B.21 in Section B.3.3 of the Appendix B. The performance of the rest of the procedures is completely similar. As a result, one can say that the inclusion of irrelevant covariates with scales even greater than the ones of the important terms does not affect the selection procedure when the dependence relation is not so great, like taking $\rho = 0.5$. Nevertheless, things change when there is a strong dependence relation. An example of this can be seen in Figure 3.20 for $\rho = 0.9$. In this, we can see as it is the first time that the LASSO techniques (LASSO.min, LASSO.1se, and LASSO.BIC) do not search for the complete recovery of S. Furthermore, the without ScalL selects more noise than Scenario 3.a. Only AdapL.min, SCAD, RelaxL, SqrtL, and ScalL seem to follow this purpose, adding a lot of noisy terms. In contrast, the remaining procedures use the dependence structure and select a bunch of covariates. Only the AdapL.1se procedure includes just relevant covariates. Concerning the employed standardization, we can see that results are quite similar for both approaches.

Lastly, in terms of prediction accuracy (Tables B.13 and 3.18), there have been appreci-

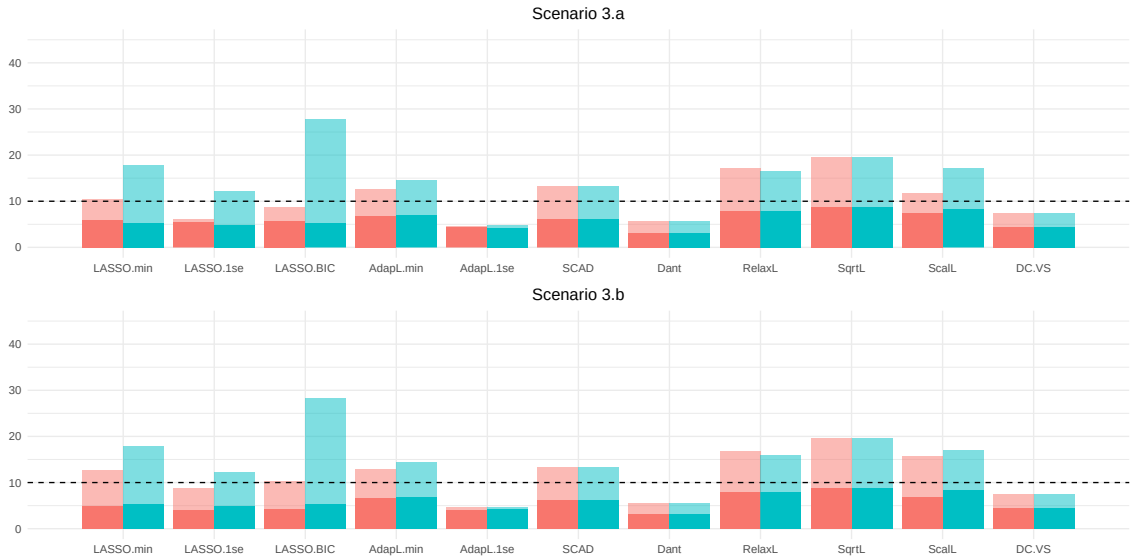


Figure 3.20: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row) for $\rho = 0.9$ and $n = 300$. The dashed line marks the $s = 10$ value.

ated similar results as the ones for Scenario 2.b. All procedures overestimate their results, obtaining MSE and %Dev values fewer and larger than the oracle ones, respectively, except for AdapL.1se and Dant algorithms, jointly with DC.VS for $\rho = 0.9$.

3.2.4 Discussion: scale effects on LASSO under dependence

Along Section 3.1, some limitations of the LASSO and derivatives under different dependence frameworks have been displayed. This analysis concludes with a discussion in Section 3.1.4 about the best possible option based on the studied dependence scenarios. Nevertheless, in all of them, covariates are assumed to have unit scales, resulting in covariates in equal scale. This framework contrasts with real problems where dependence patterns, and covariates in different scales, are expected. Motivated by this fact, throughout Section 3.2, an analysis of the LASSO and alternatives is carried out by modifying the covariates scales under dependence and allowing relevant, as well as unimportant ones, to have different magnitudes. We introduce these scenarios in Section 3.2.1. This study brings a gap between the dependence study of Section 3.1 and scale effects for LASSO and derivatives. Next, we give some guidelines for the best procedure selection in the scenarios considered based on the obtained results.

First of all, we observe that the type of employed standardization technique has an effect only when there are covariates in different scales. Conversely, similar results hold when some dependence structure exists, but the assumption of the equality of scales in the covariates is guaranteed, as in Scenario 2. Furthermore, the main differences are appreciated for the considered LASSO versions (LASSO.min, LASSO.1se, and LASSO.BIC), especially

METHOD	Scenario 3.a				Scenario 3.b			
	WITHOUT		**UNIVARIATE**		**WITHOUT**		**UNIVARIATE**	
	MSE (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)	**MSE** (7.818)	**% Dev** (0.9)
LASSO.min	7.100	0.908	6.920	0.910	7.022	0.909	6.926	0.911
LASSO.1se	7.561	0.902	7.302	0.906	7.563	0.902	7.302	0.906
LASSO.BIC	7.643	0.901	7.527	0.903	7.546	0.903	7.526	0.903
AdapL.min	7.060	0.909	6.920	0.91	7.052	0.909	6.919	0.911
AdapL.1se	8.924	0.885	8.790	0.886	8.936	0.885	8.773	0.887
SCAD	7.149	0.907	7.149	0.907	7.146	0.908	7.146	0.908
Dant	11.804	0.847	11.804	0.847	11.906	0.846	11.906	0.846
RelaxL	7.182	0.907	7.221	0.907	7.197	0.907	7.245	0.906
SqrtL	7.192	0.907	7.192	0.907	7.186	0.907	7.186	0.907
ScalL	7.337	0.905	7.184	0.907	7.259	0.906	7.181	0.907
DC.VS	8.059	0.896	8.059	0.896	8.050	0.896	8.050	0.896

Table 3.18: Comparison of all proposed algorithms for $p = 100$, $n = 300$ and $\rho = 0.9$ using different standardization techniques in Scenario 3. Oracle values are in brackets.

for the $p > n$ framework. In Scenarios 1.b, 1.c, 3.a, and 3.b both standardizations select a similar number of relevant terms, but the univariate version adds more noise in the selection process. The remaining procedures perform quite similarly for both types of standardizations. Thus, the choice of whether or not to apply a previous standardization step does not seem very relevant to the selection results.

In the case of orthogonal scenarios with different scales of the covariates (Scenario 1), there is no clear winner between compared procedures. A proper selection would depend on the research aim. If one wants to guarantee the recovery of as much relevant covariates as possible, no matter the additional noise, LASSO.min, SCAD and RelaxL for the $p > n$ case or LASSO.BIC, RelaxL, SqrtL, ScalL, and DC.VS for the $n > p$ case seem to be the most suitable procedures. However, if it is more valuable to ensure that all selected terms are important, procedures like AdapL.1se or Dant are the best options.

Conversely, different results are obtained when all covariates are related among them and on the same scale. An example of this situation is Scenario 2, simulating under a Toeplitz covariance structure with unitary terms. Now, it depends on the relevant covariates' location, the dependence strength ($\rho = 0.5$ or $\rho = 0.9$), and if $n > p$, or not, performing an adequate recovery.

In the case of $p > n$, a perfect recovery of S without noise addition is not possible, no matter the correlation strength ($\rho = 0.5$ or $\rho = 0.9$) or the relevant covariates disposition (Scenarios 2.a and 2.b). AdapL.1se, Dant, and DC.VS are the most remarkable procedures in the $p > n$ context. These algorithms include more relevant terms than noisy ones. Nevertheless, these also add noise for Scenario 2.b. In the case of a low or moderate

correlation, $\rho = 0.5$, approaches as the LASSO versions, RelaxL, SqrtL, and ScaL almost include all the s important terms but add too much noise as a trade-off.

In the case of $n > p$, we distinguish between results for Scenario 2.a and those for Scenario 2.b. For the case of Scenario 2.a, the best option would depend again on what is the main aim. If one wants to verify that S is recovered although some noise is included, procedures such as RelaxL, SqrtL, and ScalL are more suitable. Nevertheless, if the main interest is to guarantee that all selected covariates are relevant and to obtain prediction results without overestimation, AdapL.1se and Dant procedures are a better option. Concerning Scenario 2.b, one has to resort to procedures that may add noise to the model (AdapL.min, SCAD, RelaxL, SqrtL, or ScalL) to perform the complete recovery of S. Instead, if one is interested in guaranteeing that all selected covariates are relevant, the AdapL.1se seems the best possible option. In this last case, one has also the benefit that overestimation for prediction is corrected.

Eventually, we move on to the case of adding covariates with different scales to Scenario 2. This results in the Toeplitz covariance structure considering covariates in different scales (Scenarios 3.a and 3.b). It is not possible the recovery of S in the $p > n$ context, not even allowing noise to enter the model. Instead, some procedures like AdapL.1se, Dant, or DC.VS try to use the covariance structure and scale effects to select only a portion of the relevant terms with the associated highest scales. When the correlation strength increases to $\rho = 0.9$, the Dant and DC.VS interchange these relevant terms with some noisy ones strongly correlated. In contrast, in the $n > p$ context, procedures such as RelaxL, SqrtL, ScalL, or even DC.VS for $\rho = 0.5$ are capable of recovering S, adding not too much noise in the process. In contrast, AdapL.1se and Dant selector keep choosing fewer terms but guaranteeing that the selected ones are relevant with high probability. Again, only these last procedures correct a bit the overestimation of the prediction.

Summing up all the information, to chose a proper selector algorithm will depend on our objective. There are two possibilities. The first possibility focuses on minimizing the number of false discoveries, guaranteeing that all selected terms are relevant, although we can not assure the complete recovery of S. The second option is to maximize the number of true positive discoveries, recovering the highest possible number of terms of S. In this last case, noise addition has to be allowed as a trade-off for a proper recovery. For the first objective, AdapL.1se and Dant have displayed the best qualities for all dependence structures and different covariates scales. These procedures add the least amount of noise to the model and, as a result, guarantee more consistency in the recovery. A drawback of these approaches is that they do not tend, in general, to a complete recovery of S. Instead, they usually select fewer than s terms. Besides, this quantity decreases for the $p > n$ selection case. Nevertheless, if one is interested in guaranteeing recovery of a large number of true positives, procedures such as the RelaxL, SqrtL, or ScalL are more suitable. However, there are some cases where these do not verify that the complete set S is retrieved, as in Scenarios 2.a and 2.b for $p > n$ or in Scenarios 3.a and 3.b for $p > n$ or for $n > p$ with $\rho = 0.9$. This behavior happens because of dependence and scale effects. These last scenarios are the trickiest, and none of the considered algorithms can get the s influential

terms. The strong dependence structure and the confusing phenomenon of different scales on covariates could explain this situation. Finally, the DC.VS algorithm is in the middle of both philosophies. For the $p > n$ case, its behavior resembles the first group, whereas, for $n > p$, this is more similar to the second one.

3.3 A first screening step based on some coefficient of relevance

As displayed throughout Sections 3.1 and 3.2, the LASSO algorithms suffer from different drawbacks in practice. Apart from those treated in Section 2.2, it has been seen that the LASSO tends to select a lot of noisy covariates even under orthonormal design. Furthermore, this has extra difficulties when there exist dependence structures and/or covariates with different scales. As a result, a preliminary step would be of interest to reduce the dimension of the covariates, discarding some irrelevant ones to facilitate the selection procedure a posteriori. For this purpose, it is very common in the literature to resort to screening procedures. Hence, in this section, we study if a first screening step would be useful under dependence frameworks with covariates in different scales. Thus, we try to sort covariates' relevance to establish a proper cutoff or threshold to define a first screening step. For this aim, the performance of the coefficient of determination (R^2), the distance correlation coefficient (DC), and partial least squares (PLS) values are tested as measures of relevance. First, their performance under independence with different scales (Scenario 1. c) is analyzed, and then, more difficult frameworks, considering the Toeplitz dependence structure (Scenario 2. b) and its version with different scales on relevant as well as unimportant covariates (Scenario 3. b), are studied.

In all cases, it has been only considered the without standardization framework. This decision is because univariate standardization equals relevance between all covariates. As a result, the univariate standardization does not allow us to detect what covariates are the relevant ones. For each of the $M = 500$ Monte Carlo replicates, we calculate the associated coefficients of relevance (R^2, DC, and PLS) for all terms. Then, their sample distributions are compared, using boxplots. The averaged value over the $M = 500$ replicates is computed for each covariate to determine if recovering important covariates based on their relevance coefficients is possible. One expects relevant covariates to have the highest values of relevance and then, to be able to apply a proper threshold.

Making use of the independence framework of Scenario 1.c introduced in Section 3.2.1, where relevant ($s = 1, \ldots, 10$) as well as some unimportant covariates ($s = 11, \ldots, 22$) have different scales, we implement a first screening step. For this purpose, we employ the coefficient of determination (R^2), the distance covariance coefficient (DC), and partial least squares (PLS) values in the without standardization case. The coefficient associated with each covariate is calculated in all the $M = 500$ Monte Carlo replicates, and their resulting boxplots are displayed in Figure B.5 in Section B.3.1 of the Appendix B. A good performance of the coefficients translates into relevant covariates with higher values than irrelevant ones. One can appreciate that this is verified for R^2 and DC coefficients in this framework. Nevertheless, PLS tends to select covariates with the highest scales, regardless

of whether these are relevant or not. Remarkably, we only use the without framework because PLS performs very poorly in the univariate case, equaling all boxplots and being unable to discriminate between important covariates and noisy ones. In contrast, similar results are obtained for R^2 and DC in the univariate standardized framework as these quantities do not depend on the covariates scales.

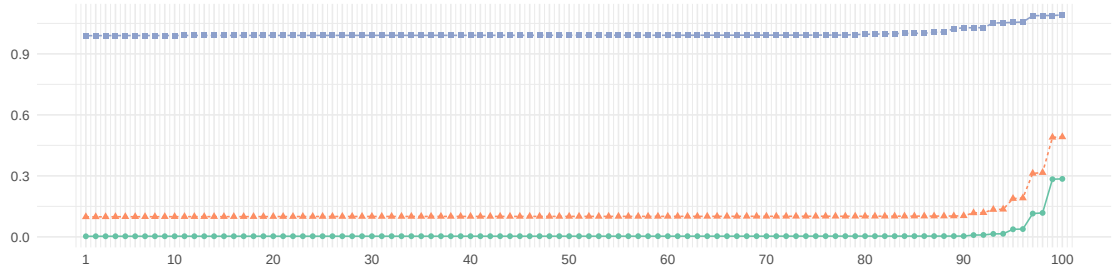


Figure 3.21: Ordered mean values of covariates in terms of R^2 (●), DC (▲) and PLS (■) coefficients in Scenario 1.c without standardization.

We calculate the averaged values of the Monte Carlo simulations for each covariate, and these values are considered in increasing order to achieve a proper threshold for recovering relevant covariates. Results are shown in Figure 3.21. Given the results, an appropriate threshold can be defined for the R^2 and DC values in case of independence, although there are covariates with different scales. Nevertheless, one needs to pay the price of noise addition to verify that the s relevant terms are added to the model for PLS components. In particular, noisy covariates with the highest scales enter the model. This fact illustrates the poor behavior of the PLS values, even for the independence case. Thus, this procedure is avoided for the study henceforth.

Next, the R^2 and DC values discriminant performances are analyzed under dependence. For this purpose, we consider Scenario 2.b. In this case, there are $s = 10$ relevant covariates placed every ten locations ($j = 3, 6, \ldots, 30$). We expect a suitable screening procedure to have correlation values for these variables greater than the rest. Boxplots of resulting quantities are displayed in Figure B.13 for $\rho = 0.5$ and in Figure B.14 for $\rho = 0.9$ collected in Section B.3.2 of the Appendix B. First, differences in the dependence structures based on the ρ value are appreciated. When the dependence structure is not too strong, as in the $\rho = 0.5$ case, we see as both procedures can discriminate the relevant covariates. However, unimportant ones pretty related to relevant terms obtain large correlation values, as expected, although these are smaller than the ones associated with covariates in S. These allow one to establish a proper cutoff to recover relevant covariates without noise addition completely. This phenomenon is observed again if we pay attention to the ordered mean values displayed in Figure 3.22. The $s = 10$ relevant covariates get higher mean values than the remaining ones.

In contrast, taking $\rho = 0.9$, things dramatically change. In view of the boxplots (Figure B.14 in Section B.3.2 of the Appendix B), one observes that, due to strong dependence,

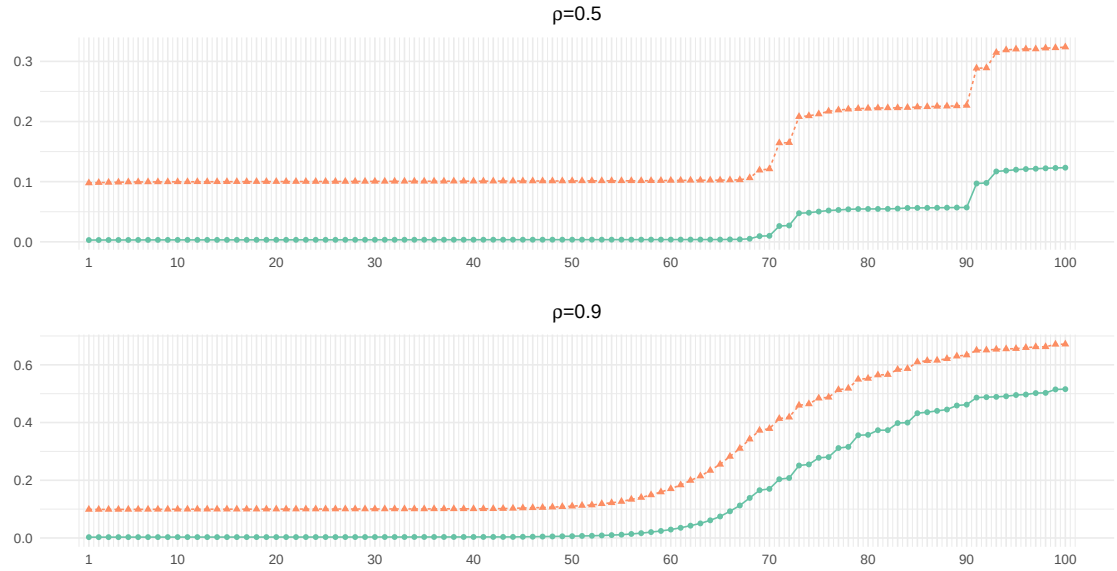


Figure 3.22: Ordered mean values of covariates in terms of R^2 (●) and DC (▲) coefficients in Scenario 2.b.

all values of the first 30 variables are quite high. This fact is also asserted if their mean values are calculated and ordered (Figure 3.22). As a result, we can conclude that a proper threshold is not allowed without noisy covariates addition in the model. Thus, a second step employing an additional covariates selection procedure can be desirable for this class of scenarios. At this point, it is possible to differentiate two approaches to "correctly" select relevant covariates: i) trying to recover S with the least noise inclusion or ii) searching for a bunch of covariates between the first thirty able to represent the significant information. See Section 3.2.3 for examples and discussion.

Eventually, we add more complexity to the previous model of Scenario 2.b. Now, different values for covariates scales are considered. For this aim, we employ Scenario 3.b. The variance of relevant as well as unimportant covariates is modified. These last irrelevant terms are selected as ones related to the $s = 10$ variables of S. Again, there are considered "medium" dependence taking $\rho = 0.5$ and high dependence with $\rho = 0.9$ scenarios.

Results for boxplots are shown in Figures B.24 and B.25 in Section B.3.3 of the Appendix B for $\rho = 0.5$ and $\rho = 0.9$, respectively. Figure 3.23 displays their averaged and ordered versions. We appreciate for $\rho = 0.5$ that it is now impossible to completely recover S without noise addition, which contrasts with the results for scenario 2.b with $\rho = 0.5$. It can be seen in Figure B.24 as the covariates with the highest values correspond to relevant ones with the greatest scales. These are followed by irrelevant ones quite correlated with these. In fact, relevant covariates with the lowest scales have relevance values similar to irrelevant ones. Then, for correct recovery of S, one needs to allow irrelevant covariates to enter the model even for the $\rho = 0.5$ case. This requirement is easily appreciated by paying attention to the ordered values displayed in Figure 3.23. Besides, despite this fact,

it does not always seem possible to distinguish important covariates with the lowest scales from noisy ones. It seems that the DC values are capable of remarking a bit better than the R^2 coefficients of the S terms in this last situation.

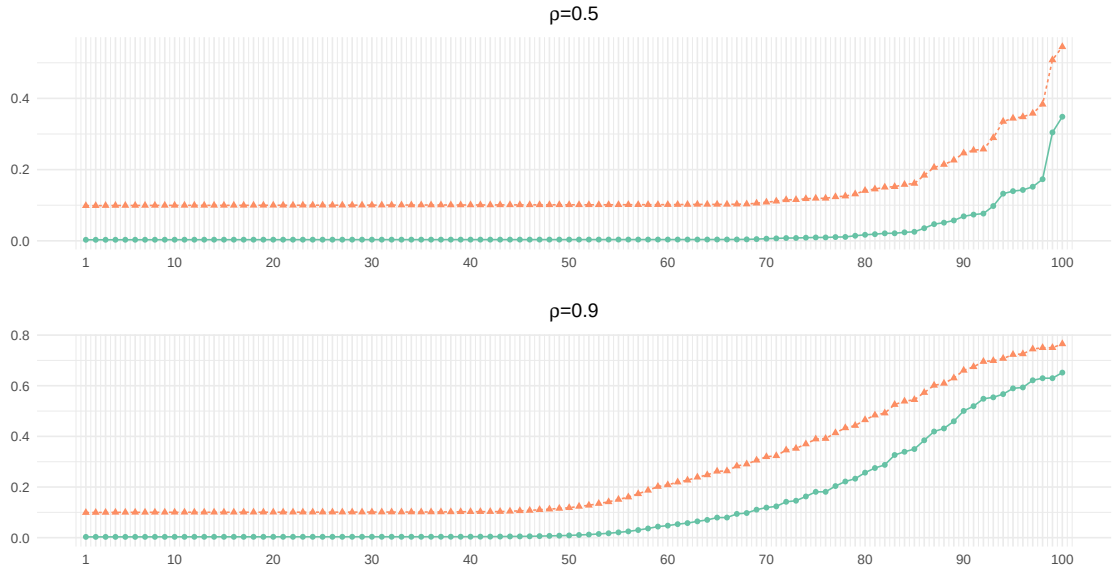


Figure 3.23: Ordered mean values of covariates in terms of R^2 (●) and DC (▲) coefficients in Scenario 3.b.

Finally, the dependence relation is strengthened in Scenario 3.b, taking $\rho = 0.9$. This scenario is a really tough framework because of dependence and different scales between covariates. If the strong dependence case is just quite tricky, as was mentioned above for Scenario 2.b with $\rho = 0.9$, its complexity increases by adding different scales. This fact can be seen in Figure B.25 in Section B.3.3 of the Appendix B. Here, these procedures need to include about 40 variables in the model to guarantee the full recovery of the $s = 10$ relevant term. Similar to Scenario 2.b with $\rho = 0.9$, the increase in dependence translates into difficulties distinguishing between values, finding that relevance coefficients with the highest values are those in the middle of relevant covariates locations (over the 15^{th} term). We also appreciate this phenomenon by seeing the ordered values displayed in Figure 3.23. Hence, one should employ a similar strategy to scenario 2.b. Firstly, apply a screening procedure to reduce noise. Next, perform some covariates selection techniques to detect relevant covariates. Again, one has to decide which is the most appropriate algorithm based on their goals for this second step.

In summary, we have seen that R^2, as well as DC coefficients, are good options for a first screening step which helps to clean the data. Nevertheless, screening techniques based on these ideas should be followed for a second step, using some covariate selection algorithm. Particularly, this procedure is necessary when some class of dependence between covariates exists. Otherwise, so much noise would be added to the model trying to recover S. As a result, screening procedures also suffer from dependence and scale effects no matter

the employed threshold. Hence, these have similar limitations to penalization techniques under dependence and/or covariates in different scales. Besides, it is interesting to notice that the R^2 coefficient only applies under linear structures. If we guess other types of relations, other approaches like the DC coefficient are more suitable. Coefficients that apply for more global dependence structures are treated in Chapter 4.

3.4 Critical analysis of results in some real data sets

Finally, the covariates selection capability of the considered algorithms in Section 3.1.3 is tested in real data examples. For this aim, LASSO procedures and competitors are implemented over the four real data sets introduced before in Section 2.5. These data sets are examples of real problems where different dependence structures and scale effects arise. We refer the reader to Section 2.5 for more details. Thus, the performance of these algorithms is compared, analyzing which covariates are selected in each case. Furthermore, the results of the riboflavin and prostate cancer data sets are compared with those previously obtained in the existing literature.

3.4.1 Riboflavin

The riboflavin data set contains a total of $p = 4088$ expression levels of genes. These are believed to be related to the rate of production of riboflavin (vitamin B2) by the bacterium Bacillus subtilis. A total of $n = 71$ samples have been collected to determine which of the total genes are related to riboflavin production. As a result, this is a high dimensional example where $p > n$. It has been displayed in Section 2.5.1 that all covariates have a similar range of scale values, and there are different types of dependence patterns between them. Next, considered LASSO versions and competitors are applied to select covariates.

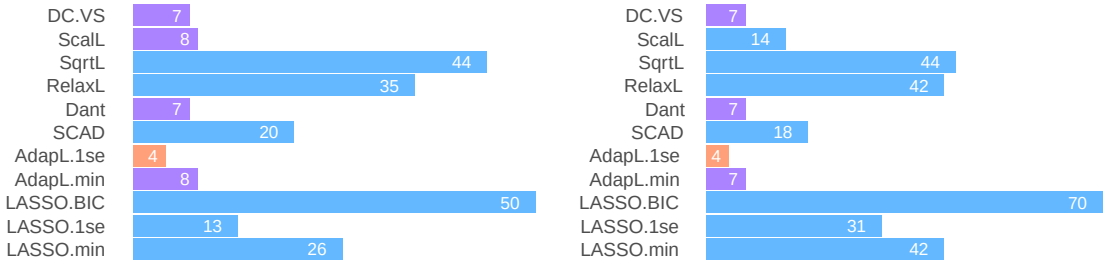


Figure 3.24: Number of selected covariates for the considered procedures for raw data (left) and univariate standardized data (right) of the riboflavin data set.

This data has been centered to avoid the intercept in the model without loss of generality. Although it seems reasonable to assume that all covariates are on a similar scale, small differences have been appreciated in Section 2.5.1. As a result, two different frameworks are considered: working with the raw data and with its univariate standardized version.

The number of selected covariates for each of the eleven considered algorithms is displayed in Figure 3.24. There are quite big differences between the number of selected

LASSO.min	ABH_at, ACOA_at, AMYC_at, ARGF_at, LACA_at, LYTA_at, NDK_at, PCKA_at, PURC_at, RPLL_at, XLYA_at, YBGB_at, YCDH_at, YCGN_at, YCGO_at,YCKE_at, YHDS_r_at, YHFH_r_at, YHZA_at, YRBA_at, YRZI_r_at, YTGB_at, YVAY_at, YWMC_at, YXLE_at, YYDF_i_at
LASSO.1se	ARGF_at, GAPB_at, XHLA_at, XLYA_at, YCDH_at, YCGN_at, YCKE_at, YHFH_r_at, YHZA_at, YRZI_r_at, YTGD_at, YXLD_at, YXLE_at
LASSO.BIC	AADK_at, ABH_at, ACOB_at, ADAB_at, ALD_at, AMYC_at, ARGF_at, ARGH_at, bhlB_at, BIOB_at, GSIB_at, LACA_at, LYTA_at, NADC_at, PCKA_at, PURC_at, RPLL_at, spoOM_at, SPOVAA_at, SPOVG_at, XLYA_at, YBGB_at, YCDH_at, YCGN_at, YCGO_at, YCKE_at, YCSE_at, YCZF_at, YDAG_at, YDBI_at, YDDK_at, YFMH_r_at, YHDS_r_at, YHDX_r_at, YHFH_r_at, YHZA_at, YLXQ_at, YOAB_at, YORB_i_at, YPTA_at, YPUD_at, YPUF_at, YQCE_at, YRHD_at, YRZI_r_at, YTGB_at, YWFO_at, YXLE_at, YYBJ_at, YYBN_at
AdapL.min	AMYC_at, ARGF_at, PCKA_at, XLYA_at, YCGN_at, YHZA_at, YTGB_at, YXLE_at
AdapL.1se	GAPB_at, XLYA_at, YHZA_at, YXLE_at
SCAD	AADK_at, ARGC_at, IOLE_at, mrpD_at, PCP_at, SPOVAB_at, sspM_r_at, YFIJ_at, YHDT_at, YHEH_at, YKCA_at, YMFF_at, YOAC_at, YOSV_r_at, YPZE_at, YRVM_at, YXIC_at, YXLF_at
Dant	XHLA_at, XTRA_at, YCGN_at, YCKE_at, YDAR_at, YOAB_at, YXLD_at
RelaxL	ARGF_at, DNAJ_at, GAPB_at, LYSC_at, PKSA_at, PRIA_at, SPOVAA_at, XHLB_at, XTRA_at, YACN_at, YBFI_at, YCDH_at, YCGO_at, YCKE_at, YCLB_at, YCLF_at, YDDH_at, YDDK_at, YEBC_at, YEZB_at, YFHE_r_at, YFII_at, YFIO_at, YFIR_at, YHDS_r_at, YKBA_at, YOAB_at, YQJU_at, YRVJ_at, YTGB_at, YURQ_at, YXLD_at, YXLE_at, YYBG_at, YYDA_at
SqrtL	LYSC_at, METB_at, PHRI_r_at, RPLJ_at, RPLL_at, RPLO_at, RPLP_at, RPLX_at, RPSN_at, SIGY_at, XHLA_at, XKDS_at, XTRA_at, YBGB_at, YCDH_at, YCDI_at, YCEA_at, YCGM_at, YCGN_at, YCGO_at, YCGP_at, YCKE_at, YCLF_at, YDAR_at, YDBM_at, YDDK_at, YDDM_at, YEBC_at, YHFH_r_at, YHZA_at, YOAB_at, YODF_at, YRPE_at, YRVJ_at, YTGA_at, YTGB_at, YTGD_at, YTIA_at, YXLC_at, YXLD_at, YXLE_at, YXLF_at, YXLG_at, YXLJ_at
ScalL	GAPB_at, XHLA_at, XLYA_at, YCDH_at, YCGN_at, YCKE_at, YHZA_at, YXLD_at
DC.VS	FLHO_at, RPLX_at, xepA_at, YCKE_at, YQKD_at, YRHC_at, YWRO_at

Table 3.19: Selected genes in the raw riboflavin data for each of the eleven considered procedures. Genes selected 8 times are highlighted in **coral** color, the ones selected 7 times in **violet** and those corresponding to 6 times in **blue**.

terms for these procedures. The LASSO.BIC is the one that selects the greatest number of terms, as it would be expected for the $p > n$ framework based on observed results in Sections 3.1 and 3.2. This is followed for the SqrtL, RelaxL, LASSO.min, and LASSO.1se. This applies in both, without and univariate standardization scenarios.

It is expected for these five algorithms to be able to recover a great part of the S set, which is unknown in practice, but adding quite a noise in the process. In contrast, AdapL.1se, Dant, DC.VS and AdapL.min are the algorithms that select fewer covariates. These last are expected to be more conservative, guaranteeing that a high percentage of the selected covariates are relevant. Nevertheless, as it has been seen along Sections 3.1 and 3.2, when $p > n$ and there exists strong dependence structures and covariates with different scales, some important terms could be interchanged with irrelevant ones.

The genes selected by the approaches in the without standardization frameworks are displayed in Table 3.19. Table 3.20 collects those for the univariate standardization case. Most of the procedures change their selection within the without and univariate standardization contexts. Only the Dant and the DC.VS algorithms keep their selection.

The AdapL.1se and AdapL.min always select a large percentage of popular genes, in the sense of those most selected by the eleven algorithms. However, the Dant and the DC.VS pick a smaller percentage of these genes.

LASSO.min	ARGF_at, DNAJ_at, GAPB_at, LYSC_at, PRIA_at, SPOIIAA_at, SPOVAA_at, THIA_at, THIK_at, XHLB_at, XKDP_at, YACN_at, YBFI_at, YCDH_at, YCGO_at, YCKE_at, YCLB_at, YCLF_at, YDDH_at, YDDK_at, YEBC_at, YFHE_r_at, YFIO_at, YFIR_at, YHDS_r_at, YKBA_at, YKVJ_at, YLXW_at, YMFE_at, YOAB_at, YPGA_at, YQJT_at, YQJU_at, YRVJ_at, YTGB_at, YUID_at, YURQ_at, YWRO_at, YXLD_at, YXLE_at, YYBG_at, YYDA_at
LASSO.1se	ARGF_at, DNAJ_at, GAPB_at, LYSC_at, PKSA_at, SPOIISA_at, SPOVAA_at, XHLB_at, XKDS_at, XTRA_at, YBFI_at, YCDH_at, YCGO_at, YCKE_at, YCLB_at, YCLF_at, YDDH_at, YDDK_at, YEBC_at, YEZB_at,YFHE_r_at, YFIR_at, YHDS_r_at, YKBA_at, YOAB_at, YQJU_at, YRVJ_at, YURQ_at, YXLD_at, YXLE_at, YYDA_at
LASSO.BIC	ADHB_at, ALD_at, ARAA_at, ARAM_at, ARAN_at, ARGF_at, ARGH_at, DEGA_at, ECSB_at, GAPB_at, GUTR_at, LEVF_at, LYSC_at, METK_at, PHOA_at, PYRAA_at, sigM_at, SPOIVA_at, SPOVAA_at, XHLB_at, XKDB_at, XKDP_at, XLYA_at, YACN_at, YBFI_at, YBXA_at, YCLB_at, YDAO_at, YDDH_at, YDDK_at, YEBC_at, YESV_at, YETH_at, YFHE_r_at, YFIO_at, YHDS_r_at, YIST_at, YISU_at, YKBA_at, YKNV_at, YKVJ_at, YLXW_at, YMAH_i_at, YOAB_at, YOSU_at, YPGA_at, YPUI_at, YQED_at, YQGJ_at, YQJT_at, YQJU_at, YRVJ_at, YTGB_at, YTSA_at, YUID_at, YULB_at, YULC_at, YURR_at, YUSJ_at, YVFM_at, YVHJ_at, YWBI_at, YWJG_at, YWRO_at, YXAF_at, YXIB_at, YXLD_at, YXLE_at, YYBI_at, YYCO_at
AdapL.min	ARGF_at, SPOVAA_at, XHLB_at, YCLB_at, YEBC_at, YOAB_at, YXLD_at
AdapL.1se	ARGF_at, XHLB_at, YOAB_at, YXLD_at
SCAD	AADK_at, ARGC_at, IOLE_at, mrpD_at, PCP_at, SPOVAB_at, sspM_r_at, YFIJ_at, YHDT_at, YHEH_at, YKCA_at, YMFF_at, YOAC_at, YOSV_r_at, YPZE_at, YRVM_at, YXIC_at, YXLF_at
Dant	XHLA_at, XTRA_at, YCGN_at, YCKE_at, YDAR_at, YOAB_at, YXLD_at
RelaxL	ARGF_at, CTAA_at, DNAJ_at, GAPB_at, LYSC_at, PRIA_at, SPOIIAA_at, SPOVAA_at, THIA_at, THIK_at, XHLB_at, XKDB_at, YACN_at, YBFI_at, YCKE_at, YCLB_at, YCLF_at, YDDH_at, YDDK_at, YEBC_at, YFHE_r_at, YFIO_at, YFIR_at, YHDS_r_at, YKBA_at, YKVJ_at, YLXW_at, YMFE_at, YOAB_at, YPGA_at, YQJT_at, YQJU_at, YRVJ_at, YTGB_at, YUID_at, YWRO_at, YXIB_at, YXLD_at, YXLE_at, YYBG_at, YYCO_at, YYDA_at
SqrtL	LYSC_at, METB_at, PHRI_r_at, RPLJ_at, RPLL_at, RPLO_at, RPLP_at, RPLX_at, RPSN_at, SIGY_at, XHLA_at, XKDS_at, XTRA_at, YBGB_at, YCDH_at, YCDI_at, YCEA_at, YCGM_at, YCGN_at, YCGO_at, YCGP_at, YCKE_at, YCLF_at, YDAR_at, YDBM_at, YDDK_at, YDDM_at, YEBC_at, YHFH_r_at, YHZA_at, YOAB_at, YODF_at, YRPE_at, YRVJ_at, YTGA_at, YTGB_at, YTGD_at, YTIA_at, YXLC_at, YXLD_at, YXLE_at, YXLF_at, YXLG_at, YXLJ_at
ScalL	LYSC_at, SPOIISA_at, XHLA_at, XKDS_at, XTRA_at, YCGN_at, YCGO_at, YCKE_at, YDDK_at, YEBC_at, YHCL_at, YOAB_at, YURQ_at, YXLD_at
DC.VS	FLHO_at, RPLX_at, xepA_at, YCKE_at, YQKD_at, YRHC_at, YWRO_at

Table 3.20: Selected genes in the univariate standardized riboflavin data for each of the eleven considered procedures. Genes selected 9 times are highlighted in **coral** color, the ones selected 7 times in **violet** and those corresponding to 6 times in **blue**.

Bühlmann et al. (2014) apply stability selection with randomized LASSO over the riboflavin data and detect three stable genes: `LYSC_at`, `YOAB_at`, and `YXLD_at`. The most similar selection is the one performed by the adaptive versions, AdapL.min and AdapL.1se, considering the univariate standardization. These techniques select 7 and 4 genes of the $p = 4088$ available, respectively, including `YOAB_at` and `YXLD_at`. The LASSO versions, jointly with the RelaxL, the SqrtL, and the ScaL select the `LYSC_at` gen under

the univariate standardization framework. However, only the Relax and the SqrtL choose this in the selection process using the raw data. As it has been seen for different examples in Sections 3.1 and 3.2, this gen could be a relevant one, but a greater sample size could be needed to recover this. Furthermore, other important genes could be avoided in the Bühlmann et al. (2014) selection because of the same reason. Some possible candidates would be those repeated a great number of times or the ones selected for procedures that have displayed a more robust behavior in simulations, such as the AdapL.1se or the Dant.

In conclusion, although a larger sample size would be necessary to ensure adequate recovery of the relevant terms, the covariates selection algorithms for the context $p > n$ allow a great dimensionality reduction. From $p = 4088$ genes, all algorithms select fewer than $n = 71$ terms. Besides, one can work with fewer than 10 covariates using the selections of the AdapL.min, AdapL.1se, Dant, or DC.VS approaches. This transforms the problem into a tractable one.

3.4.2 Prostate cancer

Here, an additional medical study is analyzed. In this case, eight clinical measures are employed to explain the level of prostate-specific antigen before surgery in men which suffer from prostate cancer. These results in $p = 8$ covariates measured in $n = 97$ patients. This example has covariates with similar scale ranges, except for two of them. Besides, these mainly have medium and strong positive dependence relations between them. More details and an explanatory analysis can be found in Section 2.5.2. This data is introduced in Stamey et al. (1989) and has been previously studied in works as Hastie et al. (2009) (Chapter 3) or Székely and Rizzo (2014) applying covariates selection techniques. In Hastie et al. (2009) and Székely and Rizzo (2014), a training sample of $n = 67$ individuals is employed to select covariates and then, the remaining 30 samples are used to perform prediction. Next, following their same guidelines, results of the studied covariates selection procedures introduced in Section 3.1.3 are obtained and compared with theirs.

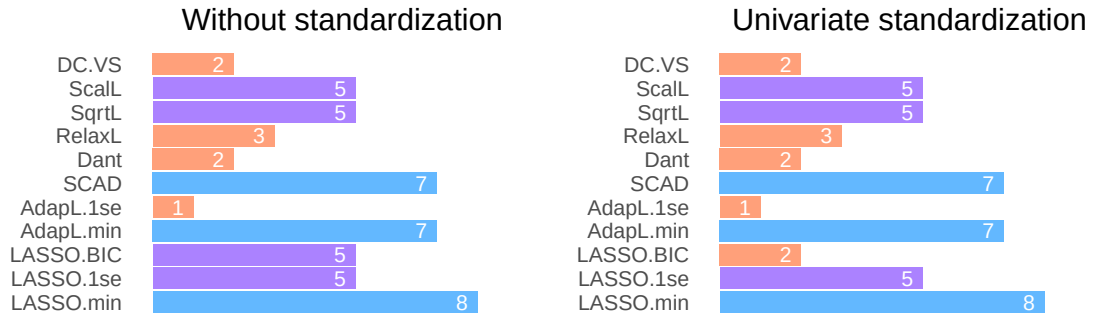


Figure 3.25: Number of selected covariates for the considered procedures for raw data (left) and univariate standardized data (right) of the prostate cancer data set.

An analysis of the data under without and univariate standardizations is performed. As mentioned above, the training set is employed for selecting which covariates are the

relevant ones, and then, the test set is used to measure the models' prediction capability following Hastie et al. (2009) guidelines. In contrast with Hastie et al. (2009) and Székely and Rizzo (2014), we do not start standardizing each variable considering all $n = 97$ samples. Instead, we work with the raw data in the without standardization case, and we standardize separately, the training and test sets, in the univariate version. In both scenarios, we work with all variables centered to avoid the intercept in the regression model. This is done without loss of generality. Again, we distinguish between training and test sets to center the variables. This is motivated to correctly guarantee that the intercept is null in the covariates selection as well as prediction steps. We calculate the mean squared error using the test set to measure the prediction capability of each considered procedure in terms of their selected covariates. For this purpose, we adjust a classical linear model considering only the covariates obtained in the first step. This is a common way to proceed for guaranteeing a better estimation of the parameters vector (see, for example, Belloni and Chernozhukov (2013)).

In this data set, we appreciate first that the number of selected covariates is the same as in the without and univariate frameworks for all procedures except for the LASSO.BIC. This last approach selects 5 covariates in the without standardization version and two when applying the univariate one. This information is summarized in Figure 3.25. Moreover, as it is displayed in Table 3.22, all algorithms select the same covariates for without/univariate standardization but for the LASSO.1se, changing age for svi, the LASSO.BIC, removing age, lbph and pgg45, and the ScalL switching age by svi. At this point, it is interesting to note that all covariates are in similar scales except age and gleason, which have higher scale values (see Section 2.5.2 for more details). Hence, when applying univariate standardization, it makes sense for an algorithm to discard these if they are not relevant. Furthermore, to detect relevant covariates, one can start studying which are the ones more times selected for all algorithms. See these results in Table 3.21. We appreciate that the covariates most relevant, understanding these as the ones selected equal or greater than 6 times, seem to be the lcavol and the lweight, followed by lbph, svi, pgg45, and age. In contrast, lcp and gleason can be assumed as noise.

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45
WITHOUT	11	10	6	7	5	3	1	7
UNIVARIATE	11	10	3	6	7	3	1	6

Table 3.21: Number of times each covariates is selected for the 11 considered procedures.

Assuming that lcavol, as well as lweight, are the relevant covariates, it is appreciated in Table 3.22 as only LASSO.BIC in the univariate case, AdapL.1se, Dant, and DC.VS select these without adding noise to the model. This fact properly corresponds with results for dependence data with different scales. As we saw in Sections 3.1 and 3.2 only these procedures perform well when there exists dependence between covariates. If these results are compared with the ones of Hastie et al. (2009) and Székely and Rizzo (2014), it is observed that this pair of covariates are always selected for all procedures too. In the

LASSO performance of Hastie et al. (2009) lbph and svi are selected as well and for the pdCor procedure of Székely and Rizzo (2014) this adds gleason and svi. At this point, it is important to remark on some things. First, a different selection is done between our LASSO.1se algorithm and the LASSO implementation of Hastie et al. (2009) (see Table 3.22). This can be explained by two different facts: the way data is processed and the descent coordinates algorithm employed to obtain the penalization value of λ. This last gets different results for every run, even for the same data and context, changing the set of selected variables. Thus, it is not so surprising that one can get different results. Second, related to pdCor, it is important to take into consideration the differences in the selection process. Whereas pdCor searches for all types of partial relation in a forward way between covariates and response, considered penalization techniques assume linearity in the regression model. As it is argued in Székely and Rizzo (2014), they found that the relation between lpsa and gleason seems not to be perfectly linear. As a result, it makes sense for the considered algorithm devoted to linear regression, to not take into account this last covariate.

Eventually, in terms of prediction, one can compare obtained MSE (see Table 3.22) in an illustrative and exploratory way. Procedures labeled as optimal ones in terms of adding only relevant covariates, LASSO.BIC in the univariate case, AdapL.1se, Dant, and DC.VS, gets the highest MSE values. Nevertheless, if one applies observed results for simulations with dependence structures and covariates in different scales, it has been seen that the rest of the procedures tend to overestimate the results. Besides, it is interesting to consider that a good trade-off between a fewer number of covariates and an increment in MSE is obtained. The best prediction result is achieved for LASSO.min adding all covariates and getting a $MSE = 0.323$. In contrast, the worst result is achieved for the AdapL.1se with a $MSE = 0.479$ but this only takes into consideration a covariate. As a result, it seems worth it to lose some MSE accuracy to get harmony. If the MSE calculation is applied to the selection made by the LASSO algorithm of Hastie et al. (2009) and the pdCor of Székely and Rizzo (2014), it is obtained values of $MSE = 0.349$ and $MSE = 0.455$, respectively, for both without and univariate standardizations. Because of these results, it seems that the inclusion of more covariates does not improve too much the prediction results. For the LASSO of Hastie et al. (2009) one can see simpler models as the one of the RelaxL obtains a similar MSE avoiding an extra covariate. In contrast, for the pdCor, models which only consider lcavol and lweight, as the LASSO.BIC in the univariate case, Dant or DC.VS, improve the MSE value with two fewer covariates. This last can be explained because of the nonlinear relation between lpsa and gleason variables.

3.4.3 Body fat

Next, the body fat data set introduced in Section 2.5.3 of Chapter 2 is used for covariates selection. In this case, it is wanted to detect relevant covariates for the body fat explanation. In particular, $p = 14$ covariates are considered. These are body measures taken in $n = 252$ men. After applying some data transformation to avoid skewness and possible outliers, the considered sample size is $n = 239$ individuals. Details of the employed processes are

		SELECTED MODEL	MSE
LASSO.min	WITHOUT:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + gleason + pgg45	0.323
	UNIVARIATE:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + gleason + pgg45	0.323
LASSO.1se	WITHOUT:	psa ∼ lcavol + lweight + age + lbph + pgg45	0.438
	UNIVARIATE:	lpsa ∼ lcavol + lweight + lbph + svi + pgg45	0.345
LASSO.BIC	WITHOUT:	lpsa ∼ lcavol + lweight + age + lbph + pgg45	0.438
	UNIVARIATE:	lpsa ∼ lcavol + lweight	0.471
AdapL.min	WITHOUT:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + pgg45	0.323
	UNIVARIATE:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + pgg45	0.323
AdapL.1se	WITHOUT:	lpsa ∼ lcavol	0.479
	UNIVARIATE:	lpsa ∼ lcavol	0.479
SCAD	WITHOUT:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + pgg45	0.323
	UNIVARIATE:	lpsa ∼ lcavol + lweight + age + lbph + svi + lcp + pgg45	0.323
Dant	WITHOUT:	lpsa ∼ lcavol + lweight	0.471
	UNIVARIATE:	lpsa ∼ lcavol + lweight	0.471
RelaxL	WITHOUT:	lpsa ∼ lcavol + lweight + svi	0.354
	UNIVARIATE:	lpsa ∼ lcavol + lweight + svi	0.354
SqrtL	WITHOUT:	lpsa ∼ lcavol + lweight + lbph + svi + pgg45	0.345
	UNIVARIATE:	lpsa ∼ lcavol + lweight + lbph + svi + pgg45	0.345
ScalL	WITHOUT:	lpsa ∼ lcavol + lweight + age + lbph + pgg45	0.438
	UNIVARIATE:	lpsa ∼ lcavol + lweight + lbph + svi + pgg45	0.345
DC.VS	WITHOUT:	lpsa ∼ lcavol + lweight	0.471
	UNIVARIATE:	lpsa ∼ lcavol + lweight	0.471
LASSO		lpsa ∼ lcavol + lweight + lbph + svi	0.349
pdCor		lpsa ∼ lcavol + lweight + gleason + svi	0.455

Table 3.22: Selected covariates for the considered procedures using the prostate data training set and for the LASSO approach of Hastie et al. (2009) and the pdCor of Székely and Rizzo (2014). MSE: mean squared error obtained using the test set with the previous selected covariates.

given in Section 2.5.3. This data set has covariates highly correlated between them as it is displayed through its correlation matrix values (see Section 2.5.3). Moreover, these are in different scales as well. Then, covariates selection is performed using the procedures introduced in Section 3.1.3. Next, the obtained results are analyzed following the criteria of Sections 3.1 and 3.2.

Again, we work with the data center without loss of generality and consider both: without and univariate standardizations. The number of covariates selected by each of the studied algorithms is shown in Figure 3.26. In this example, all procedures tend to select fewer covariates applying the univariate standardization. Dant and DC.VS algorithms are the only procedures that keep selecting a covariate in both frameworks. This fact may be motivated by the notable differences in the scale values of the covariates. In the univariate standardization case, most algorithms select fewer than 4 covariates. This last may be because of the existence of strong correlations between covariates. Thus, just a bunch of covariates may explain all the information. In contrast, for the without standardization

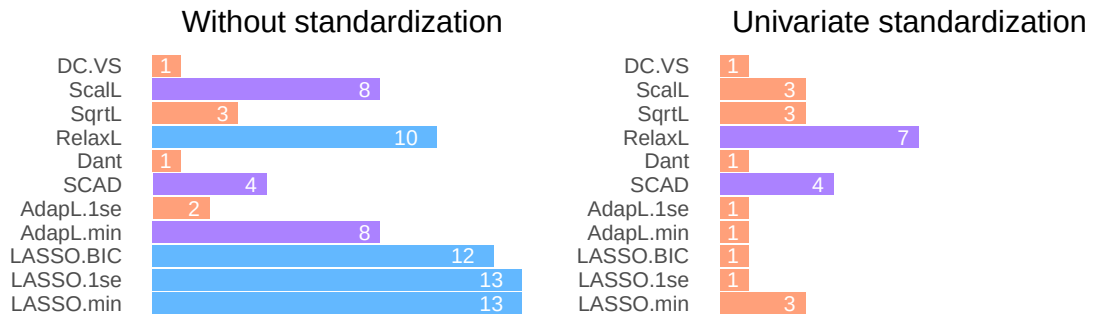


Figure 3.26: Number of selected covariates for the considered procedures for raw data (left) and univariate standardized data (right) of the body fat data set.

case, only the AdapL.1se, Dant, SqrtL, and DC.VS select fewer than 4 terms. Given the results of Sections 3.1 and 3.2, we see that the AdapL.1se, the Dant, and sometimes the DC.VS approach tend to be more conservative in the sense that these select fewer covariates but guarantee that these are relevant with a high probability. This contrast with other procedures that select more features for a proper recovery of the complete S set, but these add quite a noise in exchange.

Model	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}
LASSO.min	✓✓	✓✓	✓	✓✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
LASSO.1se	✓✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
LASSO.BIC	✓✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓
AdapL.min	✓	✓	✓	✓	✓		✓		✓			✓		✓
AdapL.1se	✓		✓				✓							
SCAD	✓✓	✓✓		✓✓					✓✓					
Dant	✓✓													
RelaxL	✓✓	✓✓	✓✓	✓✓		✓		✓✓	✓✓		✓	✓	✓✓	
SqrtL	✓✓	✓✓		✓✓										
ScalL	✓	✓✓	✓	✓✓	✓		✓		✓			✓		✓
DC.VS	✓✓													
TOTAL:	8/11	8/5	7/1	8/5	5	4	6	4/1	7/2	3	3	6	1/1	5

Table 3.23: Selected covariates without standardization (✓) and with univariate standardization (✓). Covariates are denoted as X_1: Density, X_2: Age, X_3: Weight, X_4: Height, X_5: Neck, X_6: Chest, X_7: Abdomen, X_8: Hip, X_9 : Thigh, X_{10} : Knee, X_{11} : Ankle, X_{12} : Biceps, X_{13} : Forearm, X_{14} : Wrist.

Covariates selected for each algorithm, distinguish between without and univariate frameworks, are collected in Table 3.23. The terms most selected for both types of standardizations are density, age, and height, followed by thigh and weight. In particular, in the univariate standardization scenario, several procedures select only the density covariate. Some of these are the LASSO.BIC, the AdapL.1se, the Dant, and the DC.VS. This fact makes sense taking into consideration that Siri's equation (see Siri (1956)) claims

that the BodyFat can be explained by the Density parameter as

$$\text{BodyFat} = 4.95/\text{Density} - 4.50.$$

Thus, this is an example where consideration of data without standardization when there are covariates on very different scales can lead to wrong results, selecting irrelevant terms and avoiding important ones. In addition, due to strong dependence patterns between covariates (see Section 2.5.3), confusion phenomena arise. This translates into procedures adding irrelevant terms even in the univariate standardized framework. As a consequence, this real data example asserts the result previously obtained along Sections 3.1 and 3.2.4. These claim that the best options for scenarios having covariates with different scales under dependence are the LASSO.BIC (only for the $n > p$ case), univariate standardized AdapL.1se, univariate standardized Dant, and DC.VS algorithms.

3.4.4 Portuguese wine

Eventually, covariates selection is applied over a Portuguese red wine data set. This is introduced jointly with an exploratory analysis in Section 2.5.4 of Chapter 2. In this case, a total of $p = 10$ physicochemical parameters are employed to explain the alcohol content. These are measured for $n = 1599$ wines. After some transformation of the data to avoid skewness and outliers (the reader is referred to Section 2.5 for more details), a total of $n = 1519$ samples are considered in the study. This is also an example where there exists some, but not too much, strong correlations between covariates. Besides, there are only two considered covariates that have quite different ranges of values.

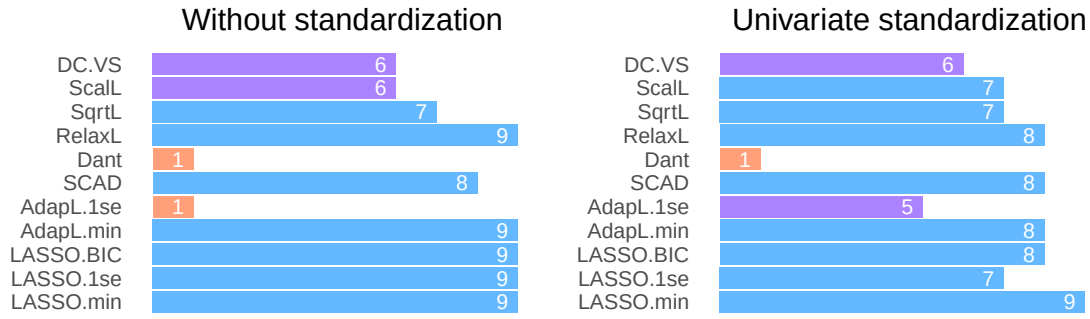


Figure 3.27: Number of selected covariates for the considered procedures for raw data (left) and univariate standardized data (right).

After centering the data, we start analyzing results for the considered covariates selection techniques under both: without and univariate standardization frameworks. In this case, results about the number of selected covariates do not differ too much between the two types of standardization as it can be appreciated in Figure 3.27. They vary between four units at most from one standardization to another in each methodology. Algorithms such as the LASSO.min, SCAD, Dant, SqrtL, and DC.VS select the same number in both frameworks. The rest of the procedures select fewer covariates in the univariate version

in comparison with the without case except for the AdapL.1se as well as the ScalL. It is shown in Figure 3.27 how all algorithms add at least six covariates except for the without AdapL.1se and Dant procedures.

Model	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
LASSO.min	✓✓	✓✓	✓✓	✓✓	✓✓	✓	✓✓	✓	✓✓	✓✓
LASSO.1se	✓✓	✓	✓✓	✓✓	✓	✓	✓✓	✓	✓✓	✓✓
LASSO.BIC	✓✓	✓✓	✓✓	✓✓	✓	✓	✓✓	✓	✓✓	✓✓
AdapL.min	✓✓	✓✓	✓✓	✓✓	✓	✓	✓✓	✓	✓✓	✓✓
AdapL.1se	✓			✓			✓	✓	✓	✓
SCAD	✓✓	✓✓	✓✓	✓✓			✓✓	✓✓	✓✓	✓✓
Dant								✓✓		
RelaxL	✓✓	✓✓	✓✓	✓✓	✓		✓✓	✓✓	✓✓	✓✓
SqrtL	✓✓		✓✓	✓✓			✓✓	✓✓	✓✓	✓✓
ScalL	✓✓	✓	✓✓	✓✓		✓	✓✓	✓	✓	✓
DC.VS	✓✓			✓✓			✓✓	✓✓	✓✓	✓✓
TOTAL:	9/10	7/5	8/8	9/10	5/1	5	10/9	5/11	8/10	8/10

Table 3.24: Selected covariates without standardization (✓) and with univariate standardization (✓). Covariates are denoted as X_1: fixed acidity, X_2: volatile acidity, X_3: citric acidity, X_4: sugar, X_5: chlorides, X_6: free sulfur, X_7: total sulfur, X_8: density, X_9: pH, X_{10}: sulphates.

Next, it is analyzed which covariates are the most selected in each scenario. These depend on the type of employed standardization as expected. These results are summarized in Table 9. In terms of all standardized versions, fixed acidity, sugar, total sulfur, pH, and sulphates are the ones repeated between the five most selected. These are followed by citric acid and volatile acidity. The covariates most times excluded are the free sulfur and chlorides. The density feature is in the middle: this is quite relevant using univariate standardization but not for the without case. One can see that, in global terms, selection results are quite similar between both standardizations. Nevertheless, the covariates' importance, in terms of the percentage of times selected, changes from one procedure to another as normal.

The procedures which select most of the popular terms, without the inclusion of too many noisy ones, are the univariate AdapL.1se and both DC.VS versions. These two procedures also choose the density term, which is popular under the univariate selection, and the univariate AdapL.1se does not select total sulfur. Conversely, without AdapL.1se only chooses this covariate. The fact that the Dant algorithm only selects the density covariate in both frameworks is striking, which contrasts with previous results where this procedure obtained similar results to the AdapL.1se and DC.VS procedures. Considering these facts and that there are dependence and scale effects, DC.VS or univariate AdapL.1se covariates selection seems a reasonable option for alcoholic context explanation.

Chapter 4

Novel distance-based dependence measures for complex data

Until now, we have displayed how to apply covariates selection in the high dimensional framework when a model structure is assumed, as shown for the linearity assumption in Chapters 2 and 3. However, it is not always possible to know the structure of the model in advance, and one has to go one step further. In particular, we focus here on covariates selection techniques without any assumption in the regressor function. For this purpose, novel distance-based dependence measures are employed to construct appropriate statistics and perform significance tests. Specifically, these ideas are used to select covariates in complex models where estimating a sufficiently flexible regression function is a tough problem. These novel coefficients are based on modifications of the innovative distance covariance of Székely et al. (2007). We start motivating the necessity of these new dependence measures in Section 4.1, reviewing existing coefficients for detecting dependence patterns and their drawbacks. Next, some of the resulting distance coefficients used to test different types of dependence are introduced throughout Section 4.2. Eventually, a discussion on their use and advantages in complex models is carried out in Section 4.3.

4.1 Classical measures of dependence

The first dependence measure for random vectors is the well-known correlation coefficient. See, for example, Pearson (1920). This can be used to perform covariate selection in regression models by selecting those covariates with the highest correlation values with the response. Nevertheless, this is only capable of detecting linear relations. In order to identify other types of dependence patterns, new measures arose, like the ones based on ranks. Examples of these are the Spearman's coefficient (Wissler (1905)) or the Kendall's τ (Kendall (1938)). These coefficients are robust to outliers and can detect any type of monotone dependence structure. Despite these improvements, many non-monotonic dependence patterns still cannot be detected by these coefficients, being unsuitable for some regression models. Besides, all these coefficients only measure the grade of dependence of every covariate separately, ignoring the information provided by the remaining ones in the process. As a result, their computational cost increases in terms of the p size, having to apply a total of p comparisons. An example of their performance is displayed in Figure 4.1.

In 2007, Székely et al. (2007) introduced the concept of distance covariance (DC) and its scale-invariant version, the distance correlation. As its name suggests, this last computes the existing correlation between sample distances. This coefficient is able to detect all types of possible dependence relations, solving the previous limitations of the correlation

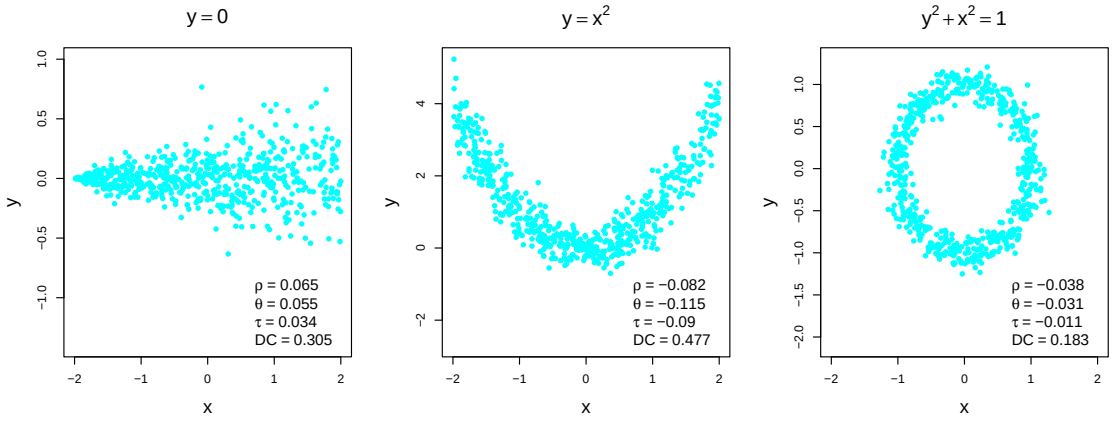


Figure 4.1: Value of the correlation coefficient (ρ), Spearman's coefficient (θ), Kendall's one (τ) and distance correlation (DC) for different simulated scenarios.

coefficients. Moreover, no preliminary assumption about the model structure is needed. This contrasts with the widely employed use of regularization techniques (see Chapters 2 and 3). As a result, it is possible to perform covariates selection using DC procedures for any regression model. Some examples are the approach of Székely et al. (2007), the DC-SIS (distance covariance sure independence screening) procedure of Li et al. (2012), applying the SIS (sure independence screening) algorithm for linear models of Fan and Lv (2008), or the partial distance correlation methodology of Székely and Rizzo (2014). The first and third procedures apply tests of independence by constructing a suitable statistic based on DC ideas. In contrast, the DC-SIS algorithm sorts the covariates using their associated DC value and then applies a threshold to keep only the most relevant ones in terms of model explanation. Another procedure iteratively using DC is the one proposed by Febrero-Bande et al. (2019) for additive formulations of the regression function. See Section 2.4.10 for more details about this method.

Recently, two new modifications of the DC coefficient were proposed to test different types of dependence. These are the martingale difference divergence (MDD) of Shao and Zhang (2014) and the conditional distance covariance (CDC) of Wang et al. (2015). The MDD coefficient is employed to test the causality of a vector $Y \in \mathbb{R}^q$ conditioned to a scalar random variable $X \in \mathbb{R}$, whereas the CDC tests the conditional dependence of two random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$, given a third one $Z \in \mathbb{R}^r$. Both coefficients can be used for specification testing and to perform covariates selection procedures. Some examples are the works of Shao and Zhang (2014) as well as Zhang et al. (2018) for the MDD case, and the procedure proposed in Wang et al. (2015) for the CDC performance. Furthermore, partial versions of the DC (see Székely and Rizzo (2014)) and the MDD coefficients (see Park et al. (2015)) have been proposed as well. These last test the corresponding independence between X and Y once the effect of a covariate Z has been removed. A summary of all these tests is displayed in Table 4.1 at the end of Section 4.2.

4.2 Novel distance-based dependence measures

In this section, the DC, MDD, and CDC coefficients are introduced in detail in Sections 4.2.1, 4.2.2, and 4.2.3, respectively. Some of their most notable properties are displayed and proper estimators are obtained for each coefficient.

4.2.1 Distance covariance (DC)

The DC coefficient introduced in Székely et al. (2007) is a new measure of dependence that detects all possible types of relations between two random vectors of arbitrary dimension. Thus, given $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ with $p, q \geq 1$, the DC tests if this pair of random vectors is independent. This test is defined by

$$H_0 : X \perp Y \quad \text{vs.} \quad H_1 : X \not\perp Y, \tag{4.1}$$

where $X \perp Y$ denotes independence between X and Y.

In particular, two random vectors are said to be independent when it is verified that $F_{X,Y} = F_X F_Y$, where F_X and F_Y are the distribution functions of X and Y, respectively, and $F_{X,Y}$ their joint distribution. Rewriting this condition in terms of their associated characteristic functions, the independence test can be formulated as

$$H_0 : \varphi_{X,Y} = \varphi_X \varphi_Y \quad \text{vs.} \quad H_1 : \varphi_{X,Y} \neq \varphi_X \varphi_Y \tag{4.2}$$

where $\varphi_{X,Y}$ is the joint characteristic function and φ_X, φ_Y the marginal versions of X, Y.

In order to test the null hypothesis of (4.2), a statistic measuring whether the difference $\varphi_{X,Y} - \varphi_X \varphi_Y$ is significant, is needed. This is the main motivation for the construction of the DC coefficient (Székely et al. (2007), Szekely and Rizzo (2017)).

Thus, a weighted L_2 norm ($\|\cdot\|_w^2$) defined in the $\mathbb{R}^p \times \mathbb{R}^q$ space of complex functions is considered to measure the difference between $\varphi_{X,Y}$ and $\varphi_X \varphi_Y$. This norm is defined as

$$\|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)\|_w^2 = \int_{\mathbb{R}^p \times \mathbb{R}^q} |\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)|^2 w(t,s)\, dt\, ds, \tag{4.3}$$

where $w(\cdot,\cdot)$ is a weight function that must be correctly selected to ensure the existence of the above integral and $|f| = f\bar{f}$, being $f(\cdot)$ a complex value function with conjugate $\bar{f}(\cdot)$.

Therefore, once a suitable weight function $w(\cdot,\cdot)$ is selected, one can consider as a measure of dependence $DC^2(X,Y;w) = \|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)\|_w^2$. This quantity satisfies that $DC^2(X,Y;w) = 0$ if and only if X and Y are independent. Particularly, dividing $DC^2(X,Y;w)$ by $\sqrt{DC(X;w)DC(Y;w)}$, where

$$DC^2(X;w) = \int_{\mathbb{R}^{2p}} |\varphi_{X,X}(t,s) - \varphi_X(t)\varphi_X(s)|^2 w(t,s)\, dt\, ds, \tag{4.4}$$

it is obtained a type of unsigned correlation coefficient, $DCor_w$. This term, $DCor_w$, is an extension of the correlation coefficient for all types of dependence relations.

Following these guidelines, Székely et al. (2007) take

$$w(t,s) = (c_p c_q \|t\|_p^{1+p} \|s\|_q^{1+q})^{-1} dt\, ds \;\text{ for }\; c_p = \frac{\pi^{(p+1)/2}}{\Gamma((p+1)/2)} \;\text{ and }\; c_q = \frac{\pi^{(q+1)/2}}{\Gamma((q+1)/2)}, \tag{4.5}$$

where $\|\cdot\|_p$ and $\|\cdot\|_q$ are the euclidean norms in $\mathbb{R}^p$ and $\mathbb{R}^q$ and $\Gamma(\cdot)$ the gamma function.

Making a little abuse of notation, it is written $\|\cdot\|^2$ henceforth for short, instead of $\|\cdot\|_\omega^2$, as the L_2 norm using the weight function defined above. Thus, to guarantee the finiteness of $\|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)\|^2$, it is sufficient that $\mathbb{E}[\|X\|_p] < \infty$ and $\mathbb{E}[\|Y\|_q] < \infty$. Hence, the DC between two random vectors X and Y with finite first moments is the nonnegative number $DC(X,Y)$ defined by

$$DC^2(X,Y) = \|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)\|^2 = \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)|^2}{\|t\|_p^{p+1} \|s\|_q^{q+1}} dt\, ds, \tag{4.6}$$

and the distance variance coefficient is given as the square root of

$$DC^2(X) = DC^2(X,X) = \|\varphi_{X,X}(t,s) - \varphi_X(t)\varphi_X(s)\|^2. \tag{4.7}$$

In a similar way, the distance correlation coefficient between two random vectors X and Y with finite first moments is the nonnegative number $DCor(X,Y)$ given by

$$DCor^2(X,Y) = \begin{cases} \frac{DC^2(X,Y)}{\sqrt{DC^2(X)DC^2(Y)}}, & DC^2(X)DC^2(Y) > 0, \\ 0, & DC^2(X)DC^2(Y) = 0. \end{cases} \tag{4.8}$$

This term verifies that $0 \leq DCor(X,Y) \leq 1$, and $DCor(X,Y) = 0$ if and only if X and Y are independent.

Alternative expressions for the squared DC coefficient given in (4.6) are

$$\begin{aligned} DC^2(X,Y) =& \mathbb{E}\left[\|X' - X''\|_p \|Y' - Y''\|_q\right] \\ &+ \mathbb{E}\left[\|X' - X''\|_p\right] \mathbb{E}\left[\|Y' - Y''\|_q\right] - 2\mathbb{E}\left[\|X' - X''\|_p \|Y' - Y'''\|_q\right] \end{aligned} \tag{4.9}$$

and

$$\begin{aligned} DC^2(X,Y) =& \mathbb{E}_{X'Y'}\left[\mathbb{E}_{X''Y''}\left[\|X' - X''\|_p \|Y' - Y''\|_q\right]\right] \\ &+ \mathbb{E}_{X'X''}\left[\|X' - X''\|_p\right] \mathbb{E}_{Y'Y''}\left[\|Y' - Y''\|_q\right] \\ &- 2\mathbb{E}_{X'Y'}\left[\mathbb{E}_{X''}\left[\|X' - X''\|_p\right] \mathbb{E}_{Y''}\left[\|Y' - Y''\|_q\right]\right] \end{aligned} \tag{4.10}$$

where (X', Y'), (X'', Y'') and (X''', Y''') are iid copies of (X, Y). The reader is referred to Székely et al. (2007) for more details.

Next, empirical estimators of all these quantities are introduced to be able to perform the test (4.1) in practice. Given $(\mathbf{X_n}, \mathbf{Y_n}) = \{(X_i, Y_i), i = 1, \ldots, n\}$ an iid sample from the joint distribution function of $(X,Y) \in \mathbb{R}^p \times \mathbb{R}^q$, it is defined $A_{il} = a_{il} - \bar{a}_{i\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$ by

means of quantities

$$a_{il} = \|X_i - X_l\|_p, \quad \bar{a}_{i\cdot} = \frac{1}{n}\sum_{l=1}^{n} a_{il}, \quad \bar{a}_{\cdot l} = \frac{1}{n}\sum_{i=1}^{n} a_{il} \quad \text{and} \quad \bar{a}_{\cdot\cdot} = \frac{1}{n^2}\sum_{i,l=1}^{n} a_{il}, \tag{4.11}$$

and similarly $B_{il} = b_{il} - \bar{b}_{i\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}$ with $b_{il} = \|Y_i - Y_l\|_q$. Then, the squared empirical distance covariance $DC_n^2(\mathbf{X_n}, \mathbf{Y_n})$, being the empirical estimator of (4.6), is the nonnegative number defined by

$$DC_n^2(\mathbf{X_n}, \mathbf{Y_n}) = \frac{1}{n^2}\sum_{i,l=1}^{n} A_{il}B_{il}. \tag{4.12}$$

Respectively, the empirical distance variance $DC_n(\mathbf{X_n})$ is the square root of the nonnegative number given by

$$DC_n^2(\mathbf{X_n}) = DC_n^2(\mathbf{X_n}, \mathbf{X_n}) = \frac{1}{n^2}\sum_{i,l=1}^{n} A_{il}^2. \tag{4.13}$$

In summary, the estimator of $DC^2(X, Y)$ given in (4.12) is obtained as the multiplication of the matrices resulting from centering the sample distances twice.

Moreover, the empirical distance correlation coefficient, $DCor_n(\mathbf{X_n}, \mathbf{Y_n})$, is defined as the square root of

$$DCor_n^2(\mathbf{X_n}, \mathbf{Y_n}) = \begin{cases} \frac{DC_n^2(\mathbf{X_n}, \mathbf{Y_n})}{\sqrt{DC_n^2(\mathbf{X_n})DC_n^2(\mathbf{Y_n})}}, & DC_n^2(\mathbf{X_n})DC_n^2(\mathbf{Y_n}) > 0, \\ 0, & DC_n^2(\mathbf{X_n})DC_n^2(\mathbf{Y_n}) = 0. \end{cases} \tag{4.14}$$

This coefficient verifies that $0 \leq DCor_n(\mathbf{X_n}, \mathbf{Y_n}) \leq 1$, similar to the population version introduced in (4.8). Besides, if $DCor_n(\mathbf{X_n}, \mathbf{Y_n}) = 1$, then there exist a vector a, a nonzero real number b and an orthogonal matrix C such that $\mathbf{Y_n} = a + b\mathbf{X_n}C$. Furthermore, it is almost surely guaranteed that $\lim_{n\to\infty} DCor_n^2(\mathbf{X_n}, \mathbf{Y_n}) = DCor^2(X, Y)$. More properties about $DC_n^2(\mathbf{X_n}, \mathbf{Y_n})$, $DC_n(\mathbf{X_n})$ and $DCor_n(\mathbf{X_n}, \mathbf{Y_n})$ are derived in Székely et al. (2007).

In terms of asymptotic distribution, under the null hypothesis of independence, $nDC_n^2(\mathbf{X_n}, \mathbf{Y_n})/S_2$ converges in distribution to a quadratic form $Q \overset{D}{=} \sum_{m=1}^{\infty} c_m G_m^2$, being S_2 a normalizing factor defined in Székely et al. (2007), $\{G_m\}_{m=1}^{\infty}$ independent standard normal random variables and $\{c_m\}_{m=1}^{\infty}$ nonnegative constants that depend on the distribution of (X, Y). In contrast, if the independence hypothesis is violated, $nDC_n^2(\mathbf{X_n}, \mathbf{Y_n}) \to \infty$ in probability as $n \to \infty$. As a result, a test that rejects the null hypothesis of independence for large values of $nDC_n^2(\mathbf{X_n}, \mathbf{Y_n})$ is consistent in an omnibus way against dependence alternatives. In practice, it is possible to approximate the limiting distribution by resampling techniques, for example using permutation tests or bootstrap algorithms.

It is interesting to note that DC can be used not only for independence tests but also for Goodness-of-Fit (GoF) ones. An example is the work of Xu and He (2021). In this, a procedure based on DC is employed to test the null hypothesis $H_0 : X \perp \varepsilon$ and $m \in \mathcal{M}_\beta$ in the regression model $Y = m(X) + \varepsilon$ with $m \in \mathcal{M}_\beta = \{g(x)^\top \beta : \beta \in \mathbb{R}^p\}$ for a given known function $g(\cdot)$. In this context, the corresponding $\mathbf{Y_n}$ term is obtained using the residuals of the fitted model.

Another interesting extension of the DC coefficient is its use for partial tests. In particular, Székely and Rizzo (2014) introduced the concept of partial distance covariance (pDC) and an analogous correlation version. Then, given an extra random vector $Z \in \mathbb{R}^r$, which is known to contribute to the variation of Y, it is measured the dependence between Y and X after removing their respective dependence on Z. This can be denoted as $H_0 \colon \mathcal{P}_{Z^\perp}(X) \perp \mathcal{P}_{Z^\perp}(Y)$, where $\mathcal{P}_{Z^\perp}(X)$ as well as $\mathcal{P}_{Z^\perp}(Z)$ are the orthogonal projections of the $\mathcal{U}$-centered distance matrix of X, respectively Y, onto $Z^\perp$, being this last the orthogonal space generated by the $\mathcal{U}$-centered distance matrix of Z. This translates into testing if pDC can be assumed to be null.

Despite all the good qualities displayed by the empirical versions of the distance covariance and correlation coefficients, these exhibit some disadvantages as well. The squared empirical distance covariance $DC_n^2(\mathbf{X_n}, \mathbf{Y_n})$, introduced in (4.12), is a biased estimator of (4.6). In particular, its bias increases with the dimensions of X and Y, i.e. when $p, q \to \infty$. As a result, this translates into the fact that $DC_n(\mathbf{X_n}, \mathbf{Y_n})$ and $DCor_n(\mathbf{X_n}, \mathbf{Y_n})$ are biased estimators of $DC(X, Y)$ and $DCor(X, Y)$, respectively. Furthermore, quoting Székely and Rizzo (2013), although distance correlation characterizes independence, interpretation of the size of $DCor_n(\mathbf{X_n}, \mathbf{Y_n})$ without a formal test is difficult in high dimensions. This is owing to the fact that $DCor_n^2(\mathbf{X_n}, \mathbf{Y_n}) \longrightarrow 1$ as $p, q \to \infty$, even though X and Y are independent. To face these problems, a new unbiased sample estimator for distance covariance/ variance and a modified distance correlation statistic are proposed by Székely and Rizzo (2013). This new statistic is based on plug-in the new unbiased versions of DC in the numerator and denominator of expression (4.14). This statistic verifies that, under the null hypothesis of independence, this converges to a Student t distribution. As a result, this new approach also solves the inconsistency problem in high dimensions.

An additional problem is the computational associated with constructing the distance matrices. Some recent works, such as Huo and Székely (2016) or Chaudhuri and Hu (2019), have proposed some alternatives to reduce this cost in the case of considering two univariate random variables. New solutions applying to the vectorial framework are desirable and need to be considered in the future.

Eventually, it is interesting to remark on the natural relation between DC and the Hilbert-Schmidt Independence Criterion (HSIC) of Gretton et al. (2005). The HSIC employs the cross-covariance operator between two random vectors, defined in different reproducing kernel Hilbert spaces (RKHSs) with the universal kernel, to measure if there exists some type of dependence between them. The independence is verified when the HSIC operator takes the null value. The DC coefficient is just a particular case of the HSIC operator replacing general kernel distances with Euclidean ones. Through the years, a parallel evolution has been observed for both procedures. Some examples of papers linking both ideas are the works of Sejdinovic et al. (2013), Hua and Ghosh (2015), Zhu et al. (2020), or Edelmann and Goeman (2022), among others. As a result, the HSIC measure can also be employed to perform independence tests, as in the work of Song et al. (2012), or GoF tests, as it is done in Sen and Sen (2014) for simultaneous GoF and error-predictor independence tests in linear models.

4.2.2 Martingale difference divergence (MDD)

The MDD coefficient is another measure of dependence that was introduced by Shao and Zhang (2014). This coefficient is a natural extension of the DC of Székely et al. (2007), Szekely and Rizzo (2017) but to measure the departure from conditional mean independence between a scalar response variable $Y \in \mathbb{R}$ and a predictor vector $X \in \mathbb{R}^p$. The resulting test problem is now

$$H_0 : \mathbb{E}[Y|_X] = \mathbb{E}[Y] \quad \text{vs.} \quad H_1 : \mathbb{E}[Y|_X] \neq \mathbb{E}[Y]. \tag{4.15}$$

Its name is inherited from the interpretation of the martingale difference concept in probability. This means that, if the null hypothesis in (4.15) is verified, then $Y - \mathbb{E}[Y]$ is a martingale difference concerning the X vector.

Thus, following similar ideas and argumentation of the DC of Székely et al. (2007), the MDD coefficient is designed to measure the difference between the conditional mean and the unconditional one to perform (4.15). Then, the MDD of Y, given X, is the nonnegative number $MDD^2(Y|_X)$ defined as

$$MDD^2(Y|_X) = \frac{1}{c_p} \int_{\mathbb{R}^p} \frac{|\psi_{Y,X}(t) - \psi_Y \psi_X(t)|^2}{\|t\|_p^{p+1}} dt, \tag{4.16}$$

where $\psi_{Y,X}(t) = \mathbb{E}[Ye^{i<t,X>}]$, $\psi_Y = \mathbb{E}[Y]$ and $\psi_X(t) = \varphi_X(t)$.

The MDD coefficient defined in (4.16) verifies that $MDD^2(Y|_X) \geq 0$. Besides, this takes the null value if and only if the H_0 hypothesis of (4.15) holds. This is called divergence instead of distance because $MDD^2(Y|_X) \neq MDD^2(X|_Y)$ in general.

Similar to the distance correlation coefficient displayed in (4.8), it is possible to define a scale-invariant coefficient with the MDD ideas. This gives place to the martingale difference correlation (MDC), being the square root of

$$MDC^2(Y|_X) = \begin{cases} \frac{MDD^2(Y|_X)}{\sqrt{\text{var}^2(Y)\mathcal{V}^2(X)}}, & \text{var}^2(Y)\mathcal{V}^2(X) > 0, \\ 0, & \text{var}^2(Y)\mathcal{V}^2(X) = 0. \end{cases} \tag{4.17}$$

where $\mathcal{V}^2(X)$ is the distance variance of X defined in (4.7). The MDC verifies that $0 \leq MDC^2(Y|_X) \leq 1$. Similar properties as the ones of DC for $MDD^2(Y|_X)$ and $MDC^2(Y|_X)$ are collected in Shao and Zhang (2014).

Park et al. (2015) proved that, if it is guaranteed that $\mathbb{E}[|Y|^2] < \infty$, then an alternative formulation for the MDD coefficient can be given. This is defined as

$$MDD^2(Y|_X) = \frac{1}{c_p} \int_{\mathbb{R}^p} \left(\frac{1}{2} \Delta_s |\psi_{Y,X}(t) - \psi_Y \psi_X(t)|^2 \Big|_{s=0} \right) \frac{1}{\|t\|_p^{p+1}} \, dt, \tag{4.18}$$

where $\varphi_{X,Y}$ is now the joint characteristic function and φ_X, as well as φ_Y, the marginal versions of X and Y, respectively. Here, $\Delta_s f$ denotes the Laplacian operator of a function f. We refer to Park et al. (2015) for more details.

Equation (4.18) establishes a close connection between the MDD and the DC coefficients. In particular, expression (4.18) is directly related to the theoretical definition of the DC coefficient given in (4.6).

Furthermore, it can be seen in Shao and Zhang (2014) or Park et al. (2015) that, if it is verified that $\mathbb{E}[\|X\|_p^3] + \mathbb{E}[|Y|^3] < \infty$ and $\mathbb{E}[\|X\|_p^2] + \mathbb{E}[|Y|^2] < \infty$, an alternative form to the definition (4.16) is given by

$$\begin{aligned} MDD^2(Y|_X) =& \mathbb{E}\left[\|X - X'\|_p L(Y, Y')\right] + \mathbb{E}\left[\|X - X'\|_p\right] \mathbb{E}\left[L(Y, Y')\right] \\ & - 2\mathbb{E}\left[\|X - X'\|_p L(Y, Y'')\right] \\ =& - \mathbb{E}\left[\left(Y - \mathbb{E}[Y]\right)\left(Y' - \mathbb{E}[Y']\right)\|X - X'\|_p\right] \end{aligned} \tag{4.19}$$

being (X', Y') and (X'', Y'') iid copies of (X, Y) and $L(y, y') = (y - y')^2/2$.

In this case, the first formulation of equation (4.19) is similar to the one associated with the DC coefficient given in (4.9), just changing $\|Y - Y'\|_q$ by $L(Y, Y')$ respectively. This establishes another link between the MDD and the DC coefficients. The reader is referred to the paper of Park et al. (2015) for a more in-depth analysis of the connection between MDD and DC coefficients.

Taking a sample of $i = 1, \ldots, n$ iid observations $(\mathbf{X_n}, \mathbf{Y_n}) = \{(X_i, Y_i), i = 1, \ldots, n\}$ from the joint distribution of $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, it is defined now A_{il} as in (4.11) and $B_{il} = b_{il} - \bar{b}_{i\cdot} - \bar{b}_{\cdot l} + \bar{b}_{\cdot\cdot}$, where $b_{il} = |Y_i - Y_l|^2/2$, $\bar{b}_{i\cdot} = \frac{1}{n}\sum_{l=1}^n b_{il}$, $\bar{b}_{\cdot l} = \frac{1}{n}\sum_{i=1}^n b_{il}$ and $\bar{b}_{\cdot\cdot} = \frac{1}{n^2}\sum_{i,l=1}^n b_{il}$ for $i, l = 1, \ldots, n$. The empirical estimator of $MDD^2(Y|_X)$, i.e. the sample martingale difference divergence version, can be defined as the nonnegative number

$$MDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}}) = \frac{1}{n^2}\sum_{i,l=1}^n A_{il}B_{il} \tag{4.20}$$

and its associated sample martingale difference correlation coefficient version is given by

$$MDC_n^2(\mathbf{Y_n}|_{\mathbf{X_n}}) = \begin{cases} \frac{MDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}})}{\sqrt{\mathrm{var}_n^2(\mathbf{Y_n})\mathcal{V}_n^2(\mathbf{X_n})}}, & \mathrm{var}_n^2(\mathbf{Y_n})\mathcal{V}_n^2(\mathbf{X_n}) > 0, \\ 0, & \mathrm{var}_n^2(\mathbf{Y_n})\mathcal{V}_n^2(\mathbf{X_n}) = 0, \end{cases} \tag{4.21}$$

where $\mathrm{var}_n(\mathbf{Y_n}) = \frac{1}{n}\sum_{i=1}^n (Y_i - \bar{Y})^2$ with $\bar{Y} = \frac{1}{n}\sum_{i=1}^n Y_i$, and $\mathcal{V}_n^2(\mathbf{X_n})$ is defined in (4.13).

If $\mathbb{E}\left[\|X\|_p + |Y|^2\right] < \infty$, it is guaranteed that both estimators, $MDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}})$ and $MDC_n^2(\mathbf{Y_n}|_{\mathbf{X_n}})$, converge to their population versions displayed in (4.16) and (4.17), respectively, in an almost sure way. Proof of these results can be found in Shao and Zhang (2014). Furthermore, under the null hypothesis of independence in mean, it is verified that $nMDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}}) \longrightarrow \|\Gamma(t)\|^2$ in distribution when $n \to \infty$, where $\Gamma(\cdot)$ is a Gaussian process. Additionally, if $\mathbb{E}[Y^2|_X] = \mathbb{E}[Y^2]$ is also guaranteed, $nMDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}})/S_n \longrightarrow Q$ in distribution when $n \to \infty$, where Q is a nonnegative quadratic form of centered Gaussian random variable with $\mathbb{E}[Q] = 1$ and $S_n = \frac{1}{n^2}\sum_i \sum_l \|X_i - X_l\|_p \frac{1}{n}\sum_i (Y_i - \bar{Y}_n)^2$. In contrast, if the null hypothesis is not verified, one has that $nMDD_n^2(\mathbf{Y_n}|_{\mathbf{X_n}})/S_n \longrightarrow \infty$ in probability when $n \to \infty$. The reader is referred to Shao and Zhang (2014) for more details. Even though we know the asymptotic distribution for both H_0 and H_1 hypotheses, resampling

procedures can be used in practice to calibrate the distribution of the test statistic. This can be especially useful for small sample sizes.

As a result, using the estimators of the MDD or MDC introduced in (4.20) and (4.21), respectively, it is possible to perform covariates selection in regression models, specifying which covariates are the relevant ones. An example of this is the work of Shao and Zhang (2014). They propose a screening procedure sorting out the covariates' relevance in terms of the regressor function explanation, i.e. based on $\mathbb{E}\left[Y|_X\right]$ explanation, and then a proper cut-off is established to detect the significative covariates. Authors make use of the MDC criteria to measure covariates' relevance and establish an order. A different approach is introduced in Zhang et al. (2018), performing covariates selection in terms of causality. A statistic based on the MDD ideas is proposed to test the null hypothesis of $H_0 : \mathbb{E}\left[Y|_{X_j}\right] = \mathbb{E}[Y]$ almost surely for all $j = 1 \ldots, p$. A wild bootstrap scheme is also provided to approximate the statistics distribution.

All these ideas can be transferred to GoF testing as well. An example is the work of Su and Zheng (2017). These authors test the null hypothesis of $H_0 : \mathbb{P}\left(\mathbb{E}\left[Y|_X\right] = g(X, \beta)\right) = 1$ for some $\beta \in \mathcal{B}$, being $\mathcal{B}$ the parameter space and assuming $Y = g(X, \beta) + \varepsilon$, with $g(\cdot)$ a known function. The MDD is applied making use of the covariates and the residuals calculated under the null hypothesis. Calibration of the test is again done using a wild bootstrap. A similar, but broader approach, is also provided by Teran Hidalgo et al. (2018) making use of HSIC techniques.

Similar to the DC case, another extension of the MDD use is the implementation of partial tests. Specifically, Park et al. (2015) proposed the partial martingale difference divergence (pMDD) coefficient and its correlation analogous. These new coefficients measure the departure from mean independence of two random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ once the effect of a third one, $Z \in \mathbb{R}^r$, has been removed. With some abuse of notation, the resulting mean independence test can be written as the null hypothesis $H_0 : \mathbb{E}\left[\mathcal{P}_{Z^\perp}(Y)|_{\mathcal{P}_{Z^\perp}(X)}\right] = \mathbb{E}[\mathcal{P}_{Z^\perp}(Y)]$. Thus, this null hypothesis of mean independence will be accepted when these coefficients take the null value, i.e. pMDD=0.

Finally, it is interesting to mention that the MDD inherits some of the problems of the DC coefficient. These are related to the high computational cost required and the bias of the MDD and MDC estimators displayed in (4.20) and (4.21), respectively. As proposed by Park et al. (2015) or Zhang et al. (2018), the bias can be corrected by applying $\mathcal{U}$-centered ideas. The resulting new unbiased estimator for the MDD coefficient is introduced later in Section 5.2. Furthermore, Park et al. (2015) generalize the biased and unbiased versions of the MDD and MDC coefficients, by allowing $Y \in \mathbb{R}^q$ where $q \geq 1$.

4.2.3 Conditional distance covariance (CDC)

The CDC is a different coefficient of dependence introduced by Wang et al. (2015). This term measures the dependence of two random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ conditioned to a third one, $Z \in \mathbb{R}^r$, resulting in a conditional version of the DC coefficient introduced above in Section 4.2.1. For this aim, conditional characteristic functions are employed, and ideas of the DC coefficient are adapted to the conditional framework. As a result, this

problem translates now into testing

$$H_0 : X \perp_{|Z} Y \quad \text{vs.} \quad H_1 : X \not\perp_{|Z} Y \tag{4.22}$$

where $X \perp_{|Z} Y$ denotes independence of X and Y conditioned to Z.

Using similar DC arguments, it is possible to rewrite (4.22) in terms of conditional characteristic functions. This results in the new test given by

$$H_0 : \varphi_{X,Y|Z} = \varphi_{X|Z}\varphi_{Y|Z} \quad \text{vs.} \quad H_1 : \varphi_{X,Y|Z} \neq \varphi_{X|Z}\varphi_{Y|Z}, \tag{4.23}$$

where $\varphi_{X,Y|Z}$, $\varphi_{X|Z}$ and $\varphi_{Y|Z}$ are the joint and marginal conditional characteristic functions.

Then, the CDC with finite first moments given Z ($\mathbb{E}[\|X\|_p + \|Y\|_q|Z] < \infty$), is defined as the square root of

$$\begin{aligned} CDC^2(X,Y|_Z) =& \|\varphi_{X,Y|Z}(t,s) - \varphi_{X|Z}(t)\varphi_{Y|Z}(s)\|^2 \\ =& \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|\varphi_{X,Y|Z}(t,s) - \varphi_{X|Z}(t)\varphi_{Y|Z}(s)|^2}{\|t\|_p^{p+1}\|s\|_q^{q+1}} dt\, ds \end{aligned} \tag{4.24}$$

being $\|\cdot\|$ the weighted norm defined in Section 4.2.1 and c_p as well as c_q the constants defined in (4.5).

Similarly, the conditional distance variance is the square root of

$$CDC^2(X|_Z) = CDC^2(X,X|_Z) = \|\varphi_{X,X|Z}(t,s) - \varphi_{X|Z}(t)\varphi_{X|Z}(s)\|^2.$$

The CDC coefficient, defined in (4.24), has analogous properties to the unconditional version of (4.6). Particularly, it is verified that $CDC(X,Y|_Z) = 0$ if and only if X and Y are conditionally independent given Z.

Following an argument similar to that of the DC case, the conditional distance correlation (CDCor) can be defined as the square root of

$$CDCor^2(X,Y|_Z) = \begin{cases} \frac{CDC^2(X,Y|_Z)}{\sqrt{CDC^2(X|_Z)CDC^2(Y|_Z)}}, & CDC^2(X|_Z)CDC^2(Y|_Z) > 0, \\ 0, & CDC^2(X|_Z)CDC^2(Y|_Z) = 0. \end{cases} \tag{4.25}$$

and this verifies that $0 \leq CDCor(X,Y|_Z) \leq 1$ and $CDCor(X,Y|_Z) = 0$ if and only if X and Y are conditionally independent given Z.

In order to construct an estimator of $CDC^2(X,Y|_Z)$, the empirical characteristic functions conditioned to Z can be plugged in (4.24). Note that, for the estimation of conditional characteristic functions, one needs to resort to some kind of smoothing techniques like, for example, kernel-type estimators. We refer the reader to Wang et al. (2015) for more details. Thus, given $W_i = (X_i, Y_i, Z_i)$, $i = 1, \ldots, n$, an iid sample from a random vector $W = (X,Y,Z) \in \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^r$, denote by $\mathbf{X_n} = \{X_1, \ldots, X_n\}$, $\mathbf{Y_n} = \{Y_1, \ldots, Y_n\}$, $\mathbf{Z_n} = \{Z_1, \ldots, Z_n\}$, and $\mathbf{W_n} = (\mathbf{X_n}, \mathbf{Y_n}, \mathbf{Z_n})$. As a result, the sample

conditional distance covariance $CDC_n(\mathbf{X_n}, \mathbf{Y_n}|\mathbf{Z_n})$ is the square root of

$$\widetilde{CDC}_n^2(\mathbf{X_n}, \mathbf{Y_n}|\mathbf{Z_n}) = \|\varphi^n_{X,Y|_Z}(t,s) - \varphi^n_{X|_Z}(t)\varphi^n_{Y|_Z}(s)\|^2, \tag{4.26}$$

where $\varphi^n_{X,Y|_Z}$, $\varphi^n_{X|_Z}$ and $\varphi^n_{Y|_Z}$ are the corresponding empirical conditional characteristic functions for (X,Y), X and Y, respectively.

Following Wang et al. (2015), let $d_{ijkl} = \left(a^X_{ij} + a^X_{kl} - a^X_{ik} - a^X_{jl}\right)\left(b^Y_{ij} + b^Y_{kl} - b^Y_{ik} - b^Y_{jl}\right)$ and $d^S_{ijkl} = d_{ijkl} + d_{ijlk} + d_{ilkj}$ for $i,j,k,l = 1,\ldots,n$, where a_{ij} and b_{ij} are defined as in (4.11), and Z_1, Z_2, Z_3 and Z_4 are iid copies of Z. Then, it is verified that

$$CDC^2(X,Y|_{Z=z}) = \frac{1}{12}\mathbb{E}[d^S_{1234}|_{Z_1=z,Z_2=z,Z_3=z,Z_4=z}].$$

In consequence, the conditional dependence coefficients can be estimated by applying kernel regression smoothing ideas to the above expectation. This will result in a $\mathcal{V}$-process. Using these ideas, the resulting sample conditional distance covariance is defined as the square root of

$$CDC^2_n(\mathbf{W_n}|_Z) = CDC^2_n(\mathbf{X_n}, \mathbf{Y_n}, \mathbf{Z_n}|_Z) = \frac{1}{n^4}\sum_{ijkl}\Psi_n(W_i, W_j, W_k, W_l; Z) \tag{4.27}$$

where Ψ_n is the symmetric random kernel of degree 4 defined in Schick (1997):

$$\Psi_n(W_i, W_j, W_k, W_l; Z) = \frac{n^4\Phi_i(Z)\Phi_j(Z)\Phi_k(Z)\Phi_l(Z)}{12\Phi^4(Z)}d^S_{ijkl},$$

where $\Phi_i(Z) = K_H(Z - Z_i)$ and $\Phi(Z) = \sum_{i=1}^n \Phi_i(Z)$, being K a kernel function and H a bandwidth matrix r-dimensional.

Besides, letting $\mathbf{W_{X_n}} = (\mathbf{X_n}, \mathbf{X_n}, \mathbf{Z_n})$ and $\mathbf{W_{Y_n}} = (\mathbf{Y_n}, \mathbf{Y_n}, \mathbf{Z_n})$, the sample conditional distance correlation can be analogously defined as the square root of

$$CDCor^2_n(\mathbf{W_n}|_Z) = \begin{cases} \frac{CDC^2_n(\mathbf{W_n}|_Z)}{\sqrt{CDC^2_n(\mathbf{W_{X_n}}|_Z)CDC^2_n(\mathbf{W_{Y_n}}|_Z)}}, & CDC^2_n(\mathbf{W_{X_n}}|_Z)CDC^2_n(\mathbf{W_{Y_n}}|_Z) > 0, \\ 0, & CDC^2_n(\mathbf{W_{X_n}}|_Z)CDC^2_n(\mathbf{W_{Y_n}}|_Z) = 0. \end{cases}$$

It is verified that $\widetilde{CDC}^2_n(\mathbf{W_n}|_Z) = CDC^2_n(\mathbf{W_n}|_Z)$ for a given sample $\mathbf{W_n} = \{W_1,\ldots,W_n\}$ from the joint distribution of (X,Y,Z). In addition, if $\mathbb{E}\left[\|X\|_p + \|Y\|_q|_Z\right] < \infty$ and $\Phi(Z)/n$ is a consistent density function estimator of Z, then $CDC^2_n(\mathbf{W_n}|_Z) \longrightarrow CDC^2(X,Y|_Z)$ in probability for each value of Z as $n \to \infty$. See Wang et al. (2015) for more details and properties of $CDC^2_n(\mathbf{W_n}|_Z)$.

Summing up, the CDC coefficient can be employed to perform covariates selection conditioned to the Z term. Some examples are the works of Wang et al. (2015), Song et al. (2020), or Lu and Lin (2020), to say a few. Wang et al. (2015) use the CDC to perform the conditional independence test displayed in (4.22), applying conditioned covariates selection. In particular, a statistic based on the CDC coefficient is defined, and a test

is implemented calibrating this utilizing a local bootstrap. Other procedures related to screening techniques for conditional dependence are the recent works of Song et al. (2020) as well as Lu and Lin (2020). The first one adapts the ideas of Liu et al. (2014) using the CDCor to specify significant covariates for general varying-coefficient regression models. All covariates are sorted out based on their CDCor values, and then a threshold is applied. In contrast, Lu and Lin (2020) start selecting an initial set of covariates, and measure the relevance of the remaining terms conditioned to this subset. For this aim, they use the CDCor, resulting in the CDC-SIS (conditional distance correlation sure independence screening) algorithm.

Similar to the DC or the MDD coefficients, the CDC term suffers from a high-computational cost. Moreover, the estimator displayed in (4.27) is biased. Nevertheless, an unbiased version of the CDC can be defined analogously by applying ideas of $\mathcal{U}$-processes theory. This last is introduced in Wang et al. (2015) and has similar properties as the ones exposed for (4.27).

Eventually, it is interesting to note that, as far as we know, there is still no equivalence of the CDC criteria for the HSIC techniques. Furthermore, partial tests for conditional independence have not yet been derived using CDC insights.

METHOD	H_0 :
HSIC Gretton et al. (2005)	$X \perp Y \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q$
DC Székely et al. (2007)	$X \perp Y \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q$
pDC Székely and Rizzo (2014)	$\mathcal{P}_{Z^\perp}(X) \perp \mathcal{P}_{Z^\perp}(Y) \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q, Z \in \mathbb{R}^r$
MDD Shao and Zhang (2014), Park et al. (2015)	$\mathbb{E}[Y\|_X] = \mathbb{E}[Y] \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q$
pMDD Park et al. (2015)	$\mathbb{E}\left[\mathcal{P}_{Z^\perp}(Y)\|_{\mathcal{P}_{Z^\perp}(X)}\right] = \mathbb{E}[\mathcal{P}_{Z^\perp}(Y)] \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q, Z \in \mathbb{R}^r$
CDC Wang et al. (2015)	$X \perp_{\|Z} Y \quad X \in \mathbb{R}^p, Y \in \mathbb{R}^q, Z \in \mathbb{R}^r$

Table 4.1: Comparison of the independence tests performed by the Hilbert-Schmidt independence criterion (HSIC), distance correlation (DC), partial distance correlation (pDC), martingale difference divergence (MDD), partial martingale difference divergence (pMDD) and conditional distance correlation (CDC). Here, $\perp$ denotes orthogonality/independence, $\perp_{|Z}$ independence conditioned to Z and $\mathcal{P}_{Z^\perp}(X)$ as well as $\mathcal{P}_{Z^\perp}(Z)$ the orthogonal projection of the $\mathcal{U}$-centered distance matrix of X, respectively Y, onto $Z^\perp$, being this last the orthogonal space generated by the $\mathcal{U}$-centered distance matrix of Z.

4.3 Application in complex models

All of the novel dependence measures mentioned throughout Section 4.2 allow for independence tests of a different nature. These apply to random vectors of arbitrary dimensions, and no requirement of the type $n > p$ is needed. Thus, the resulting coefficients can be employed to perform different covariates selection techniques for regression models without previous assumptions about the regressor function. As a result, no preliminary estimation of the model is required. This last contradicts other approaches designed for the high dimensional framework. An example is penalization techniques, where some structure (linear, additive, etc.) needs to be assumed. See Chapters 2 and 3 for a study of their implementation under the linearity assumption. In addition, these coefficients protect against the curse of dimensionality. See Section 1.2 for more details.

Furthermore, these coefficients can be adapted to more complex frameworks, such as metric spaces or situations when the variables are not necessarily vectors. An extension of the DC results in Euclidean spaces to general metric ones can be found in Lyons (2013). In particular, these ideas are extended to strong negative type metric spaces, collecting the case of separable Hilbert spaces. Another related work is the one of Jansen (2021), which expands the developments of Lyons (2013) to all Hilbert spaces. Regarding this last, new dependence measures for functional data have been developed for specification testing. Lee et al. (2020) assume a Hilbert framework and extend the MDD vectorial coefficient to this functional case to apply significance tests. In contrast, Lai et al. (2020) use the DC adaptation to semimetric spaces of negative type to perform specification tests under the linearity assumption. Other recent works, such as the ones of Hu et al. (2020) or Zhao et al. (2022), adapt the MDD to the functional context for independence testing. Additional and interesting applications of the distance-based dependence measures are their use in quantile regression (see Xu and Chen (2020), Zhang et al. (2018)), in time series (see Edelmann et al. (2019), Davis et al. (2018), Dehling et al. (2020) or Lee and Shao (2018)) or in cure models from the Survival Analysis (see Zhang and Cui (2021), Chen (2021) and Edelmann et al. (2022)), to say a few.

Therefore, these novel dependence coefficients result in quite versatile tools that allow different covariates selection procedures to be performed in complex models. Moreover, different types of dependence structures are considered in terms of the employed coefficient. These are related to independence testing, using the DC, conditional mean independence, employing the MDD, or conditional independence tests resorting to the CDC coefficient.

As was mentioned at the beginning of this chapter, it is desirable to consider more complex formulations for the regression model structure than the ones presented in Chapter 1. Specifically, the high dimensional problem of covariates selection in the linear case, especially for the $p > n$ context, is treated along Chapters 2 and 3 using regularization techniques. A natural extension of the linear model is the varying-coefficient regression model of Hastie and Tibshirani (1993). Assuming that variables $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$ are centered without loss of generality, this is given by the expression

$$Y = \beta(t)X + \varepsilon \tag{4.28}$$

being t a variable taking values in a domain $\mathcal{D} \in \mathbb{R}$.

Furthermore, more complexity can be added to the above formulation (4.28). In particular, one can consider Y, X, and ε as functional data. A particular case is to assume that these three terms are functions of the same argument $t \in \mathcal{D}$ and that their relation is concurrent or point by point, given place to the model known as the functional concurrent model (FCM). Besides, not only linearity but other structures can be considered as well. In general, this results in the functional concurrent model formulation given by

$$Y(t) = m(t, X(t)) + \varepsilon(t), \tag{4.29}$$

where $m(\cdot)$ is the unknown regressor function. Particular cases of the (4.29) formulation are the linear structure taking $m(t, X(t)) = \beta(t)X(t)$, or the additive formulation given by $m(t, X(t)) = \sum_{i=1}^{p} F_j(X_j(t))$. As a result, the concurrent model of (4.29) also collects the varying-coefficients model introduced in (4.28) taking $Y(t) = Y$, $X(t) = X$ and $\varepsilon(t) = \varepsilon$ for all $t \in \mathcal{D}$. Besides, the model (4.29) is just a functional extension of the model displayed in equation (1.1) of Chapter 1. The same applies to its linear and additive versions concerning models studied in Sections 1.1.1 and 1.1.2, respectively.

Next, using the novel dependence coefficients introduced above, we develop new covariates selection approaches for the FCM. Specifically, new significance tests are introduced for the synchronous FCM in Chapter 5 and its asynchronous version in Chapter 6. For this purpose, the MDD and the CDC coefficients are employed, respectively.

Chapter 5

New significance tests for the synchronous functional concurrent model based on the MDD coefficient

In Chapter 4, novel dependence distances coefficients are proposed to test covariates' significance in complex models without previous estimation of the regression function. In this chapter, a novel implementation is proposed to test the significance of the synchronous version of an additive functional concurrent model (FCM). This new approach is based on an unbiased version of the MDD coefficient introduced in Section 4.2.2. The synchronous FCM is introduced in Section 5.1 jointly with a motivation for the need for dimensionality reduction. Subsequently, the unbiased version of the MDD is introduced in Section 5.2. Next, in Section 5.3, the new dependence tests are proposed. Theoretical justification of their good behavior is given and a bootstrap scheme is proposed to calculate its p-values in practice. A simulation study to test their performance is presented in Section 5.4, jointly with a comparison involving Ghosal and Maity (2022a) and Kim et al. (2018) competitors. Next, the proposed tests are applied in three real data sets in Section 5.5. Eventually, some discussion arises in Section 5.6. The contents of this chapter are collected in Freijeiro-González et al. (2022b).

5.1 The functional concurrent model (FCM): the need for significance tests

A general concurrent model is a regression model where the response $Y = (Y_1, \ldots, Y_q) \in \mathbb{R}^q$, for $q \geq 1$, and $p \geq 1$ covariates $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$ are all functions of the same argument $t \in \mathcal{D}$, and the influence is concurrent, simultaneous or point-wise in the sense that X is assumed to only influence $Y(t)$ through its value $X(t) = (X_1(t), \ldots, X_p(t)) \in \mathbb{R}^p$ at time t by the relation

$$Y(t) = m(t, X(t)) + \varepsilon(t), \tag{5.1}$$

where $m(\cdot)$ is an unknown function collecting the $\mathbb{E}\left[Y(t)|_{X(t)}\right]$ information and $\varepsilon(t)$ is the model error. This last is a process which is assumed to have mean zero, independent of X and with covariance function $\Omega(s,t) = \mathbb{C}\left[\varepsilon(s), \varepsilon(t)\right]$, being $\mathbb{C}[\cdot,\cdot]$ the covariance operator.

The concurrent model displayed in (5.1) is in the middle of longitudinal and functional data. This classification depends on the number of observed time instants in the t domain $\mathcal{D}$. When this number is dense enough, the sample data can be treated as curves, translating into a functional data framework. Otherwise, if time instants are spaced respective to the t domain and not dense, a longitudinal framework will be more suitable. Determining the inflection point between both situations is still an open problem. For a discussion on this topic, we refer the reader to the work of Wang et al. (2017).

There are plenty of contexts where the (5.1) formulation arises both in functional or longitudinal framework form. The functional concurrent model can be employed in any situation where data can be monitored, like in health, environmental or financial issues among others. Some examples can be seen in works such as the ones of Xue and Zhu (2007) or Jiang et al. (2011) for the longitudinal data context. They perform epidemiology studies of AIDS data sets. Other real data examples in medicine can be found in Goldsmith and Schwartz (2017) or Wang et al. (2017). Goldsmith and Schwartz (2017) perform a blood pressure study to detect masked hypertension. For their part, authors in Wang et al. (2017) use the concurrent model in a data study of flu prevalence in the United States. Furthermore, they model Alzheimer's disease progression using brain neuroimaging data. More examples of health and nutrition are displayed in Kim et al. (2018) and Ghosal and Maity (2022a). They perform studies related to gait deficiency, dietary calcium absorption, and the relation between child mortality and financial power in different countries. Examples in the environmental field are collected in works such as Zhang et al. (2011) or Ospína-Galindez et al. (2019). These studies are based on describing forest nitrogen cycling and modeling the rainfall ground, respectively. A completely different example is the work of Ghosal and Maity (2022b), where casual bike rentals in Washington, D.C., are concurrently explained using meteorological variables. This extensive list of examples reveals that the concurrent model is a very transversal and wide-employed tool.

An inconvenience of the concurrent model general formulation, displayed in (5.1), is that the $m(\cdot)$ structure is quite difficult to be estimated in practice. For this reason, it is common to consider some assumptions about its form. In the literature, it is quite common to assume linearity, which translates into taking $m(t, X(t)) = \beta(t)X(t)$ in (5.1), and work under this premise. However, this assumption can be quite restrictive in practice. Thus, more general structures are needed to model real examples properly. This last results in a gain in flexibility but adds complexity to the estimation process. Maity (2017) discusses the effort made for estimating different concurrent model structures. This paper highlights that more information is needed to correctly estimate the function $m(\cdot)$. In conclusion, it is crucial to guarantee that there exists useful information on the covariates X to model the behavior of Y as a preliminary step. Therefore, covariates selection algorithms for the concurrent model are of interest to avoid irrelevant covariates entering the model and simplify the estimation process.

As a result, the first step to assure the veracity of the model structure displayed in (5.1) is to verify if all p covariates $\{X_1(t), \ldots, X_p(t)\}$ contribute to the correct explanation of $Y(t)$, or some can be excluded from the model formulation.

To the best of our knowledge, there is no literature on significance tests for the additive concurrent model that avoids previous model estimation or extra tuning parameters. We refer to Wang et al. (2017) and Ghosal and Maity (2022a) for these in the linear formulation. They both propose effect tests over the $\beta(t)$ function making use of the empirical likelihood. Thus, once the model parameters are estimated in the linear framework, the authors provide tools to test if all p covariates are relevant or, on the contrary, if some can be excluded from the model. Nevertheless, a suitable effects estimation involves several tuning parameters

and the linearity hypothesis. These are necessary to guarantee the adequate performance of the cited procedures. In terms of the $\beta(t)$ structure estimation, different approaches arise. For example, Wang et al. (2017) propose using a local linear estimator, which depends on a proper bandwidth selection. In contrast, Ghosal and Maity (2022a) employs an expansion into a finite number of elements of a functional basis. This expansion requires the number of considered terms selection. In addition, this last procedure needs to estimate the error model structure. This process translates into an additional functional basis representation and estimation of extra parameters. All this translates into difficulties in the estimation process, even if the linear hypothesis can be accepted. Currently, Kim et al. (2018) developed a new significance test in a more general framework to alleviate the linear hypothesis assumption: additive effects are considered in (5.1). This work employs F-test techniques over a functional basis representation of the additive effects to detect relevant covariates. Again, this technique depends on an adequate preliminary estimation of the model effects to be able to select relevant covariates by applying significance tests. However, the correct selection of the number of basis functions for each considered covariate/effect representation is still an open problem. These quantities play the role of tuning parameters. Furthermore, a proper error variance estimation is needed to standardize the covariates as an initial step. As this structure is unknown in practice, Kim et al. (2018) assumes that this can be decomposed as a sum of two terms. The first one is a zero-mean smooth stochastic process, and the second term is a zero-mean white noise measurement error with variance σ^2, resulting in the autocovariance function $\Omega(s,t) = \Sigma(s,t) + \sigma^2\mathbb{I}\{s = t\}$. Nevertheless, this assumption can be restrictive in practice. In consequence, significance tests without any assumption in the model structure and no necessity of a preliminary estimation step are desirable.

Other procedures for covariates selection with a different methodology are the Bayesian selectors and the penalization techniques used in the concurrent model estimation process. We can highlight the works of Goldsmith and Schwartz (2017) or Ghosal et al. (2020) in the linear formulation and the one of Ghosal and Maity (2022b) for general additive effects. While Goldsmith and Schwartz (2017) uses the spike-and-slab regression covariates selection procedure, Ghosal et al. (2020) and Goldsmith and Schwartz (2017) implement penalizations based on LASSO (Tibshirani (1996)), SCAD (Fan and Li (2001)), MCP (Zhang (2010)) or its grouped versions (Yuan and Lin (2006)), respectively. As a result, the selection of covariates is implemented together with estimation. Nevertheless, some tuning parameters are needed in all these methodologies: it is necessary to determine the number of basis functions to represent the effects in all of them, jointly with prior parameters, in case of the spike-and-slab regression, or the amount of penalization otherwise. As a result, the estimation of tuning parameters applies in these approaches as well.

In this chapter, we deal with this concern by bridging a gap for significance tests without previous model estimation. The new proposal for specification testing can assess the usefulness of a vector X for modeling the expectation of the Y vector in a pretty general formulation. Besides, this approach avoids extra tuning parameters estimation, as well as the need to model the error structure. For this aim, we propose a novel statistic

for the concurrent model based on the martingale difference divergence ideas of Shao and Zhang (2014) (see Section 4.2.2) is proposed. As a result, this approach tests the effect of the covariates in the explanation of Y no matter the underlying form of $m(\cdot)$ while assuming additive effects in a synchronous FCM. This is a functional extension of the additive model introduced in Section 1.1.2 of Chapter 1 for the vectorial framework.

5.1.1 The synchronous FCM

As its name suggests, in a synchronous FCM it is assumed that all points of the curves are observed in the same time instants. However, the preliminary assumption that all trajectories are completely observed can be quite restrictive in practice. In Section 5.1.2 it is shown how to adapt this requirement to contexts where some points are missed, adjusting the procedure to more realistic situations.

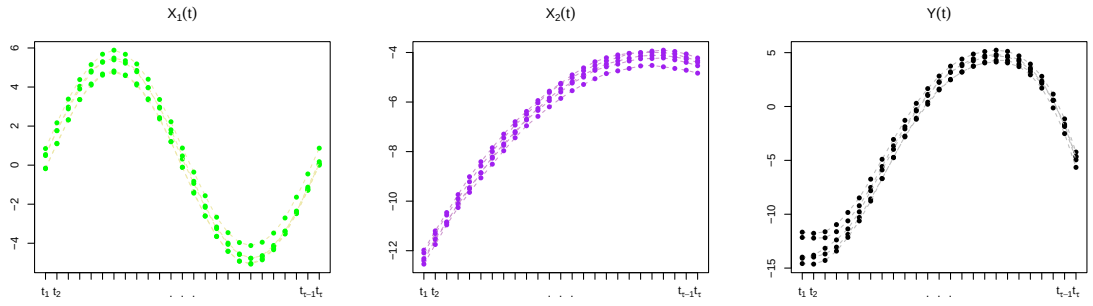


Figure 5.1: Example of a sample of five curves measured at same time instants $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$ considering $p = 2$ covariates ($X_1(t)$ and $X_2(t)$) to explain $Y(t)$. Filled points simulate a total of $n_u = 3$ observed points at each instant t_u.

Thus, a total of $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$ time instants are considered and there are n_u observed samples, each of them of the form $\{Y_{i_u}(t_u), X_{i_u}(t_u)\}_{i_u=1}^{n_u}$. As mentioned before, assuming all curves observed at the same time instants translates into $n_u = n$ for all $u = 1, \ldots, \mathcal{T}$. Then, we have a sample of the form $(\mathbf{Y_n(t)}, \mathbf{X_n(t)}) = \{(Y_i(t_u), X_i(t_u)),\ u = 1, \ldots, \mathcal{T}\}_{i=1}^{n}$. A graphic example of our current situation considering $q = 1$ and $p = 2$ covariates in a synchronous FCM is displayed in Figure 5.1.

5.1.2 Some missing points in curves trajectories

As mentioned above, the assumption that we observe the complete trajectories of the curves can be quite restrictive in practice. In contrast, it is common to find some missing points. Then, for each time point t_u there are $1 \leq n_u \leq n$ observed samples of the form $\{Y_{i_u}(t_u), X_{i_u}(t_u)\}_{i_u=1}^{n_u}$. A graphic example for the case considering $q = 1$ and $p = 2$ covariates is displayed in the first row of Figure 5.2. In this example, we have $n = 5$ curves and a different number of observations. For instance, there are $n_1 = 4$ points for t_1.

In this context, the method proposed in Section 5.3 can not be applied directly. This is because it is not verified $n_u = n$ for all $u = 1, \ldots, \mathcal{T}$. However, we can solve this problem

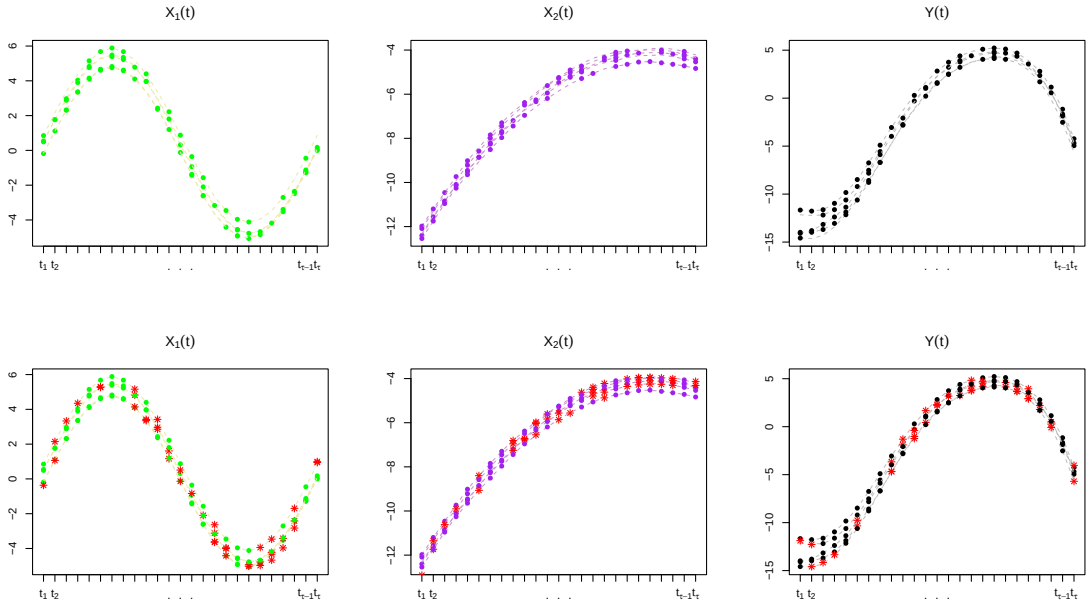


Figure 5.2: First row: sample of five curves measured at different time instants $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$ considering $p = 2$ covariates ($X_1(t)$ and $X_2(t)$) to explain $Y(t)$. Second row: same example adding the recovered points by means of splines interpolation. Filled dots ($\bullet$) represent the n_u observed points at each instant t_u and asterisks ($*$) the recovered ones.

by estimating the missing curve values when is possible. This option translates into a recovering of the whole curve trajectories on the grid $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$, verifying now that $n_u = n$ for all $u = 1, \ldots, \mathcal{T}$.

A simple but efficient idea is to recover the complete trajectory of the curves using some interpolating method with enough flexibility. For example, making use of cubic spline interpolation ideas for each of the $1, \ldots, n$ curves. Results of this recovery for the example introduced above in Section 5.1.1 are displayed in the second row of Figure 5.2. In this case, the `spline` function of the `stats` library of the R software (R Core Team (2019)) has been employed.

In addition, other approaches for recovering the missing points are also available. Next, we propose one based on functional basis representation following the guidelines of Kim et al. (2018), Ghosal et al. (2020), and Ghosal and Maity (2022b). If it is possible to assume that the total number of time observations $\bigcup_{u=1}^{\mathcal{T}} t_u$ is dense in $\mathcal{D}$, then the eigenvalues and eigenfunctions corresponding to the original curves can be estimated using functional principal component analysis (see Yao et al. (2005)). We refer to Yao et al. (2005) for more details about the procedure. As a result, one can get the estimated trajectory $\hat{X}_{ij}(\cdot)$ of the true curves $X_{ij}(\cdot)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, given by $\hat{X}_{ij}(t) = \hat{\mu}_j(t) + \sum_{q=1}^{Q} \hat{\zeta}_{iqj} \hat{\Psi}_{qj}(t)$. Here, Q denotes the number of considered eigenfunctions, which can be chosen using a predetermined percentage of explained variance criterion. Consequently, it is possible to recover the value of $X_1(\cdot), \ldots, X_p(\cdot)$ on all grid $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$. In the same way, the values

of $Y_1(\cdot), \ldots, Y_q(\cdot)$ can also be recovered. Thus, it is possible to work again in the context of synchronously measured data. This procedure is implemented in the `fpca.sc` function belonging to the library `refund` of R (see Goldsmith et al. (2021)). For our proposed naive example, we have obtained similar results to the splines interpolation methodology displayed in Figure 5.2. As a result, these are omitted.

5.2 Unbiased MDD

Next, it is presented an unbiased version of the MDD estimator of Shao and Zhang (2014) obtained in Section 4.2.2. This has been introduced by Park et al. (2015) in the vectorial framework considering random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ with $p, q \geq 1$. In particular, their relation with $\mathcal{U}$-statistics is displayed, providing theoretical properties which will be of interest to obtain the asymptotic distribution of the MDD-based statistic obtained in Section 5.3 for the synchronous FCM.

Given a sample of $i = 1, \ldots, n$ iid observations $(\mathbf{X_n}, \mathbf{Y_n}) = \{(X_i, Y_i), i = 1, \ldots, n\}$ from the joint distribution of $(X, Y) \in \mathbb{R}^p \times \mathbb{R}^q$, define $A = (A_{il})_{i,l=1}^n$ and $B = (B_{il})_{i,l=1}^n$ as in Section 4.2.2. Then, following the $\mathcal{U}$-centered ideas of Székely and Rizzo (2014), it is possible to define the $\mathcal{U}$-centered versions of A and B, $\overline{A}$ and $\overline{B}$ respectively, given by

$$\overline{A}_{il} = A_{il} - \frac{1}{n-2}\sum_{q=1}^{n} A_{iq} - \frac{1}{n-2}\sum_{q=1}^{n} A_{ql} + \frac{1}{(n-1)(n-2)}\sum_{q,r=1}^{n} A_{qr}$$

$$\overline{B}_{il} = B_{il} - \frac{1}{n-2}\sum_{q=1}^{n} B_{iq} - \frac{1}{n-2}\sum_{q=1}^{n} B_{ql} + \frac{1}{(n-1)(n-2)}\sum_{q,r=1}^{n} B_{qr}$$

where $A_{il} = \|X_i - X_l\|_p$ and $B_{il} = \|Y_i - Y_l\|_q^2/2$.

As a result, an unbiased estimator for MDD is defined as

$$MDD_n^2(\mathbf{Y_n}|\mathbf{X_n}) = (\overline{A} \cdot \overline{B}) = \frac{1}{n(n-3)}\sum_{i \neq l} \overline{A}_{il}\overline{B}_{il}. \tag{5.2}$$

A proof that $MDD_n^2(\mathbf{Y_n}|\mathbf{X_n})$ is an unbiased estimator for $MDD^2(Y|_X)$ can be found in Section 1.1 of the Supplementary Material of Zhang et al. (2018) for the $q = 1$ case. An extension for the case considering $q \geq 1$ is displayed in the proof of Proposition 3.4 given in Park et al. (2015).

An important characteristic of the $MDD_n^2(\mathbf{Y_n}|\mathbf{X_n})$ unbiased estimator defined in (5.2) is that this is a $\mathcal{U}$-statistic of order four. In fact, with some calculation, it can be proved that

$$MDD_n^2(\mathbf{Y_n}|\mathbf{X_n}) = \frac{1}{\binom{n}{4}}\sum_{i<k<l<r} \phi(Z_i, Z_k, Z_l, Z_r) \tag{5.3}$$

with symmetric kernel function

$$\begin{aligned}\phi(Z_i, Z_k, Z_l, Z_r) &= \frac{1}{4!}\sum_{(s,w,u,v)}^{(i,k,l,r)}(A_{sw}B_{uv} + A_{sw}B_{sw} - 2A_{sw}B_{su}) \\ &= \frac{1}{6}\sum_{s<w,u<v}^{(i,k,l,r)}(A_{sw}B_{uv} + A_{sw}B_{sw}) - \frac{1}{12}\sum_{(s,w,u)}^{(i,k,l,r)} A_{sw}B_{su}\end{aligned}$$

where $Z_i = (X_i, Y_i)$ for $i = 1, \ldots, n$ and the summation is over all permutation of the 4-tuples of indices (i, k, l, r). A guideline about this calculation is provided in Section 1.1 of the Supplementary Material of Zhang et al. (2018).

In view of the (5.3) formulation, one can directly notice that $MDD_n^2(Y|_X)$ is a $\mathcal{U}$-statistic of order four by proper definition. Then, this statistic can be employed to perform the independence in mean test displayed in (4.15). Next, significance tests are proposed for the synchronous FCM to select covariates. An adaptation of the unbiased MDD coefficient displayed in (5.2) is used to obtain a proper statistic.

5.3 Significance tests based on MDD

In this section, new significance tests are proposed for the synchronous FCM using the unbiased MDD coefficient of Zhang et al. (2018) introduced above in Section 5.2. Once a tool to measure conditional mean independence between $Y \in \mathbb{R}$ and a vector $X = (X_1, \ldots, X_p)^\top \in \mathbb{R}^p$ has been provided, these ideas are adapted to the concurrent model case. For this aim, ideas presented in the work of Zhang et al. (2018) in the vectorial framework are adapted.

Taking $D \subset \{1, \ldots, p\}$, the significance problem in the FCM translates into testing

$$\begin{aligned}&H_0\colon \mathbb{E}\left[Y(t)|_{X_D(t)}\right] = \mathbb{E}\left[Y(t)\right] \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \\ &H_1\colon \mathbb{P}\left(\mathbb{E}\left[Y(t)|_{X_D(t)}\right] \neq \mathbb{E}\left[Y(t)\right]\right) > 0 \quad \forall t \in \mathcal{P}\end{aligned}$$

where $X_D(t)$ denotes the subset of $X(t)$ considering only the covariates with index in D, $\mathcal{D} \setminus \mathcal{N}$ is the domain of t minus a null set $\mathcal{N} \subset \mathcal{D}$ and $\mathcal{P} \subset \mathcal{D}$ is a positive measure set.

Quoting Zhang et al. (2018), the above problem is very challenging in practice without assuming any structure of $m(\cdot)$. This drawback is due to the vast class of alternatives targeted, related to growing dimension and nonlinear dependence. To solve this inconvenience, the authors propose testing the nullity of the main effects first, keeping a type of hierarchical order. Then, it is tested if additive and separate effects first enter the model before considering interactive structures. This results in the new test displayed in (5.4).

$$\begin{aligned}&H_0\colon \mathbb{E}\left[Y(t)|_{X_j(t)}\right] = \mathbb{E}\left[Y(t)\right] \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \text{ and } \forall j \in D \\ &H_1\colon \mathbb{P}\left(\mathbb{E}\left[Y(t)|_{X_j(t)}\right] \neq \mathbb{E}\left[Y(t)\right]\right) > 0 \quad \forall t \in \mathcal{P} \text{ and some } j \in D\end{aligned} \tag{5.4}$$

Then, rejection of the null hypothesis of (5.4) automatically implies the rejection of the $H_0 : \mathbb{E}\left[Y(t)|_{X_D(t)}\right] = \mathbb{E}\left[Y(t)\right]$ hypothesis. It is important to highlight that the reciprocal is not always true. In this way, the model (5.1) only makes sense if it is possible to reject the H_0 hypothesis of (5.4). Otherwise, the covariates do not supply relevant information to explain Y. It is notorious that formulation (5.4) collects a wide range of dependence structures between X and Y in terms of additive regression models, where $m\left(t, X(t)\right) = F_1\left(t, X_1(t)\right) + \cdots + F_p\left(t, X_p(t)\right)$. Moreover, it is no need to know the real form of $m(\cdot)$ to determine whether the effect of X is significant.

It is important to notice that one can consider $D = \{1, \ldots, p\}$ to perform (5.4), which translates in testing if all p covariates are relevant, or only a subset $D \subset \{1, \ldots, p\}$ with cardinality $1 \leq d < p$. In this last case, one tests if only a bunch of covariates are relevant, excluding the rest from the model. A special case is to consider $D = \{j\}$ for some $j = 1, \ldots, p$. This approach allows to implement covariates screening with no need to estimate the regressor function. Thus, it is possible to test the effect of every covariate. This results in $j = 1, \ldots, p$ partial tests of the form

$$\begin{aligned} &H_{0j}\colon \mathbb{E}\left[Y(t)|_{X_j(t)}\right] = \mathbb{E}\left[Y(t)\right] \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \\ &H_{aj}\colon \mathbb{P}\left(\mathbb{E}\left[Y(t)|_{X_j(t)}\right] \neq \mathbb{E}\left[Y(t)\right]\right) > 0 \quad \forall t \in \mathcal{P} \end{aligned} \tag{5.5}$$

Thus, one can test if a small subset of $\{1, \ldots, p\}$ is suitable to fit the model or if all covariates need to be considered. As a result, it is possible to avoid noisy covariates entering the model and reduce the problem dimension.

In this way, we want to include all the information provided by the observed time instants $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$ in a new statistic. Besides, as mentioned above, we can be interested in testing dependence not only considering all covariates but a subset $D \subset \{1, \ldots, p\}$. As a result, an integrated dependence test is applied over the complete trajectory, considering the information provided by D. Rewriting (5.4), this gives place to the test

$$\begin{aligned} &H_0\colon \int_{\mathcal{D}\setminus\mathcal{N}} MDD^2(Y(t)|_{X_j(t)})dt = 0 \text{ almost surely for every } j \in D \\ &H_1\colon \mathbb{P}\left(\int_{\mathcal{P}} MDD^2(Y(t)|_{X_j(t)})dt \neq 0\right) > 0 \text{ for some } j \in D \end{aligned} \tag{5.6}$$

To implement the new test introduced in (5.6) a proper estimator of $\int_{\mathcal{D}} MDD^2(Y(t)|_{X_j(t)})dt$ for every $j \in D$ is needed. For this purpose, we propose an integrated statistic based on

$$T_D = \sqrt{\binom{n}{2}} \frac{\sum_{j\in D} \widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})}{\widehat{\widetilde{\mathcal{S}}}_D}, \tag{5.7}$$

being $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = \int_{\mathcal{D}} MDD_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})dt$ and

$$\widehat{\widetilde{\mathcal{S}}}_D^2 = \frac{2}{n(n-1)c_n} \sum_{1\leq k<l\leq n} \sum_{j,j'\in D} \int_{\mathcal{D}} \left(\overline{A}_{kl}(t)\right)_j \left(\overline{A}_{kl}(t)\right)_{j'} \overline{B}_{kl}^2(t)dt \tag{5.8}$$

a suitable variance estimator of $\sum_{j\in D}\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$ with c_n

$$c_n = \frac{(n-3)^4}{(n-1)^4} + \frac{2(n-3)^4}{(n-1)^4(n-2)^3} + \frac{2(n-3)}{(n-1)^4(n-2)^3} \approx \frac{(n-3)^4}{(n-1)^4}. \tag{5.9}$$

See Section 5.3.1 for in-depth details about $\widetilde{\widetilde{\mathcal{S}}}_D^2$ calculation.

The integrated version $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$ remains a $\mathcal{U}$-statistic of order four. This is because, denoting by $Z_{ij}(t) = (X_{ij}(t), Y_i(t))$ and $(\widetilde{A_{sw}B_{uv}})_j = \int_{\mathcal{D}} (A_{sw}(t))_j (B_{uv}(t))_j \, dt$ for all (s,w,u,v), we have that $\phi(Z_{ij}(t), Z_{kj}\widetilde{(t), Z_{lj}}(t), Z_{rj}(t))$ equals to

$$\begin{aligned}
&\int_{\mathcal{D}} \phi(Z_{ij}(t), Z_{kj}(t), Z_{lj}(t), Z_{rj}(t))dt \\
&= \frac{1}{4!} \sum_{(s,w,u,v)}^{(i,k,l,r)} \left\{ \left(\widetilde{A_{sw}B_{uv}}\right)_j + \left(\widetilde{A_{sw}B_{sw}}\right)_j - 2\left(\widetilde{A_{sw}B_{su}}\right)_j \right\} \\
&= \frac{1}{6} \sum_{s<w,u<v}^{(i,k,l,r)} \left\{ \left(\widetilde{A_{sw}B_{uv}}\right)_j + \left(\widetilde{A_{sw}B_{sw}}\right)_j \right\} - \frac{1}{12} \sum_{(s,w,u)}^{(i,k,l,r)} \left(\widetilde{A_{sw}B_{su}}\right)_j
\end{aligned} \tag{5.10}$$

and this remains a measurable and symmetric function. Then, similar to (5.3) argumentation, it is easy to see that it is possible to write

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = \frac{1}{\binom{n}{4}} \sum_{i<k<l<r} \phi(Z_{ij}(t), Z_{kj}\widetilde{(t), Z_{lj}}(t), Z_{rj}(t))$$

which keeps the structure of a $\mathcal{U}$-statistic of order 4. It can be proved that $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$ is an unbiased estimator of $\widetilde{MDD}^2(Y(t)|_{X_j(t)})$. See Section C.3.1 of the Appendix C.3.

Theorem 5.1. *Under the assumption of H_0 and verifying*

$$\frac{\mathbb{E}\left[G(\widetilde{Z(t), Z'(t)})^2\right]}{\left\{\mathbb{E}\left[H(\widetilde{Z(t), Z'(t)})^2\right]\right\}^2} \longrightarrow 0$$

$$\frac{\mathbb{E}\left[H(\widetilde{Z(t), Z'(t)})^4\right]/n + \mathbb{E}\left[H(\widetilde{Z(t), Z''(t)})^2 H(\widetilde{Z'(t), Z''(t)})^2\right]}{n\left\{\mathbb{E}\left[H(\widetilde{Z(t), Z'(t)})^2\right]\right\}^2} \longrightarrow 0$$

$$\frac{\mathbb{E}\left[\dot{U}(\widetilde{X(t), X''(t)})^2 V(\widetilde{Y(t), Y'(t)})^2\right]}{\widetilde{\mathcal{S}}_D^2} = o(n)$$

$$\frac{\sum_{j,j'\in D}\int_{\mathcal{D}} \mathbb{V}[Y(t)]^2 dcov(X_j(t), X_{j'}(t))^2 dt}{\widetilde{\mathcal{S}}_D^2} = o(n^2)$$

for $\mathbb{V}[\cdot]$ *the variance operator and* $dcov(\cdot,\cdot)$ *the distance covariance, it is guarantee that* $T_D \longrightarrow^d N(0,1)$ *when* $n \longrightarrow \infty$ *and* $\widetilde{\widetilde{\mathcal{S}}}_D^2/\widetilde{\mathcal{S}}_D^2 \longrightarrow^p 1$.

Theorem 5.1 guarantees the asymptotic convergence of the T_D statistic displayed in (5.7) to a normal distribution under some assumptions. Proof of this result is collected in Section C.3.3 of the Appendix C, which makes use of the Hoeffding decomposition for $\mathcal{U}$-statistics carried out in Section C.3.2 of the same document.

One drawback is that the asymptotic convergence of the T_D statistic can be very slow in practice. To solve this issue we approximate the p-value using a wild bootstrap scheme. Its scheme is collected in Algorithm 5.2. The proof of the consistency related to the proposed wild bootstrap procedure and that of the variance estimator for the concurrent model case is omitted due to extension. However, the proof results from plugging the integrated version in that in Section 1.6 of the Supplementary Material of Zhang et al. (2018).

Algorithm 5.2 (Wild bootstrap scheme for global dependence test using MDD)**.**

1. For $u = 1\ldots,\mathcal{T}$:

   1.1. Calculate
   $$(T_u)_D = \sqrt{\binom{n}{2}} \sum_{j\in D} MDD_n^2(Y(t_u)|_{X_j(t_u)}).$$

   1.2. Obtain
   $$(\hat{\mathcal{S}}_u)_D = \sqrt{\frac{2}{n(n-1)c_n} \sum_{1\le k<l\le n} \sum_{j,j'\in D} \left(\overline{A}_{kl}(t_u)\right)_j \left(\overline{A}_{kl}(t_u)\right)_{j'} \overline{B}_{kl}^2(t_u)},$$

   where $\left(\overline{A}_{kl}(t_u)\right)_j$ and $\overline{B}_{kl}(t_u)$ are the $\mathcal{U}$-centered versions of $(A_{kl}(t_u))_j = |X_{kj}(t_u) - X_{lj}(t_u)|$ and $B_{kl}(t_u) = \|Y_k(t_u) - Y_l(t_u)\|_q^2/2$, respectively.

   1.3. Generate the sample $\{e_i\}_{i=1}^n$, where e_i are i.i.d. N(0,1).

   1.4. Define the bootstrap $MDD_n^{*2}(Y(t_u)|_{X_j(t_u)})$ version as
   $$MDD_n^{*2}(Y(t_u)|_{X_j(t_u)}) = \frac{1}{n(n-1)} \sum_{k\neq l} \left(\overline{A}_{kl}(t_u)\right)_j \overline{B}_{kl}(t_u) e_k e_l$$

   1.5. Obtain the bootstrap statistic numerator
   $$(T_u^*)_D = \sqrt{\binom{n}{2}} \sum_{j\in D} MDD_n^{*2}(Y(t_u)|_{X_j(t_u)}).$$

1.6. Calculate the bootstrap variance estimator

$$(\hat{\mathcal{S}}_u^*)_D = \sqrt{\frac{1}{\binom{n}{2}} \sum_{1\leq k<l\leq n} \sum_{j,j'\in D} \left(\overline{A}_{kl}(t_u)\right)_j \left(\overline{A}_{kl}(t_u)\right)_{j'} \overline{B}_{kl}^2(t_u) e_k^2 e_l^2}.$$

1.7. Repeat steps 1.3-1.6 a number B of times obtaining the sets $\{(T_u^*)_D^{(1)}, \ldots, (T_u^*)_D^{(B)}\}$ and $\{(\hat{\mathcal{S}}_u^*)_D^{(1)}, \ldots, (\hat{\mathcal{S}}_u^*)_D^{(B)}\}$.

2. Approximate the sample statistic $(\tilde{E})_D = \int_{\mathcal{D}} (T_t)_D / (\hat{\mathcal{S}}_t)_D dt$ value by means of numerical techniques using $\{(T_1)_D, \ldots, (T_{\mathcal{T}})_D\}$ and $\{(\hat{\mathcal{S}}_1)_D, \ldots, (\hat{\mathcal{S}}_{\mathcal{T}})_D\}$.

3. For every $b = 1, \ldots, B$, approximate the bootstrap statistic value given by $(\tilde{E}^*)_D^{(b)} = \int_{\mathcal{D}} (T_t^*)_D^{(b)} / (\hat{\mathcal{S}}_t^*)_D^{(b)} dt$, by means of numerical techniques using $\{(T_1^*)_D^{(b)}, \ldots, (T_{\mathcal{T}}^*)_D^{(b)}\}$ and $\{(\hat{\mathcal{S}}_1^*)_D^{(b)}, \ldots, (\hat{\mathcal{S}}_{\mathcal{T}}^*)_D^{(b)}\}$.

4. Obtain the bootstrap p-value as $\frac{1}{B} \sum_{b=1}^{B} \mathbb{I}\{(\tilde{E}^*)_D^{(b)} \geq (\tilde{E})_D\}$, where $\mathbb{I}(\cdot)$ is the indicator function.

Moreover, the test is guaranteed to be powerful under local alternatives. A characterization of local alternatives is given in Section 1.7 of the Supplementary Material of Zhang et al. (2018). This result can be proved simply by plugging in the corresponding integrated versions in Theorem 2.4 of Zhang et al. (2018).

In terms of D, a particular case is to consider all covariates, $D = \{1, \ldots, p\}$. First of all, one must check if, at least, some covariates supply relevant information to model Y. Considering D the set of all covariates indices, we can verify this premise performing (5.6). In case of not having evidence to reject the null hypothesis of conditional mean independence, it does not make sense to model Y with the available information. Otherwise, if one discards the conditional mean independence in this initial step, one can be interested in searching for an efficient subset of covariates to reduce the problem dimension.

Then, for a subset $D \subset \{1, \ldots, p\}$ with cardinality d, $1 \leq d < p$, it is possible to test if these d covariates play a role in terms of the concurrent regression model by means of (5.6). If not, it is possible to discard them and reduce the problem dimensionality to $p - d$. In case we are interested in covariates screening one by one, which corresponds with the case where $D = \{j\}$, we can apply the $j = 1, \ldots, p$ tests displayed in (5.5). This results in p consecutive partial tests for $j = 1, \ldots, p$ considering $H_{0j} \colon \mathbb{E}\left[Y(t)|_{X_j(t)}\right] = \mathbb{E}\left[Y(t)\right]$ almost surely $\forall t \in \mathcal{D} \setminus \mathcal{N}$ or equivalently $H_{0j} \colon \widetilde{MDD}^2\left(Y(t)|_{X_j(t)}\right) = 0$ almost surely $\forall t \in \mathcal{D} \setminus \mathcal{N}$. One drawback of carrying out p consecutive tests is that the initial prefixed significance level is violated if this is not modified considering the total number of tests to be performed. As a result, the significance level has to be adequately corrected. Some techniques, such as the classic but conservative Bonferroni's correction, or the false discovery rate alternative (see Benjamini and Yekutieli (2001), and Cuesta-Albertos et al. (2019)) can be easily applied to avoid this inconvenience.

5.3.1 Derivation of $\widetilde{\widetilde{\mathcal{S}}}^2$

In this section, we prove that the estimator of the variance considered in (5.8) for the term $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = \int_{\mathcal{D}} MDD_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})dt$ correctly estimates this quantity.

As mentioned above, $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$ is a $\mathcal{U}$-statistic of order four. This result implies that using the Hoeffding decomposition, this quantity can be expressed as

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = \frac{1}{\binom{n}{2}} \sum_{1\leq k<l\leq n} \widetilde{U_j(X_{kj}(t), X_{lj}(t))} \cdot \widetilde{V(Y_k(t), Y_l(t))} + (\mathcal{R}_n)_j$$

where $\widetilde{U_j(x, x')}$ is equal to

$$\int_{\mathcal{D}} \left\{ \mathbb{E}\left[J(x, X_j'(t))\right] + \mathbb{E}\left[J(X_j(t), x')\right] - J(x, x') - \mathbb{E}\left[J(X_j(t), X_j'(t))\right] \right\} dt$$

and $\widetilde{V(y, y')} = \int_{\mathcal{D}} (y - \mu_Y)^\top (y' - \mu_Y)dt$ for $\mu_Y = \mathbb{E}[Y(t)]$, being $(\mathcal{R}_n)_j$ a remainder term.

Calculation about Hoeffding decomposition for our framework is collected in Section C.3.2 of the Appendix C.

If we define the theoretical test statistic

$$\breve{T}_n = \sqrt{\binom{n}{2}} \frac{\sum_{j\in D} \widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})}{\widetilde{\mathcal{S}}},$$

considering $\widetilde{\mathcal{S}}$ the true integrated version of the variance, we can see that

$$\begin{aligned}\breve{T}_n =& \sum_{j\in D} \frac{1}{\sqrt{\binom{n}{2}}\widetilde{\mathcal{S}}} \sum_{1\leq k<l\leq n} \widetilde{U_j(X_{kj}(t), X_{lj}(t))} \cdot \widetilde{V(Y_k(t), Y_l(t))} + \frac{\sqrt{\binom{n}{2}}}{\widetilde{\mathcal{S}}} \sum_{j\in D} (\mathcal{R}_n)_j \\ =& \frac{1}{\widetilde{\mathcal{S}}}(D_{n,1} + D_{n,2})\end{aligned}$$

where $D_{n,1} = \binom{n}{2}^{-1/2} \sum_{1\leq k<l\leq n} \sum_{j\in D} \widetilde{U_j(X_{kj}(t), X_{lj}(t))} \cdot \widetilde{V(Y_k(t), Y_l(t))}$ is the leading term and $D_{n,2} = \binom{n}{2}^{1/2} \sum_{j\in D} (\mathcal{R}_n)_j$ is the remainder one. Under the H_0 assumption of (5.4) it is verified that

$$\mathbb{V}\left[D_{n,1}\right] = \sum_{j,j'\in D} \mathbb{E}\left[\widetilde{V(Y(t), Y'(t))}^2\right] \widetilde{U_j(X_j(t), X_j'(t))} \cdot \widetilde{U_{j'}(X_{j'}(t), X_{j'}'(t))}$$

Since the contribution from the term $D_{n,2}$ is asymptotically negligible, we may set $\widetilde{\mathcal{S}}^2 = \mathbb{V}\left[D_{n,1}\right]$, and then construct the variance estimator displayed in equation (5.8).

5.4 Simulation studies

In this section, we consider two simulated concurrent model scenarios to assess the performance in the practice of the new significance tests introduced above. We distinguish between linear (Scenario A) and nonlinear (Scenario B) formulation of the model (5.1). For the sake of simplicity, we consider only the case where the data are measured at the same instants of time. For this aim, a Monte Carlo study with $M = 2000$ replicas in each case is performed using the R software (R Core Team (2019)). Besides, we compare the performance of our test with two competitors. These are the procedure introduced in Ghosal and Maity (2022a), developed in the linear framework, and the method of Kim et al. (2018) for the additive formulation. Henceforth, we refer to them by FLCM and ANFCM, respectively. We refer to Section C.1 of the Appendix C for more details about competitors' implementation.

- **Scenario A (Linear model)**: We assume linearity in (5.1), take $t \in \mathcal{D} = [0, 1]$ and consider $q = 1$ and $p = 2$ covariates entering the model.

  As a result, the simulated model is given by the structure

  $$Y(t) = \beta_1(t)X_1(t) + \beta_2(t)X_2(t) + \varepsilon(t)$$

  with

  $$X_1(t) = 5\sin\left(\frac{24\pi t}{12}\right) + \varepsilon_1(t), \quad X_2(t) = \frac{-(24t-20)^2}{50} - 4 + \varepsilon_2(t).$$

  Here, $\beta_1(t) = -\left(\frac{24t-15}{10}\right)^2 - 0.8$ and $\beta_2(t) = 0.01((24t-12)^2 - 12^2 + 100)$. The error terms represented by $\varepsilon_1(t), \varepsilon_2(t)$ and $\varepsilon(t)$ are simulated as random gaussian processes with exponential variogram $\Omega(s,t) = 0.1\exp\left(-\frac{24|s-t|}{10}\right)$. We assume that a total number of $\mathcal{T} = 25$ equispaced instants are observed in $\mathcal{D} = [0, 1]$ ($\{t_u\}_{u=1}^{25}$) and there are $n = 20, 40, 60, 80, 100$ curves available for each of them. An example of these functions is displayed in Figure 5.3. We remark that we have not included intercept in our linear formulation because this can be done without loss of generality just centering both $Y(t)$ and $X(t) = (X_1(t), X_2(t))^\top \in \mathbb{R}^2$ for all $t \in \mathcal{D}$.

- **Scenario B (nonlinear model)**: a nonlinear structure of (5.1) is assumed for this scenario. Again, we take $t \in \mathcal{D} = [0, 1]$ and consider $q = 1$ and $p = 2$ covariates to explain the model.

  Then, this model has the expression

  $$Y(t) = F_1(t, X_1(t)) + F_2(t, X_2(t)) + \varepsilon(t)$$

  being $F_1(t, X_1(t)) = \exp((24t+1)X_1(t)/20) - 2$ and $F_2(t, X_2(t)) = -1.2\log(X_2(t)^2)\sin(2\pi t)$, with $X_1(t)$ and $X_2(t)$ equally defined as in the linear case (Scenario A) and us-

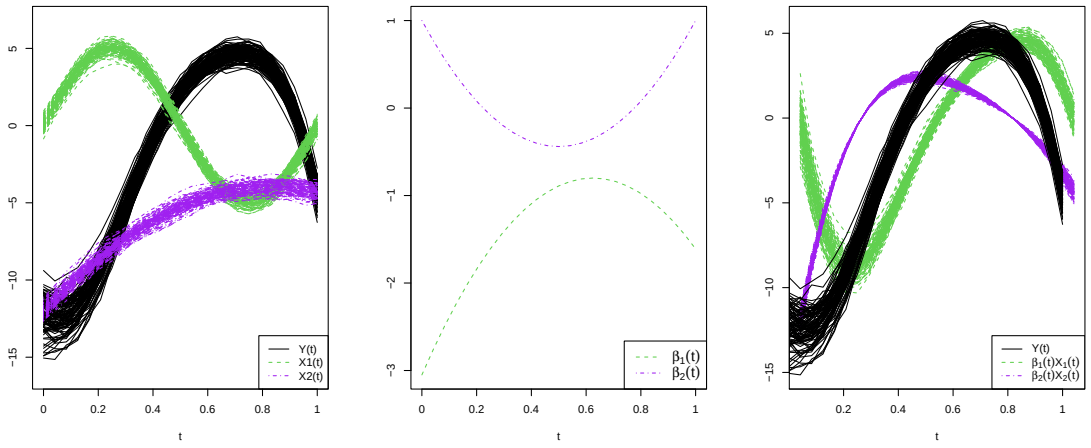


Figure 5.3: Left: simulated sample values of the functional variables along the grid $[0, 1]$ taking $n = 20$. Middle: real partial effects corresponding to $X_1(t)$ $(\beta_1(t))$ and $X_2(t)$ $(\beta_2(t))$. Right: simulated regression model components $\beta_1(t)X_1(t)$ and $\beta_2(t)X_2(t)$.

ing the same observed discretization time points. Now, the errors $\varepsilon_1(t), \varepsilon_2(t)$ and $\varepsilon(t)$ are assumed to be random gaussian processes with exponential variogram $\Omega(s,t) = 0.02 \exp\left(-\frac{24|s-t|}{10}\right)$. An example of this scenario is displayed in Figure 5.4.

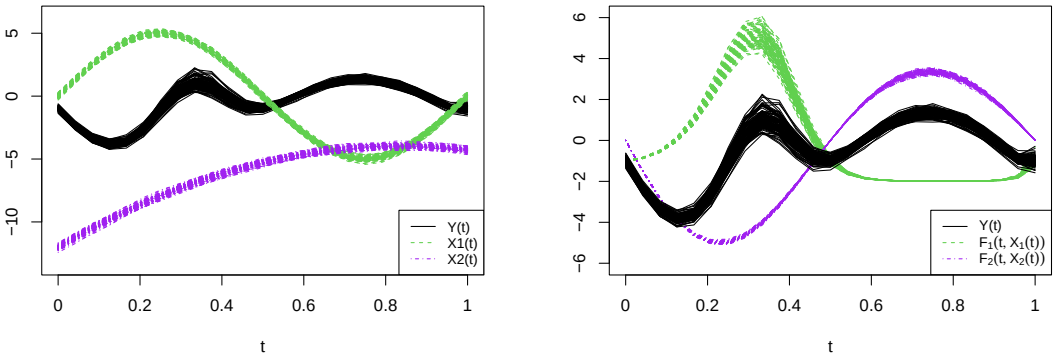


Figure 5.4: Left: simulated sample values of the functional variables along the grid $[0, 1]$ taking $n = 20$. Right: real $Y(t)$ structure jointly with partial effects corresponding to $X_1(t)$ $(F_1(t, X_1(t)))$ and $X_2(t)$ $(F_2(t, X_2(t)))$.

In all tests, we make use of the wild bootstrap techniques introduced above in Section 5.3 to approximate the p-values. We have employed $B = 1000$ resamples on each case. Besides, as we mentioned before, sample test size and power are obtained by Monte Carlo techniques. In order to know if the p-values under the null take an adequate value, the 95% confidence intervals of the significance levels are obtained by making use of

expression $\left[\alpha \mp 1.96\sqrt{\frac{\alpha(1-\alpha)}{M}}\right]$. Here, α is the expected level and M is the number of Monte Carlo simulated samples. As a result, we consider that a p-value is acceptable for levels $\alpha = 0.01, 0.05, 0.1$ if this is within the values collected in Table 6.1 for the Monte Carlo replicates. We highlight the values out of these scales in **bold** for simulation results.

M	$\boldsymbol{\alpha = 0.01}$	$\boldsymbol{\alpha = 0.05}$	$\boldsymbol{\alpha = 0.1}$
1000	$[0.004, 0.016]$	$[0.036, 0.064]$	$[0.081, 0.119]$
2000	$[0.006, 0.014]$	$[0.040, 0.060]$	$[0.087, 0.113]$

Table 5.1: Confidence intervals at 95% of the Monte Carlo proportions for M replicates.

5.4.1 Results for scenario A (linear model)

We start analyzing the performance of the global mean dependence test in the linear model formulation, using Scenario A introduced above in Section 5.4. For this purpose, we consider three different scenarios. In the first one, the null hypothesis of mean independence is verified by simulating under the assumption that $\beta_1(t) = \beta_2(t) = 0$. Next, the remaining two cases are simulated under the alternative hypothesis. This claims that information provided by $X(t) = (X_1(t), X_2(t))^\top$ is useful in some way: only the $X_2(t)$ covariate is relevant (fixing $\beta_1(t) = 0$) or both covariates $X_1(t)$ and $X_2(t)$ support relevant information to correctly explain $Y(t)$.

Model:	$\boldsymbol{\beta_1(t) = \beta_2(t) = 0}$ (H_0)			$\boldsymbol{\beta_1(t) = 0, \beta_2(t) \neq 0}$ (H_a)			$\boldsymbol{\beta_1(t) \neq 0, \beta_2(t) \neq 0}$ (H_a)		
	1%	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
$n = 20$	0.010	0.045	0.092	0.574	0.797	0.882	1	1	1
$n = 40$	0.013	0.050	0.093	0.984	0.998	1	1	1	1
$n = 60$	0.007	0.052	0.103	1	1	1	1	1	1
$n = 80$	0.009	0.045	0.094	1	1	1	1	1	1
$n = 100$	0.012	0.050	0.088	1	1	1	1	1	1

Table 5.2: Empirical sizes and powers of the MDD-based global test for mean independence testing using wild bootstrap approximation with $B = 1000$ resamples in Scenario A.

Obtained results are collected in Table 5.2 for $n = 20, 40, 60, 80, 100$. In view of the results, it is appreciated as the empirical sizes approximate the significance levels under H_0 ($H_0\colon \beta_1(t) = \beta_2(t) = 0$) as n increases. Moreover, the empirical distribution of the p-values seems to be a $U[0, 1]$ as it is appreciated in Figure C.1 of Section C.2 of the Appendix C. In contrast, simulating under the alternative hypothesis, $H_a\colon \beta_1(t) = 0, \beta_2(t) \neq 0$ and $H_a\colon \beta_1(\mathfrak{t}) \neq 0, \beta_2(\mathfrak{t}) \neq 0$ scenarios, the test power tends to one as the sample size increases. As a result, we can claim that the test is well-calibrated and has power.

Once we have rejected the null hypothesis that all covariates are irrelevant in practice, we can detect which of them play a role in terms of data explanation. For this aim, partial

tests can be carried out, testing if every covariate is irrelevant, $H_{0j}\colon \beta_j(t) = 0\ \forall t \in \mathcal{D}$, or not, $H_{aj}\colon \beta_j(t) \neq 0$ for some $t \in \mathcal{V}$, being $j = 1, \ldots, p$.

Again, we consider different scenarios. First of all, it is assumed that $X(t)$ is not significant taking $\beta_1(t) = \beta_2(t) = 0$. Then, we move to the situation where only $X_2(t)$ is relevant. Finally, we consider the model including both $X_1(t)$ and $X_2(t)$ effects to explain $Y(t)$. Results of these scenarios are displayed in Table 5.3. Here, we appreciate as the empirical sizes tend to the significance levels simulating under the null hypothesis that both covariates have not got a relevant effect on the response, separately. Besides, we see as in case of having $\beta_1(t) = 0$ and $\beta_2(t) \neq 0$, these tests help us to select relevant information, $X_2(t)$, and discard noisy one, $X_1(t)$. Otherwise, when both covariates are relevant, the partial tests clearly reject the H_{0j} hypothesis of null effect, tending their powers to the unit as sample size increases.

Model:	$\boldsymbol{\beta_1(t) = \beta_2(t) = 0}$		$\boldsymbol{\beta_1(t) = 0,\ \beta_2(t) \neq 0}$		$\boldsymbol{\beta_1(t) \neq 0,\ \beta_2(t) \neq 0}$	
	H_{01}	H_{02}	H_{01}	H_{02}	H_{01}	H_{02}
	5%/10%	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**
$n = 20$	0.040/**0.078**	0.043/0.101	0.041/0.087	0.919/0.966	1/1	0.330/0.490
$n = 60$	0.048/0.101	0.049/0.103	0.047/0.098	1/1	1/1	0.935/0.971
$n = 100$	0.046/0.089	0.047/0.096	0.046/**0.086**	1/1	1/1	0.998/1

Table 5.3: Empirical sizes and powers of the partial MDD-based global tests for mean independence testing considering as null hypothesis $H_{01}\colon \mathbb{E}[Y(t)|_{X_1(t)}] = \mathbb{E}[Y(t)]$ and $H_{02}\colon \mathbb{E}[Y(t)|_{X_2(t)}] = \mathbb{E}[Y(t)]$, and using wild bootstrap approximation with $B = 1000$ resamples in Scenario A.

5.4.2 Results for scenario B (nonlinear model)

In this section, we analyze the performance of the MDD global mean independence test in a more difficult framework: a nonlinear effects formulation. For this purpose, Scenario B introduced in Section 5.4 is employed. Again, three different situations of dependence are considered, following the same arguments of Section 5.4.1. As a result, we simulate under the no effect case ($H_0\colon F_1(t, X_1(t)) = F_2(t, X_2(t)) = 0$), which corresponds with independence, and two dependence frameworks: where only one covariate is relevant ($H_a\colon F_1(t, X_1(t)) = 0, F_2(t, X_2(t)) \neq 0$) or both of them are ($H_a\colon F_1(t, X_1(t)) \neq 0, F_2(t, X_2(t)) \neq 0$).

Results of the $M = 2000$ Monte Carlo simulations for the MDD-test taking $n = 20, 40, 60, 80, 100$ are displayed in Table 5.4. We appreciate simulating under the null hypothesis H_0 that the p-values tend to stabilize around the significance levels. Figure C.2, collected in Section C.2 of the Appendix C, shows as these seem to follow a uniform distribution in $[0, 1]$. So, we can conclude that our test is well calibrated even for nonlinear approaches. Concerning the power, when the independence assumption is violated, the p-values tend to 1 as the sample size increases. Two examples of this phenomenon are displayed in Table 5.4 simulating the different alternative hypotheses. Summing up, our proposal is also a well-calibrated and powerful test in a nonlinear framework.

Model:	$F_1(\cdot) = F_2(\cdot) = 0$ (H_0)			$F_1(\cdot) = 0, F_2(\cdot) \neq 0$ (H_a)			$F_1(\cdot) \neq 0, F_2(\cdot) \neq 0$ (H_a)		
	1%	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
$n = 20$	0.011	0.049	0.096	0.215	0.426	0.563	0.989	1	1
$n = 40$	0.013	0.05	0.094	0.564	0.793	0.886	1	1	1
$n = 60$	0.009	0.053	0.105	0.871	0.956	0.979	1	1	1
$n = 80$	0.01	0.046	0.096	0.974	0.996	1	1	1	1
$n = 100$	0.013	0.054	0.093	0.994	1	1	1	1	1

Table 5.4: Empirical sizes and powers of the MDD-based global test for mean independence testing using wild bootstrap approximation with $B = 1000$ resamples in Scenario B.

Next, our interest focuses on partial tests to apply covariates selection in this nonlinear scenario. Again, we consider the three different dependence scenarios introduced above. However, we test the independence for each covariate separately. This results in applying a total of $j = 1, \ldots, p$ tests. In this way, we expect the test in a situation as $F_1(t, X_1(t)) = 0$, $F_2(t, X_2(t)) \neq 0$ to be capable of detecting relevant covariates ($X_2(t)$), rejecting its corresponding H_{0j} hypothesis, and excluding noisy ones from the model otherwise ($X_1(t)$). Results for partial tests are collected in Table 5.5. One can see as these tests allow us to determine which covariates play a relevant role in each scenario, being those with p-values higher than the significance levels and tending to 1 as sample size increases. Conversely, those verifying that its associated p-values are less or equal to significance levels are assumed irrelevant.

Model:	$F_1(\cdot) = F_2(\cdot) = 0$		$F_1(\cdot) = 0,\ F_2(\cdot) \neq 0$		$F_1(\cdot) \neq 0,\ F_2(\cdot) \neq 0$	
	H_{01}	H_{02}	H_{01}	H_{02}	H_{01}	H_{02}
	5%/10%	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**
$n = 20$	0.04/**0.078**	0.043/0.101	0.04/**0.077**	0.567/0.692	1/1	0.18/0.299
$n = 60$	0.048/0.101	0.049/0.103	0.053/0.107	0.987/0.995	1/1	0.621/0.783
$n = 100$	0.046/0.089	0.047/0.096	0.044/0.09	1/1	1/1	0.915/0.971

Table 5.5: Empirical sizes and powers of the partial MDD-based global tests for mean independence testing considering $H_{01}\colon \mathbb{E}[Y(t)|_{X_1(t)}] = \mathbb{E}[Y(t)]$ and $H_{02}\colon \mathbb{E}[Y(t)|_{X_2(t)}] = \mathbb{E}[Y(t)]$, and using wild bootstrap approximation with $B = 1000$ resamples in Scenario B.

5.4.3 Comparison with FLCM and ANFCM algorithms

Next, our novel procedure is compared with existing competitors in the literature. For this aim, we have considered the FLCM algorithm of Ghosal and Maity (2022a) for the linear framework and the ANFCM procedure of Kim et al. (2018) for a more flexible model, assuming additive effects. Both have displayed excellent results in practice considering a proper selection of the tuning parameters. We refer the reader to Section C.1 of the Appendix C for more details.

In the simulation scenarios introduced in Section 5.4, we consider a dependence structure

where all instants relate between them. This structure emulates a real functional dataset. Nevertheless, this does not apply in the simulation scenarios of Ghosal and Maity (2022a) and Kim et al. (2018). Conversely, they consider independent errors. As a result, to perform a fair competition, we start analyzing the behavior of our MDD-based tests in their simulation scenarios. Specifically, we compare the performance of our proposal with the results of FLCM in Scenario A of Ghosal and Maity (2022a). Next, we implement a comparison with the ANFCM procedure. For this purpose, we consider Scenario (B) of Kim et al. (2018), taking the error E^3. In this last case, we implement a modification to perform Algorithm 1. In particular, we only consider the second covariate associated with the nonlinear effect. In both borrowed scenarios, we simulate under the dense assumption being $\{t_u\}_{u=1}^{81}$ a total of $m = 81$ equidistant time points in $[0, 1]$. We keep the authors' parameters selection and perform a Monte Carlo study with $M = 1000$ samples in all cases, obtaining the p-values through $B = 200$ bootstrap replicates. Besides, following the author's recommendation after a preliminary study to determine the optimal number of basis functions for these examples, we work with 7 components for FLCM and ANFCM procedures. More details can be found in Ghosal and Maity (2022a) or Kim et al. (2018), respectively. We remind the structure of the scenarios and explain implementation issues in Section C.1 of the Appendix C.

Results of the comparison between FLCM and MDD effect tests for scenario A of Ghosal and Maity (2022a) are collected in Table 5.6. We appreciate that simulating under the null ($d = 0$), one value of the FLCM algorithm is out of the 95% confidence interval. In contrast, the MDD procedure does not suffer from this issue. Moreover, paying attention to the p-values distributions under the null, which are displayed in Figure C.3 (see Section C.2 of the Appendix C), one can see the FLCM p-values do not follow a uniform distribution. In contrast, the MDD-based test corrects this phenomenon. As a result, it seems that our test provides a better calibration than the FLCM approach. Regarding the power, levels for both algorithms tend to 1 as sample size increases, and their values are higher for the $d = 7$ scenario than for the $d = 3$ one, as would be expected. Now, the FLCM algorithm outperforms the MDD results in all scenarios. However, our procedure is still quite competitive even considering that the data is simulated under the linear assumption, giving an advantage to the FLCM procedure.

Model:		$\mathbf{H_0}$ **(d = 0)**			$\mathbf{H_a}$ **(d = 3)**			$\mathbf{H_a}$ **(d = 7)**		
		1%	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
n=60	FLCM	0.007	0.054	0.103	0.776	0.888	0.937	0.999	1	1
	MDD	0.014	0.052	0.097	0.341	0.550	0.671	0.992	0.997	1
n=100	FLCM	0.005	0.038	**0.077**	0.964	0.979	0.992	1	1	1
	MDD	0.013	0.049	0.103	0.619	0.796	0.871	1	1	1

Table 5.6: Summary of empirical sizes and powers of the FLCM and MDD effect tests.

Next, we compare the performance of the MDD with the ANFCM approach in an additive framework. Table 5.7 collects the simulation results for both procedures. We can

see as both methodologies are well calibrated under the null ($d = 0$) for all levels, except for the 1%, where their values are out of the 95% confidence interval for $n = 60$. Nevertheless, taking greater values of n, as $n = 100$, solves this issue. Moreover, simulating under H_0, the p-values follow a uniform distribution. This is illustrated in Figure C.4 displayed in Section C.2 of the Appendix C. If we simulate under the alternative hypotheses ($d = 3$ and $d = 7$), we see that these quantities tend to 1 as the sample size increases. In addition, as the covariate effect becomes more noticeable, going from $d = 3$ to $d = 7$, the power of ANFCM and MDD procedures increases. Again, the power of the ANFCM algorithm is always higher than the MDD one. At this point, we should notice that the ANFCM algorithm takes advantage of the fact that an additive structure with an intercept function is assumed. In contrast, our MDD test does not consider any model structure, not even the inclusion of intercept in the model. As a result, our competitor has to measure all possible forms of departure from conditional mean independence.

Model:		$\mathbf{H_0}$ **(d = 0)**			$\mathbf{H_a}$ **(d = 3)**			$\mathbf{H_a}$ **(d = 7)**		
		1%	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
n=60	ANFCM	**0.021**	0.063	0.117	1	1	1	1	1	1
	MDD	**0.019**	0.058	0.102	0.410	0.811	0.944	0.747	0.984	1
n=100	ANFCM	0.014	0.056	0.094	1	1	1	1	1	1
	MDD	0.008	0.046	0.095	0.929	0.999	1	0.999	1	1

Table 5.7: Summary of empirical sizes and powers of the ANFCM and MDD effect tests.

It is relevant to notice that, in both previous scenarios, covariates are related to the response employing trigonometric functions when it corresponds. Then, modeling the effects takes advantage of the B-spline basis representation. In addition, the errors are assumed to be time-independent between them in the FLCM and ANFCM scenarios. These considerations are a clear advantage for the FLCM and the ANFCM algorithms compared to our procedure. Thus, to test the FLCM and ANFCM performance in a functional context with time-correlated errors and when the model structure does not depend on only trigonometric functions, we apply these to the simulation scenarios introduced in Section 5.4. For this purpose, a partial approach is considered, testing the effect of the covariates separately using the FLCM procedure in Scenario A and the ANFCM one in Scenario B. To compare our results with theirs, we simulate now $M = 2000$ Monte Carlo replications and use $B = 1000$ bootstraps resamples for ANFCM. Again, we follow the authors' recommendation and use $Q = 7$ basis terms in both procedures [1]. We refer to Section C.1 of the Appendix C for a summary of the simulation parameters selection.

Results of partial FLCM tests in scenario A are displayed in Table 5.8. It can be seen how, regardless of the size of the sample used, the test is always poorly calibrated. In fact, all obtained p-values are out of the 95% confidence intervals. These results contrast with

[1] In this setup, we have $\mathcal{T} = 25$ time instants. Then, for the ANFCM procedure, as the function `fpca.face` employs by default a total of 35 knots to carry out FPCA, we have to reduce this. We decided to take 12 knots to solve this issue.

the MDD ones displayed in Table 5.3, where the test is well calibrated. This phenomenon may be because, as mentioned above, it is considered a different dependence structure more related to a functional nature. In terms of power, there is not a clear winner. Our test is more powerful in the $H_a\colon \beta_1(t) = 0,\ \beta_2(t) \neq 0$ scenario in test H_{02}, but FLCM is a bit more powerful in the last scenario for H_{02}. However, this difference is small, and considering that the FLCM is not well calibrated, it makes sense to conclude that the MDD-based procedure outperforms this.

Model:	$\boldsymbol{\beta_1(t) = \beta_2(t) = 0}$		$\boldsymbol{\beta_1(t) = 0,\ \beta_2(t) \neq 0}$		$\boldsymbol{\beta_1(t) \neq 0,\ \beta_2(t) \neq 0}$	
	H_{01}	H_{02}	H_{01}	H_{02}	H_{01}	H_{02}
	5%/10%	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**
$n = 20$	**0.1/0.174**	**0/0.002**	**0.104/0.172**	0.622/0.758	1/1	1/1
$n = 60$	**0.09/0.152**	**0/0.004**	**0.09/0.158**	0.709/0.875	1/1	1/1
$n = 100$	**0.074/0.125**	**0/0.003**	**0.1/0.17**	0.913/0.983	1/1	1/1

Table 5.8: Empirical sizes and powers of the FLCM effect test considering $H_{01}\colon \beta_1(t) = 0$ and $H_{02}\colon \beta_2(t) = 0$ in Scenario A.

Next, the performance of the ANFCM algorithm is tested by simulating under Scenario B of Section 5.4. Results are collected in Table 5.9. Again, comparing the ANFCM results with the ones of the MDD test (Table 5.5), we see that the MDD test is well calibrated even for small values as $n = 20$ (except for a couple of cases). This fact contrasts with the results of the ANFCM procedure. In this last, most of the values are out of the 95% confidence intervals. Moreover, the MDD test has more power than ANFCM in almost all cases. As a particularity, the ANFCM algorithm is not able to detect the relevance of $X_2(t)$ in the $H_a\colon F_1(\cdot) = 0,\ F_2(\cdot) \neq 0$ scenario. Instead, the percentage of rejections is around the significance values and does not provide significant evidence to reject the null hypothesis H_{02} of independence. Thus, we can conclude that the MDD outperforms the ANFCM procedure.

Model:	$\boldsymbol{F_1(\cdot) = F_2(\cdot) = 0}$		$\boldsymbol{F_1(\cdot) = 0,\ F_2(\cdot) \neq 0}$		$\boldsymbol{F_1(\cdot) \neq 0,\ F_2(\cdot) \neq 0}$	
	H_{01}	H_{02}	H_{01}	H_{02}	H_{01}	H_{02}
	5%/10%	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**	**5%/10%**
$n = 20$	**0.098/0.17**	**0.102/0.176**	**0.134/0.203**	0.043/0.08	0.662/0.8	0.123/0.191
$n = 60$	**0.068/0.118**	0.058/0.113	**0.071/0.128**	0.013/0.028	1/1	0.117/0.216
$n = 100$	0.047/0.102	0.055/0.106	0.049/0.1	0.063/0.114	1/1	0.147/0.293

Table 5.9: Empirical sizes and powers of the ANFCM effect test considering $H_{01}\colon F_1\left(t, X_1(t)\right) = 0$ and $H_{02}\colon F_2\left(t, X_2(t)\right) = 0$ and using $B = 1000$ bootstrap resamples in Scenario B.

In summary, we have proved that the MDD algorithm performs pretty well in scenarios where the FLCM and the ANFCM procedures have an advantage, considering uncorrelated errors and trigonometric functions. Moreover, our test outperforms these when we move

on to a more functional context, as in scenarios A and B introduced in Section 5.4. In these scenarios, we consider related errors and other types of relations different from trigonometric functions.

5.5 Application in some real data sets

In this section, we test the performance of the proposed algorithms in three real data sets. Firstly, the well-known gait dataset of Olshen et al. (1989) is considered. This data set is an example of a linear effects model and has already been studied in the concurrent model framework in works as the one of Ghosal and Maity (2022a) or Kim et al. (2018). Next, a google flu database from the USA, borrowed from Wang et al. (2017), is studied. In this work, Wang et al. (2017) assume a linear formulation to model the data. Eventually, an example of a model with nonlinear effects and some missing points is studied. For this purpose, the bike sharing dataset of Fanaee-T and Gama (2014) is analyzed. Obtained results are compared with the ones of Ghosal and Maity (2022b) in this concurrent model framework.

5.5.1 Gait data

Here, we analyze the performance of the new dependence test in a well-known dataset from the functional data context. This data is the gait database (Olshen et al. (1989), Ramsay and Silverman (2005)), in which the objective is to understand how the joints in the hip and the knee interact during a gait cycle in children. This problem has already been studied in the concurrent model context using a different methodology (see Ghosal and Maity (2022a), or Kim et al. (2018)). As a consequence, we compare our results with theirs.

The data consist of longitudinal measurements of hip and knee angles taken on 39 children with gait deficiency. These are measured as they walk through a single gait cycle. This data can be found in the `fda` library (Ramsay et al. (2020)) of the R software (R Core Team (2019)). The hip and knee angles are measured at 20 evaluation points $\{t_u\}_{u=1}^{20}$ in $[0, 1]$. These values correspond to the completed percentage of a single gait cycle. Following previous studies, we have considered as response $Y(t)$ the knee angle and as explanatory covariate $X(t)$ the hip angle. Data is displayed in Figure 5.5.

Applying our dependence test, we obtain a p-value close to 0. Thus, we have strong enough evidence to reject the independence hypothesis to the usual significance levels. This conclusion translates into a dependency between knee and hip angle in one cycle of gait data in children with poor gait. This result agrees with the ones of Kim et al. (2018) or Ghosal and Maity (2022a), among others, in the concurrent model framework. They obtain p-values less than 0.004 and 0.001, respectively. Summing up, the hip angle measured at a specific time point in a gait cycle has an effect on the knee angle at the same time point in children with gait deficiency.

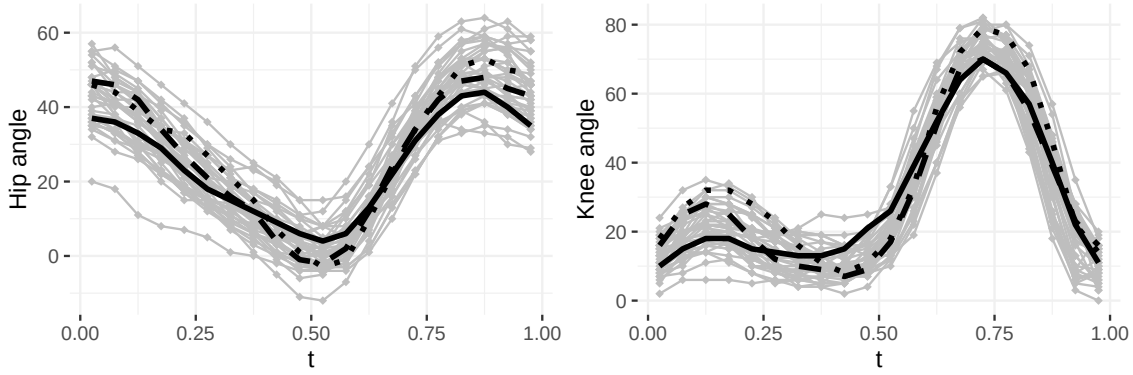


Figure 5.5: Hip (left) and knee (right) angles measurements of a complete gait cycle.

5.5.2 Google flu data from U.S.A.

Google flu data is used in Wang et al. (2017) to model the relationship between flu activity and temperature fluctuation in the USA. For this purpose, influenza-like illness (ILI) cases per 100000 doctor visits are considered in the 2013–2014 flu season (July 2013–June 2014). This information is got from the Google flu trend Website. Moreover, daily maximum and minimum temperature averaged over weather stations within each continental state is obtained by means of the US historical climatology network. The daily temperature variation (MDTV) is considered the explanatory covariate, being the difference between the daily maximum and daily minimum. The temperature fluctuation is aggregated to the same resolution as the flu activity data by taking the MDTV each week. Only 42 states are considered due to missed records. We refer to Wang et al. (2017) for more details.

The original dates from July 1st, 2013, to June 30th, 2014, were numbered by integers from 1 to 365. Then, time t is rescaled to the $[0, 1]$ interval by dividing the numbers by 365. Besides, we consider regional effects by dividing the data into four sets in terms of midwest, northeast, south, or west region to study them separately. Following Wang et al. (2017), the ILI percentage and MDTV are standardized at each time point t by dividing the variables by their root mean squares. Data of study is shown in Figure 5.6 separating this by the considered regions.

Therefore, we want to test if the MDTV has relevant information in the flu tendency modeling of the four considered regions. For this aim, we can apply a global test for each one separately. Results of dependence tests are displayed in Table 5.10. In view of all p-values being higher than 0.1, we can conclude that we do not have enough evidence to reject the null hypothesis of mean conditional independence for levels as 10%. As a result, the MDTV does not play a relevant role in the ILI modeling, no matter the US region. We can argue that perhaps the regional effect is unimportant, and we should consider the data as a whole. For this purpose, we implement a global test considering all the states, obtaining a p-value close to 0. This result highlights that there is strong evidence to reject the conditional mean independence between MDTV and ILI. As a result, MDTV provides notable information to explain the ILI behavior, but this is equal in the four considered

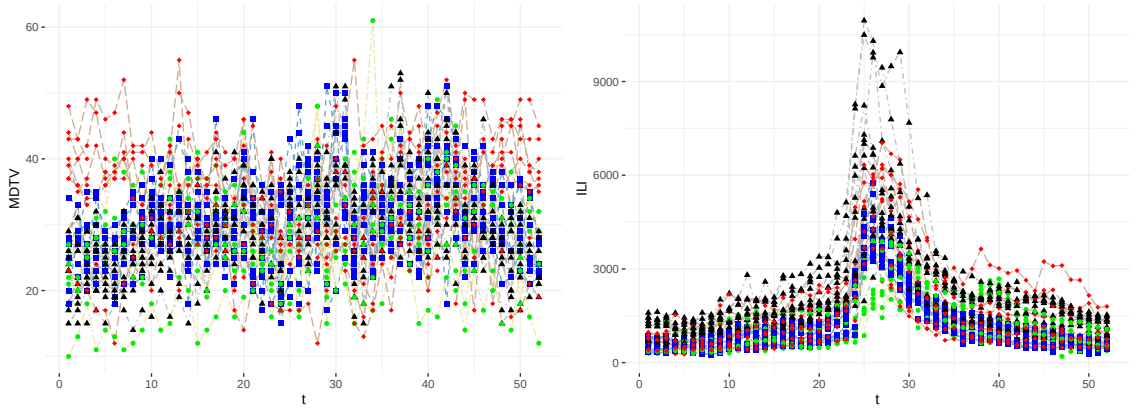


Figure 5.6: MDTV (left) and flu activity or ILI (right) data in terms of their corresponding regions: northeast (●), midwest (■), south (▲) and west (♦).

regions, so a distinction does not make sense.

p-value	midwest	northeast	south	west
	0.106	0.761	0.623	0.667

Table 5.10: P-values of the MDD-based tests for the different regions.

Our results agree with the ones of Wang et al. (2017). First, they reject the location effect for the linear model formulation. Secondly, they claim that one can avoid the MDTV covariate from the linear model for a 10% significance level but not for the 5% (p-value=0.052). Thus, they have moderately significant evidence that the MDTV plays a role in the ILI explanation, at least in the linear context. It is important to remark that differences may be because they assume linearity in their regression model. Furthermore, a first preprocessing step is applied in their case to remove spatial correlations.

5.5.3 Bike sharing data from Washington, D.C.

Next, a bike-sharing dataset of the Washington, D.C., program is analyzed. This is introduced in Fanaee-T and Gama (2014). The data is obtained daily by the Capital bike-share system in Washington, D.C., from 1 January 2011 to 31 December 2012. The aim is to explain the number of casual rentals in terms of meteorological covariates. As a result, this dataset contains information on casual bike rentals in the cited period along with other meteorological variables such as temperature in Celsius (temp), the feels-like temperature in Celsius (atemp), relative humidity in percentage (humidity), and wind speed in Km/h (windspeed) on an hourly basis. In particular, only the data corresponding with Saturdays are considered because of the dynamic changes between working and weekend days. This selection results in a total of 105 Saturdays barring some exceptions (8 missings). All covariates are normalized by formula $(t - t_{\min})/(t_{\max} - t_{\min})$ in case of temp and atemp, and these are divided by the maximum for the humidity and windspeed case. In order to

correct the skewness of the hourly bike rentals distribution ($Y(t)$), a log transformation is applied considering as response variable $Y(t) = log(Y(t) + 1)$. These are shown in Figure 5.7.

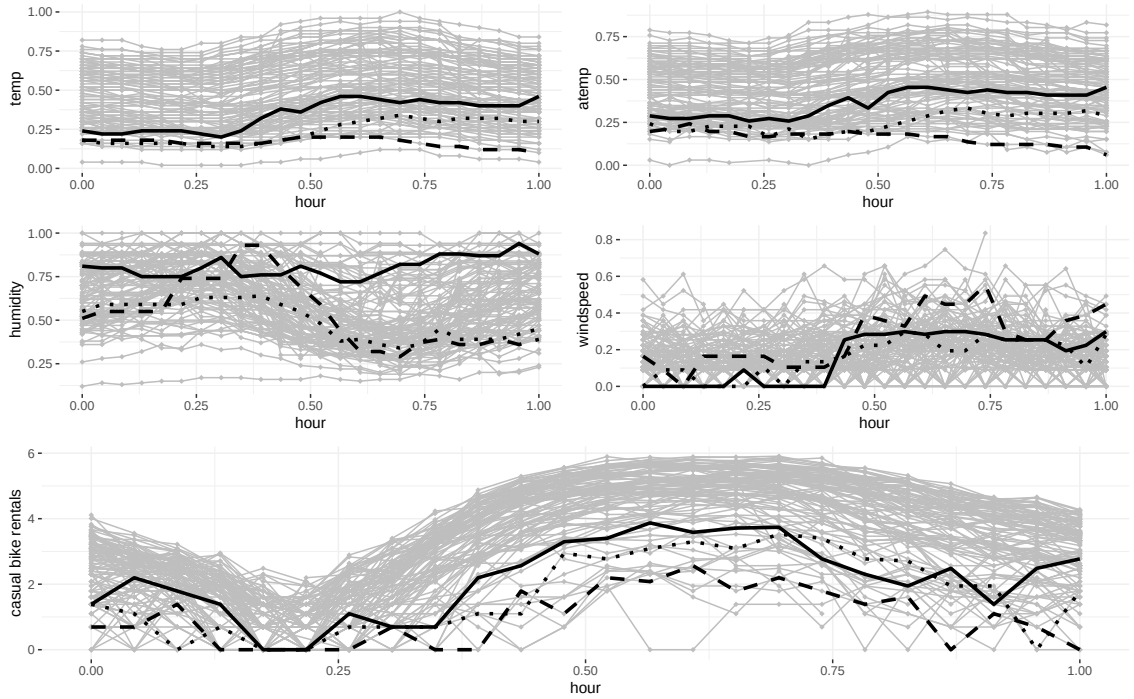


Figure 5.7: Daily temperature (temp), feeling temperature (atemp), humidity, wind speed and casual bike rentals on an hourly basis in Washington D.C. on Saturdays.

First, the missing data is recovered employing splines interpolation as described in Section 5.1.2. Then, once we have a total of $n = 105$ data points at each time instant, the global significance MDD-based test is performed. We obtain a p-value close to 0, which rejects the null hypothesis of independence for usual significant levels as the 5% or the 1%.

Next, we perform partial tests to detect if any of the four considered covariates (temp, atemp, humidity, and windspeed) can be excluded from the model. We obtain p-values of 0, 0, 0.007, and 0.001 for temperature (temp), feels-like temperature (atemp), relative humidity (humidity), and wind speed (windspeed), respectively. Thus, we can claim that all of these affect the number of casual rentals at significance levels as the 1%. This last result agrees with other studies, like the one of Ghosal and Maity (2022b). In this study, different covariates are selected by the distinct considered penalizations. In an overview of their results, each covariate is selected at least two times over the five considered procedures. As a result, all covariates seem to play a relevant role separately.

5.6 Conclusions

We propose novel significance tests for the additive functional concurrent model, which collects a wide range of different structures between functional covariates and response.

As a result, the relevance of a subset of covariates to model the response in a regression setting is tested, including global and partial tests to apply covariates screening. This approach allows one to detect irrelevant variables and reduce the problem dimensionality, facilitating the subsequent estimation procedure. For this aim, we construct test statistics based on MDD insights and taking into consideration all observed time instants. This process results in general significance tests able to determine the covariates' relevance over the complete trajectory. In contrast with existing methodology in literature for significance tests in the concurrent model, as the FLCM (Ghosal and Maity (2022a)) or the ANFCM (Kim et al. (2018)) procedures among others, our approach has the novel property that there is no need of a preliminary estimation of the model structure. Besides, this new procedure allows multivariate responses $Y(t) \in \mathbb{R}^q$ for $q \geq 1$ and $t \in \mathcal{D}$. Furthermore, no tuning parameters are involved in contrast with previous methodologies. Instead, it is only needed to compute a $\mathcal{U}$-statistic version of the MDD to be able to apply the tests. Using the theory of $\mathcal{U}$-statistics, good properties of this estimator are guaranteed in practice, as its unbiasedness. In addition, its asymptotic distribution is obtained both under the null and local alternative hypotheses. Eventually, bootstrap procedures are implemented to obtain its p-values in practice.

The new tests proposed have displayed good performance in linear formulations and in nonlinear structures. This is appreciated by means of the results of scenarios A and B considered in the simulation study of Section 5.4. These procedures are well calibrated under the null hypothesis of no effect, tending to the significance level as the sample size increases. Moreover, they have power under alternatives, which one can deduce from observing that p-values tend to the unit as sample size increases when associated covariates are relevant. Besides, these procedures seem to perform well in real data sets too. We display an example of this result in Section 5.5, where we analyze three real datasets. Other authors have already studied these, so we compare our outcomes with existing literature, obtaining similar results when these are comparable. As a result, the MDD-based test is a pretty transversal tool to detect additive effects in the concurrent model framework without the need for previous assumptions or model structure estimation. Moreover, notice that all these ideas could be extended to conditional quantile dependence testing in the concurrent model framework. For this purpose, a similar development would be enough, following the guidelines and adapting the ideas of Section 3 in Zhang et al. (2018).

In terms of performance comparison with existing literature, the MDD-based test methodology is put together with Ghosal and Maity (2022a) (FLCM) and Kim et al. (2018) (ANFCM) algorithms in the linear and additive model framework, respectively. Based on the obtained results, it is possible to claim that the new procedure is quite competitive. Even when the FLCM and ANFCM procedures have the advantage of being implemented assuming the correct model structure and an optimal number of the basis components, the new procedure results are comparable to theirs. These results arise in Section 5.4.3. In contrast, our procedure outperforms their results by simulating a more functional scenario and avoiding only trigonometric expressions in the model. Besides, another disadvantage of the competitors is that $m(t, X(t))$ is unknown in practice, so a misguided assumption

of the model structure could lead to poor results. In addition, as discussed in Ghosal and Maity (2022a) and Kim et al. (2018), a suitable selection of the number of the basis components is problematic in practice. This issue is still an open problem. This quantity plays the role of tuning parameter, so an appropriate value is needed to guarantee a proper adjustment. In contrast, our proposal has the novelty that this does not require previous estimation or tuning parameters selection. Our approach bridges a gap and solves the problems mentioned above.

One limitation of the present form of our test is that this only admits the study of numerical covariates. This restriction is quite common for the concurrent model framework. Some examples are the works of Ghosal and Maity (2022a) or Kim et al. (2018). If one wants to be able to include categorical variables, as in other works such as in Wang et al. (2017), a different metric is needed to correctly define the $\mathcal{U}$-statistic of the MDD test. Some solutions for this problem have already been proposed for the distance covariance approach in the presence of noncontinuous variables. Similar ideas could be translated to the MDD context to solve this issue. An option is to extend the ideas proposed in Lyons (2013) for general metric spaces to this case. We leave this topic for future research.

A drawback of our methodology is the statistics computational time, being of the order of $O(n(n-1)(n-2)(n-3)\mathcal{T})$ operations. Then, this procedure is quite competitive for "moderate" values of n and $\mathcal{T}$. However, for large values of these quantities, especially those related to n, the statistic has a high computational cost. Consequently, simplification techniques in the number of required operations are of interest to make the procedure more tractable.

Furthermore, it is interesting to remark that, because of the statistics structures, the tests collect only additive effects. Although this formulation embraces a huge variety of different structures, this does not consider some complex relations like interactions without a prespecified definition of new variables collecting these. Nevertheless, it is thought that, using projections, these ideas can be extended to the general concurrent model formulation, where all possible relations are considered. This is a complete new line for future research.

Eventually, an important drawback is related to the disposal of the observed time instants. It is necessary to monitor the same number of curves at each instant of time to be able to construct our proposed statistic. This restriction translates into synchronous observations with $n_t = n$ points of the observed curves for all $t \in \mathcal{D}$. When the number of missed points is small, we can impute these using interpolation techniques. An example is given in Section 5.1.2. However, in a sparse context where one observes each curve in a different number of time points, and these measures may not agree (asynchronous pattern), it is not possible preprocessing the data to obtain our starting point. In conclusion, a new methodology is needed for these scenarios based on different dependence measures. We face this problem next, in Chapter 6. We develop new significance tests for the asynchronous version of the FCM using the CDC coefficient introduced in Section 4.2.3 of Chapter 4.

Chapter 6

New significance tests for the asynchronous functional concurrent model based on the CDC coefficient

The interest in covariates selection techniques in the FCM is motivated by the growth of functional or high-frequency data studies. In Chapter 5, new covariates selection approaches were proposed for the FCM. Nevertheless, these only work for synchronous time observations. This chapter presents other novel ideas to implement covariates selection techniques for the general version of the FCM in the asynchronous context. These selection procedures are implemented through conditional independence tests, using the CDC coefficient introduced in Section 4.2.3. The chapter is organized as follows. The asynchronous FCM is introduced in Section 6.1. In Section 6.2, new ideas for significance tests are presented, justifying their good behavior. Next, a simulation study is implemented to test their performance in Section 6.3. Eventually, some conclusions arise in Section 6.4.

6.1 The asynchronous FCM

The FCM introduced in Section 5.1, and given by expression (5.1), states that given two functional variables $Y(t) = (Y_1(t), \ldots, Y_q(t)) \in \mathbb{R}^q$ and $X(t) = (X_1(t), \ldots, X_p(t)) \in \mathbb{R}^p$, with $q, p \geq 1$ and some $t \in \mathcal{D}$, their relation is concurrent or point by point. This relation is given by a function $m(t, X(t))$, which is unknown.

In practice, a total of n curves of the form $\{\mathbf{Y_i(t)}, \mathbf{X_i(t)}\}_{i=1}^n$ are registred as independent realizations of $\{Y(t), X(t)\}$. Nevertheless, only part of the curve‘s trajectory can be observed. If, for each of the $i = 1, \ldots, n$ curves, there is information in a total of $u_i = 1, \ldots, \mathcal{T}_i$ different time points, having $\mathcal{T}_i > 1$ instant values for each curve, the data translates into $\{\mathbf{Y_i(t_{iu_i})}, \mathbf{X_i(t_{iu_i})}\}_{u_i=1}^{\mathcal{T}_i}$. Thus, the two possible scenarios can be classified in the synchronous case, understanding that it is assumed $\mathcal{T}_i = \mathcal{T}$ and $t_{iu} = t_{ku}$ for all $i, k = 1, \ldots, n$ and $u = 1, \ldots, \mathcal{T}$; or in the asynchronous one, when $\mathcal{T}_i$ can differ in terms of i, and it is not always verified that $t_{iu_i} = t_{ku_k}$ for $i \neq k$. The first case also contains scenarios where some points are missed, but these can be recovered using some interpolation technique. In a word, in the synchronous class, one is able to obtain all curve values for some $\{t_u\}_{u=1}^{\mathcal{T}}$ instants. In contrast, each curve can be observed at different time points in the asynchronous framework, obtaining asynchronous grids. This last consideration translates into varied t_{iu_i} values. Moreover, another different classification can be done based on the total number $N = \sum_{i=1}^n \mathcal{T}_i$ of observed points, differentiating between functional or longitudinal nature. Some comments and references about this topic arise in Section 5.1.

In real-world scenarios, the difference between synchronous and asynchronous cases

lies in the data collection procedure. Thus, synchronous observations can be obtained when a suitable device is provided to monitor functional values or the data is measured continuously. Examples of this situation are the use of monitoring devices in a hypertension study (see Goldsmith and Schwartz (2017)), clinical studies where data collection is completed quickly, such as gait disturbance studies (Kim et al. (2018), Ghosal and Maity (2022a)), countries data collected over the years (Wang et al. (2017), Ghosal and Maity (2022a)) or data provided by weather stations (Ospína-Galindez et al. (2019)). In contrast, asynchronous versions tend to appear in clinical studies where the data collection requires patient implication. Therefore, these data sets are often the result of information provided by medical checkups. As the dates of medical checkups differ from one patient to another, the data is collected at different time points. Additionally, a different number of checks is possible for each patient in terms of medical necessity or availability of the patient. This last results in curves with a different number of points between them. In the concurrent model framework, some examples of the asynchronous version are AIDS studies (Xue and Zhu (2007), Jiang et al. (2011)) or modeling of Alzheimer's disease (Wang et al. (2017)), to say a few. As it is expected that not much information will be provided for a given instant, it is in asynchronous cases where an extra effort has to be made. As expected, not much information is provided for each given moment, so an extra effort must be made in the asynchronous case. In this chapter, we develop new covariates selection procedures under asynchronous design. For this purpose, we assume that the observed time grid is dense enough to borrow information from neighbors at a given time point. Under these assumptions, we employ new nonparametric techniques for specification testing on the $m(\cdot)$ function of (5.1) for the asynchronous FCM case. In particular, ideas about novel significance tests are provided to determine which covariates are relevant in the regression model explanation.

In terms of the general formulation displayed in (5.1), no structure of the regressor function $m(\cdot)$ is assumed. This flexibility contrasts with the assumptions of most of the existing literature, where it is usual to consider some formulation, like linearity, and work under this premise. A discussion about the effort made in concurrent model estimation for different structures can be seen in Maity (2017). As a result, under no model assumption, a proper estimator in an asynchronous scenario will depend on a bandwidth parameter h for both: allow flexibility and use neighbors' information. Thus, a preliminary screening step determining if all p explanatory covariates $\{X_1(t), \ldots, X_p(t)\}$ are relevant, or we can exclude some from the model, is desirable for problem dimensionality reduction.

Then, to ensure the veracity of the model, it is necessary to verify if all p covariates $\{X_1(t),\ldots, X_p(t)\}$ are relevant. For this purpose, a global dependence test can be performed by means of testing

$$\begin{aligned} &H_0 : Y(t) \perp_{|_t} X(t) \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \\ &H_a : \mathbb{P}\left(Y(t) \not\perp_{|_t} X(t)\right) > 0 \quad \forall t \in \mathcal{P} \end{aligned} \tag{6.1}$$

where $\perp_{|_t}$ applies for conditional independence on t, $\mathcal{D} \setminus \mathcal{N}$ is the domain of t minus a null

set $\mathcal{N} \subset \mathcal{D}$ and $\mathcal{P} \subset \mathcal{D}$ is a nonnull set.

Here, the conditional independence denoted by $\perp_{|_t}$ is understood in terms of t. This translates into the condition $\varphi_{Y(t),X(t)|_t}(s,u) = \varphi_{Y(t)|_t}(s)\varphi_{X(t)|_t}(u)$, where $\varphi_{Y(t),X(t)|_t}(s,u)$ is the conditional joint characteristic function of $Y(t)$ and $X(t)$ and $\varphi_{Y(t)|_t}(s)$ as well as $\varphi_{X(t)|_t}(u)$ the marginal characteristic functions of $Y(t)$ and $X(t)$, respectively.

In this way, model (5.1) considering the p covariates only makes sense if one can reject the H_0 hypothesis of (6.1). Otherwise, the considered covariates do not supply relevant information to explain Y. It is notorious that formulation (6.1) collects a wide range of dependence structures between X and Y, conditioned to t. In addition, knowing the real form of $m(\cdot)$ is not necessary to determine when the effect of X is or is not significant. In conclusion, this formulation collects all types of conditional dependence patterns.

In this chapter, we focus on the development of new significance tests for the general concurrent model formulation under asynchronous design. For this purpose, a novel nonparametric statistic based on the conditional distance covariance coefficient (CDC) of Wang et al. (2015) is used. We refer the reader to Section 4.2.3 for more details about the CDC coefficient. This statistic has the novelty that preliminary model estimation is unnecessary in this framework. As a result, this tests procedure if the covariates have an effect on the explanation of Y no matter the underlying form of $m(\cdot)$. In contrast, other procedures, such as the ones of Wang et al. (2017) and Ghosal and Maity (2022a) in the linear formulation, or the work of Kim et al. (2018) for the additive structure, require the corresponding $m(\cdot)$ estimation to implement significance tests. Furthermore, the proposed procedure has the extra novelty that all conditional dependencies can be detected, including possible interactions. To the best of our knowledge, there is no literature for significance tests in the concurrent model considering any general formulation or possible interactions.

6.2 Significance tests based on CDC

From now on, we assume an asynchronous context where curves can be measured at different time points for each sample, having or not repetitive instants between them. Thus, for each curve $i = 1, \ldots, n$, there are a total of $\mathcal{T}_i > 1$ observed time points. This translates into samples of the form $\{(\mathbf{Y_i}(\mathbf{t_{iu_i}}), \mathbf{X_i}(\mathbf{t_{iu_i}}))\}_{u_i=1}^{\mathcal{T}_i}$ where $(\mathbf{Y_i}(\mathbf{t_{iu_i}}), \mathbf{X_i}(\mathbf{t_{iu_i}})) \in \mathbb{R}^q \times \mathbb{R}^p$ and $i = 1, \ldots, n$. A graphic example of the current situation considering $q = 1$ and $p = 2$ covariates for a FCM with a structure similar to (5.1) is displayed in Figure 6.1. In this case, there are $n = 5$ curves and, as an example, for instant t_1, a total of 2 points have been observed, whereas, for $t_{\mathcal{T}}$, a total of 3 applies.

As a result, putting all the data together and considering $N = \sum_{i=1}^{n} \mathcal{T}_i$, a new sample with triplet structure of length N can be obtained. This is given by

$$\left\{ \left\{ \left(\mathbf{Y_1}(\mathbf{t_{1u_1}}), \mathbf{X_1}(\mathbf{t_{1u_1}}), \mathbf{t_{1u_1}} \right) \right\}_{u_1=1}^{\mathcal{T}_1}, \ldots, \left\{ \left(\mathbf{Y_n}(\mathbf{t_{nu_n}}), \mathbf{X_n}(\mathbf{t_{nu_n}}), \mathbf{t_{nu_n}} \right) \right\}_{u_n=1}^{\mathcal{T}_n} \right\}.$$

Here $(\mathbf{Y_i}(\mathbf{t_{iu_i}}), \mathbf{X_i}(\mathbf{t_{iu_i}}), \mathbf{t_{iu_i}}) \in \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^1$ for all $i = 1, \ldots, n$ and $u_1 = 1, \ldots, \mathcal{T}_i$.

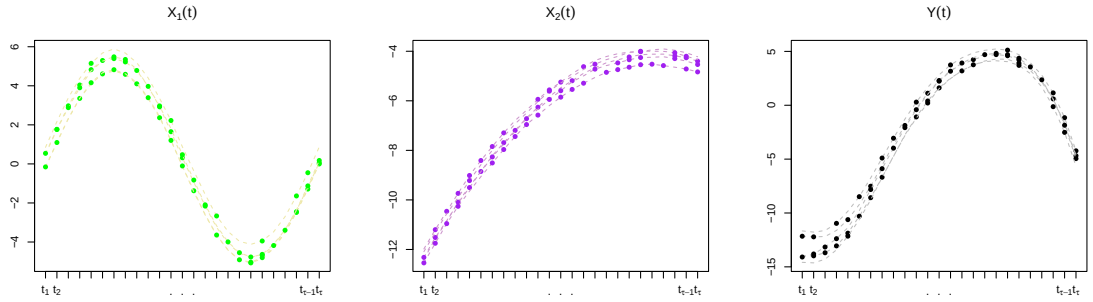


Figure 6.1: Example of a sample of five curves measured at different time instants $\{t_u\}_{u=1}^{\mathcal{T}} \in \mathcal{D}$ considering $p = 2$ covariates ($X_1(t)$ and $X_2(t)$) to explain $Y(t)$. Filled points simulate the total observed points for each curve.

This last can also be understood as

$$\begin{aligned}
&\text{curve } i = 1: \quad \Big(\mathbf{Y_1}(\mathbf{t_{11}}), \mathbf{X_1}(\mathbf{t_{11}}), \mathbf{t_{11}}\Big), \quad \dots \quad , \Big(\mathbf{Y_1}(\mathbf{t_{1\mathcal{T}_1}}), \mathbf{X_1}(\mathbf{t_{1\mathcal{T}_1}}), \mathbf{t_{1\mathcal{T}_1}}\Big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\text{curve } i = n: \quad \Big(\mathbf{Y_n}(\mathbf{t_{n1}}), \mathbf{X_n}(\mathbf{t_{n1}}), \mathbf{t_{n1}}\Big), \quad \dots \quad , \Big(\mathbf{Y_n}(\mathbf{t_{n\mathcal{T}_n}}), \mathbf{X_n}(\mathbf{t_{n\mathcal{T}_n}}), \mathbf{t_{n\mathcal{T}_n}}\Big)
\end{aligned}$$

Henceforth, for sake of simplicity, notation $\mathbf{W_N}$ is considered for samples, where $\mathbf{W_N} = \{(\mathbf{Y_1}, \mathbf{X_1}, \mathbf{t_1}), \dots, (\mathbf{Y_N}, \mathbf{X_N}, \mathbf{t_N})\}$ denotes the set of all N triplets.

At this point, it is possible to resort to Wang et al. (2015) techniques over the N-dimensional sample vector to perform (6.1) using the CDC coefficient introduced in Section 4.2.3. The resulting test is given by

$$\begin{aligned}
&H_0 : CDC^2(Y(t), X(t)|_t) = 0 \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \\
&H_a : \mathbb{P}\left(CDC^2(Y(t), X(t)|_t) \neq 0\right) > 0 \; \forall t \in \mathcal{P}
\end{aligned}$$

which, considering that $CDC^2(Y(t), X(t)|_t) \geq 0$, is equivalent to

$$\begin{aligned}
&H_0 : \int_{\mathcal{D}\setminus\mathcal{N}} CDC^2(Y(t), X(t)|_t)\omega(t)f(t)dt = 0 \text{ almost surely} \\
&H_a : \mathbb{P}\left(\int_{\mathcal{P}} CDC^2(Y(t), X(t)|_t)\omega(t)f(t)dt \neq 0\right) > 0
\end{aligned} \tag{6.2}$$

being $\omega(t)$ a weighting function and $f(t)$ the density function of t.

A first interpretation of the expression for the statistic displayed in (6.2) is that this is just a weighted expectation of the form $\mathbb{E}\left[CDC^2(Y(t), X(t)|_t)\omega(t)\right]$. The weighting function $\omega(t)$ represents the relevance of each given $t \in \mathcal{D}$ value. As a result, this can be selected as an expression of the density function. Wang et al. (2015) propose to take $\omega(t) = 12f^4(x)$ because easy calculation, but another choice is possible. Once a $\omega(t)$ function is established and suitable estimators of $CDC^2(Y(t), X(t)|_t)$ as well as $f(t)$ are

achieved, a statistic based on these terms can be employed to implement the conditional independence test given in (6.2). Without loss of generality, one can always set $\mathcal{D} = [0, 1]$ just rescaling the $\mathcal{D}$ domain, which translates into integrating into the $[0, 1]$ interval. In terms of $f(t)$, there are two options: knowing the real distribution in advance, for example, that time points follow a uniform distribution in $t \in [0, 1]$ ($f(t) = 1$), or to estimate this employing non-parametric techniques and plug-in its estimation. On the other hand, a $\mathcal{U}$-statistic of order 4 can be obtained for $CDC^2(Y(t), X(t)|_t)$, given a t value, following calculation displayed in Section 4.2.3. Nevertheless, as in the vectorial case, this estimator has a pretty high computational cost. Consequently, simpler estimators are useful to lower the computational burden. The construction of a more tractable version is discussed below in Section 6.2.1, resulting in the estimator given in equation (6.3). As a result, this last is employed to implement the test.

Moreover, a kernel function, as well as a bandwidth value, are involved in the CDC estimation. Then, a suitable selection of the kernel function and the bandwidth parameter is needed to estimate this coefficient correctly. A discussion about kernel and bandwidth selection is carried out in Section 6.2.2.

For a proper selection of the bandwidth parameter and density estimator $\hat{f}(t)$, Wang et al. (2015) develop an estimator for the corresponding $\mathbb{E}\left[CDC^2(Y(t), X(t)|_t)12f^4(t)\right]$ quantity in the vectorial framework and proved its asymptotic normality. See Theorem 7 of Wang et al. (2015). In particular, considering the N triplets structure given by $\mathbf{W_N}$, this result also extends to our context. As a result, under some assumptions, the asymptotic normality of the statistic developed for the asynchronous FCM version is guaranteed. However, the variance term associated with this distribution is difficult to obtain in practice. Moreover, the convergence to this normal distribution may be slow in practice. Therefore, a bootstrap procedure is an appealing alternative to calibrating the statistic distribution. Following Wang et al. (2015) ideas, the local bootstrap of Paparoditis and Politis (2000) is adapted to the asynchronous FCM context to obtain the test p-value in practice. This results in the scheme proposed in Algorithm 6.1.

Algorithm 6.1 (Local bootstrap scheme for significance tests using CDC)**.** Given a kernel function $K(\cdot)$ and some proper bandwidth parameter h:

1. For $i = 1 \ldots, N$ estimate $CDC^2\left(Y(t), X(t)|_{t=t_i}\right)$ by means of $\mathcal{V}_N(t_1), \ldots, \mathcal{V}_N(t_N)$ as defined in expression (6.3).

2. Approximate the sample statistic $E = \int_{\mathcal{D}\setminus\mathcal{N}} CDC^2(Y(t), X(t)|_t)\omega(t)f(t)dt$ by means of numerical techniques using $\{\mathcal{V}_N(t_1), \ldots, \mathcal{V}_N(t_N)\}$.

3. For $i = 1 \ldots, N$, draw Y_i^* and X_i^* from the Nadaraya-Watson estimators of the distribution functions given by

$$\hat{F}_{Y|_{t=t_i}}(y) = \frac{\sum_{l=1}^{N} K_h(t_i - t_l)\mathbb{I}_{(-\infty, Y_l]}(y)}{\sum_{l=1}^{N} K_h(t_i - t_l)} \quad \text{and} \quad \hat{F}_{X|_{t=t_i}}(x) = \frac{\sum_{l=1}^{N} K_h(t_i - t_l)\mathbb{I}_{(-\infty, X_l]}(x)}{\sum_{l=1}^{N} K_h(t_i - t_l)},$$

respectively, being $\mathbb{I}(\cdot)$ the indicator function. Roughly speaking, each observed value

Y_l or X_l, where $l = 1, \ldots, N$, has a probability $K_h(t_i - t_l)/\sum_{l=1}^{N} K_h(t_i - t_l)$ to be chosen as the i-th bootstrap sample.

4. For $i = 1, \ldots, N$ obtain $\mathcal{V}_N^*(t_1), \ldots, \mathcal{V}_N^*(t_N)$ by expression (6.3) using the local bootstrap sample $\mathbf{W_N^*} = \{(\mathbf{Y_1^*}, \mathbf{X_1^*}, \mathbf{t_1}), \ldots, (\mathbf{Y_N^*}, \mathbf{X_N^*}, \mathbf{t_N})\}$.

5. Approximate the bootstrap statistic $E^* = \int_{\mathcal{D}\setminus\mathcal{N}} CDC^{2*}(Y(t), X(t)|_t)\omega(t)f(t)dt$ making use of $\{\mathcal{V}_N^*(t_1), \ldots, \mathcal{V}_N^*(t_N)\}$.

6. Repeat steps 3-5 a number B of times obtaining $\{(E^*)^{(1)}, \ldots, (E^*)^{(B)}\}$.

7. Compute the bootstrap p-value as $\frac{1}{1+B}\left(1 + \sum_{b=1}^{B} \mathbb{I}\{(E^*)^{(b)} \geq E\}\right)$.

For the asynchronous FCM case, a local bootstrap, resampling in both variables Y and X, has been proposed. In contrast, Wang et al. (2015) use the local bootstrap of Paparoditis and Politis (2000) for the vectorial framework, just resampling in one variable. This modification is motivated by the fact that, in this context, ties may be expected in the conditioned variable (t), although this is not usual in the vectorial framework. Therefore, it is likely to have different values for $Y(t)$ and $X(t)$ given some value of t. As a result, resampling in both variables seems more appropriate for the asynchronous FCM context than resampling only in one. An illustrative comparison between both procedures and through a simulation study is carried out in Section 6.3.1. Moreover, other resampling techniques are available to calibrate the CDC-based test correctly. An alternative for calibration is the use of permutations. The performance of this option is displayed in Section D.3 of Appendix D for the simulation scenario studied in Section 6.3.1.

In this procedure (Algorithm 6.1), the same bandwidth parameter, h, is employed for estimation as well as resampling, similar to Wang et al. (2015) procedure. However, we propose a different criterion for bandwidth selection from the naive rule-of-thumb that is intended for density estimation and applied in Wang et al. (2015). Instead, a search of h considering different values in the $\mathcal{D}$ domain is performed. As mentioned above, one can rescale the domain to the range $[0, 1]$ with no loss of generality and search there. This new search is motivated since automatic selection rules based on density estimation have displayed bad behavior for this FCM context. Some examples of this phenomenon are displayed below in Section 6.3.

6.2.1 Estimation of CDC in practice

Following guidelines introduced in Section 4.2.3, we derive a proper $\mathcal{V}$-statistic (respectively, $\mathcal{U}$-statistic) of order 4 to implement the conditional independence test displayed in (6.2). Nevertheless, this statistic requires the order of $\mathcal{O}(N^4)$ calculations for a given conditional value. This cost results in $\mathcal{O}(N^5)$ operations when all t values are considered. As a result, considering $N = 20$ it is needed $3.2 \cdot 10^6$ operations and for $N = 100$, a total of 10^{10}. Since, in the concurrent model, N denotes the total number of observed points considering all curves, one expects greater values than $N = 100$. This complexity can result in intractable situations. Thus, a low-cost statistic is needed, especially using bootstrap techniques to

obtain p-values in practice. For this aim, it is possible to resort to a weighted version of the distance covariance estimator of Székely et al. (2007) to estimate (4.24). This is given by the expression

$$\mathcal{V}_N(t) = \frac{\sum_{l=1}^{N}\sum_{m=1}^{N} A_{lm|_t} B_{lm|_t} K_h(t-t_l) K_h(t-t_m)}{\left(\sum_{l=1}^{N} K_h(t-t_l)\right)^2} \tag{6.3}$$

where

$$A_{lm|_t} = a_{lm|_t} - \bar{a}_{l\cdot|_t} - \bar{a}_{\cdot m|_t} + \bar{a}_{\cdot\cdot|_t}$$
$$B_{lm|_t} = b_{lm|_t} - \bar{b}_{l\cdot|_t} - \bar{b}_{\cdot m|_t} + \bar{b}_{\cdot\cdot|_t}$$

for $l, m = 1, \ldots, N$ and where

$$\bar{a}_{l\cdot|_t} = \frac{\sum_{i=1}^{N} a_{li|_t} K_h(t-t_i)}{\sum_{i=1}^{N} K_h(t-t_i)}, \quad \bar{a}_{\cdot m|_t} = \frac{\sum_{i=1}^{N} a_{mi|_t} K_h(t-t_i)}{\sum_{i=1}^{N} K_h(t-t_i)},$$
$$\bar{a}_{\cdot\cdot|_t} = \frac{\sum_{i=1}^{N} K_h(t-t_i)\left(\sum_{j=1}^{N} a_{ij|_t} K_h(t-t_j)\right)}{\left(\sum_{i=1}^{N} K_h(t-t_i)\right)^2}$$

and similar for $\bar{b}_{l\cdot|_t}$, $\bar{b}_{\cdot m|_t}$ and $\bar{b}_{\cdot\cdot|_t}$, being $a_{lm|_t} = \|X_l(t) - X_m(t)\|_p$ and $b_{lm|_t} = \|Y_l(t) - Y_m(t)\|_q$, where $\|\cdot\|_p$ and $\|\cdot\|_q$ denote the euclidean norms of $\mathbb{R}^p$ and $\mathbb{R}$, respectively.

Then, the estimator $\mathcal{V}_N(t)$ of (6.3) results in a $\mathcal{V}$-statistic of order 2. This calculation translates into $\mathcal{O}(N^3)$ operations, reducing the computational cost and resulting in a cheaper alternative. Other authors have already applied this idea in the vectorial framework. For example, Wang et al. (2015) use this type of estimator to perform their CDIT criterion.

6.2.2 Kernel function and bandwidth selection

For the $CDC^2(Y(t), X(t)|_t)$ estimation, a kernel function, as well as a proper bandwidth parameter, are needed. It is remarkable that now, in the FCM case, the conditional covariate is $t \in \mathcal{D} \subset \mathbb{R}$, then, both kernel and bandwidth are one-dimensional.

In the vectorial framework, Wang et al. (2015) choose the Gaussian kernel to construct their statistic. This selection is because this criterion is easy to extend to the multidimensional case. For this purpose, they consider a diagonal bandwidth matrix H with the same h value. Furthermore, using the Gaussian kernel, $\omega(t)/N$ results in a consistent density estimator under some regularity conditions. We refer the reader to Wang et al. (2015) for more details. In their study, the values of the conditioned variable, $Z \in \mathbb{R}^r$, are assumed to be random. Then, to avoid randomness from their statistic, they consider the expectation of the conditional distance covariance weighted in terms of the Z values. As a result, a proper density estimator is required, and this last condition is desirable. Nevertheless, different situations may arise for the FCM, such as knowing the distribution of t in advance. An example of this situation is when one knows that the points of the rescaled functional curves follow some distribution in $[0, 1]$ concerning time, but only some points are recorded

because of the capability of the monitoring device.

Furthermore, it is possible to consider observed time instants $\{t_i\}_{i=1}^N$ as deterministic in the sense that measures of variables $Y(\cdot)_1, \ldots, Y(\cdot)_q$ and $X_1(\cdot), \ldots, X_p(\cdot)$ are obtained in fixed time points. An example of this situation may appear in medical checkups where these are prespecified with a given distance between medical appointments. Some examples can be found in Xue and Zhu (2007) or Jiang et al. (2011), in AIDS studies, or in Wang et al. (2017) for Alzheimer's disease. Thus, one can avoid the expectation operator interpretation in (6.2) because of the lack of randomness in t. As a result, all the information can be collected employing an integral approximation for all time points. Moreover, given a particular instant and due to the FCM nature, it is natural to consider only neighbors' information. For this reason, a compact kernel seems to be a better option than the Gaussian one for the asynchronous FCM. Henceforth, we employ the uniform or rectangular kernel for our study, which is given by the expression

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right) = \frac{1}{2h} \mathbb{I}\{|x/h| \leq 1\}.$$

Other options for compact kernels, such as triangular or Epanechnikov kernels, would also be possible. A representation of these functions is collected in Figure 1.2 of Chapter 1.

Furthermore, as is shown below in Section 6.3, bandwidth selection criteria, based on density estimation, do not perform well for the FCM framework. In particular, this fact is illustrated by considering the rule-of-thumb and the unbiased cross-validation approaches for density estimation. As a result, another criterion is necessary. In practice, we propose to prove with different values of $h \in \mathcal{D}$, respectively in $[0, 1]$, for statistic calculation and extract conclusions based on these results. Next, this idea is implemented and tested for the asynchronous FCM model through a simulation study in Section 6.3.

6.3 Simulation studies

In this section, we consider two simulation scenarios for assessing the performance of the CDC-based tests displayed in (6.2). These are a linear (Scenario A) and a nonlinear (Scenario B) formulation of the model (5.1) in the asynchronous framework. For this purpose, a Monte Carlo study with $M = 500$ replicas is performed using the R software (R Core Team (2019)).

In particular, Scenarios A and B introduced in Section 5.4 are employed. Now, it is assumed that, for each curve $i = 1, \ldots, n$, a total of $\mathcal{T}_i = 4$ different time instants are observed in $\mathcal{D}_t = [0, 1]$, taking sample sizes of $n = 20, 60, 100$. These time points are randomly generated following a uniform distribution.

Then, in both frameworks, the density function is assumed to be known, following a $U[0, 1]$. Thus, as $f(t) = 1$, it is defined $\omega(t) = 1$ for all $t \in [0, 1]$. If this information is not available, the density function can be easily estimated using nonparametric techniques. Nevertheless, it is assumed in simulations that the density distribution is known in advance just for the sake of simplicity.

We use the local bootstrap of Algorithm 6.1 to approximate the p-values, employing $B = 500$ resamples in each case. The sample size and power of the test are obtained using Monte Carlo techniques. To ensure that the p-values under the null take on a suitable value, we calculate the 95% confidence intervals of the significance levels using the expression $\left[\alpha \mp 1.96\sqrt{\frac{\alpha(1-\alpha)}{M}}\right]$. Here α is the expected level, and $M = 500$ is the number of Monte Carlo simulated samples. Thus, we consider that a p-value is acceptable for levels $\alpha = 0.01, 0.05, 0.1$ when this is within the values collected in Table 6.1. Then, the resulting p-values outside these scales for simulation results are highlighted in **bold**.

M	$\boldsymbol{\alpha = 0.01}$	$\boldsymbol{\alpha = 0.05}$	$\boldsymbol{\alpha = 0.1}$
500	$[0.001, 0.019]$	$[0.031, 0.069]$	$[0.074, 0.126]$

Table 6.1: Confidence intervals of the Monte Carlo proportions at different levels for a total of $M = 500$ replicates.

6.3.1 Results for scenario A (linear model)

The behavior of the CDC-based test for asynchronous FCM linear formulation is analyzed in the first place. For this purpose, Scenario A of Section 5.4 is employed. Three different simulation frameworks are considered: simulating under the null hypothesis of conditional independence ($\beta_1(t) = \beta_2(t) = 0$) and the other two violating this through conditional dependence on X_2 ($\beta_1(t) = 0, \beta_2(t) \neq 0$) or in both covariates ($\beta_1(t) \neq 0, \beta_2(t) \neq 0$). The test is calibrated employing the local bootstrap introduced above in Section 6.2.

As mentioned above in Section 6.2.2, the rectangular kernel is employed, and a proper bandwidth needs to be picked. This selection translates into a value that guarantees that the test is well-calibrated under the null hypothesis of conditional independence. To select this term, we start proving with a wide grid of values along $[0, 1]$, taking $h = 0.1, 0.2, \ldots, 0.8$. Next, we determine the most optimal value of those considered in the grid and refine the search using a more refined grid around this quantity. A criterion to select this first value is to choose an h value whose associated p-values distribution resembles the $U[0, 1]$ under the null hypothesis. Eventually, the most suitable value in the new denser grid is picked based on this criterion. These selections can be made formally using some uniformity GoF test. We perform Kolmogorov-Smirnov tests[1] in this study. It is important to remark that, as expected for the optimal h quantity to vary in terms of the sample size, n, this procedure is performed for all different considered values of n. Besides, the rule-of-thumb estimator, h_{RoT}, and the unbiased cross-validation criteria for density estimation, h_{UCV}, are also considered to show the malfunction of automatic bandwidths.

For all considered sample sizes (n=20, 60, 100), in the first search, one can realize that the optimal value seems near $h = 0.5$. An example of this situation, for $n = 100$, can be appreciated in Figure D.2 of Section D.1 in the Appendix D in terms of the obtained

[1] For this purpose, we have employed the `ks.test(·,"punif")` test of the base package `stats` of R (R Core Team (2019)).

p-values histograms. As a result, we narrow down the search between the interval $(0.4, 0.6)$ taking $h = 0.42, 0.44, 0.46, 0.48$ and $h = 0.52, 0.54, 0.56, 0.58$. The percentage of rejections for significance values $\alpha = 0.01, 0.05, 0.1$, for the considered h quantities, are displayed in Figure D.1 in Section D.1 of the Appendix D. There, it is appreciated as optimal calibration seems to happen when h is near 0.5. In fact, we got that suitable values in the considered grid are $h = 0.48, 0.46, 0.5$ for $n = 20, 60, 100$, respectively. These quantities obtain associated p-values of 0.103, 0.6728, and 0.3195 in the uniformity Kolmogorov-Smirnov test. As a result, there is no evidence to reject the null hypothesis of uniformity for low significance levels for these bandwidth values. Then, these h values can be employed to perform the CDC-based global test. Results using these tuning parameters are collected in Table 6.2. It is possible to appreciate as the test is well calibrated, having all values between the 95% confidence interval.

Model:		$\boldsymbol{\beta_1(t) = \beta_2(t) = 0}$ (H_0)			$\boldsymbol{\beta_1(t) = 0, \beta_2(t) \neq 0}$ (H_a)			$\boldsymbol{\beta_1(t) \neq 0, \beta_2(t) \neq 0}$ (H_a)		
h	**n**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
0.48	20	0.010	0.034	0.080	1	1	1	1	1	1
0.46	60	0.002	0.048	0.112	1	1	1	1	1	1
0.5	100	0.008	0.052	0.118	1	1	1	1	1	1

Table 6.2: Empirical sizes and powers of the CDC-based global test for conditional dependence testing using a local bootstrap approximation with $B = 500$ resamples in Scenario A for fixed bandwidth values (h) and a total of $N = n \cdot 4$ sample points.

Regarding the rule-of-thumb estimator, h_{RoT}, and the one based on the unbiased cross-validation criteria, h_{UCV}, one observes a pretty poor performance for both in practice. For both parameters, the test is always poorly calibrated. An example of this last fact is shown in Figure D.2 of Section D.1 in the Appendix D for sample size $n = 100$. The optimal bandwidth values selected concerning density estimation are quite small, far away from optimal values for calibration. Besides, it is appreciated as their p-values do not follow a uniform distribution. In fact, p-values$< 2.2 \cdot 10^{-16}$ are obtained for both cases in the Kolmogorov-Smirnov test of uniformity.

Once a bandwidth value is selected, verifying that the test is well-calibrated under H_0 in each case, it must be verified if the test has power under the alternative hypothesis. For this aim, two different conditional dependence scenarios are simulated: having only dependence on X_2 $(\beta_1(t) = 0, \beta_2(t) \neq 0)$ or in both covariates $(\beta_1(t) \neq 0, \beta_2(t) \neq 0)$. It can be seen in Table 6.2 as the test is very powerful for both scenarios, always getting a percentage of rejections equal to the unit. As a curiosity, it is interesting to mention that, for all considered bandwidth values in $[0, 1]$, we always appreciate a really high power simulating under the alternative hypothesis.

Finally, the local bootstrap approach applied by resampling in both covariates compares with the one employed by Wang et al. (2015), using the version of Paparoditis and Politis (2000). Then, the local bootstrap scheme introduced in Algorithm 6.1 is implemented, resampling only the response variable Y. Calibration results for different bandwidth values

$h = 0.1, 0.2, \ldots, 0.8$ are displayed in Figures D.6 and D.5 in Section D.2 of the Appendix D. Given the results, one can appreciate as the optimal value for h seems to change in this case. In particular, the test is well-calibrated for bandwidth parameter values close to 0.3. Then, it is possible to conclude that a calibration resampling only on Y is also adequate. However, because of the FCM nature, it seems more reasonable to resample on both covariates. Moreover, these results bring out the fact that the bandwidth selection for density function criterion, like the rule-of-thumb (h_{RoT}) or unbiased cross-validation (h_{UCV}), performs poorly in this context again.

Furthermore, a different point of view on implementing permutations to calibrate the test is displayed in Section D.3 of the Appendix D.

6.3.2 Results for scenario B (nonlinear model)

Next, we analyze the performance of the CDC-based tests in a nonlinear framework. For this purpose, the Scenario B of Section 5.4 is implemented. Details about its implementation in the asynchronous case are given above, in Section 6.3. Following similar guidelines as for the linear case (Scenario A considered in Section 6.3.1), three different contexts are studied: simulating under conditional independence ($F_1(\cdot) = F_2(\cdot) = 0$), when there is only an important covariate ($F_1(\cdot) = 0, F_2(\cdot) \neq 0$) and when both are relevant ($F_1(\cdot) \neq 0, F_2(\cdot) \neq 0$). As a result, the first option is useful for detecting if the test is well-calibrated, and the remaining ones are for the power study related to the detection of alternatives.

Again, we consider in first place values of $h = 0.1, 0.2, \ldots, 0.8$, and a posterior narrower search is done accordingly with the optimal values obtained. It can be seen in Figure D.4, collected in Section D.1 of the Appendix D, as the optimal values are around $h = 0.5$. In fact, for all sample sizes, the value which obtains the best results for the Kolmogorov-Smirnov test of uniformity is $h = 0.5$. For sample sizes $n = 20, 60, 100$, p-values of 0.051, 0.129 and 0.283 were obtained, respectively. An example of how the p-values resemble a uniform distribution is appreciated for $n = 100$ in Figure D.3 in Section D.1 of the Appendix D. Thus, results considering $h = 0.5$ are summarized in Table 6.3.

Model:		$\boldsymbol{F_1(\cdot) = F_2(\cdot) = 0}$ (H_0)			$\boldsymbol{F_1(\cdot) = 0, F_2(\cdot) \neq 0}$ (H_1)			$\boldsymbol{F_1(\cdot) \neq 0, F_2(\cdot) \neq 0}$ (H_1)		
h	**n**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
0.5	20	0.012	0.036	0.090	1	1	1	1	1	1
0.5	60	0.008	0.066	**0.140**	1	1	1	1	1	1
0.5	100	0.008	0.050	0.118	1	1	1	1	1	1

Table 6.3: Empirical sizes and powers of the CDC-based global test for conditional dependence testing using a local bootstrap approximation with $B = 500$ resamples in Scenario B for fixed bandwidth values (h) and a total of $N = n \cdot 4$ sample points.

In view of the results, one appreciates that simulating under the H_0 hypothesis all values are within the confidence interval, except for $n = 60$ and $\alpha = 0.1$. This drawback solves by increasing the sample size. Besides, it can be seen in Figure D.4 in Section D.1 of the Appendix D as other selections as $h = 0.46$ or $h = 0.48$, arrange this. Again, the test

is very powerful, obtaining a percentage of rejections equal to one for both alternatives and all considered settings.

Thus, the CDC-based test performs well no matter the underlying structure of the FCM model displayed in (5.1).

6.4 Conclusions

We have proposed new ideas to perform significance tests in the asynchronous FCM with the global formulation. In particular, given $p > 1$ covariates, it is possible to determine if these contain relevant information or can be discarded from the model adjustment, conditioned to the time variable. These global tests adapt the ideas of Wang et al. (2015) developed for the vectorial framework to the FCM. Specifically, the CDC coefficient introduced in Section 4.2.3 is used to test for conditional independence considering all available time points. As a result, given an instant t, this test employs the neighborhood information of the asynchronous version to calculate the statistic value. To the best of our knowledge, this is the first time that such a local approach to these characteristics has been proposed for the synchronous version of FCM. Moreover, this procedure has the novel advantage that no assumption about the model structure is needed, detecting all possible types of conditional dependence. In addition, this methodology allows us to consider a multivariate response, taking $Y(t) \in \mathbb{R}^q$ for $q \geq 1$ and $t \in \mathcal{D}$. Besides, under some assumptions, it is guaranteed that the distribution of the test statistic is asymptotically normal using Theorem 7 of Wang et al. (2015). Furthermore, some bootstrap schemes are proposed to calibrate this in practice. Specifically, an adaptation of the local bootstrap of Paparoditis and Politis (2000) is introduced in Section 6.2, and calibration using permutations is treated in Section D.3 of the Appendix D as an alternative. Other resampling procedures would be possible as well. The good behavior of the proposed global test is displayed through a simulation study in Section 6.3.

A suitable selection of a kernel function and a bandwidth parameter is needed to correctly estimate the CDC coefficient using the local character of the data. Estimation of the CDC coefficient and selection of the tuning parameters are discussed in Sections 6.2.1 and 6.2.2. In this case, paying attention to the FCM nature, the choice of a compact kernel seems the best option to keep the concurrent nature. In consequence, functions as the uniform kernel can be employed. A different hurdle is the correct choice of the bandwidth parameter. Wang et al. (2015) propose the use of the rule-of-thumb for density estimation. This results in an automatic selection of the h value. Nevertheless, this selection does not perform well for the asynchronous FCM. This drawback is proved through the simulation results of Section 6.3 employing the rule-of-thumb option and the optimal bandwidth for density estimation obtained by the unbiased cross-validation criterion. Given the result, a search in the t domain is proposed in practice to solve this drawback. In Sections 6.3.1 and 6.3.2, it is displayed as a proper value for the bandwidth can guarantee that the test is well calibrated under the null and. In contrast, this selection is powerful when simulating different alternatives. A criterion for the selection of the bandwidth value automatically is

still an open problem that needs further research.

A natural extension of the problem displayed in (6.2) is its adaptation for partial testing. Thus, one could test if the covariates in a subset $D \subset \{1, \ldots, p\}$, with cardinal equal or greater than one, can be assumed to be conditional independent or if some of its components are relevant in the model explanation. In particular, considering $D = \{j\}$, for some $j = 1, \ldots, p$, would allow performing covariates selection in the asynchronous FCM. Although the adaptation of expression (6.2) is straightforward, just considering $X_D(t)$ instead of $X(t)$, some problems concerning the proper bandwidth selection appear. In particular, we have noticed that selecting a suitable bandwidth for partial tests is a hard problem in practice. As a result, the extension to partial tests is an interesting open problem for future research.

An extra inconvenience of this procedure is the computational cost. As commented in Section 6.2.1, the operations required to estimate the CDC coefficient are of order $\mathcal{O}(N^5)$ or $\mathcal{O}(N^3)$ if the low-cost version is employed. This is still a high computational cost, and some alternatives to reduce this would be desirable. This drawback is a usual limitation of the dependence coefficients based on distances, introduced throughout Chapter 4, and as a result, this is an interesting topic for future research.

Results, conclusions and future work

This thesis project entitled "New covariates selection approaches in high dimensional or functional regression models" is devoted to studying and developing new covariates selection techniques in recent and challenging high dimensional or functional data contexts for regression models. Next, obtained results and conclusions are commented on, along with possible future work related to open lines of research.

Results and conclusions

The results and conclusions obtained in this thesis have been commented on throughout the different chapters of the manuscript. Specifically, details are given in the conclusion or discussion sections, respectively. We refer the reader to Section 1.3 for an overview of Chapter 1 and Section 2.6 for the resulting discussion of Chapter 2. Conclusions for the LASSO study under dependence are displayed in Section 3.1.4 and those for covariates with different scales under dependence scenarios in Section 3.2.4, both in Chapter 3. Related to Chapter 4, obtained results and conclusions arise in Section 4.3. Eventually, those concerning the synchronous and asynchronous versions of the FCM are collected in Sections 5.6 and 6.4 of Chapters 5 and 6, respectively. Next, a summary of these results is presented for each section.

Chapter 1: Problems of regression models in the high dimensional framework: the need for dimensionality reduction. Here, we develop a brief introduction to the main topic of the thesis. In particular, some motivation for the need for covariates selection in regression models, especially for high dimensional or complex frameworks, is given. For this aim, a review of the problems that appear in high dimensions is carried out, explaining their reasons and implications. These drawbacks motivate the studies performed in consecutive chapters.

Chapter 2: The Least Absolute Shrinkage and Selection Operator (LASSO). The Least Absolute Shrinkage and Selection Operator (LASSO) is the most employed penalization technique to adjust linear models when $p > n$. In particular, this penalization problem simultaneously allows both: to estimate the vector of parameters and perform covariates selection. Its convex formulation and attractive properties are introduced and analyzed in this chapter. Moreover, all its limitations as a covariates selector technique are also collected and studied in detail. We provide an extensive review of adaptations of the LASSO and alternative procedures designed to correct some of these problems. We analyze their properties, prominent advantages, and drawbacks from a critical point of view. Next, we propose some real examples to motivate the need for dimensionality reduction. Finally, this chapter closes with a discussion considering all the options studied.

This analysis results in a pretty comprehensive review of the LASSO properties, limitations, adaptations, and alternative procedures. This review is part of the content of the published article Freijeiro-González et al. (2022a).

Chapter 3: LASSO regression as a variable selector. Performance under dependence structures and different scales on covariates. We test the efficiency of the LASSO regression as a variable selector from novel points of view. Firstly, we examine the performance of this algorithm under different dependence structures through a simulation study. In view of the LASSO problems and limitations, we implement a comparison with proper adaptations and competitors. Based on the obtained results, some guidance is given about the best option based on the data nature, paying attention to its dependence structure. This simulation study results in the second part of the paper Freijeiro-González et al. (2022a).

Secondly, we add more complexity to the study. Now, there are considered distinct dependence frameworks and covariates in different scales. Again, we perform an extensive simulation study, comparing the performance of the LASSO with that of the competitors, following similar guidelines as those of the dependence study. This analysis finishes with a discussion about what to expect and which is the best procedure in these situations.

Next, a study to determine if performing a threshold, as in screening techniques, could be adequate to recover the relevant covariates is also included. Some conclusions arise based on the observed results.

Finally, we apply the considered competitors to some real datasets where covariates are correlated and have different scales. We consider the results and conclusions obtained from previous studies in their analysis.

Chapter 4: Novel distance-based dependence measures for complex data. A complete review of novel dependence measures based on distances is developed. In particular, we start exposing the weaknesses of the classical dependence measures and the necessity of new coefficients to measure dependence is motivated. Next, we carry out a detailed review of the distance covariance (DC), martingale difference divergence (MDD), and conditional distance covariance (CDC). Their coefficients expression, jointly with their properties, corresponding estimators, and possible extensions, are introduced. For this purpose, an exhaustive bibliographical review of the current literature performs, collecting all existing dependence coefficients based on distances. These coefficients allow one to apply covariate selection without any assumption about the structure of the model. As a result, in the last section, some conclusions about their use in complex models are given.

Chapter 5: New significance tests for the synchronous functional concurrent model based on the martingale difference divergence coefficient. New specification tests, particularly for significance testing, are developed for the synchronous version of the functional concurrent model (FCM). Global tests, as well as partial ones, are proposed to make use of the MDD coefficient. This results in the first procedure for covariates selection

in the FCM that does not need previous model estimation or tuning parameters. We obtain the asymptotic distribution of the test statistic and propose a bootstrap procedure to estimate the p-values in practice. Its good behavior is illustrated through a simulation study. Besides, their performance is compared to existing competitors, outperforming their results. Eventually, we test this approach employing three real data sets, and some conclusions are given based on the observed results. This study translates into the paper Freijeiro-González et al. (2022b).

Chapter 6: New significance tests for the asynchronous functional concurrent model based on the conditional distance covariance coefficient. A novel global significance test for the asynchronous version of the FCM is introduced. This procedure makes use of the CDC coefficient. We adapt the construction of the statistic to the FCM context and discuss a proper selection of the required kernel and bandwidth parameters. In particular, we show how automatic bandwidths that supposedly work well in the vector framework do not perform for the synchronous FCM. Then, we propose a new bandwidth selection procedure to solve this drawback. The asymptotic convergence of the statistic to a Gaussian distribution is guaranteed, and a local bootstrap is adapted to calibrate the test in practice. Its proper performance is displayed using a simulation study, and some conclusions arise in view of the results.

Future work

Apart from the obtained results and conclusions, some new topics that require further research have appeared during the development of this thesis. These are left as future lines of research. Next, these lines of work are detailed.

Study of penalization techniques and competitors for covariates selection in more general formulations. Penalization techniques have been studied under the linearity assumption in the structure of the vectorial regression model, given by (1.2), in Chapter 2. In particular, the L_1 type penalty has brought special attention in this framework, giving place to the LASSO regression. Nevertheless, one can extend the use of penalties for covariates selection to more general models. Thus, it is an interesting future line of research to study the application of penalization techniques in other regression structures, not only the linear case: additive effects, general linear model (logistic regression), or generalized additive model, to say a few. Therefore, a complete review of different covariates selection techniques for these models, including penalizations and other competitive approaches, would be interesting. Some comments about existing ideas related to the use of the L_1 penalty in these models arise in Section 2.6 of Chapter 2.

Performance of the new penalization techniques for more general formulations under dependence structures and different scales on covariates. Once we finish the collection of penalty techniques jointly with other suitable approaches devoted to covariates

selection in non-linear structures, it is interesting to compare their performance in practice under different scenarios. This study is motivated because the problems of the LASSO regression, or the use of the L_1 penalty, can also be inherited for more general structures. Then, similar to the work of Chapter 3, a study and analysis of the performance of these new proposed procedures under dependence structures and different scales on covariates is a future line of research.

The use of penalization techniques in the FCM. Another topic for future work is to apply penalization techniques in both synchronous and asynchronous versions of the FCM. In particular, a time-varying penalization $\lambda(t)$ can be assumed to detect if some of the studied covariates are relevant only in some nonnull time sets. If a linear structure of the FCM is assumed, the regression problem can be expressed as

$$\min_{\beta(t)} \ \|\mathbf{Y_n(t)} - \mathbf{X_n(t)}\beta(t)\|_2^2 + \lambda(t)\|\beta(t)\|_1,$$

and making use of the point-by-point nature, for each $t \in \mathcal{D}$, a problem of vectorial type has to be solved. Thus, comparing different penalization techniques' performance is interesting to find out how these approaches select covariates and how these perform when some consistency condition is violated. For example, if a model with only two out of four relevant covariates is generated by $Y(t) = \beta_1(t)X_1(t) + \beta_2(t)X_2(t) + \varepsilon(t)$, we want to know if some regularization technique can detect the relevant terms and exclude the noisy ones. A simulated example is displayed in Figure 6.2.

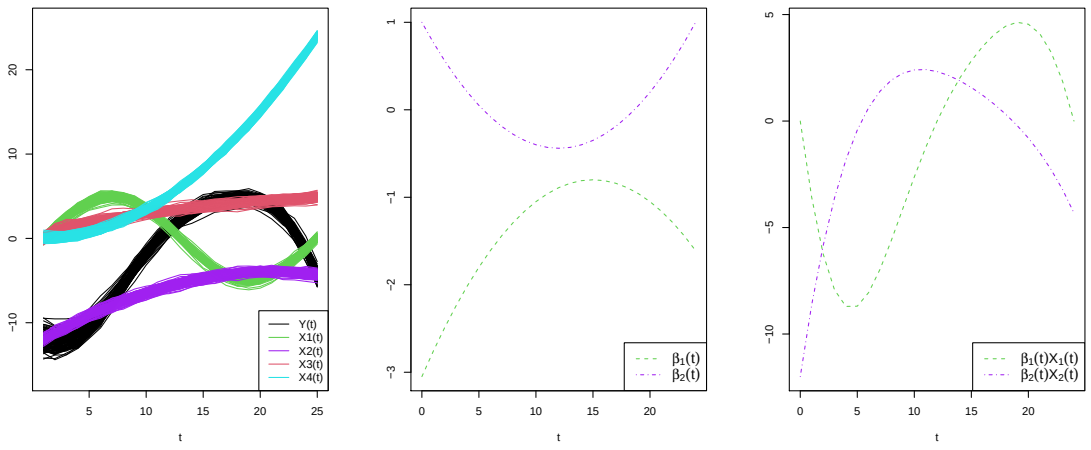


Figure 6.2: Left: simulated sample values of the functional variables along the grid $[0, 24]$ taking $n = 20$ curves. Middle: real partial effects corresponding to $X_1(t)$ ($\beta_1(t)$) and $X_2(t)$ ($\beta_2(t)$). Right: regression model components $\beta_1(t)X_1(t)$ and $\beta_2(t)X_2(t)$.

Results applying LASSO, AdapL.1se, and DC.VS techniques (see Chapters 2 and 3 for more details) are displayed in Figure 6.3 for this example.

Other ways of estimating the $\beta(t)$ function in the FCM are available. An example is

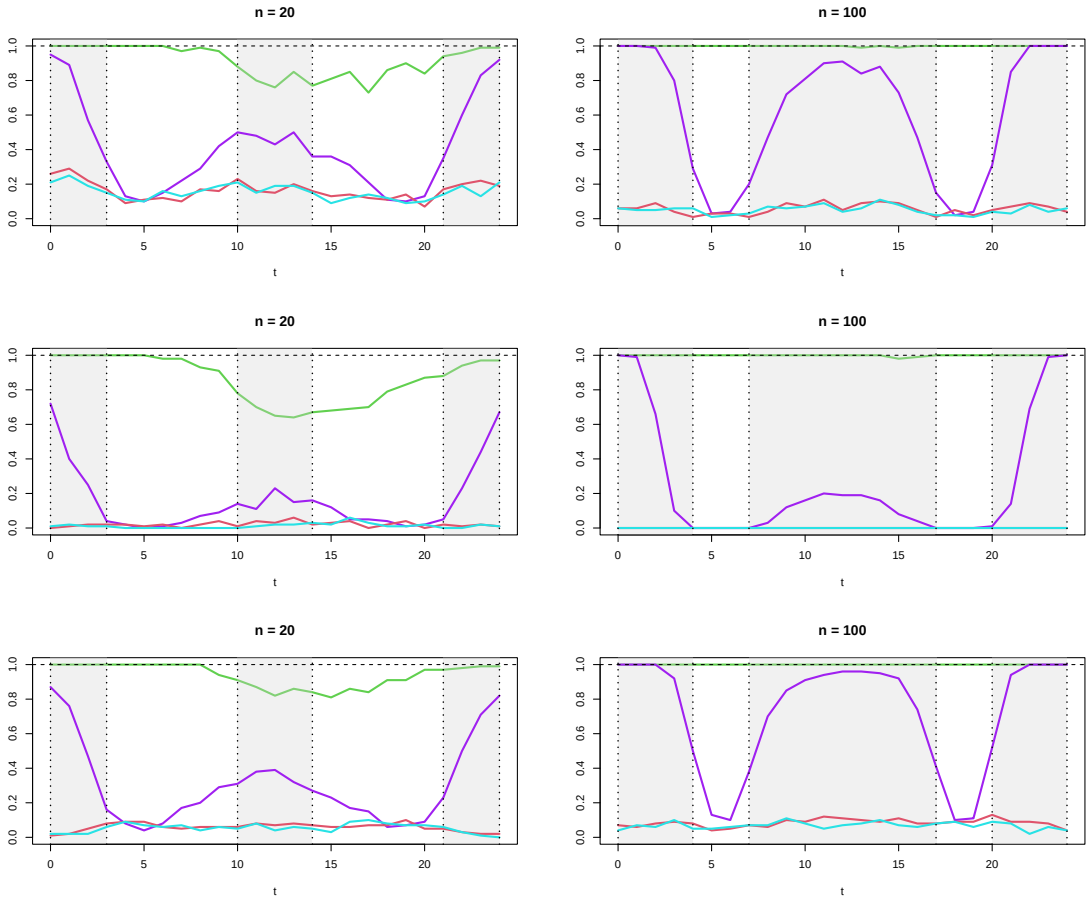


Figure 6.3: Percentage of times each covariate is selected over $M = 100$ simulations, taking $n = 20, 100$, by the LASSO (the first row), AdapL (the second row) and DC.VS (the third row). The vertical lines divide the gray zones where the $X_2(t)$ covariate verify the LASSO beta-min condition. The green curve corresponds with $X_1(t)$, the violet one with $X_2(t)$, the red with $X_3(t)$ and the cyan curve with $X_4(t)$.

the work of Wang et al. (2017), using local linear regression. In this setting, penalizations different from the L_1 type can be applied too. One can do this last by adapting ideas as the ones of Vidaurre et al. (2012), Lee et al. (2016), Fan (1997) or Yuan and Lin (2006) methodologies to the FCM.

Furthermore, more general structures of the regression model in the FCM case can be assumed. For these formulations, some penalties can be adapted to apply covariates selection as well. We consider this topic an interesting idea for future work.

Adaptation of the MDD-based significance tests of the synchronous FCM for additive quantile regression. In Chapter 5, new MDD-based tests for significance testing in the synchronous FCM are proposed. These are based on the MDD coefficient for conditional mean dependence presented in Chapter 4. In particular, we propose these tests

for the additive formulation of the FCM regression model $Y(t) = \sum_{j=1}^{p} F_j(X_j(t)) + \varepsilon(t)$, where the regressor function collects the information of the conditional expectation, i. e. $\mathbb{E}[Y(t)|_{X(t)}] = \sum_{j=1}^{p} F_j(X_j(t))$. However, information about some quantile $\mathbb{Q}[Y(t)|_{X(t)}]$ could be of interest as well, translating into the quantile regression context. One can adapt the ideas of Zhang et al. (2018) for quantile regression in the vectorial case for conditional quantile dependence testing in the FCM, following similar guidelines as the ones treated in Chapter 5. This proposal is a new and open line for future work.

Extension of the MDD-based significance tests of the synchronous FCM for the general formulation. The significance tests introduced in Chapter 5, based on the MDD coefficient studied in Chapter 4, apply to the additive formulation of the synchronous FCM. This is given by $Y(t) = \sum_{j=1}^{p} F_j(X_j(t)) + \varepsilon(t)$. Although this model is flexible and captures several types of relations, an even broader formulation is desirable. Specifically, significance tests for the general model $Y(t) = m(t, X(t)) + \varepsilon(t)$, introduced in (5.1), would be really useful. An idea to achieve this purpose is to resort to random projections. Thus, instead of capturing the relevance of the covariates "separately" in an additive way in the statistic of (5.7), these covariates are randomly projected. The resulting dummy variable is considered instead of the initial covariates. This approach avoids the sum effect of the MDD coefficients. If we launch random projections appropriately and repeat this procedure several times, the new statistic version considers the general FCM formulation. Again, this test could also calibrate using wild bootstrap, but the derivation of its asymptotic distribution is a more difficult problem that would need further research.

Development of new GoF tests for the synchronous FCM. Similar to the significance tests developed for the synchronous FCM in Chapter 5, new GoF tests can be proposed using the MDD coefficient. These result in testing

$$
\begin{aligned}
H_0 &\colon \mathbb{E}\left[\varepsilon(t)|_{X_j(t)}\right] = \mathbb{E}\left[\varepsilon(t)\right] \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \text{ and } \forall j \in D \\
H_1 &\colon \mathbb{P}\left(\mathbb{E}\left[\varepsilon(t)|_{X_j(t)}\right] \neq \mathbb{E}\left[\varepsilon(t)\right]\right) > 0 \quad \forall t \in \mathcal{P} \text{ and some } j \in D
\end{aligned}
$$

where $X_D(t)$ denotes the subset of $X(t)$ considering only the covariates with index in $D \subset \{1, \ldots, p\}$, $\mathcal{D} \setminus \mathcal{N}$ is the domain of t minus a null set $\mathcal{N} \subset \mathcal{D}$, $\mathcal{P} \subset \mathcal{D}$ is a positive measure set and $\varepsilon(t)$ is the error of the model.

This formulation is equivalent to the test displayed in (6.1) just changing $Y(t)$ by $\varepsilon(t)$. Here, if the error $\varepsilon = Y(t) - m(t, X(t))$ assumes some structure over the regressor function $m(\cdot)$, this results in a GoF test. In particular, this test can be rewritten as

$$
\begin{aligned}
H_0 &\colon \int_{\mathcal{D}\setminus\mathcal{N}} MDD^2(\varepsilon(t)|_{X_j(t)})dt = 0 \text{ almost surely for every } j \in D \\
H_1 &\colon \mathbb{P}\left(\int_{\mathcal{P}} MDD^2(\varepsilon(t)|_{X_j(t)})dt \neq 0\right) > 0 \text{ for some } j \in D
\end{aligned}
$$

where $MDD^2(\cdot)$ is the MDD coefficient introduced in Chapter 4.

Here the error term estimation under the GoF assumption is necessary to construct a suitable statistic. This process implies model estimation. Maity (2017) discuss techniques to estimate different formulations of the FCM. In particular, if an additive formulation is assumed, a similar statistic as the one displayed in (5.7) can be constructed considering $\hat{\varepsilon}(t)$. One can also calibrate this procedure using wild bootstrap techniques. Nevertheless, one can not directly resort now to the ideas of Zhang et al. (2018) to obtain the asymptotic distribution of the statistic. Then, it is not possible to guarantee its good behavior theoretically. Instead, the error structure has to be taken into consideration now. This development is a new topic for future research.

Construction of partial significance tests for the asynchronous FCM based on the CDC. In Chapter 6, a new global test based on the CDC coefficient treated in Chapter 4 is proposed to test conditional significance in the asynchronous FCM. This test applies to the general formulation of the FCM, detecting all types of conditional dependence between the response $Y(t) = (Y_1(t), \ldots, Y_q(t))$ and the $X(t) = (X_1(t), \ldots, X_p(t))$ covariates, for $q, p \geq 1$. Furthermore, one can extend the employed ideas for global testing to develop partial tests and perform covariates selection. This extension would result in the problem

$$
\begin{aligned}
&H_0 : Y(t) \perp_{|_t} X_j(t) \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \text{ and } \forall j \in D \\
&H_a : \mathbb{P}\left(Y(t) \not\perp_{|_t} X_j(t)\right) > 0 \quad \forall t \in \mathcal{P} \text{ and some } j \in D
\end{aligned}
$$

where $\perp_{|_t}$ applies for conditional independence on t, $X_D(t)$ denotes the subset of $X(t)$ considering only the covariates with index in $D \subset \{1, \ldots, p\}$, $\mathcal{D} \setminus \mathcal{N}$ is the domain of t minus a null set $\mathcal{N} \subset \mathcal{D}$ and $\mathcal{P} \subset \mathcal{D}$ is a nonnull set.

The global test corresponds with $D = \{1, \ldots, p\}$ and a special case is to consider $D = \{j\}$ for some $j = 1, \ldots, p$. This last approach allows us to implement covariates selection with no need for model estimation, testing the effect of every covariate separately. One can rewrite this test using the CDC coefficient, obtaining a similar test to the one displayed in equation (6.2). This new formulation is given by

$$
\begin{aligned}
&H_0 : \int_{\mathcal{D}\setminus\mathcal{N}} CDC^2(Y(t), X_j(t)|_t)\omega(t)f(t)dt = 0 \text{ almost surely } \text{ for every } j \in D \\
&H_a : \mathbb{P}\left(\int_{\mathcal{P}} CDC^2(Y(t), X_j(t)|_t)\omega(t)f(t)dt \neq 0\right) > 0 \text{ for some } j \in D
\end{aligned}
$$

Following the guidelines of Wang et al. (2015), the Gaussian distribution for the new statistics is asymptotically guaranteed. However, some problems arise related to the CDC coefficient estimation and the statistics calibration in practice. Specifically, it is quite tricky to obtain a proper bandwidth value for a suitable calibration of the test using the same arguments of Chapter 6. We have observed that the procedures employed for global tests do not work well now. Further research on this topic is necessary to provide an optimal calibration system for partial tests.

Development of new GoF tests for the asynchronous FCM. Using the ideas developed in Chapter 6 for the asynchronous FCM, new GoF tests could be proposed using the CDC coefficient studied in Chapter 4. In particular, the resulting new test would be

$$\begin{aligned} H_0 &: \varepsilon(t) \perp_{|t} X(t) \quad \text{almost surely } \forall t \in \mathcal{D} \setminus \mathcal{N} \\ H_a &: \mathbb{P}\left(\varepsilon(t) \not\perp_{|t} X(t)\right) > 0 \quad \forall t \in \mathcal{P} \end{aligned}$$

where $\perp_{|t}$ applies for conditional independence on t, $\mathcal{D} \setminus \mathcal{N}$ is the domain of t minus a null set $\mathcal{N} \subset \mathcal{D}$, $\mathcal{P} \subset \mathcal{D}$ is a nonnull set and $\varepsilon(t)$ is the error of the model.

It is easy to see that this is similar to the problem (6.1), just changing $Y(t)$ by the model error. Thus, this formulation can be expressed in terms of the CDC coefficient as

$$\begin{aligned} H_0 &: \int_{\mathcal{D}\setminus\mathcal{N}} CDC^2(\varepsilon(t), X(t)|_t)\omega(t)f(t)dt = 0 \text{ almost surely} \\ H_a &: \mathbb{P}\left(\int_{\mathcal{P}} CDC^2(\varepsilon(t), X(t)|_t)\omega(t)f(t)dt \neq 0\right) > 0, \end{aligned}$$

resulting in a GoF test after assuming a model structure in $m(\cdot)$ and calculating the model error as $\varepsilon(t) = Y(t) - m\left(t, X(t)\right)$.

Estimating the $m(\cdot)$ function and the model error is necessary now to construct a suitable statistic. We obtain an error estimator using the model residuals once the regressor function is estimated employing the underlying form. Maity (2017) collects some ideas for FCM estimation under different structures. Next, the estimation of the CDC coefficient using $\hat{\varepsilon}(t)$ and $X(t) = (X_1(t), \ldots, X_p(t))$ is needed. Again, proper kernel and bandwidth values have to be selected. In practice, one can calibrate the test employing resampling techniques, as the local bootstrap introduced in Chapter 6. In this case, as the error model is approximated by the residuals $\hat{\varepsilon}(t)$, this has to be considered when obtaining the asymptotic distribution of the statistic. Then, the adaptation of the Wang et al. (2015) results for the significance tests in the asynchronous FCM needs to be carefully reviewed and modified. This framework opens another possible line of research.

Appendix A

Extra results for LASSO under dependence

A.1 Calculation of σ

We are interested in the adjusted models being able to recover, at most, 90% of the explained deviance. So, it is necessary to establish the variance of the error distribution, ε, using this criterion. Then, once the vector β and the correlation matrix Σ are determined, it is needed to calculate the value of the σ parameter taking into account all this information. As a result, a different value for σ is obtained depending in each of the simulated scenarios introduced in Section 3.1.1.

In particular, these quantities are obtained verifying the condition (A.1) of

$$\begin{aligned} &\%Dev = \frac{\mathbb{V}(< X, \beta >)}{\mathbb{V}(< X, \beta >) + \sigma^2} \Rightarrow \mathbb{V}(< X, \beta >) = \%Dev \cdot (\mathbb{V}(< X, \beta >) + \sigma^2) \\ &\Rightarrow \sigma^2 = \frac{1 - \%Dev}{\%Dev} \mathbb{V}(< X, \beta >), \end{aligned} \tag{A.1}$$

where $\mathbb{V}(\cdot)$ is the variance operator. Similarly, the $\mathbb{C}(\cdot)$ operator is going to denote the covariance henceforth.

A.1.1 Scenario 1 (Orthogonal scenario)

The formula for σ in Scenario 1 is

$$\sigma = \sqrt{\frac{1 - 0.9}{0.9} \sum_{j=1}^{s} \beta_j^2}. \tag{A.2}$$

Then, for $s = 10$ it is needed to take $\sigma \simeq 1.317616$, in the case of $s = 15$ its value is $\sigma \simeq 1.613743$, and for $s = 20$ this quantity results in $\sigma \simeq 1.86339$.

This is due to the fact that

$$\sigma^2 \overset{(a)}{=} \frac{1 - \%Dev}{\%Dev} \sum_{j=1}^{p} \beta_j^2 \overset{(b)}{\Rightarrow} \sigma^2 = \frac{1 - \%Dev}{\%Dev} \sum_{j=1}^{s} \beta_j^2$$

where (a) is true because $\mathbb{V}(< X, \beta >) = \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) = \beta_1^2\mathbb{V}(X_1) + \cdots + \beta_p^2\mathbb{V}(X_p)$, and $\mathbb{V}(X_j) = 1$ since X_j *i.i.d.* $X \in N_n(0, I_p)$. Besides, (b) arises because of β structure.

A.1.2 Scenario 2 (Dependence by blocks)

The value of σ in Scenario 2 is

$$\sigma = \sqrt{\frac{1-0.9}{0.9}\left(s + 2\rho\left[\sum_{j=1}^{s}\beta_j\left(\sum_{\substack{k=j+1\\ k\equiv j_{mod10}}}^{s}\beta_k\right)\right]\right)}. \tag{A.3}$$

Now, this quantity depends on the ρ value. Then, it is necessary to distinguish between

- $\rho = 0.5$: in this situation we have that, for $s = 10$, we get $\sigma \simeq 0.745356$, for $s = 15$, we obtain $\sigma \simeq 1.178511$, and for $s = 20$, the value $\sigma \simeq 1.490712$.
- $\rho = 0.9$: in this scenario we see that, for $s = 10$, it is needed to take $\sigma = 1$, for $s = 15$, we get $\sigma \simeq 1.581139$, and for $s = 20$, the quantity $\sigma = 2$.

It is owing to

$$\begin{aligned}
\mathbb{V}(< X, \beta >) &= \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p)\\
&= \beta_1^2\mathbb{V}(X_1) + \mathbb{V}(X_2\beta_2 + \cdots + X_p\beta_p) + 2\mathbb{C}(X_1\beta_1, X_2\beta_2 + \ldots X_p\beta_p)\\
&= \beta_1^2\mathbb{V}(X_1) + \beta_2^2\mathbb{V}(X_2) + \mathbb{V}(X_3\beta_3 + \cdots + X_p\beta_p) + 2\mathbb{C}(X_2\beta_2, X_3\beta_3 + \cdots + X_p\beta_p)\\
&+ 2\mathbb{C}(X_1\beta_1, X_2\beta_2 + \cdots + X_p\beta_p)\\
&= \ldots\\
&\overset{(a)}{=} \sum_{j=1}^{p}\beta_j^2 + 2\rho\left[\sum_{j=1}^{p}\beta_j\left(\sum_{\substack{k=j+1\\ k\equiv j_{mod10}}}^{p}\beta_k\right)\right] = s + 2\rho\left[\sum_{j=1}^{s}\beta_j\left(\sum_{\substack{k=j+1\\ k\equiv j_{mod10}}}^{s}\beta_k\right)\right]
\end{aligned}$$

where (a) is due to $\mathbb{C}(X_j, X_k) = \rho$ for $mod_{10}(j) = mod_{10}(k)$ and $\beta_j = 1$ for $j = 1, \ldots, s$.

A.1.3 Scenario 3 (Toeplitz covariance)

In the case of Scenario 3, the σ value changes in relation to considering the Scenario 3.a or the Scenario 3.b.

- Scenario 3.a: only the first $s = 15$ covariates are important.

$$\sigma = \sqrt{\frac{1-0.9}{0.9}\left(15\cdot 0.5^2 + 2(0.5^2)\sum_{\substack{j=1\\ 15\geq k>j}}^{15}\rho^{|j-k|}\right)} \simeq \sqrt{\frac{1}{9}\left(3.75 + 0.5\sum_{\substack{j=1\\ 15\geq k>j}}^{15}\rho^{|j-k|}\right)},$$

then, taking $\rho = 0.5$ we get $\sigma \simeq 1.067189$ and for $\rho = 0.9$, $\sigma \simeq 1.951213$.

This is because

$$\begin{aligned}\mathbb{V}(< X, \beta >) &= \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) \\ &= \dots \\ &= \sum_{j=1}^{p} \beta_j^2 + 2\sum_{\substack{j=1 \\ k>j}}^{p} \beta_j \rho^{|j-k|} \beta_k = 15 \cdot 0.5^2 + 2(0.5^2) \sum_{\substack{j=1 \\ 15 \geq k > j}}^{15} \rho^{|j-k|}.\end{aligned}$$

- Scenario 3.b: the $s = 10$ relevant variables are placed every 10 sites.

$$\sigma = \sqrt{\frac{1-0.9}{0.9}\left(10 \cdot 0.5^2 + 2(0.5^2) \sum_{\substack{j,k=1 \\ j \neq k \\ j,k \equiv 1 \bmod 10}}^{99} \rho^{|j-k|}\right)} \simeq \sqrt{\frac{1}{9}\left(2.5 + 0.5 \sum_{\substack{j,k=1 \\ j \neq k \\ j,k \equiv 1 \bmod 10}}^{99} \rho^{|j-k|}\right)},$$

then, for $\rho = 0.5$ we have that $\sigma \simeq 0.5275097$, while for $\rho = 0.9$, $\sigma \simeq 0.7276863$.

Now

$$\begin{aligned}\mathbb{V}(< X, \beta >) &= \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) \\ &= \dots \\ &= \sum_{\substack{j=1 \\ j \equiv 1 \bmod 10}}^{99} \beta_j^2 + 2 \sum_{\substack{j,k=1 \\ j \neq k \\ j,k \equiv 1 \bmod 10}}^{99} \beta_j \rho^{|j-k|} \beta_k = 10 \cdot 0.5^2 + 2(0.5^2) \sum_{\substack{j,k=1 \\ j \neq k \\ j,k \equiv 1 \bmod 10}}^{99} \rho^{|j-k|}.\end{aligned}$$

A.2 Consistency conditions

In order to guarantee consistency for the proper recovery of S, some assumptions about the parameters values are needed. In Section 2.1.2 and Section 2.1.4 of the document, some conditions are introduced for this aim. These requirements are collected in Table A.1 for the $p = 100$ value used in the simulations.

The $|S| \log(100) = o(n)$ condition is only verified for simulations taking values of $n = 100, 200, 400$. For $n = 50$, the consistency is only guaranteed in the case of $s = 10$. All the simulation scenarios with $n = 25$ are inconsistent.

Paying attention to the $\inf_{j \in S} |\beta_j| > \sqrt{s \log(100)/n}$ beta-min condition, in the Scenario 1, the algorithm is able for signal recovery for $s = 10, 15$ with $n = 50, 100, 200, 400$ and for $s = 20$ with $n = 100, 200, 400$. In the Scenario 2, it is needed a value of $n = 50, 100, 200, 400$ when $s = 10$ and take $n = 100, 200, 400$ samples for $s = 15, 20$. Eventually, for the Scenario 3.a only important covariates are distinguishes from the zero ones with $n = 400$ samples

	n					$n > \log(100)s$
s	**25**	**50**	**100**	**200**	**400**	
$s = 10$	1.36	0.96	0.68	0.48	0.34	$n \geq 47$
$s = 15$	1.66	1.18	0.83	0.59	0.42	$n \geq 70$
$s = 20$	1.92	1.36	0.96	0.68	0.48	$n \geq 93$
$\sigma\sqrt{\log(100)/n}$	0.43σ	0.3σ	0.21σ	0.15σ	0.11σ	

Table A.1: Values $\sqrt{s\log(100)/n}$ to guarantee the beta-min condition. Last column shows the number of samples needed to guarantee $|S|\log(100) = o(n)$. In the last row the oracle scale of λ ($\sigma\sqrt{\log(100)/n}$) is given.

and with $n = 200, 400$ for Scenario 3.b.

Finally, a guidance about the oracle scale of λ, $\sigma\sqrt{\log(100)/n}$, is displayed. This quantity depends on the sample size n as well as the error variance σ. This last changes in every simulation scenario. In Section A.1, the value of σ for recovering the 90% of explained deviance is calculated in the simulated scenarios.

A.3 Tuning parameters selection

In this section, the selection of tuning parameters is explained, displaying the considered grid of values for every methodology. Moreover, a more deep analysis of the suitable selection of λ for the proper recovery of relevant covariates is carried out. For this purpose, greater values than the optimal one provided by the 10-fold cross-validation criterion minimizing the MSE are tested. Besides, the performance of the LASSO selection taking the penalization value which minimizes the BIC criterion in analyzed.

A.3.1 Grid values of the tuning parameters

The choice of the grid values of tuning parameters will depend on the methodology employed and the sample data. In this work, a grid of length 100 is consider for every algorithm. Next, we explain the selection of the grid made by the employed libraries of R (R Core Team (2019)) for the different procedures:

LASSO: `glmnet` of Friedman et al. (2010), last update November 27, 2022.

SCAD: `ncvreg` of Breheny and Huang (2011), last update October 13, 2022.

AdapL: `glmnet` of Friedman et al. (2010), last update November 27, 2022.

Dant: `flare` of Li et al. (2019), last update October 13, 2022.

RelaxL: `relaxo` of Meinshausen (2012), last update May 23, 2022.

SqrtL: `flare` of Li et al. (2019), last update October 13, 2022.

ScalL: `scalreg` of Sun (2019), last update October 14, 2022.

Distance correlation algorithm for variable selection (DC.VS): `fda.usc` of Febrero-Bande and Oviedo de la Fuente (2012), last update October 17, 2022.

In the LASSO case, to choose a correct grid for λ, we follow the considerations of Friedman et al. (2010). They take 100 values on a grid $[\lambda_{\min}, \lambda_{\max}]$ where $\lambda_{\max}$ is the smallest value for which the entire vector $\hat{\beta} = 0$ and $\lambda_{\min} = \alpha\lambda_{\max}$, where $\alpha = 0.01$ if $p > n$ and $\alpha = 0.0001$ otherwise. This algorithm is implemented in the `glmnet` library (Friedman et al. (2010)) of R (R Core Team (2019)). The $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ minimizing the penalized maximum likelihood, $\lambda^{\min}$, is used to adjust the model (LASSO.min). Besides, other larger values than $\lambda^{\min}$ are tested for the LASSO adjustment (see Section 3.2). Particularly, the LASSO making use of λ^{1se} (LASSO.1se), which is the largest value of λ such that error is within one standard error of $\lambda^{\min}$, is included in the simulation comparison. The selection of λ^{1se} is already implemented in the `glmnet` library (Friedman et al. (2010)). Eventually, we consider the selection of the optimal λ by means of minimizing the BIC criterion (see Section 3.3). Its search performs in the same grid of λ values as the LASSO one.

The same scheme is following for the AdapL. Here, a first estimator of β is needed. This is obtained by means of a RIDGE regression, $\hat{\beta}^{RIDGE}$. For this purpose, we make use of the library `glmnet` (Friedman et al. (2010)). As for the LASSO case, we consider two different adjustments for the RIDGE and posterior weighted LASSO: using the penalization parameters which correspond with $\lambda^{\min}$ (AdapL.min) or making use of the λ^{1se} penalties (AdapL.1se).

For the SCAD procedure, due to the concave nature of the penalization, a linear approximation of the problem is employed to get the penalty (see Zou and Li (2008)). Then, the LARS algorithm introduced in Efron et al. (2004) is used. More information is provided in Breheny and Huang (2011).

The grid values for the Dant is similar to the LASSO ones, just following recommendations of Li et al. (2019). Now $\lambda_{\min} = \alpha\lambda_{\max}$ with $\alpha = 0.5$ for Dant algorithm. In case of SqrtL the authors suggest to take $\lambda_{\max} = \pi\sqrt{\log(p)/n}$ and $\alpha = 0.3$ for $\lambda_{\min} = \alpha\lambda_{\max}$.

For the RelaxL, we take a sequence of 10 equispaced points in $[0.0001, 1]$ for estimating the ϕ parameter. In terms of λ, the LARS algorithm (Efron et al. (2004)) is employed. See Meinshausen (2007) for more details.

Eventually, the ScalL procedure estimates the penalization λ by means of a estimation of σ and recompute these quantities iteratively. The initial value for the penalization is $\lambda_0 = \sqrt{(2/n)\log(p)}$ and the posterior ones are recomputed as $\lambda = \hat{\sigma}\lambda_0$. More information is given in Sun and Zhang (2013).

Next, the performance of LASSO for greater penalization values than the λ minimizing the MSE ($\lambda^{\min}$) is tested.

A.3.2 LASSO performance for greater values of λ

As we can see in the simulation results of Section A.6, the selection of λ by means of 10-fold cross-validation criterion minimizing the MSE adds to much noise. Thus, a greater value of λ may be needed to properly recover the relevant covariates, S, without adding irrelevant ones. As a result, we test the performance of the LASSO for greater values of λ.

For this aim, taking into consideration that the library `glmnet` provides an estimation of the mean cross-validated error for every value of the λ grid, this information can be employed to select a larger λ. In fact, this supplies the λ^{1se} term, which is the largest value of λ such that this is within 1 standard error of the minimum ($\lambda^{\min}$).

Then, we test the LASSO performance for the different simulation scenarios in terms of covariates selection for values of $\lambda = \lambda^{\min} + \alpha \cdot dist$ taking $\alpha = 0.5, 1, 1.5, 2, 2.5, 4$, being $dist = \lambda^{1se} - \lambda^{\min}$ and considering $n = 400$.

Results for Scenario 1 are displayed in Figure A.1, for Scenario 2 in Figures A.2 and A.3, and for Scenario 3 in Figures A.4 and A.5. Complete results for the $\alpha = 1$ case, taking $\lambda = \lambda^{1se}$ are collected in Section A.6.

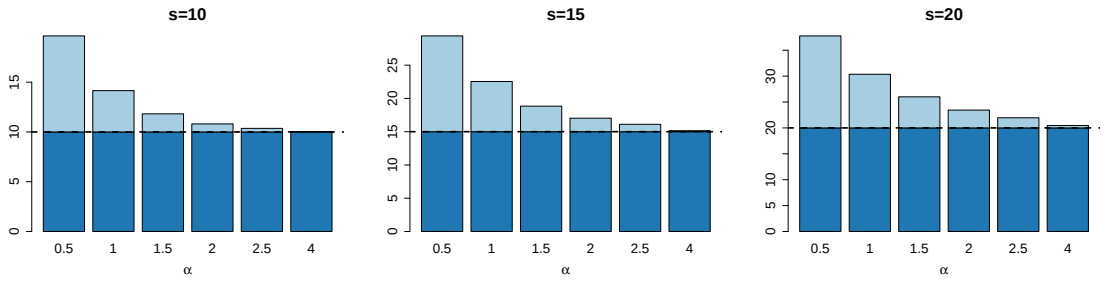


Figure A.1: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO in Scenario 1 for $n = 400$ and $\lambda = \lambda^{\min} + \alpha \cdot dist$, being $dist = \lambda^{1se} - \lambda^{\min}$. The dashed line marks the s value.

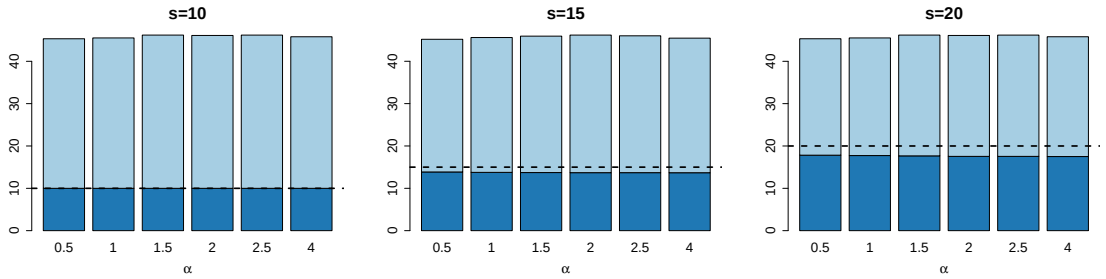


Figure A.2: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO in Scenario 2 with $\rho = 0.5$ for $n = 400$ and $\lambda = \lambda^{\min} + \alpha \cdot dist$, being $dist = \lambda^{1se} - \lambda^{\min}$. The dashed line marks the s value.

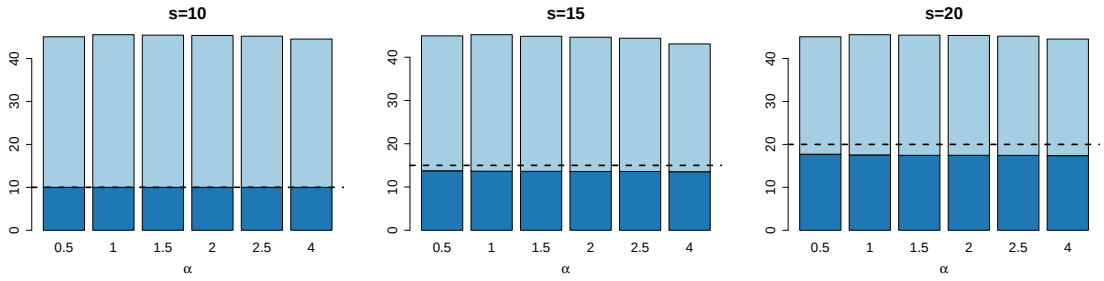


Figure A.3: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO in Scenario 2 with $\rho = 0.9$ for $n = 400$ and $\lambda = \lambda^{\min} + \alpha \cdot dist$, being $dist = \lambda^{1se} - \lambda^{\min}$. The dashed line marks the s value.

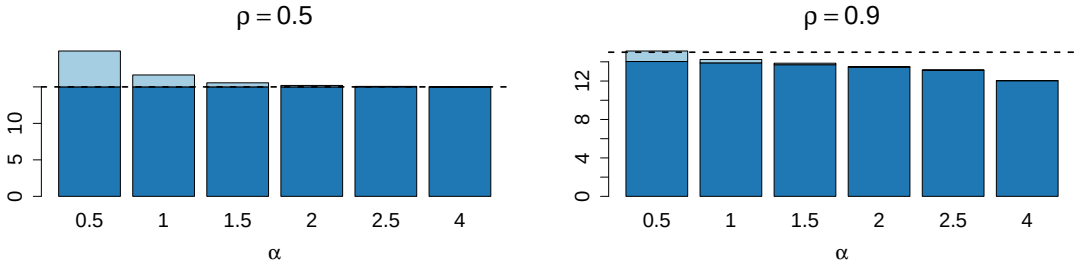


Figure A.4: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO in Scenario 3.a for $n = 400$ and $\lambda = \lambda^{\min} + \alpha \cdot dist$, being $dist = \lambda^{1se} - \lambda^{\min}$. The dashed line marks the s value.

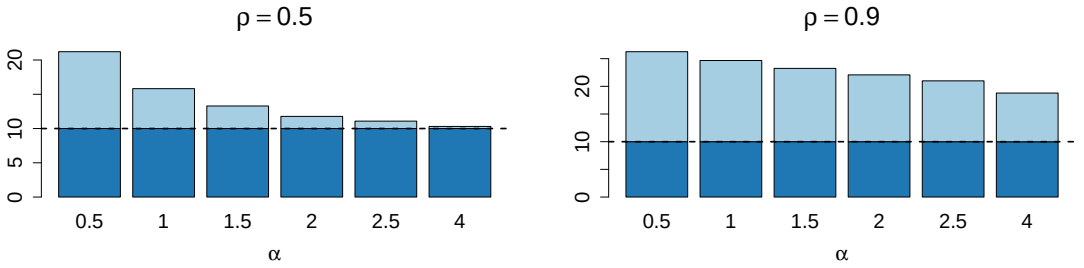


Figure A.5: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO in Scenario 3.b for $n = 400$ and $\lambda = \lambda^{\min} + \alpha \cdot dist$, being $dist = \lambda^{1se} - \lambda^{\min}$. The dashed line marks the s value

Next, the selection of a proper value of the λ parameter for the LASSO performance employing the BIC criterion is treated.

A.3.3 LASSO performance employing BIC criterion

In order to test if the addition of a great amount of noisy covariates by the LASSO is due to cross-validation (CV), a comparison by means of minimizing the BIC criterion is carried out. Then, the optimal λ value is selected as the one which minimizes the BIC criterion. For this purpose, we have made use of the λ grid provided by the LASSO path of the `glmnet` function (Friedman et al. (2010)) implemented in R (R Core Team (2019)). See Section A.3 for more details. As we are working with a multiple linear model, the BIC value is estimated by $BIC = n\log(MSE) + k\log(n)$, being MSE the mean square error and $k \leq p$ the number of covariates which enters the model. To calculate the MSE, σ has been estimated employing the residuals of the model.

Next, supplementary figures and tables for the simulated scenarios that result from the LASSO.BIC adjustment are displayed. Results for Scenario 1 are collected in Figure A.6 and Table A.2. In case of Scenario 2 with $\rho = 0.5$, they are displayed in Figure A.7 and Table A.3, and for $\rho = 0.9$ in Figure A.8 and Table A.4. Eventually, the number of selected covariates in the Scenario 3.a and 3.b are shown in Figures A.9 and A.10 respectively, whereas the prediction results are collected in Table A.5.

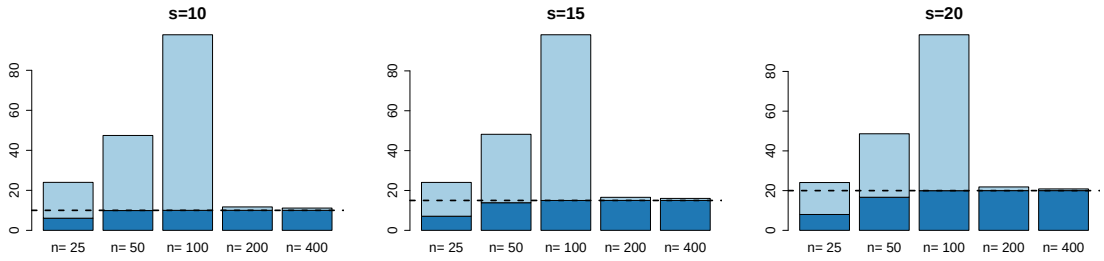


Figure A.6: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 1. The dashed line marks the s value.

	s = 10		**s = 15**		**s = 20**	
	MSE (1.736)	**% Dev**	**MSE** (2.604)	**% Dev**	**MSE** (3.472)	**% Dev**
$n = 25$	0	1	0	1	0	1
$n = 50$	0.001	1	0.001	1	0.001	1
$n = 100$	0.014	0.999	0.011	1	0.005	1
$n = 200$	1.544	0.910	2.272	0.911	2.950	0.913
$n = 400$	1.646	0.904	2.443	0.905	3.224	0.906

Table A.2: Summary of the LASSO results for Scenario 1 making use of the BIC criterion. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

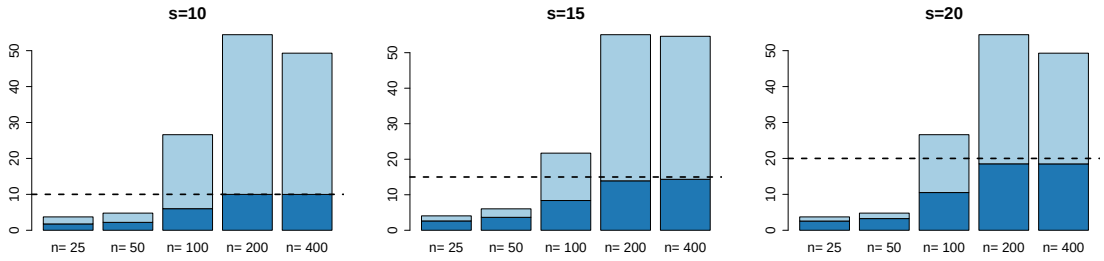


Figure A.7: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 2 with $\rho = 0.5$. The dashed line marks the s value.

	s = 10		**s = 15**		**s = 20**	
	MSE (0.556)	**% Dev**	**MSE** (1.389)	**% Dev**	**MSE** (2.222)	**% Dev**
$n = 25$	5.366	0.450	12.436	0.489	21.466	0.450
$n = 50$	6.056	0.396	12.826	0.482	24.222	0.396
$n = 100$	3.375	0.649	6.829	0.721	13.500	0.649
$n = 200$	0.524	0.950	1.575	0.939	2.097	0.950
$n = 400$	0.538	0.949	1.346	0.949	2.154	0.949

Table A.3: Summary of the LASSO results for Scenario 2 with $\rho = 0.5$ making use of the BIC criterion. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

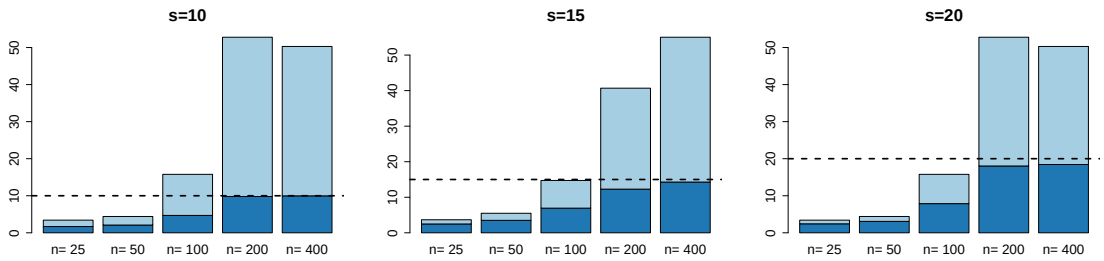


Figure A.8: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 2 with $\rho = 0.9$. The dashed line marks the s value.

	s = 10		**s = 15**		**s = 20**	
	MSE (1)	**% Dev**	**MSE** (2.5)	**% Dev**	**MSE** (4)	**% Dev**
$n = 25$	5.715	0.431	13.187	0.472	22.858	0.431
$n = 50$	6.533	0.392	14.055	0.464	26.134	0.392
$n = 100$	4.864	0.541	9.721	0.633	19.455	0.541
$n = 200$	1.071	0.899	3.909	0.854	4.285	0.899
$n = 400$	0.968	0.912	2.455	0.910	3.872	0.912

Table A.4: Summary of the LASSO results for Scenario 2 with $\rho = 0.9$ making use of the BIC criterion. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

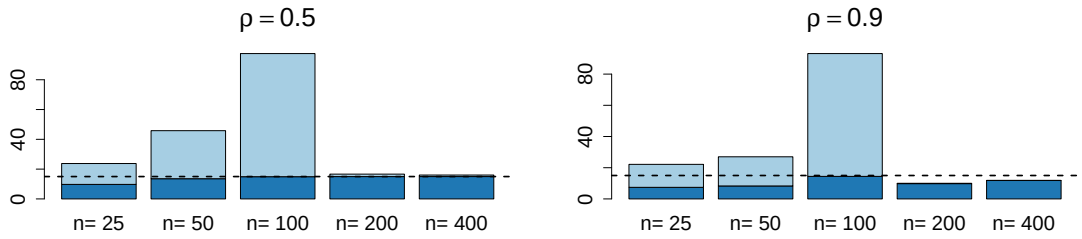


Figure A.9: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 3.a. The dashed line marks the $s = 15$ value.

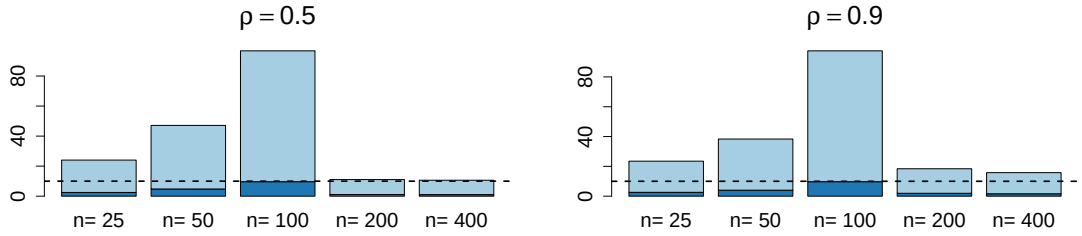


Figure A.10: Number of important covariates (dark area) and noisy ones (soft area) selected by the LASSO.BIC in Scenario 3.b. The dashed line marks the $s = 10$ value.

	Scenario 3.a				Scenario 3.b			
	$\rho = 0.5$		$\rho = 0.9$		$\rho = 0.5$		$\rho = 0.9$	
	MSE (1.139)	**% Dev**	**MSE** (3.807)	**% Dev**	**MSE** (0.278)	**% Dev**	**MSE** (0.53)	**% Dev**
$n = 25$	0	1	0.010	1	0	1	0	1
$n = 50$	0.002	1	0.854	0.978	0	1	0.021	0.996
$n = 100$	0.012	0.999	0.206	0.995	0.004	0.998	0.007	0.999
$n = 200$	0.990	0.912	3.721	0.905	0.254	0.908	0.471	0.910
$n = 400$	1.066	0.906	3.824	0.903	0.267	0.903	0.501	0.905

Table A.5: Summary of the LASSO results for Scenario 3.a and Scenario 3.b making use of the BIC criterion. The oracle value for the deviance is 0.9 and those for the MSE are in brackets.

A.4 Computational time

Just to illustrate the computational time of the implemented algorithms, a total of $M = 100$ replicates of Scenario 1 taking $s = 10$ (introduced in Section 3.1.1), are carried out. Results are displayed in Figure A.11 for different sample sizes. We refer the reader to Sections 3.1.3 and A.3 for implementation issues. The displayed time is the "real" elapsed time since the process was started. As a result, we have measured both, the total user and system CPU times[1]. Simulations have been run on a computer HP ZBook Power with a computer processing unit 11th Gen Intel(R) Core(TM) i7-11800H.

In view of the results of Figure A.11, it is appreciated that the SqrtL is the slowest procedure. This is followed by the DC.VS and the Dant. Their computational time proportionally increases in terms of the sample size. In the simplest framework of orthogonal design, for a sample size of $n = 400$, the SqrtL reaches the 6000 s and the DC.VS the 3500 s approximately. This fact contrasts with the time spent by the remaining LASSO procedures, being in all cases, inferior to 120 s. These are the LASSO.min, LASSO.1se, LASSO.BIC, AdapL.min, AdapL.1se, RelaxL and ScalL. Furthermore, the SCAD procedure is also quite competitive in time. The Dant is a bit more expensive, having times between 200-360 s, although this outperforms the times of the DC.VS and SqrtL algorithms, especially for great sample sizes. In contrast, the DC.VS procedure is competitive for the $n > p$ case, but its computational time highly increases for $n = 100$, $n = 200$ and $n = 400$, exceeding more than 320 s. As we move to more complex scenarios, like Scenarios 2 and 3 of Section 3.1.1, the computational time increases for all algorithms. In practice, we have observed similar patterns as the ones observed in Figure A.11 and commented above. The SqrtL, the DC.VS and the Dant are always the approaches that take the longest. In comparison, the rest of LASSO versions are the fastest options.

[1]For this purpose, the function `system.time` of the base package of R Core Team (2019) has been employed. In particular, the "elapsed" time has been considered.

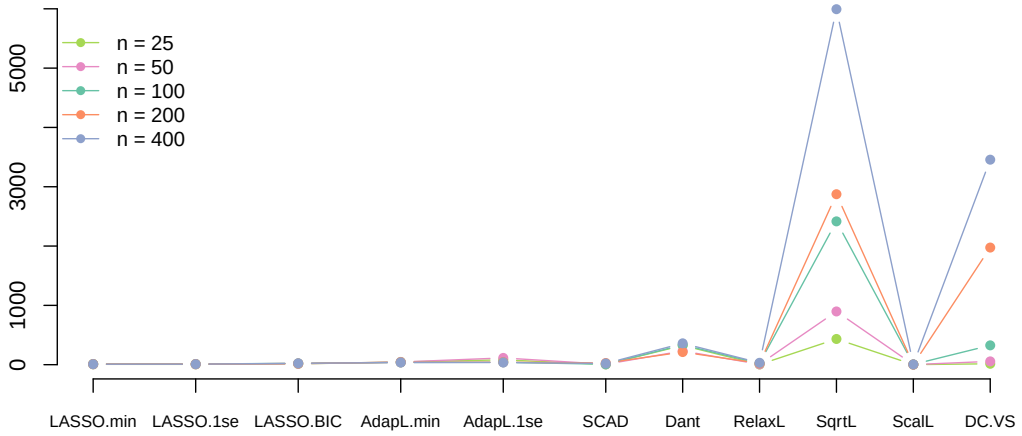


Figure A.11: Computational time in seconds of the eleven algorithms considered for $M = 100$ replicates of Scenario 1, taking $s = 10$ and different sample sizes.

A.5 Efficient covariates calculation

It is considered by efficient covariates the subset of important ones required to explain the data. When all covariates are uncorrelated, these are exactly those in S. An example is the case of Scenario 1 considered in the simulation study. However, in case of dependence among covariates, it is not trivial to know the real number of efficient ones. This quantity changes based on the dependence structure of the data. As a result, the number of efficient covariates is different for each combination of Scenario 2 and Scenario 3 parameters. An idea to figure out this quantity is to apply a singular value decomposition of Σ_S. Here, Σ_S denotes the submatrix of Σ considering the elements of S. Then, in terms of its eigenvalues study, one can get to know how many covariates are necessary to explain a certain percentage of variability previously fixed.

The resulting eigenvalues are collected in Table A.6 for Scenario 2 and in Table A.7 for Scenarios 3.a and 3.b. In consequence, we can decide how many of those s covariates are really necessary to explain the data taking into account the percentage of variability one wants to explain. A summary of this study is displayed in Table A.8 for Scenario 2 and in Table A.9 for Scenarios 3.a and 3.b.

s	$\rho = 0.5$	$\rho = 0.9$
$s = 10$	1, 1, 1, 1, 1, 1, 1, 1, 1, 1	1, 1, 1, 1, 1, 1, 1, 1, 1, 1
$s = 15$	1.5, 1.5, 1.5, 1.5, 1.5, 1, 1, 1, 1, 1, 0.5, 0.5, 0.5, 0.5, 0.5	1.9, 1.9, 1.9, 1.9, 1.9, 1, 1, 1, 1, 1, 0.1, 0.1, 0.1, 0.1, 0.1
$s = 20$	1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5	1.9, 1.9, 1.9, 1.9, 1.9, 1.9, 1.9, 1.9, 1.9, 1.9, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1

Table A.6: Eigenvalues of Σ_S for Scenario 2 in decreasing order.

	$\rho = 0.5$	$\rho = 0.9$
Scenario 3.a ($s = 15$)	2.83, 2.41, 1.92, 1.50, 1.17, 0.93, 0.76, 0.63, 0.54, 0.47, 0.42, 0.39, 0.36, 0.35, 0.34	9.55, 2.69, 1.01, 0.51, 0.30, 0.20, 0.15, 0.12, 0.09, 0.08, 0.07, 0.06, 0.06, 0.05, 0.05
Scenario 3.b ($s = 10$)	1, 1, 1, 1, 1, 1, 1, 1, 1, 1	1.96, 1.69, 1.38, 1.11, 0.90, 0.74, 0.64, 0.56, 0.52, 0.49

Table A.7: Eigenvalues of Σ_S for Scenario 3 in decreasing order.

	$\rho = 0.5$					$\rho = 0.9$				
s	**80%**	**90%**	**95%**	**98%**	**99%**	**80%**	**90%**	**95%**	**98%**	**99%**
$s = 10$	8	9	10	10	10	8	9	10	10	10
$s = 15$	10	12	14	15	15	8	9	10	12	14
$s = 20$	12	16	18	20	20	9	10	10	16	18

Table A.8: Required covariates in Scenario 2 to explain a certain percentage of variability.

	Scenario 3.a ($s = 15$)					**Scenario 3.b** ($s = 10$)				
ρ	**80%**	**90%**	**95%**	**98%**	**99%**	**80%**	**90%**	**95%**	**98%**	**99%**
$\rho = 0.5$	8	11	13	15	15	8	9	10	10	10
$\rho = 0.9$	2	4	6	10	13	7	9	9	10	10

Table A.9: Required covariates in Scenario 3 to explain a certain percentage of variability.

A.6 Simulation results

This section collects the results of the different simulation scenarios of Section 3.1.1. The average of the results over the $M = 500$ simulations for every combination of n and s is shown. The number of covariates correctly selected ($|\hat{S} \cap S|$) as well as the noisy ones ($|\hat{S} \setminus S|$) are displayed. Besides, it is measured the prediction power of the algorithm by means of the mean squared error (MSE) and the percentage of explained deviance ($\%Dev$).

A.6.1 Scenario 1 (Orthogonal scenario)

	LASSO.min					LASSO.1se				
n	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (10)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (1.736)	**% Dev** (0.9)	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (10)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (1.736)	**% Dev** (0.9)
$n = 25$	5.4	15	20.5	1.192	0.916	3.8	8	11.8	4.135	0.727
$n = 50$	9.9	27.3	37.2	0.169	0.990	9.4	14.7	24.1	0.690	0.956
$n = 100$	10	24.5	34.5	0.702	0.959	10	11	21	1.037	0.939
$n = 200$	10	22.2	32.2	1.164	0.932	10	6.8	16.8	1.410	0.918
$n = 400$	10	21.3	31.3	1.434	0.917	10	4.2	14.2	1.595	0.907
n	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (15)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (2.604)	**% Dev** (0.9)	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (15)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (2.604)	**% Dev** (0.9)
$n = 25$	6	13.5	19.5	2.520	0.888	3.9	7.2	11.1	7.388	0.679
$n = 50$	13.1	26.8	39.9	0.579	0.973	10.6	13.3	24	3.065	0.864
$n = 100$	15	29.7	44.7	0.841	0.967	15	16.3	31.3	1.261	0.950
$n = 200$	15	27.1	42.1	1.628	0.936	15	11.1	26.1	1.945	0.924
$n = 400$	15	26.3	41.3	2.091	0.919	15	7.6	22.6	2.300	0.911
n	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (20)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (3.472)	**% Dev** (0.9)	$\|\hat{\boldsymbol{S}} \cap \boldsymbol{S}\|$ (20)	$\|\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\|$	$\|\hat{\boldsymbol{S}}\|$	**MSE** (3.472)	**% Dev** (0.9)
$n = 25$	6.6	12.6	19.2	4.155	0.866	4.1	6.7	10.8	10.720	0.664
$n = 50$	14.7	23.3	38	1.747	0.944	10.5	11.4	21.9	6.865	0.782
$n = 100$	20	33.5	53.5	0.914	0.973	20	20.3	40.3	1.420	0.958
$n = 200$	20	30.8	50.8	2.060	0.940	20	14.7	34.7	2.420	0.929
$n = 400$	20	29.4	49.4	2.736	0.920	20	10.4	30.4	2.977	0.913

Table A.10: Results of LASSO.min and LASSO.1se for Scenario 1. Oracle values are in brackets.

	AdapL.min					AdapL.1se				
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.736)	**% Dev** (0.9)
$n = 25$	3.9	6.2	10.1	0.639	0.960	1.7	1.6	3.3	6.109	0.621
$n = 50$	9.4	4	13.5	0.937	0.945	7.4	0.6	8.1	2.820	0.828
$n = 100$	10	1.7	11.7	1.364	0.920	10	0	10	1.559	0.908
$n = 200$	10	12.8	22.8	1.275	0.925	10	0	10	1.633	0.904
$n = 400$	10	10.8	20.8	1.515	0.912	10	0	10	1.682	0.902
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (2.604)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (2.604)	**% Dev** (0.9)
$n = 25$	4.1	6.1	10.2	0.901	0.963	1.7	1.6	3.4	9.650	0.607
$n = 50$	10.9	6.9	17.8	1.393	0.945	6	1.6	7.6	7.230	0.707
$n = 100$	15	3.2	18.2	1.805	0.929	14.6	0.1	14.7	2.431	0.903
$n = 200$	15	13.9	28.9	1.824	0.929	15	0	15	2.384	0.907
$n = 400$	15	11.2	26.2	2.235	0.913	15	0	15	2.491	0.903
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (20)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.472)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (20)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.472)	**% Dev** (0.9)
$n = 25$	4.3	5.9	10.2	1.332	0.958	1.6	1.4	3	14.226	0.567
$n = 50$	11	7.5	18.4	2.069	0.938	5.3	2.1	7.4	11.330	0.656
$n = 100$	19.8	4.4	24.1	2.190	0.935	18.1	0.4	18.5	3.843	0.884
$n = 200$	20	14.2	34.2	2.343	0.931	20	0	20	3.089	0.909
$n = 400$	20	11.1	31.1	2.942	0.914	20	0	20	3.277	0.905

Table A.11: Results of AdapL.min and AdapL.1se for Scenario 1. Oracle values are in brackets.

	SCAD					Dant				
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.736)	**% Dev** (0.9)
$n = 25$	2.4	3.2	5.5	5.953	0.619	1.9	1.9	3.8	10.951	0.335
$n = 50$	9.1	7.6	16.7	1	0.936	5.2	1.7	7.0	9.976	0.419
$n = 100$	10	4.2	14.2	1.212	0.929	7.9	0.4	8.3	9.584	0.442
$n = 200$	10	2.3	12.3	1.536	0.910	9.6	0	9.6	8.685	0.494
$n = 400$	10	2.1	12.1	1.635	0.905	10	0	10	7.693	0.552
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (2.604)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (2.604)	**% Dev** (0.9)
$n = 25$	2.2	2.9	5.2	9.855	0.591	1.9	1.8	3.7	11.252	0.540
$n = 50$	7.9	5.7	13.6	4.750	0.802	5.3	2.2	7.5	8.741	0.653
$n = 100$	15	7.5	22.5	1.464	0.942	9.1	1	10.2	7.516	0.703
$n = 200$	15	2.9	17.9	2.188	0.914	12.3	0.1	12.4	5.410	0.788
$n = 400$	15	2	17	2.425	0.906	14.6	0	14.6	2.976	0.885
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (20)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.472)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (20)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.472)	**% Dev** (0.9)
$n = 25$	2.2	2.5	4.7	14.833	0.551	1.8	1.7	3.5	16.072	0.510
$n = 50$	6.9	4.7	11.6	9.508	0.711	5.1	2.3	7.3	13.238	0.606
$n = 100$	20	11.2	31.2	1.567	0.954	9.7	1.5	11.2	11.445	0.662
$n = 200$	20	4.1	24.1	2.746	0.919	14.2	0.2	14.4	9.175	0.731
$n = 400$	20	1.9	21.9	3.191	0.907	18.2	0	18.3	5.404	0.843

Table A.12: Results of SCAD and Dant for Scenario 1. Oracle values are in brackets.

	RelaxL					SqrtL				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)
$n = 25$	2.6	4.1	6.7	5.921	0.623	1.8	1.7	3.5	7.396	0.535
$n = 50$	9	13.5	22.5	1.049	0.933	6.4	2.5	8.9	3.651	0.771
$n = 100$	10	4.6	14.6	1.302	0.924	10	2.8	12.8	1.396	0.917
$n = 200$	10	0.7	10.7	1.600	0.906	10	2.8	12.8	1.516	0.911
$n = 400$	10	0.3	10.3	1.674	0.902	10	2.8	12.8	1.616	0.906
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.604)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.604)	**% Dev** (0.9)
$n = 25$	2.4	3.3	5.7	10.372	0.567	1.7	1.6	3.3	11.590	0.528
$n = 50$	10.5	12.8	23.3	2.860	0.879	5.3	2.1	7.4	8.987	0.636
$n = 100$	15	12.2	27.2	1.470	0.942	13.2	2.7	15.9	3.398	0.859
$n = 200$	15	1.8	16.8	2.293	0.910	15	2.6	17.6	2.242	0.912
$n = 400$	15	0.3	15.3	2.478	0.904	15	2.6	17.6	2.403	0.907
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.472)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.472)	**% Dev** (0.9)
$n = 25$	2.5	3.2	5.7	14.284	0.566	1.7	1.5	3.2	16.187	0.508
$n = 50$	10.2	10.3	20.6	6.215	0.803	4.8	2	6.8	13.985	0.579
$n = 100$	19.9	19.7	39.6	1.490	0.956	12.9	2.5	15.4	7.801	0.762
$n = 200$	20	4	24	2.861	0.916	20	2.4	22.4	2.957	0.913
$n = 400$	20	0.5	20.5	3.254	0.905	20	2.4	22.4	3.176	0.907

Table A.13: Results of RelaxL and SqrtL for Scenario 1. Oracle values are in brackets.

	ScalL					DC.VS				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)
$n = 25$	3.2	4.0	7.2	2.969	0.809	1	1	2	8.472	0.489
$n = 50$	7.5	3.9	11.4	2.432	0.848	3.5	1.5	5	6.339	0.628
$n = 100$	10	6.5	16.5	1.282	0.924	9.9	1	11	1.508	0.911
$n = 200$	10	4.1	14.1	1.475	0.914	10	6.8	16.8	1.418	0.917
$n = 400$	10	4.0	14.1	1.594	0.907	10	1.9	11.9	1.633	0.905
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.604)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.604)	**% Dev** (0.9)
$n = 25$	3.2	3.8	7	4.869	0.797	0.9	1.1	2	13.025	0.480
$n = 50$	7.2	3.8	11	5.435	0.780	3.1	1.8	5	10.998	0.572
$n = 100$	14.2	5.1	19.3	2.492	0.898	9.6	1.4	11	6.470	0.743
$n = 200$	15	3.8	18.8	2.188	0.914	15	5.4	20.4	2.119	0.917
$n = 400$	15	3.9	18.9	2.371	0.908	15	1.8	16.8	2.421	0.906
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.472)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.472)	**% Dev** (0.9)
$n = 25$	3.2	3.5	6.7	6.921	0.785	1	1	2	17.386	0.475
$n = 50$	7	3.5	10.5	8.648	0.740	3.1	1.8	5	15.423	0.549
$n = 100$	15.1	3.9	19	5.867	0.821	9	2	11	11.279	0.666
$n = 200$	20	3.5	23.5	2.879	0.915	20	2	21.9	2.941	0.914
$n = 400$	20	3.5	23.5	3.137	0.909	20	1.7	21.7	3.194	0.907

Table A.14: Results of ScalL and DC.VS for Scenario 1. Oracle values are in brackets.

A.6.2 Scenario 2 (Dependence by blocks)

	LASSO.min					LASSO.1se				
n	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)
$n=25$	9.9	57	66.9	0.307	0.967	9.8	52	61.8	0.310	0.967
$n=50$	10	53.1	63.1	0.438	0.956	10	50.1	60.1	0.438	0.956
$n=100$	10	47.3	57.3	0.495	0.951	10	45	55	0.495	0.951
$n=200$	10	40.2	50.2	0.523	0.950	10	38.4	48.4	0.523	0.950
$n=400$	10	35.9	45.9	0.538	0.949	10	35.7	45.7	0.538	0.949
n	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)
$n=25$	13.6	52	65.6	0.768	0.967	13.2	46.3	59.5	0.793	0.966
$n=50$	13.8	48.7	62.5	1.095	0.956	13.7	45.8	59.5	1.095	0.956
$n=100$	13.9	42.5	56.4	1.238	0.951	13.7	41.2	54.9	1.238	0.951
$n=200$	13.9	36	50	1.307	0.950	13.7	34.6	48.4	1.307	0.950
$n=400$	14	31.6	45.6	1.346	0.949	13.8	32.1	45.9	1.346	0.949
n	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)
$n=25$	17.7	49.1	66.9	1.228	0.967	17.3	44.6	61.8	1.241	0.967
$n=50$	18	45.1	63.1	1.752	0.956	17.7	42.4	60.1	1.752	0.956
$n=100$	18	39.3	57.3	1.981	0.951	17.7	37.2	55	1.981	0.951
$n=200$	18	32.2	50.2	2.091	0.950	17.6	30.8	48.4	2.091	0.950
$n=400$	18.1	27.8	45.9	2.154	0.949	17.8	27.9	45.7	2.154	0.949

Table A.15: Results of LASSO.min and LASSO.1se for Scenario 2 with $\rho = 0.5$. Oracle values are in brackets.

	AdapL.min					AdapL.1se				
n	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)
$n=25$	8.8	0.9	9.8	0.338	0.964	6.6	0.6	7.2	1.338	0.864
$n=50$	9.5	0.5	10	0.438	0.956	9.5	0.5	10	0.440	0.955
$n=100$	9.8	0.2	10	0.495	0.951	9.8	0.2	10	0.495	0.951
$n=200$	10	0	10	0.523	0.950	9.9	0.1	10	0.523	0.950
$n=400$	10	0	10	0.538	0.949	10	0	10	0.538	0.949
n	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)
$n=25$	8.6	0.2	8.8	1.049	0.955	5.7	0.1	5.7	3.816	0.845
$n=50$	9.8	0.1	9.9	1.117	0.955	9	0.1	9.1	1.515	0.939
$n=100$	10	0	10	1.238	0.951	9.9	0	10	1.244	0.951
$n=200$	10	0	10	1.307	0.950	9.9	0.1	10	1.307	0.950
$n=400$	10	0	10	1.346	0.949	10	0	10	1.346	0.949
n	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)
$n=25$	9.7	0	9.8	1.350	0.964	7.2	0	7.2	5.385	0.863
$n=50$	10	0	10	1.752	0.956	10	0	10	1.762	0.955
$n=100$	10	0	10	1.981	0.951	10	0	10	1.981	0.951
$n=200$	10	0	10	2.091	0.950	10	0	10	2.091	0.950
$n=400$	10	0	10	2.154	0.949	10	0	10	2.154	0.949

Table A.16: Results of AdapL.min and AdapL.1se for Scenario 2 with $\rho = 0.5$. Oracle values are in brackets.

n	SCAD					Dant				
	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)
$n=25$	9.1	3.1	12.2	0.309	0.967	4.1	37.1	41.2	3.046	0.679
$n=50$	9.6	1.7	11.2	0.438	0.956	6.3	56.7	63	2.546	0.743
$n=100$	9.8	0.4	10.2	0.495	0.951	8.2	74.2	82.4	1.706	0.832
$n=200$	10	0.2	10.1	0.523	0.950	9.7	87.5	97.2	0.753	0.928
$n=400$	10	0	10	0.538	0.949	10	89.9	99.9	0.548	0.948
n	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)
$n=25$	10.4	1.3	11.7	0.796	0.966	7.2	33.3	40.5	6.525	0.724
$n=50$	10.8	0.3	11.1	1.095	0.956	9.4	43	52.4	5.832	0.767
$n=100$	10.4	0	10.4	1.238	0.951	10.8	48.7	59.5	4.891	0.808
$n=200$	10.3	0	10.3	1.307	0.950	11.2	51.2	62.4	4.743	0.820
$n=400$	10	0	10	1.346	0.949	11.2	50.4	61.6	4.968	0.811
n	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)
$n=25$	11.8	0.3	12.2	1.238	0.967	8.2	32.9	41.1	12.183	0.679
$n=50$	11.2	0	11.2	1.752	0.956	12.6	50.4	63	10.179	0.743
$n=100$	10.1	0	10.1	1.981	0.951	16.4	65.9	82.4	6.826	0.832
$n=200$	10.1	0	10.1	2.091	0.950	19.4	77.7	97.2	3.006	0.929
$n=400$	10	0	10	2.154	0.949	20	79.9	99.9	2.194	0.948

Table A.17: Results of SCAD and Dant for Scenario 2 with $\rho = 0.5$. Oracle values are in brackets.

n	RelaxL					SqrtL				
	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (10)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)
$n=25$	2.4	8.7	11.1	0.312	0.967	2.3	21	23.3	6.995	0.297
$n=50$	2.5	9	11.5	0.440	0.955	7	63	70	3.040	0.701
$n=100$	2.5	9	11.5	0.500	0.951	10	89.9	99.9	0.497	0.951
$n=200$	2.3	9.1	11.4	0.529	0.950	10	89.9	99.9	0.523	0.950
$n=400$	2.4	8.9	11.3	0.542	0.948	10	90	100	0.538	0.949
n	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (15)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)
$n=25$	3.7	7.4	11.1	0.787	0.966	4.8	22.1	26.9	14.422	0.412
$n=50$	4.1	7.9	12	1.100	0.956	11.6	57.6	69.3	4.435	0.823
$n=100$	4	7.8	11.7	1.242	0.951	14.9	84.5	99.5	1.269	0.950
$n=200$	3.8	7.9	11.7	1.321	0.950	15	84.9	99.9	1.307	0.950
$n=400$	3.7	7.9	11.7	1.377	0.947	15	85	100	1.346	0.949
n	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)	$\lvert\hat{S}\cap S\rvert$ (20)	$\lvert\hat{S}\setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)
$n=25$	5.1	6	11.1	1.248	0.967	4.7	18.6	23.3	27.783	0.303
$n=50$	5.4	6.1	11.5	1.761	0.955	14	56	70	12.172	0.700
$n=100$	5.3	6.2	11.5	2.000	0.951	20	79.9	99.9	1.988	0.951
$n=200$	5.1	6.3	11.5	2.116	0.950	20	80	99.9	2.091	0.950
$n=400$	5.2	6.2	11.3	2.161	0.948	20	80	100	2.154	0.949

Table A.18: Results of RelaxL and SqrtL for Scenario 2 with $\rho = 0.5$. Oracle values are in brackets.

	ScalL					DC.VS				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.556)	**% Dev** (0.9)
$n = 25$	0.9	3.2	4.1	3.453	0.627	0.4	1.5	1.9	5.273	0.464
$n = 50$	1.8	6.5	8.3	1.659	0.820	1.1	3.8	5	3.404	0.662
$n = 100$	2.2	8.1	10.3	0.659	0.935	2.2	7.8	10	0.495	0.951
$n = 200$	2.1	7.9	10.1	0.782	0.926	2.3	7.7	10	0.523	0.950
$n = 400$	2.2	7.9	10.1	0.706	0.932	2.3	7.7	10	0.538	0.949
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.389)	**% Dev** (0.9)
$n = 25$	1.6	2.7	4.3	7.113	0.685	0.8	1.2	2	11.949	0.512
$n = 50$	3	5.2	8.2	2.665	0.887	2.2	2.8	5	6.114	0.756
$n = 100$	3.5	6.8	10.2	1.391	0.945	3.4	6.6	10	1.238	0.951
$n = 200$	3.4	6.8	10.2	1.320	0.950	3.5	6.5	10	1.307	0.950
$n = 400$	3.4	7.1	10.4	1.374	0.947	3.4	6.6	10	1.346	0.949
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.222)	**% Dev** (0.9)
$n = 25$	1.8	2	3.8	13.803	0.627	0.9	1	1.9	21.090	0.464
$n = 50$	3.9	4.4	8.3	5.963	0.836	2.4	2.6	5	13.616	0.662
$n = 100$	4.8	5.4	10.2	2.703	0.934	4.7	5.3	10	1.981	0.951
$n = 200$	4.8	5.5	10.2	2.261	0.946	4.9	5.1	10	2.091	0.950
$n = 400$	4.7	5.7	10.4	2.202	0.947	4.9	5.1	10	2.154	0.949

Table A.19: Results of ScalL and DC.VS for Scenario 2 with $\rho = 0.5$. Oracle values are in brackets.

	LASSO.min					LASSO.1se				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)
$n = 25$	9.8	56.8	66.7	0.548	0.943	9.6	52	62	0.561	0.941
$n = 50$	10	51.7	61.7	0.784	0.926	10	48	58	0.784	0.926
$n = 100$	10	45.8	55.8	0.888	0.918	10	44.2	54.2	0.888	0.918
$n = 200$	10	41.6	51.6	0.939	0.913	10	40	50	0.939	0.913
$n = 400$	10	36.1	46.1	0.968	0.912	10	35.5	45.5	0.968	0.912
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
$n = 25$	13.5	52.1	65.6	1.377	0.943	12.9	45.8	58.6	1.441	0.940
$n = 50$	13.8	47.4	61.2	1.960	0.925	13.5	44.1	57.7	1.962	0.925
$n = 100$	13.8	41.9	55.7	2.220	0.918	13.7	40.7	54.4	2.220	0.918
$n = 200$	13.9	37.2	51.1	2.347	0.913	13.8	36.7	50.5	2.348	0.913
$n = 400$	14	31.7	45.7	2.420	0.912	13.7	31.6	45.2	2.420	0.912
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)
$n = 25$	17.7	49	66.7	2.191	0.943	17.1	44.5	61.6	2.244	0.941
$n = 50$	17.8	43.9	61.7	3.137	0.926	17.4	40.5	58	3.136	0.926
$n = 100$	17.9	37.8	55.8	3.551	0.918	17.6	36.7	54.2	3.550	0.918
$n = 200$	18.1	33.5	51.6	3.755	0.913	17.7	32.3	50	3.756	0.913
$n = 400$	18.1	27.9	46.1	3.872	0.912	17.5	28	45.5	3.872	0.912

Table A.20: Results of LASSO.min and LASSO.1se for Scenario 2 with $\rho = 0.9$. Oracle values are in brackets.

n	AdapL.min					AdapL.1se				
	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)
$n = 25$	8.44	1	9.44	0.621	0.935	5	0.5	5.5	2.392	0.759
$n = 50$	9.46	0.54	10	0.789	0.926	9.3	0.6	9.9	0.837	0.922
$n = 100$	9.85	0.15	10	0.891	0.918	9.8	0.2	10	0.891	0.918
$n = 200$	9.98	0.04	10.02	0.941	0.913	9.7	0.3	10	0.941	0.913
$n = 400$	10	0	10	0.969	0.911	10	0	10	0.969	0.911
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
$n = 25$	8.08	0.21	8.29	1.750	0.928	4.8	0.1	4.9	5.530	0.780
$n = 50$	9.67	0.14	9.82	2.010	0.924	7.6	0.1	7.7	3.274	0.876
$n = 100$	9.97	0.03	10	2.229	0.917	9.5	0	9.5	2.453	0.909
$n = 200$	10	0	10	2.353	0.913	9.4	0.6	10	2.353	0.913
$n = 400$	10	0	10	2.423	0.912	9.9	0.1	10	2.423	0.912
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)
$n = 25$	9.42	0.02	9.44	2.485	0.935	5.5	0	5.5	9.633	0.758
$n = 50$	10	0	10	3.155	0.926	9.8	0	9.9	3.350	0.922
$n = 100$	10	0	10	3.566	0.918	10	0	10	3.566	0.918
$n = 200$	10.02	0	10.02	3.764	0.913	9.9	0.1	10	3.764	0.913
$n = 400$	10	0	10	3.876	0.911	10	0	10	3.876	0.911

Table A.21: Results of AdapL.min and AdapL.1se for Scenario 2 with $\rho = 0.9$. Oracle values are in brackets.

n	SCAD					Dant				
	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)
$n = 25$	8.9	3.4	12.3	0.562	0.942	4.1	36.7	40.8	3.224	0.671
$n = 50$	9.6	1.8	11.4	0.788	0.926	6	54.3	60.3	2.980	0.721
$n = 100$	9.8	0.4	10.2	0.891	0.918	8.2	73.6	81.8	2.090	0.806
$n = 200$	10	0.2	10.1	0.941	0.913	9.7	86.9	96.6	1.197	0.889
$n = 400$	10	0	10	0.969	0.911	10	89.8	99.8	0.984	0.910
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
$n = 25$	10	1.6	11.6	1.430	0.941	6.9	32.1	39	7.290	0.700
$n = 50$	10.9	0.4	11.3	1.972	0.925	9.4	43.3	52.7	6.630	0.750
$n = 100$	10.5	0	10.5	2.229	0.918	10.8	48.8	59.5	5.922	0.781
$n = 200$	10.4	0	10.4	2.353	0.913	11.2	50.4	61.6	5.829	0.784
$n = 400$	10.1	0	10.1	2.423	0.912	11.1	50.2	61.3	6.013	0.781
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)
$n = 25$	11.8	0.5	12.3	2.248	0.942	8.2	32.7	40.9	12.883	0.671
$n = 50$	11.4	0	11.4	3.154	0.926	12	48.3	60.3	11.916	0.721
$n = 100$	10.2	0	10.2	3.566	0.918	16.3	65.5	81.8	8.359	0.806
$n = 200$	10.2	0	10.2	3.764	0.913	19.3	77.3	96.6	4.781	0.889
$n = 400$	10	0	10	3.876	0.911	20	79.8	99.8	3.936	0.910

Table A.22: Results of SCAD and Dant for Scenario 2 with $\rho = 0.9$. Oracle values are in brackets.

	RelaxL					SqrtL				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)
$n = 25$	2.8	8.4	11.2	0.574	0.941	2.2	19.7	21.9	7.428	0.265
$n = 50$	3	8.8	11.8	0.794	0.925	6.5	58.1	64.6	3.858	0.637
$n = 100$	2.8	8.9	11.7	0.904	0.916	10	89.7	99.7	0.907	0.916
$n = 200$	2.9	8.7	11.7	0.959	0.911	10	89.9	99.8	0.939	0.913
$n = 400$	2.9	8.7	11.6	0.987	0.910	10	89.8	99.8	0.968	0.912
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
$n = 25$	4.3	6.6	10.9	1.470	0.939	4.5	20.5	25	15.906	0.372
$n = 50$	4.6	7.4	11.9	1.979	0.925	11	53.5	64.5	6.375	0.759
$n = 100$	4.6	7.4	12	2.235	0.917	14.8	82.9	97.6	2.399	0.911
$n = 200$	4.4	7.4	11.8	2.370	0.912	15	84.9	99.9	2.347	0.913
$n = 400$	4.5	7.2	11.7	2.438	0.911	15	84.9	99.9	2.420	0.912
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)
$n = 25$	6.1	5.2	11.2	2.296	0.941	4.4	17.5	21.9	29.766	0.264
$n = 50$	6.3	5.6	11.9	3.176	0.925	12.9	51.7	64.6	15.434	0.637
$n = 100$	6.2	5.6	11.7	3.618	0.916	19.9	79.8	99.7	3.627	0.916
$n = 200$	6.2	5.5	11.7	3.836	0.911	20	79.9	99.8	3.755	0.913
$n = 400$	6.1	5.5	11.6	3.919	0.910	20	79.9	99.8	3.871	0.912

Table A.23: Results of RelaxL and SqrtL for Scenario 2 with $\rho = 0.9$. Oracle values are in brackets.

	ScalL					DC.VS				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1)	**% Dev** (0.9)
$n = 25$	1	2.9	3.9	3.848	0.603	0.5	1.4	1.9	5.415	0.463
$n = 50$	1.9	6	7.9	2.072	0.795	1.2	3.7	4.9	3.682	0.656
$n = 100$	2.5	7.8	10.2	1.034	0.905	2.4	7.6	10	0.895	0.917
$n = 200$	2.4	7.6	10	1.094	0.898	2.3	7.7	10	0.941	0.913
$n = 400$	2.5	7.8	10.4	1.111	0.899	2.4	7.6	10	0.969	0.911
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.5)	**% Dev** (0.9)
$n = 25$	1.7	2.2	3.9	8.185	0.654	0.9	1.1	1.9	12.770	0.495
$n = 50$	3.1	4.7	7.8	3.915	0.844	2.3	2.7	5	7.034	0.734
$n = 100$	3.8	6.4	10.2	2.449	0.910	3.7	6.3	10	2.231	0.917
$n = 200$	3.9	6.4	10.3	2.380	0.912	3.7	6.3	10	2.353	0.913
$n = 400$	4	6.5	10.5	2.567	0.906	3.8	6.2	10	2.423	0.912
n	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (20)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (4)	**% Dev** (0.9)
$n = 25$	2	1.7	3.7	15.213	0.606	1	0.9	1.9	21.661	0.463
$n = 50$	4.2	3.7	7.8	8.220	0.796	2.6	2.3	4.9	14.728	0.656
$n = 100$	5.4	4.8	10.2	4.143	0.904	5.4	4.6	10	3.579	0.917
$n = 200$	5.4	4.6	10.1	4.068	0.906	5.3	4.7	10	3.764	0.913
$n = 400$	5.6	5	10.5	4.009	0.908	5.4	4.6	10	3.876	0.911

Table A.24: Results of ScalL and DC.VS for Scenario 2 with $\rho = 0.9$. Oracle values are in brackets.

A.6.3 Scenario 3 (Toeplitz covariance)

Scenario 3.a

	LASSO.min					LASSO.1se				
					$\rho = 0.5$					
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)
$n = 25$	9.4	12.1	21.5	0.124	0.984	7.2	4.4	11.5	0.965	0.896
$n = 50$	13.5	18.2	31.7	0.190	0.983	12.9	7	19.9	0.488	0.955
$n = 100$	14.9	16.3	31.3	0.545	0.951	14.9	5.1	20	0.766	0.932
$n = 200$	15	14.5	29.5	0.825	0.927	15	2.9	17.9	0.971	0.914
$n = 400$	15	14.4	29.4	0.971	0.914	15	1.5	16.5	1.063	0.906
					$\rho = 0.9$					
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
$n = 25$	7.4	7.8	15.2	0.676	0.982	6.9	1.8	8.7	1.639	0.956
$n = 50$	9	7.1	16.2	1.894	0.950	8.7	1.6	10.3	2.630	0.931
$n = 100$	11.2	6	17.2	2.814	0.928	10.7	0.9	11.6	3.267	0.916
$n = 200$	12.8	6.5	19.2	3.302	0.916	12.4	0.6	13	3.587	0.908
$n = 400$	14.1	5.8	19.9	3.620	0.908	13.9	0.4	14.3	3.762	0.905

Table A.25: Results of LASSO.min and LASSO.1se for Scenario 3.a. Oracle values are in brackets.

	AdapL.min					AdapL.1se				
					$\rho = 0.5$					
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)
$n = 25$	5.7	3.2	8.9	0.333	0.968	2.9	0.7	3.6	2.230	0.783
$n = 50$	9.8	2.6	12.4	0.643	0.942	5.2	0.3	5.5	1.992	0.818
$n = 100$	13.8	1.9	15.7	0.867	0.923	6.8	0	6.8	1.932	0.827
$n = 200$	15	11.5	26.5	0.834	0.926	9.7	0	9.7	1.568	0.861
$n = 400$	15	9	24	0.995	0.912	13.7	0	13.7	1.195	0.894
					$\rho = 0.9$					
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
$n = 25$	3.3	0.8	4.1	2.056	0.945	2	0	2	4.433	0.880
$n = 50$	3.9	0.4	4.3	3.144	0.917	2.4	0	2.4	4.708	0.876
$n = 100$	4.5	0.2	4.7	3.669	0.906	2.6	0	2.7	4.963	0.872
$n = 200$	8.8	4.7	13.5	3.311	0.916	3.5	0	3.6	4.526	0.885
$n = 400$	11	5.3	16.3	3.593	0.909	4	0	4	4.543	0.885

Table A.26: Results of AdapL.min and AdapL.1se for Scenario 3.a. Oracle values are in brackets.

	SCAD					Dant				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.139)	**% Dev** (0.9)
$n = 25$	5	3.4	8.4	1.157	0.880	4.3	1.1	5.3	2.313	0.769
$n = 50$	8.6	6.4	15	0.679	0.937	6.7	0.3	7	2.158	0.802
$n = 100$	12.7	11.6	24.3	0.660	0.941	9	0	9.1	1.865	0.833
$n = 200$	15	13.3	28.3	0.824	0.927	11.1	0	11.1	1.649	0.854
$n = 400$	15	6.1	21.1	1.015	0.910	12.8	0	12.8	1.443	0.873
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
$n = 25$	3.1	3	6.1	2.086	0.944	3.6	0	3.7	4.237	0.885
$n = 50$	3.8	4.1	7.8	2.716	0.929	4.1	0	4.2	4.892	0.871
$n = 100$	4.5	3.6	8.1	3.443	0.911	4.9	0	4.9	5.013	0.871
$n = 200$	6.7	6.2	12.8	3.476	0.911	5.6	0	5.6	4.969	0.873
$n = 400$	9	7	16	3.658	0.907	6.6	0	6.6	4.920	0.875

Table A.27: Results of SCAD and Dant for Scenario 3.a. Oracle values are in brackets.

	RelaxL					SqrtL				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (1.139)	**% Dev** (0.9)
$n = 25$	6.7	4.3	11	1.145	0.876	25.9	2	7.9	1.518	0.836
$n = 50$	12.2	6.8	19	0.615	0.944	11.8	2.4	14.2	0.832	0.922
$n = 100$	14.5	2.9	17.4	0.870	0.922	14.8	2.5	17.3	0.865	0.923
$n = 200$	15	0.9	15.9	1.025	0.909	15	2.5	17.5	0.982	0.913
$n = 400$	15	0.6	15.6	1.078	0.905	15	2.4	17.4	1.053	0.907
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (15)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
$n = 25$	6.3	2.3	8.6	1.914	0.948	7.1	1.3	8.4	3.101	0.916
$n = 50$	7.9	2.3	10.2	2.743	0.928	9	1.5	10.5	3.381	0.911
$n = 100$	10	1.7	11.7	3.259	0.916	11	2	13	3.207	0.917
$n = 200$	11.9	2	13.9	3.539	0.910	11	1	12	3.524	0.910
$n = 400$	13.6	1.6	15.2	3.729	0.905	14	1.8	15.9	3.717	0.906

Table A.28: Results of RelaxL and SqrtL for Scenario 3.a. Oracle values are in brackets.

	ScalL					DC.VS				
						$\rho = 0.5$				
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)
$n = 25$	6.9	3.3	10.2	0.692	0.926	1.6	0.4	2	4.249	0.598
$n = 50$	12.4	3.5	15.9	0.657	0.938	4.2	0.8	5	2.516	0.769
$n = 100$	14.9	5.3	20.1	0.795	0.929	8.6	2.4	11	1.439	0.871
$n = 200$	15	3.5	18.5	0.956	0.915	14.1	5.5	19.6	0.988	0.912
$n = 400$	15	3.4	18.4	1.039	0.908	15	1.6	16.6	1.060	0.906
						$\rho = 0.9$				
n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)
$n = 25$	7.1	1.7	8.9	1.606	0.957	1.6	0.4	2	5.683	0.847
$n = 50$	9	1.9	10.9	2.530	0.933	2.7	1.7	4.5	4.010	0.894
$n = 100$	11.1	5.8	16.9	3.219	0.918	3.7	2.2	5.8	4.031	0.896
$n = 200$	12.7	2	14.7	3.488	0.911	4.5	1.9	6.4	4.101	0.895
$n = 400$	14.1	2	16.1	3.701	0.906	4.8	0.7	5.6	4.285	0.891

Table A.29: Results of ScalL and DC.VS for Scenario 3.a. Oracle values are in brackets.

Scenario 3.b

	LASSO.min					LASSO.1se				
						$\rho = 0.5$				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.278)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.278)	**% Dev** (0.9)
$n = 25$	5.3	14.2	19.5	0.155	0.929	2.9	5.2	8.1	0.843	0.659
$n = 50$	9.8	26.5	36.3	0.034	0.986	9.3	14.3	23.6	0.117	0.952
$n = 100$	10	23.6	33.6	0.123	0.955	10	11.9	21.9	0.170	0.938
$n = 200$	10	21.8	31.8	0.195	0.929	10	8	18	0.228	0.917
$n = 400$	10	21.8	31.8	0.234	0.915	10	5.7	15.7	0.255	0.907
						$\rho = 0.9$				
n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.53)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.53)	**% Dev** (0.9)
$n = 25$	4.4	16.7	21.1	0.028	0.994	3.3	11	14.3	0.144	0.968
$n = 50$	7.3	21	28.3	0.147	0.971	6.8	15.7	22.5	0.229	0.955
$n = 100$	9.4	21.6	31.1	0.309	0.940	9.3	16.4	25.7	0.366	0.929
$n = 200$	10	20.8	30.8	0.417	0.920	10	15.2	25.2	0.449	0.914
$n = 400$	10	20.8	30.8	0.471	0.911	10	14.6	24.6	0.488	0.908

Table A.30: Results of LASSO.min and LASSO.1se for Scenario 3.b. Oracle values are in brackets.

	AdapL.min					AdapL.1se				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)
$n = 25$	3.6	6.2	9.9	0.104	0.959	1.4	1.7	3.1	0.997	0.615
$n = 50$	9.1	4.4	13.6	0.149	0.943	7.7	0.9	8.6	0.397	0.841
$n = 100$	10	1.6	11.6	0.221	0.919	10	0	10	0.248	0.909
$n = 200$	10	11.3	21.3	0.211	0.923	10	0	10	0.262	0.905
$n = 400$	10	9.5	19.5	0.246	0.911	10	0	10	0.270	0.902
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)
$n = 25$	2.3	5.4	7.7	0.153	0.967	1.1	2.7	3.8	0.741	0.839
$n = 50$	4.7	5.5	10.2	0.308	0.939	2.8	3	5.8	0.715	0.858
$n = 100$	7.9	3.4	11.4	0.418	0.919	6.2	2.1	8.3	0.618	0.880
$n = 200$	9.8	8.3	18.1	0.426	0.918	9.4	0.6	10	0.499	0.904
$n = 400$	10	7.2	17.2	0.479	0.909	9.9	0.1	10	0.513	0.903

Table A.31: Results of AdapL.min and AdapL.1se for Scenario 3.b. Oracle values are in brackets.

	SCAD					Dant				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)
$n = 25$	2.1	3.6	5.7	0.854	0.656	1.8	2	3.8	1.131	0.556
$n = 50$	8	8.7	16.7	0.187	0.921	4.7	2.8	7.6	0.798	0.693
$n = 100$	10	3.3	13.3	0.204	0.925	7.7	1.4	9.1	0.608	0.777
$n = 200$	10	2	12	0.248	0.910	9.5	0.2	9.8	0.348	0.874
$n = 400$	10	1.8	11.8	0.263	0.904	10	0	10	0.273	0.901
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)
$n = 25$	2	7.2	9.3	0.282	0.936	1.1	4.8	5.8	1.018	0.778
$n = 50$	4.4	8.9	13.3	0.281	0.945	1.8	6.1	7.9	1.091	0.786
$n = 100$	6.9	7.6	14.4	0.402	0.922	2.9	7.4	10.3	1.030	0.799
$n = 200$	8.4	4.8	13.2	0.490	0.906	4.7	7.7	12.4	0.907	0.827
$n = 400$	9	2.6	11.6	0.530	0.900	7.1	6.1	13.2	0.717	0.864

Table A.32: Results of SCAD and Dant for Scenario 3.a. Oracle values are in brackets.

	RelaxL					SqrtL				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)
$n = 25$	2.8	4.8	7.6	0.783	0.680	1.9	2.1	4	1.073	0.569
$n = 50$	9.3	15.8	25.1	0.117	0.952	5.8	3.7	9.6	0.612	0.752
$n = 100$	10	6.7	16.7	0.203	0.925	10	4.4	14.4	0.221	0.918
$n = 200$	10	1.8	11.8	0.253	0.908	10	4.2	14.2	0.244	0.912
$n = 400$	10	0.7	10.7	0.268	0.903	10	4.5	14.5	0.259	0.906
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)
$n = 25$	3.4	11.6	15	0.160	0.964	2.1	8	10.1	1.897	0.949
$n = 50$	6.7	17.1	23.8	0.216	0.957	5.5	13.3	18.9	2.681	0.93
$n = 100$	9.1	17.2	26.2	0.355	0.931	9.1	16.5	25.6	3.207	0.917
$n = 200$	9.9	14.4	24.2	0.446	0.915	10	18.2	28.1	3.524	0.910
$n = 400$	10	11.1	21.1	0.490	0.907	10	20	30	3.717	0.906

Table A.33: Results of RelaxL and SqrtL for Scenario 3.b. Oracle values are in brackets.

	ScalL					DC.VS				
					$\boldsymbol{\rho = 0.5}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.278)	**% Dev** (0.9)
$n = 25$	3	4.2	7.2	0.461	0.813	0.9	1.1	2	1.321	0.500
$n = 50$	7.1	5.1	12.2	0.385	0.844	3.1	1.9	5	0.986	0.626
$n = 100$	10	6.6	16.6	0.210	0.923	9.7	1.3	11	0.270	0.900
$n = 200$	10	5.5	15.5	0.237	0.914	10	5	15	0.235	0.915
$n = 400$	10	5.5	15.6	0.256	0.907	10	1.5	11.5	0.263	0.904
					$\boldsymbol{\rho = 0.9}$					
n	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)	$\lvert\hat{\boldsymbol{S}} \cap \boldsymbol{S}\rvert$ (10)	$\lvert\hat{\boldsymbol{S}} \setminus \boldsymbol{S}\rvert$	$\lvert\hat{\boldsymbol{S}}\rvert$	**MSE** (0.53)	**% Dev** (0.9)
$n = 25$	2.5	8.9	11.4	0.265	0.937	0.4	1.6	2	1.712	0.639
$n = 50$	5.9	13.3	19.2	0.292	0.941	1.3	3.7	5	1.081	0.787
$n = 100$	9.1	17.8	26.8	0.382	0.925	4.3	6.1	10.3	0.636	0.876
$n = 200$	10	15	25	0.449	0.914	7.5	4.5	12	0.542	0.896
$n = 400$	10	15.2	25.2	0.487	0.908	8.9	2.1	11	0.537	0.898

Table A.34: Results of ScalL and DC.VS for Scenario 3.b. Oracle values are in brackets.

A.7 Tables and figures

A.7.1 Scenario 2: dependence by blocks

	$s = 15$		
prob:	$(0.6, 0.8]$	$(0.8, 0.9]$	$(0.9, 1]$
$n = 50$	27, 29, 26, 50, **14**, **11**, **13**, **15**, 40, 18	17, 19, 16, 30	20, **9**, **7**, **10**, **8**, **6**, **5**, **4**, **3**, **2**, **1**
$n = 100$	39, 28, **13**, 29, **14**, 40, **11**, **12**, 16	17, **15**, 18, 30, 19	20, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**,**1**
$n = 200$	27, 28, 40, 29, **11**, 16, **15**, **14**, **12**	17, 30, 19, **13**, 18	20, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**, **1**
	$s = 20$		
prob:	$(0.6, 0.8]$	$(0.8, 0.9]$	$(0.9, 1]$
$n = 50$	**12**, 60, **14**, **13**, **17**, **15**, 29, 50, **11**, **18**	**16**, 40, **19**, 30	**20**, **3**, **2**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **1**
$n = 100$	50, 28, 29, **12**, **18**, **16**, **15**, **14**, **13**, 40, **17**, **11**	**19**, 30	**20**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**, **1**
$n = 200$	27, 40, 28, 29, **16**, **15**, **14**, **11**, **19**, **12**	**18**, **17**, 30, **13**	**20**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**, **1**

Table A.35: The first 25 covariates with highest selection probability for the LASSO.min in Scenario 2 with $\rho = 0.5$, in ascending order. The important covariates of the model are in **bold**.

	$s = 15$		
prob:	$(0.6, 0.8]$	$(0.8, 0.9]$	$(0.9, 1]$
$n = 50$	27, 26, **11**, 40, 29, **12**, **13**, **14**, 18, 16	**15**, 17, 19, 30	20, **10**, **8**, **7**, **6**, **9**, **5**, **4**, **3**, **2**, **1**
$n = 100$	39, 28, **13**, 40, **12**, **11**, 29, **15**, **14**, 16, 17	18, 19, 30	20, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**, **1**
$n = 200$	24, 40, 28, 29, **12**, **13**, **11**, **14**	16, **15**, 18, 19, 17, 30	20, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **2**, **1**
	$s = 20$		
prob:	$(0.6, 0.8]$	$(0.8, 0.9]$	$(0.9, 1]$
$n = 50$	27, **14**, 50, **13**, **18**, **12**, 29, **11**, **15**, **16**, 40, **17**	**19**, 30	**20**, **6**, **3**, **1**, **10**, **9**, **8**, **7**, **5**, **4**, **2**
$n = 100$	50, 28, **13**, 40, 29, **16**, **17**, **15**, **14**, **18**, **12**, **11**	**19**, 30	**20**, **2**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **1**
$n = 200$	23, 40, 29, 28, **12**, **15**, **14**, **13**, **16**, **17**	**18**, **11**, **19**, 30	**20**, **2**, **10**, **9**, **8**, **7**, **6**, **5**, **4**, **3**, **1**

Table A.36: The first 25 covariates with highest selection probability for the LASSO.min in Scenario 2 with $\rho = 0.9$, in ascending order. The important covariates of the model are in **bold**.

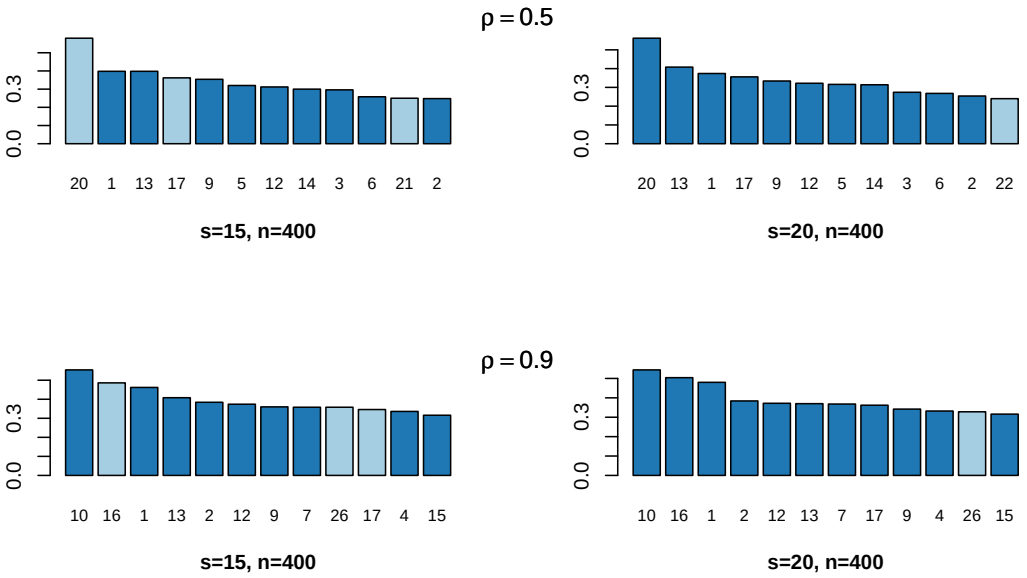


Figure A.12: The 12 covariates with highest selection probability for the RelaxL in Scenario 2 with $\rho = 0.5$ (the first row) and Scenario 2 with $\rho = 0.9$ (the second row) taking $n = 400$. The important covariates of the model are in dark color while the noisy ones in soft color.

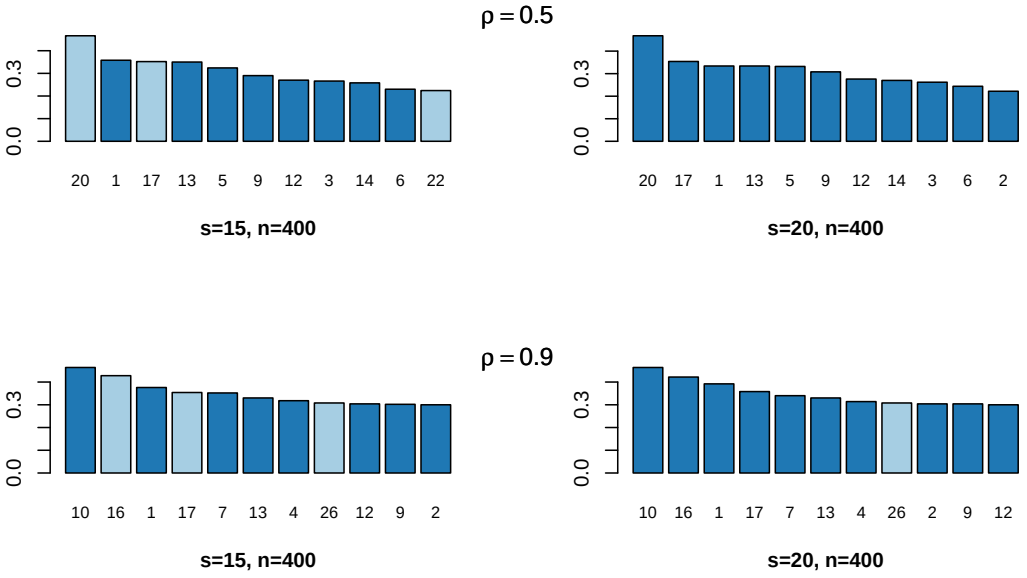


Figure A.13: The 11 covariates with highest selection probability for the ScalL in Scenario 2 with $\rho = 0.5$ (the first row) and Scenario 2 with $\rho = 0.9$ (the second row) taking $n = 400$. The important covariates of the model are in dark color while the noisy ones in soft color.

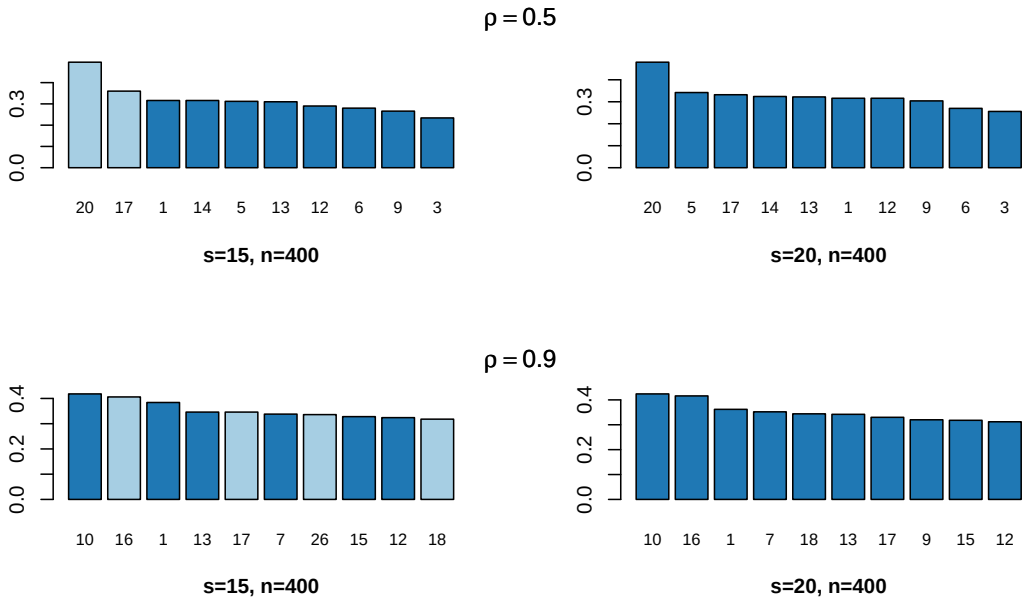


Figure A.14: The 10 covariates with highest selection probability for the DC.VS in Scenario 2 with $\rho = 0.5$ (the first row) and Scenario 2 with $\rho = 0.9$ (the second row) taking $n = 400$. The important covariates of the model are in dark color while the noisy ones in soft color.

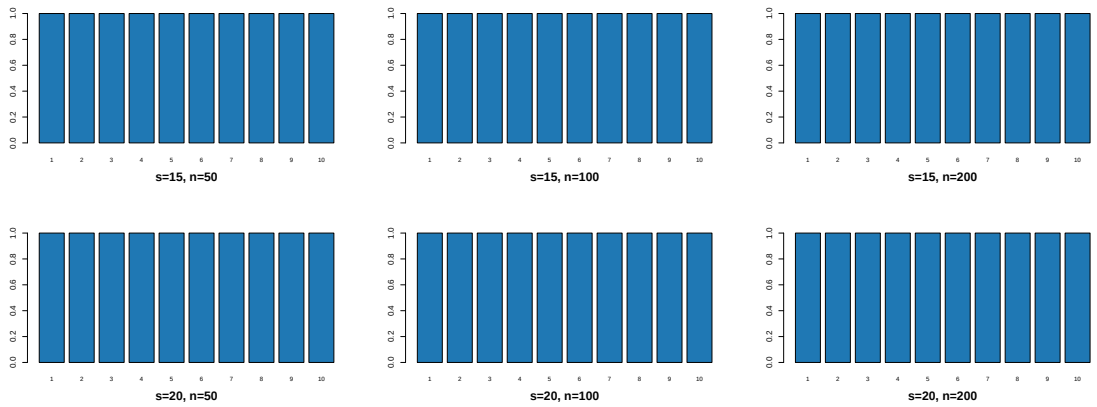


Figure A.15: Percentage of times a representative of the 10 first covariates enters the model in Scenario 2 with $\rho = 0.5$ for the LASSO.min.

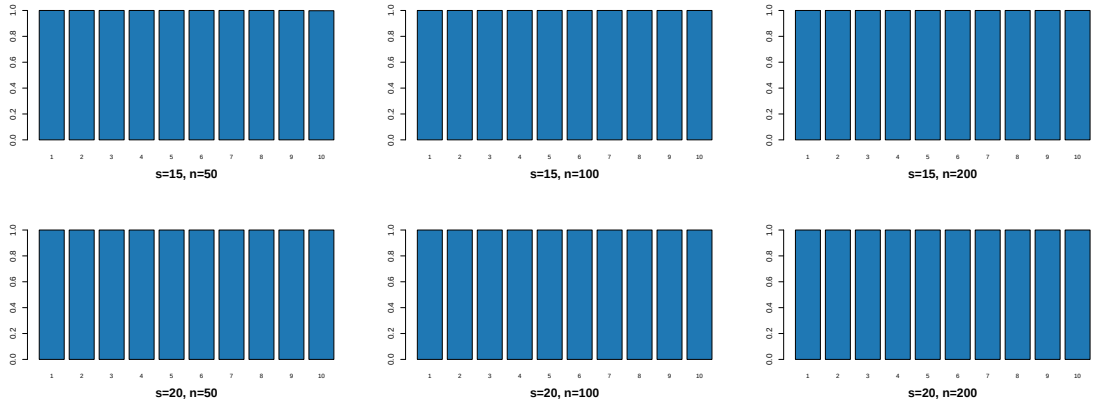


Figure A.16: Percentage of times a representative of the 10 first covariates enters the model in Scenario 2 with $\rho = 0.9$ for the LASSO.min.

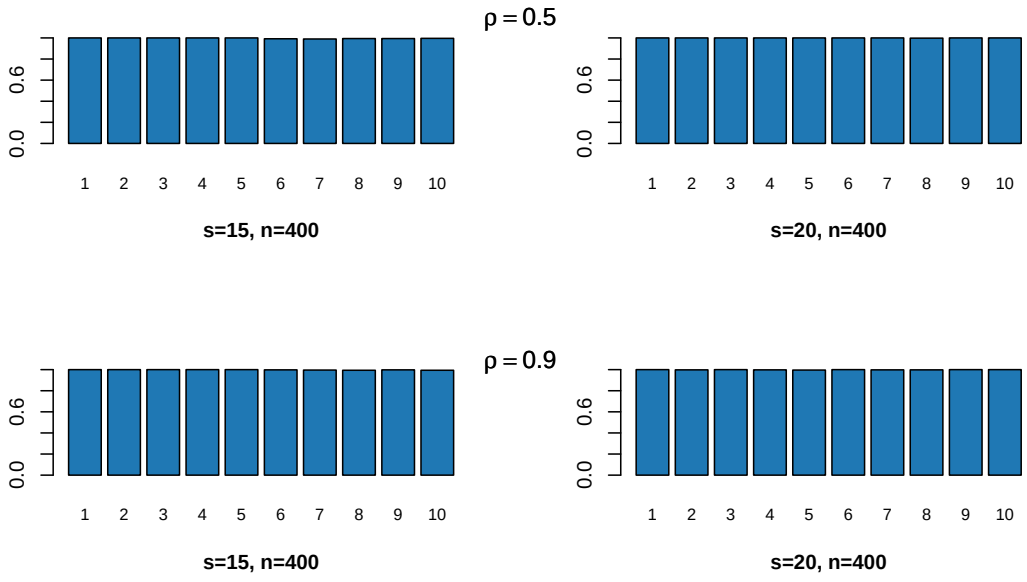


Figure A.17: Percentage of times a representative of the 10 first covariates enters the model in Scenario 2 for the RelaxL.

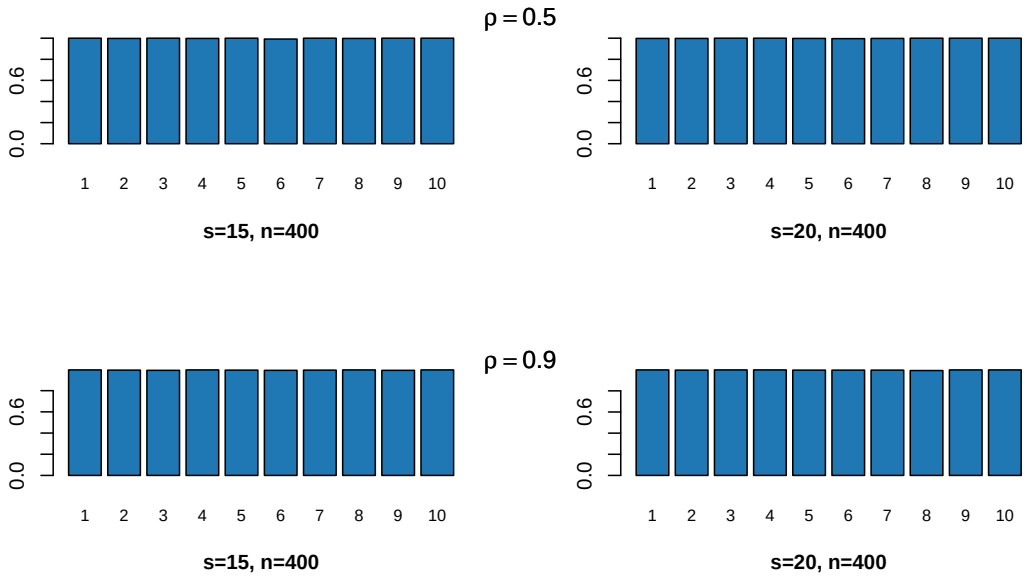


Figure A.18: Percentage of times a representative of the 10 first covariates enters the model in Scenario 2 for the ScalL.

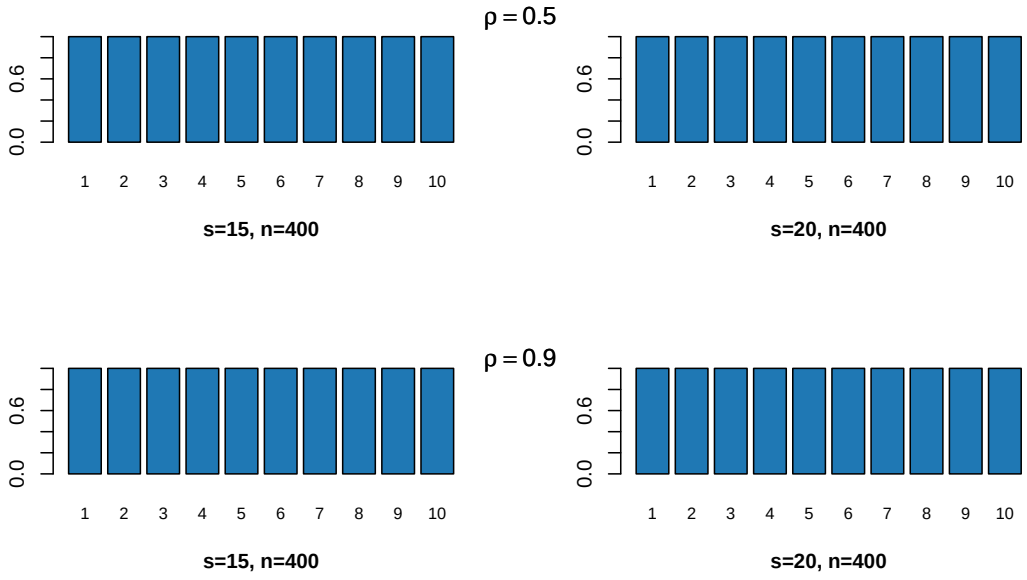


Figure A.19: Percentage of times a representative of the 10 first covariates enters the model in Scenario 2 for the DC.VS.

A.7.2 Scenario 3: Toeplitz covariance

Scenario 3.a

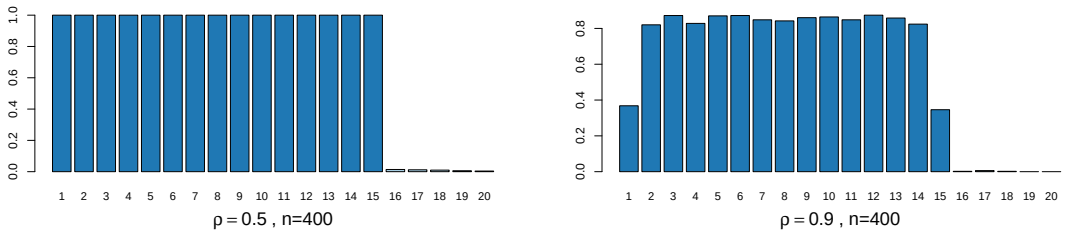


Figure A.20: Percentage of times each of the 20 first covariates enters the model in Scenario 3.a for the LASSO.BIC.

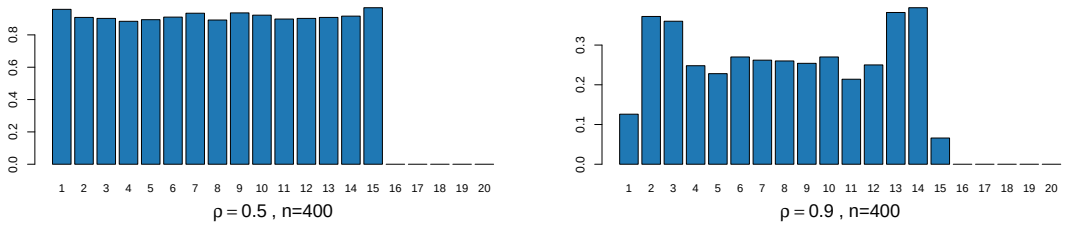


Figure A.21: Percentage of times each of the 20 first covariates enters the model in Scenario 3.a for the AdapL.1se.

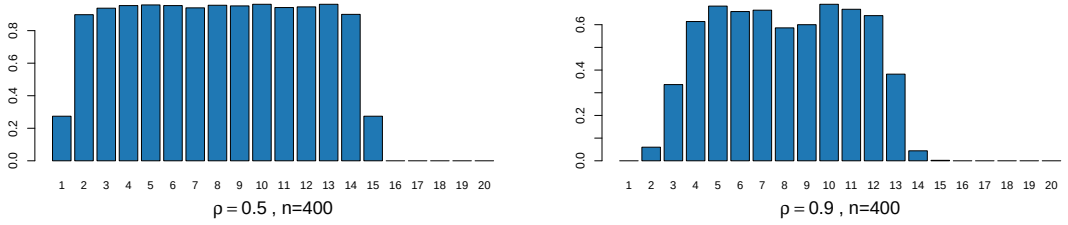


Figure A.22: Percentage of times each of the 20 first covariates enters the model in Scenario 3.a for the Dant.

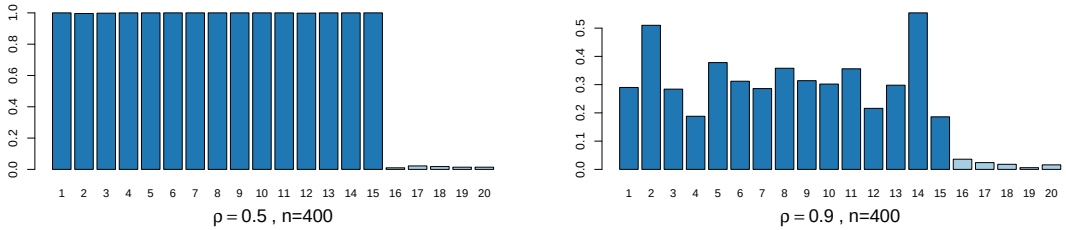


Figure A.23: Percentage of times each of the 20 first covariates enters the model in Scenario 3.a for the DC.VS.

Scenario 3.b

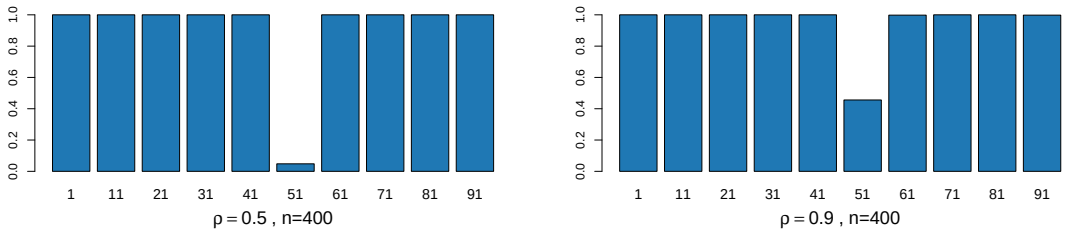


Figure A.24: Percentage of times a representative of the 10 relevant covariates enters the model in Scenario 3.b for the LASSO.BIC.

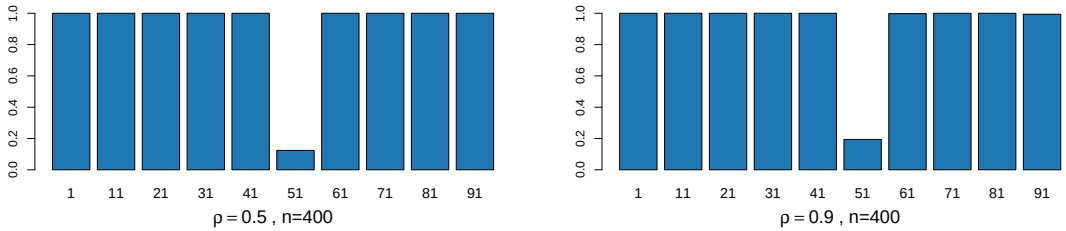


Figure A.25: Percentage of times a representative of the 10 relevant covariates enters the model in Scenario 3.b for the SCAD.

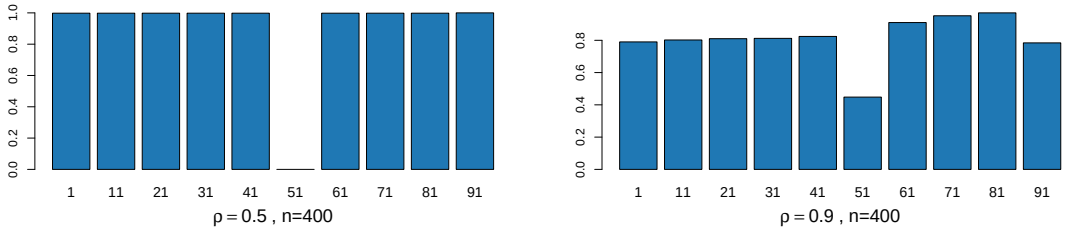


Figure A.26: Percentage of times a representative of the 10 relevant covariates enters the model in Scenario 3.b for the Dant.

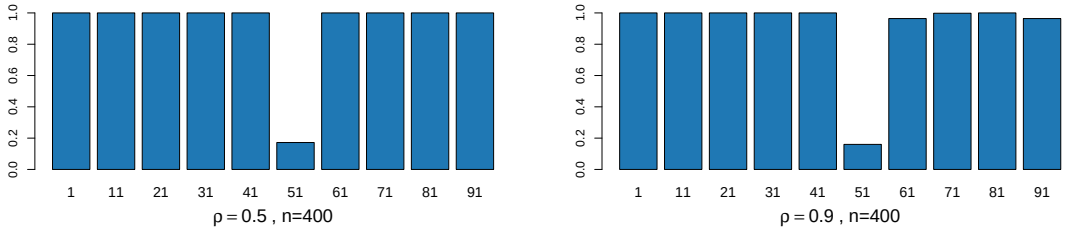


Figure A.27: Percentage of times a representative of the 10 relevant covariates enters the model in Scenario 3.b for the DC.VS.

Appendix B

Extra results for LASSO facing scale effects under dependence

B.1 Calculation of σ

Similar to Section A.1 of the Appendix A, the variance of the model error term, σ^2, is calculated for all scenarios considered in Section 3.2.1. Again, these quantities are computed allowing the model to explain, at most, the 90% of the deviance. For this aim, the variance is obtained verifying the condition displayed in equation (A.1).

B.1.1 Scenario 1 (Independence)

In the case of Scenario 1.a, the value of $\sigma \simeq 1.317616$. Similar arguments as the ones of Section A.1.1 of Appendix A for the $s = 10$ case are employed.

Otherwise, the formula of σ agrees for Scenario 1.b and 1.c. This is given by

$$\sigma = \sqrt{\frac{1-0.9}{0.9} 1.25^2 \sum_{j=1}^{s} \mathbb{V}(X_j)}. \tag{B.1}$$

Then, for $s = 10$ and having that $\mathbb{V}(X_1) = \mathbb{V}(X_2) = 0.5$, $\mathbb{V}(X_3) = \mathbb{V}(X_4) = 1$, $\mathbb{V}(X_5) = \mathbb{V}(X_6) = 3$, $\mathbb{V}(X_7) = \mathbb{V}(X_8) = 10$, $\mathbb{V}(X_9) = \mathbb{V}(X_{10}) = 25$, it is verified that $\sum_{j=1}^{s} \mathbb{V}(X_j) = 79$ and then we need to take $\sigma \simeq 3.703414$.

This is due to the fact that

$$\sigma^2 \overset{(a)}{=} \frac{1-\%Dev}{\%Dev} \sum_{j=1}^{p} \beta_j^2 \mathbb{V}(X_j) \overset{(b)}{\Rightarrow} \sigma^2 = \frac{1-\%Dev}{\%Dev} 1.25^2 \sum_{j=1}^{s} \mathbb{V}(X_j)$$

where (a) is true because $\mathbb{V}(< X, \beta >) = \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) = \beta_1^2\mathbb{V}(X_1) + \cdots + \beta_p^2\mathbb{V}(X_p)$ and (b) because of β structure with $\beta_1 = \cdots = \beta_s = 1.25$ and $\beta_j = 0$ for $j = s+1, \ldots, p$.

B.1.2 Scenario 2 (Toeplitz covariance with unit scales)

In the case of Scenario 2, the value of σ changes depending on the structure of the scenario.

- Scenario 2.a: only the first $s = 15$ covariates are important. Then, taking $\rho = 0.5$ we get that $\sigma \simeq 1.067189$ and for $\rho = 0.9$, $\sigma \simeq 1.951213$. We refer to Section A.1.3 of Appendix A for calculation guidelines.

- Scenario 2.b: there are $s = 10$ relevant variables placed every 3 sites in 3-30 locations.

$$\sigma = \sqrt{\frac{1-0.9}{0.9}\left(10 \cdot 0.5^2 + 2(0.5^2)\sum_{\substack{j,k=3 \\ j<k \\ j,k\equiv 1 \mod 3}}^{30} \rho^{|j-k|}\right)} \simeq \sqrt{\frac{1}{9}\left(2.5 + 0.5\sum_{\substack{j,k=3 \\ j<k \\ j,k\equiv 1 \mod 3}}^{30} \rho^{|j-k|}\right)},$$

then, for $\rho = 0.5$ this results in $\sigma \simeq 0.5899767$, while for $\rho = 0.9$, $\sigma \simeq 1.115418$.

The explanation is that

$$\begin{aligned}
\mathbb{V}(< X, \beta >) &= \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) \\
&= \dots \\
&= \sum_{j=1}^{p} \beta_j^2 + 2\sum_{\substack{j,k=3 \\ j<k \\ j,k\equiv 1 \mod 3}}^{30} \beta_j \rho^{|j-k|}\beta_k = 10 \cdot 0.5^2 + 2(0.5^2)\sum_{\substack{j,k=3 \\ j<k \\ j,k\equiv 1 \mod 3}}^{30} \rho^{|j-k|}.
\end{aligned}$$

B.1.3 Scenario 3 (Toeplitz covariance with different scales)

For Toeplitz covariance structure with different scales (Scenario 3), ideas of Scenario 2.b taking $\rho = 0.5$ for variance error term calculation (Section B.1.2) have been adapted. Two cases are considered: only relevant covariates have different scales from the unit (Scenario 3.a) and unimportant ones with different scales are added as well (Scenario 3.b). Here, we have that the model variance is given by

$$\begin{aligned}
\mathbb{V}(< X, \beta >) &= \mathbb{V}(X_1\beta_1 + \cdots + X_p\beta_p) \\
&= \beta_1^2\mathbb{V}(X_1) + \mathbb{V}(X_2\beta_2 + \cdots + X_p\beta_p) + 2\mathbb{C}(X_1\beta_1, X_2\beta_2 + \dots X_p\beta_p) \\
&= \beta_1^2\mathbb{V}(X_1) + \beta_2^2\mathbb{V}(X_2) + \mathbb{V}(X_3\beta_3 + \cdots + X_p\beta_p) + 2\mathbb{C}(X_2\beta_2, X_3\beta_3 + \cdots + X_p\beta_p) \\
&+ 2\mathbb{C}(X_1\beta_1, X_2\beta_2 + \cdots + X_p\beta_p) \\
&= \dots \\
&\overset{(a)}{=} \sum_{\substack{j=3 \\ j\equiv 1_{mod3}}}^{30} \beta_j^2\sigma_j^2 + 2\left[\sum_{\substack{j=3 \\ j\equiv 1_{mod3}}}^{30} \beta_j\sigma_j\left(\sum_{\substack{k>j=3 \\ k\equiv 1_{mod3}}}^{30} \beta_k\sigma_k\rho^{|j-k|}\right)\right] \\
&= (0.5)^2 \sum_{\substack{j=3 \\ j\equiv 1_{mod3}}}^{30} \sigma_j^2 + 2(0.5)^2 \sum_{\substack{k>j=3 \\ k\equiv 1_{mod3}}}^{30} \sigma_j\sigma_k\rho^{|j-k|}
\end{aligned}$$

where (a) is due to $\beta_j \neq 0$ for $mod_3(j) = 1$, and it is verified that $\mathbb{C}(X_j, X_k) = \sigma_j\sigma_k\rho^{|j-k|}$ for $mod_3(j) = mod_3(k)$.

Then, the variance of the error is given by

$$\sigma = \sqrt{\frac{1-0.9}{0.9}\left((0.5)^2 \sum_{\substack{j=3 \\ j\equiv 1_{mod3}}}^{30} \sigma_j^2 + 2(0.5)^2 \sum_{\substack{k>j=3 \\ k\equiv 1_{mod3}}}^{30} \sigma_j \sigma_k \rho^{|j-k|}\right)}$$

$$\simeq \sqrt{\frac{1}{9}\left(19.75 + 0.5 \sum_{\substack{k>j=3 \\ k\equiv 1_{mod3}}}^{30} \sigma_j \sigma_k \rho^{|j-k|}\right)},$$

and, for $\rho = 0.5$ we have that $\sigma \simeq 1.636796$ and for $\rho = 0.9$ we get $\sigma \simeq 2.796096$.

B.2 Calculation eigenvalues of covariance matrices

Scenario 1.b						**Scenario 1.c**					
70%	**80%**	**90%**	**95%**	**98%**	**99%**	**70%**	**80%**	**90%**	**95%**	**98%**	**99%**
49	66	83	91	96	98	10	31	65	82	92	95

Table B.1: Required covariates in Scenarios 1.b and 1.c to explain a certain percentage of variability.

	Scenario 2.a ($s = 15$)					**Scenario 2.b** ($s = 10$)				
$\boldsymbol{\rho}$	**80%**	**90%**	**95%**	**98%**	**99%**	**80%**	**90%**	**95%**	**98%**	**99%**
$\rho = 0.5$	8	11	13	15	15	8	9	10	10	10
$\rho = 0.9$	2	4	6	10	13	7	9	9	10	10

Table B.2: Required covariates in Scenarios 2.a and 2.b to explain a certain percentage of variability.

	Scenario 3.a						**Scenario 3.b**					
$\boldsymbol{\rho}$	**70%**	**80%**	**90%**	**95%**	**98%**	**99%**	**70%**	**80%**	**90%**	**95%**	**98%**	**99%**
$\rho = 0.5$	22	35	57	76	90	95	9	18	37	59	80	90
$\rho = 0.9$	6	8	15	26	51	71	4	6	12	19	35	53

Table B.3: Required covariates in Scenarios 3.a and 3.b to explain a certain percentage of variability.

B.3 Simulation results

Similar to Section A.6 of Appendix A, extra tables and figures of simulation results of Section 3.2.1 are collected.

B.3.1 Scenarios 1.a, 1.b and 1.c

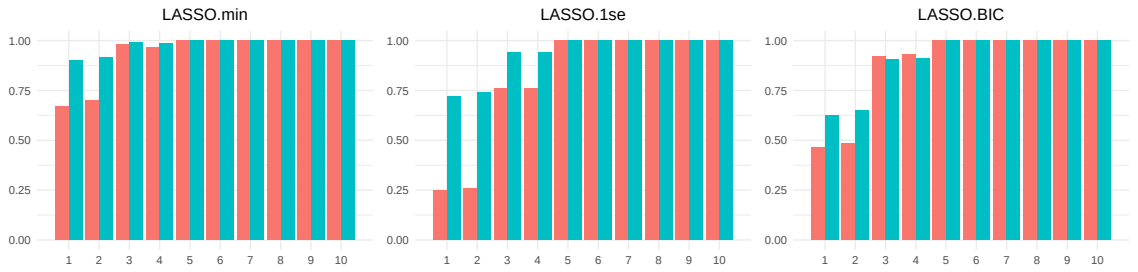


Figure B.1: Percentage of times each of the $s = 10$ relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 1.b.

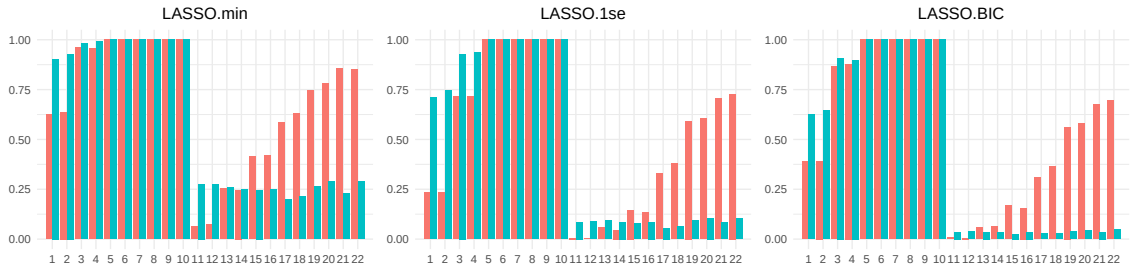


Figure B.2: Percentage of times each of the $s = 10$ relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 1.c.

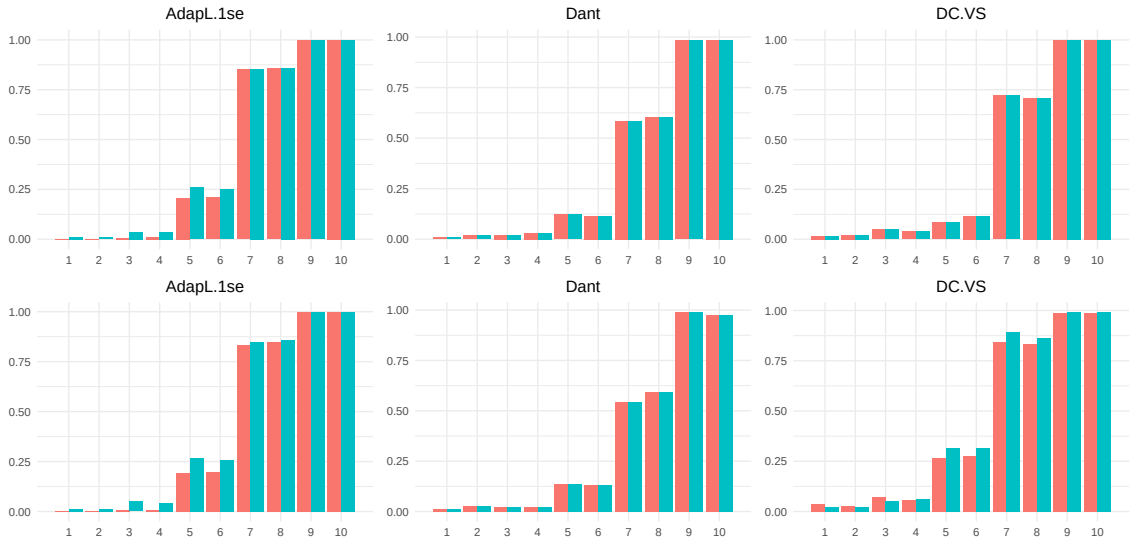


Figure B.3: Percentage of times each of the first 22 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 50$ in Scenario 1.b (the first row) and Scenario 1.c (the second row).

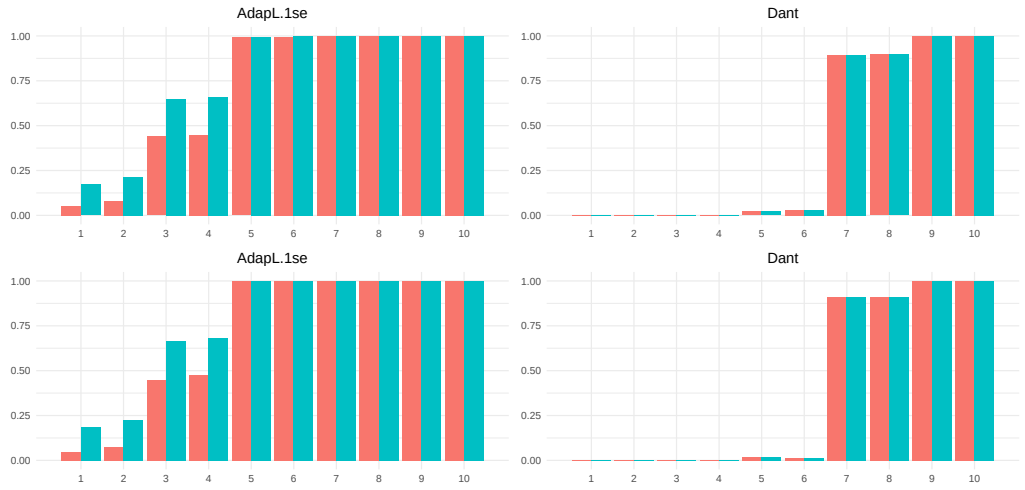


Figure B.4: Percentage of times each of the $s = 10$ relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 1.b (the first row) and Scenario 1.c (the second row).

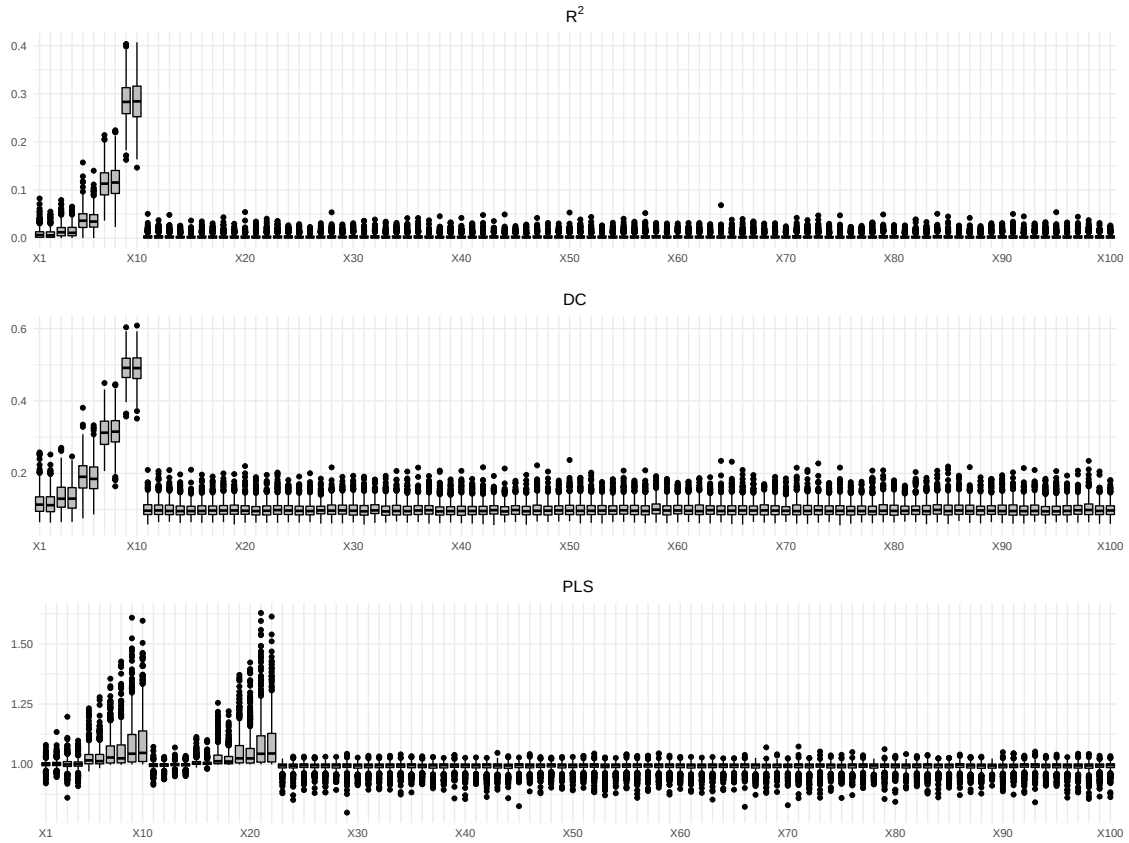


Figure B.5: Boxplots of the covariates loadings in terms of R^2, DC and PLS for $n = 300$ in Scenario 1.c with without standardization.

		Scenario 1.a					Scenario 1.b					Scenario 1.c				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (13.715)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (13.715)	**% Dev** (0.9)
LASSO.min	25	5.33	15.20	20.54	1.101	0.923	6.02	7.21	13.22	4.418	0.965	5.65	9.28	14.93	3.935	0.969
	50	9.80	27.35	37.15	0.184	0.989	7.23	9.02	16.24	6.257	0.953	7.01	11.57	18.57	6.147	0.953
	100	10	24.53	34.53	0.701	0.959	8.55	11.91	20.46	8.356	0.938	8.43	14.82	23.26	8.201	0.938
	150	10	23.47	33.47	0.986	0.942	9.31	15.23	24.54	9.060	0.933	9.18	18.21	27.39	9.025	0.932
	300	10	21.42	31.42	1.345	0.922	9.92	16.47	26.38	10.972	0.919	9.91	20.22	30.14	10.866	0.920
LASSO.1se	25	3.81	8.34	12.15	3.796	0.747	5.18	2.14	7.32	11.234	0.910	4.77	4.44	9.22	10.406	0.917
	50	9.22	14.42	23.64	0.791	0.949	6.17	1.82	7.99	12.624	0.905	5.89	4.32	10.20	12.366	0.905
	100	10	11.10	21.10	1.037	0.939	7.27	1.85	9.12	13.133	0.903	7.09	4.68	11.77	12.806	0.904
	150	10	8.58	18.58	1.276	0.926	8.02	2.29	10.31	12.961	0.904	7.90	5.66	13.56	12.626	0.883
	300	10	5.07	15.07	1.538	0.910	9.24	2.20	11.44	12.972	0.904	9.16	5.68	14.84	12.813	0.891
LASSO.BIC	25	5.99	17.99	23.98	0	1	6.50	12.41	18.92	0.259	0.998	6.12	13.84	19.95	0.172	0.999
	50	9.86	37.17	47.02	0.002	1	7.80	17.05	24.85	2.468	0.983	7.72	20.42	28.15	1.860	0.987
	100	9.99	87.73	97.72	0.013	0.999	9.79	81.81	91.60	0.714	0.995	9.79	82.75	92.54	0.647	0.995
	150	10	2.21	12.21	1.473	0.914	8.80	3.67	12.47	11.456	0.915	8.52	6.25	14.78	11.587	0.913
	300	10	1.35	11.35	1.616	0.906	9.66	2.22	11.88	12.671	0.907	9.42	4.77	14.19	12.729	0.906

Table B.4: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ without using standardization in Scenarios 1.a, 1.b and 1.c. Oracle values are in brackets.

		Scenario 1.a					Scenario 1.b					Scenario 1.c				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.736)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (13.715)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (13.715)	**% Dev** (0.9)
LASSO.min	25	5.52	15.45	20.96	0.967	0.930	5.30	16.23	21.53	2.731	0.975	5.39	16.29	21.68	2.673	0.973
	50	9.85	27.19	37.05	0.173	0.990	7.93	22.41	30.34	2.476	0.982	8.02	22.80	30.82	2.394	0.982
	100	10	24.48	34.48	0.703	0.959	9.38	22.01	31.40	5.977	0.956	9.40	21.81	31.22	6.026	0.955
	150	10	23.40	33.40	11.587	0.913	9.79	22.07	31.86	7.942	0.941	9.80	22.20	32	7.959	0.940
	300	10	21.27	31.27	12.729	0.906	9.99	21.17	31.17	10.638	0.922	9.99	20.54	30.53	10.677	0.921
LASSO.1se	25	3.86	8.25	12.11	4.003	0.733	4.28	9.44	13.72	11.469	0.897	4.27	9.42	13.69	13.205	0.880
	50	9.38	14.65	24.04	0.690	0.957	6.91	9.60	16.51	7.092	0.947	7.05	10.48	17.53	6.806	0.948
	100	10	10.91	20.91	1.036	0.939	8.69	8.16	16.84	9.359	0.931	8.72	8.04	16.76	9.406	0.930
	150	10	8.35	18.35	11.587	0.913	9.34	7.23	16.57	10.531	0.922	9.32	7.36	16.68	10.542	0.887
	300	10	5	15	12.729	0.906	9.92	4.99	14.91	12.174	0.910	9.91	4.87	14.78	12.180	0.891
LASSO.BIC	25	6.06	17.96	24.02	0	1	5.57	18.28	23.84	0.001	1	5.65	18.16	23.81	0.001	1
	50	9.89	37.47	47.36	0.001	1	8.45	37.40	45.85	0.024	1	8.52	37.69	46.21	0.022	1
	100	10	87.63	97.63	0.015	0.999	9.95	88.28	98.23	0.053	1	9.94	87.75	97.70	0.100	0.999
	150	10	2.05	12.05	11.587	0.913	9.09	3.22	12.31	11.572	0.914	9.08	3.31	12.39	11.559	0.913
	300	10	1.37	11.37	12.729	0.906	9.84	1.63	11.47	12.739	0.906	9.86	1.67	11.52	12.722	0.906

Table B.5: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ using univariate standardization in Scenarios 1.a, 1.b and 1.c. Oracle values are in brackets.

B.3.2 Scenarios 2.a and 2.b

		$\rho = 0.5$									
		Scenario 2.a					**Scenario 2.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.348)	**% Dev** (0.9)
LASSO.min	25	9.02	13.14	22.16	0.110	0.988	5.81	16.11	21.91	0.066	0.973
	50	13.31	18.56	31.87	0.193	0.982	9.65	25.19	34.84	0.049	0.985
	100	14.93	16.77	31.70	0.540	0.952	10	22.53	32.53	0.161	0.953
	150	15	15.10	30.10	0.722	0.936	10	21.90	31.90	0.215	0.937
	300	15	14.70	29.70	0.921	0.919	10	20.27	30.27	0.280	0.919
LASSO.1se	25	7.77	7.86	15.63	0.601	0.933	4.54	9.81	14.35	0.389	0.857
	50	12.51	8.06	20.57	0.485	0.955	9.21	14.34	23.54	0.144	0.954
	100	14.81	5.42	20.23	0.765	0.931	10	10.89	20.89	0.222	0.935
	150	15	3.80	18.80	0.895	0.920	10	8.93	18.93	0.267	0.922
	300	15	2	17	1.035	0.908	10	6.32	16.32	0.312	0.910
LASSO.BIC	25	9.31	14.45	23.76	0	1	6.12	17.87	23.98	0	1
	50	13.40	31.66	45.06	0.004	1	9.69	36.66	46.35	0.001	1
	100	14.92	82.77	97.69	0.011	0.999	9.99	87.71	97.70	0.003	0.999
	150	14.90	2.10	17	0.929	0.917	10	2.12	12.12	0.306	0.910
	300	15	1.30	16.30	1.042	0.908	10	0.71	10.71	0.331	0.905
		$\rho = 0.9$									
		Scenario 2.a					**Scenario 2.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.244)	**% Dev** (0.9)
LASSO.min	25	7.02	8.24	15.25	0.686	0.981	4.23	13.44	17.67	0.144	0.987
	50	8.89	7.07	15.96	1.914	0.948	6.10	13.74	19.84	0.514	0.955
	100	11.06	6.19	17.25	2.804	0.927	8.11	14.35	22.46	0.822	0.932
	150	12	6.40	18.50	3.138	0.919	9.11	14.60	23.71	0.951	0.922
	300	13.60	6	19.60	3.525	0.910	9.87	14.23	24.09	1.096	0.911
LASSO.1se	25	6.62	3.40	10.02	1.468	0.959	3.88	8.78	12.66	0.344	0.969
	50	8.34	1.81	10.15	2.643	0.928	5.62	8.35	13.97	0.728	0.937
	100	10.37	0.95	11.32	3.271	0.915	7.69	8.66	16.35	0.958	0.921
	150	11.50	0.80	12.30	3.499	0.910	8.87	8.23	17.10	1.059	0.913
	300	13.20	0.40	13.70	3.715	0.905	9.81	8.10	17.91	1.154	0.906
LASSO.BIC	25	7.21	14.26	21.48	0.022	1	4.38	17.80	22.18	0.005	1
	50	8.14	17.61	25.75	0.956	0.976	5.82	23.71	29.53	0.224	0.982
	100	14.45	79.21	93.66	0.187	0.995	9.73	85.86	95.59	0.038	0.997
	150	8.40	0.20	8.60	3.715	0.904	6.87	5.64	12.51	1.146	0.906
	300	10.40	0.10	10.50	3.822	0.902	8.79	5.79	14.57	1.183	0.904

Table B.6: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ without using standardization in Scenarios 2.a and 2.b. Oracle values are in brackets.

		$\rho = 0.5$									
		Scenario 2.a					**Scenario 2.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.139)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.348)	**% Dev** (0.9)
LASSO.min	25	9.49	12.96	22.45	0.068	0.992	5.92	16.01	21.93	0.091	0.965
	50	13.50	18.59	32.08	0.185	0.983	9.70	25.12	34.82	0.049	0.985
	100	14.94	16.27	31.21	0.546	0.951	10	22.18	32.18	0.162	0.952
	150	15	15	30	0.722	0.936	10	21.76	31.76	0.215	0.937
	300	15	14.50	29.50	0.922	0.918	10	20.15	30.15	0.280	0.919
LASSO.1se	25	8.21	7.80	16.01	0.486	0.945	4.57	9.66	14.23	0.402	0.855
	50	12.87	8.10	20.97	0.463	0.957	9.38	14.03	23.41	0.138	0.956
	100	14.86	5.12	19.98	0.768	0.931	10	10.46	20.46	0.225	0.934
	150	15	3.60	18.60	0.898	0.920	10	8.73	18.73	0.267	0.922
	300	15	2	17	1.035	0.908	10	6.26	16.26	0.312	0.910
LASSO.BIC	25	9.66	14.12	23.78	0	1	6.27	17.70	23.97	0	1
	50	13.53	32.16	45.69	0.003	1	9.72	36.98	46.70	0.001	1
	100	14.91	83.12	98.03	0.007	0.999	9.99	88.15	98.14	0.002	1
	150	14.90	1.90	16.90	0.933	0.917	10	2.03	12.03	0.305	0.910
	300	15	1.40	16.40	1.040	0.908	10	0.66	10.66	0.331	0.905
		$\rho = 0.9$									
		Scenario 2.a					**Scenario 2.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (3.807)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (1.244)	**% Dev** (0.9)
LASSO.min	25	7.29	8.18	15.47	0.675	0.981	4.43	13.82	18.25	0.129	0.988
	50	9	7.11	16.11	1.898	0.948	6.25	13.99	20.24	0.513	0.955
	100	11.10	6.02	17.13	2.813	0.927	8.14	14.27	22.41	0.824	0.932
	150	12.10	6.30	18.30	3.150	0.919	9.16	14.53	23.69	0.952	0.922
	300	13.60	5.90	19.50	3.526	0.910	9.87	14.12	23.99	1.098	0.911
LASSO.1se	25	6.91	3.54	10.45	1.353	0.962	4.22	9.13	13.35	0.299	0.973
	50	8.67	1.74	10.40	2.613	0.929	5.97	8.46	14.43	0.713	0.938
	100	10.66	0.86	11.52	3.268	0.915	7.85	8.60	16.46	0.955	0.921
	150	11.70	0.70	12.40	3.497	0.910	9	8.25	17.26	1.059	0.913
	300	13.40	0.40	13.70	3.715	0.905	9.83	8.06	17.89	1.154	0.906
LASSO.BIC	25	7.33	14.80	22.13	0.011	1	4.54	18.11	22.65	0.002	1
	50	8.29	18.93	27.22	0.847	0.979	5.84	25.32	31.15	0.188	0.985
	100	14.50	79.46	93.96	0.180	0.996	9.69	85.49	95.18	0.043	0.997
	150	8.90	0.30	9.20	3.669	0.906	7.08	5.47	12.55	1.143	0.906
	300	11	0.20	11.10	3.800	0.903	9.03	5.69	14.72	1.180	0.904

Table B.7: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ using univariate standardization in Scenarios 2.a and 2.b. Oracle values are in brackets.

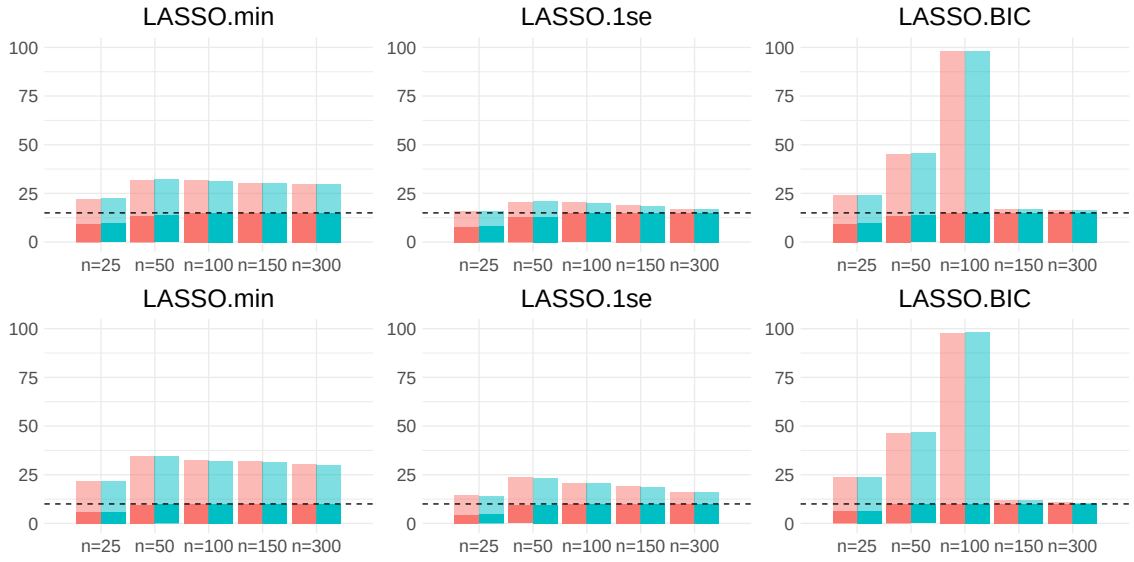


Figure B.6: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for $p = 100$ and $\rho = 0.5$ selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row). The dashed line marks the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

Figure B.7: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row) for $\rho = 0.5$ and $n = 50$. The dashed lines mark the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

METHOD	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (1.139)	% Dev (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (1.139)	% Dev (0.9)
	Scenario 2.a									
	WITHOUT					**UNIVARIATE**				
LASSO.min	13.31	18.56	31.87	0.193	0.982	13.50	18.59	32.08	0.185	0.983
LASSO.1se	12.51	8.06	20.57	0.485	0.955	12.87	8.10	20.97	0.463	0.957
LASSO.BIC	13.40	31.66	45.06	0.004	1	13.53	32.16	45.69	0.003	1
AdapL.min	9.58	2.55	12.12	0.668	0.939	9.57	2.53	12.10	0.662	0.940
AdapL.1se	5.11	0.27	5.38	2.009	0.816	5.23	0.29	5.52	1.937	0.823
SCAD	8.61	6.45	15.06	0.668	0.938	8.61	6.45	15.06	0.668	0.938
Dant	6.65	0.23	6.88	2.030	0.811	6.65	0.23	6.88	2.030	0.811
RelaxL	12.16	6.84	19	0.605	0.945	12.09	6.73	18.81	0.621	0.943
SqrtL	11.68	2.38	14.07	0.792	0.924	11.68	2.38	14.07	0.792	0.924
ScalL	11.87	3.50	15.37	0.693	0.934	12.32	3.48	15.79	0.656	0.938
DC.VS	4.19	0.81	5.00	2.475	0.771	4.19	0.81	5.00	2.475	0.771
	Scenario 2.b									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.348)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (0.348)	**% Dev** (0.9)
LASSO.min	9.65	25.19	34.84	0.049	0.985	9.70	25.12	34.82	0.049	0.985
LASSO.1se	9.21	14.34	23.54	0.144	0.954	9.38	14.03	23.41	0.138	0.956
LASSO.BIC	9.69	36.66	46.35	0.001	1	9.72	36.98	46.70	0.001	1
AdapL.min	8.95	4.15	13.10	0.189	0.943	9.08	4.14	13.22	0.183	0.945
AdapL.1se	7.05	0.95	8.01	0.488	0.849	7.33	0.86	8.19	0.458	0.857
SCAD	8.44	8.45	16.89	0.183	0.941	8.44	8.45	16.89	0.183	0.941
Dant	4.92	2.87	7.79	0.789	0.760	4.92	2.87	7.79	0.789	0.760
RelaxL	9.14	13.49	22.63	0.166	0.948	9.15	13.58	22.74	0.162	0.950
SqrtL	7.29	4.95	12.24	0.418	0.865	7.29	4.94	12.24	0.419	0.865
ScalL	7.70	6.21	13.92	0.333	0.893	8.05	6.29	14.34	0.309	0.9
DC.VS	3.16	1.84	5.00	1.084	0.674	3.16	1.84	5.00	1.084	0.674

Table B.8: Comparison of all proposed algorithms for $p = 100$, $n = 50$ and $\rho = 0.5$ using different standardization techniques in Scenario 2. Oracle values are in brackets.

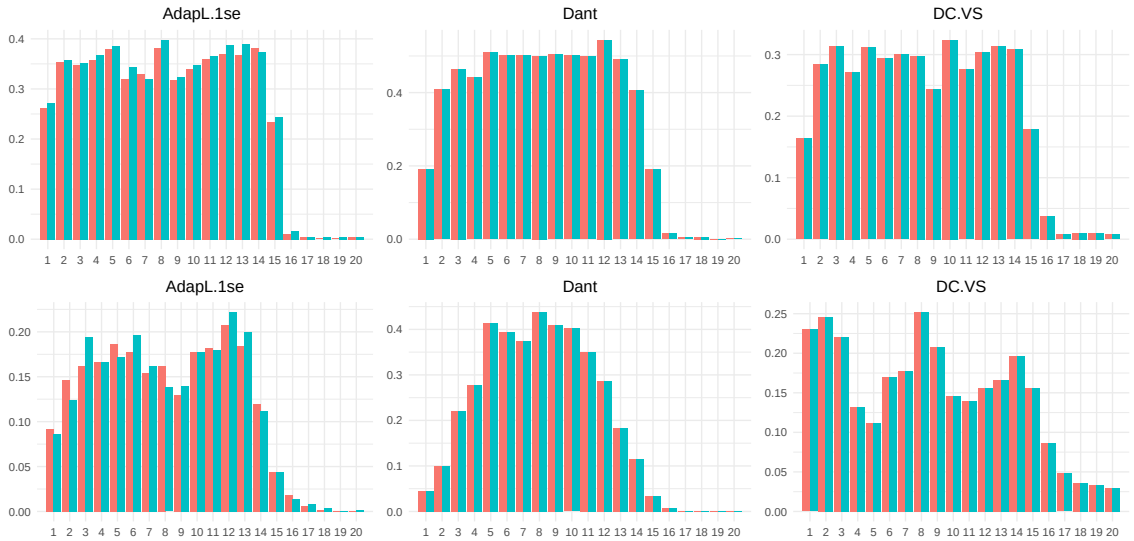


Figure B.8: Percentage of times each of the first 20 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 50$ in Scenario 2.a taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

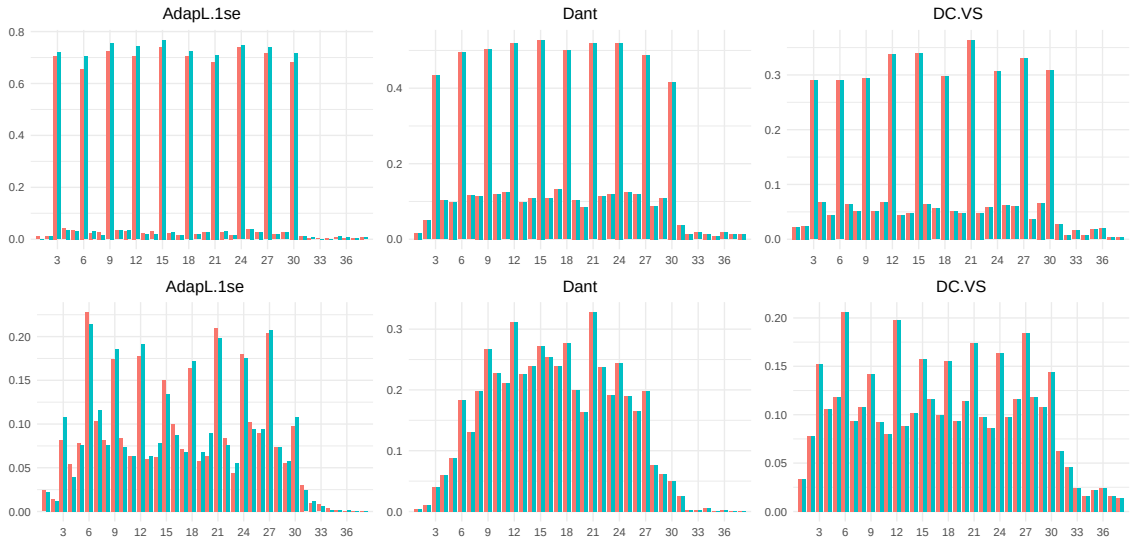


Figure B.9: Percentage of times each of the first 38 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 50$ in Scenario 2.b taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

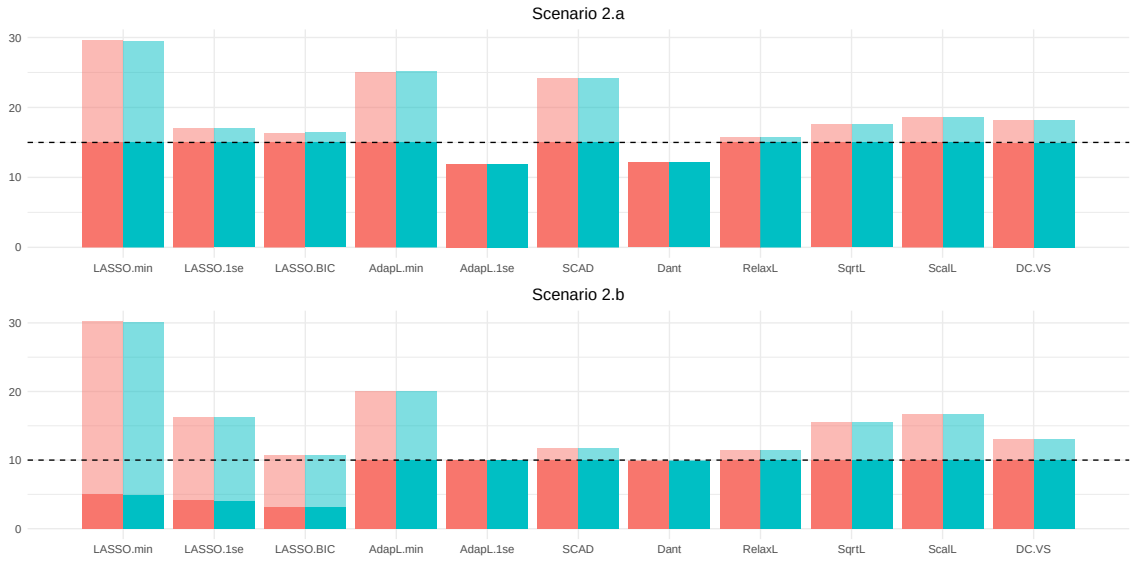


Figure B.10: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 2.a (the first row) and 2.b (the second row) for $\rho = 0.5$ and $n = 300$. The dashed lines mark the $s = 15$ and $s = 10$ value for the first and the second row, respectively.

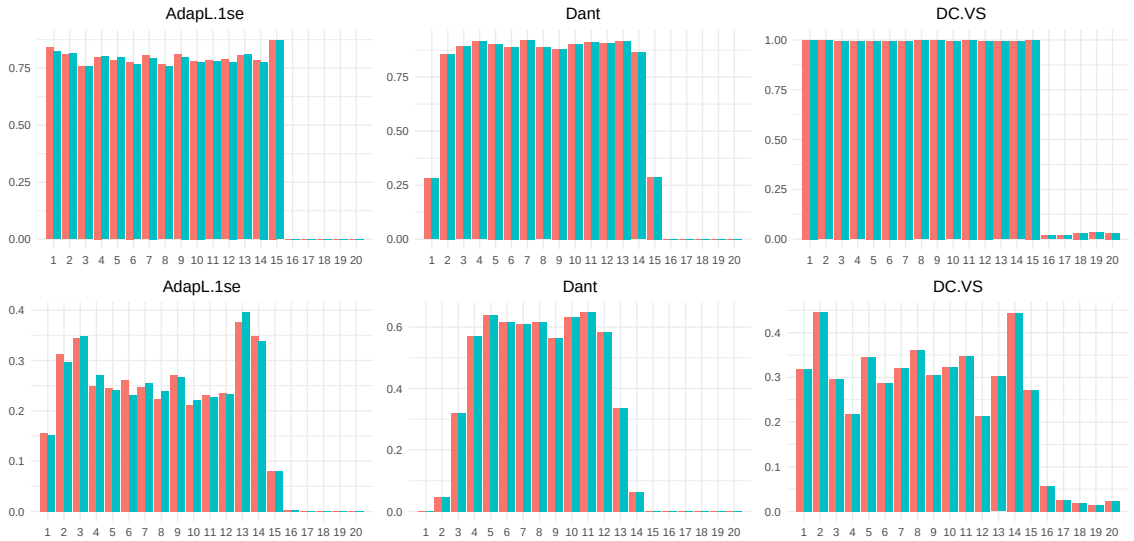


Figure B.11: Percentage of times each of the first 20 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 2.a taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

METHOD	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (1.139)	% Dev (0.9)	$\lvert\hat{S} \cap S\rvert$ (15)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (1.139)	% Dev (0.9)
	Scenario 2.a									
	WITHOUT					**UNIVARIATE**				
LASSO.min	15	14.70	29.70	0.921	0.919	15	14.50	29.50	0.922	0.918
LASSO.1se	15	2	17	1.035	0.908	15	2	17	1.035	0.908
LASSO.BIC	15	1.30	16.30	1.042	0.908	15	1.40	16.40	1.040	0.908
AdapL.min	15	10.10	25.10	0.942	0.917	15	10.18	25.18	0.941	0.917
AdapL.1se	11.97	0	11.97	1.356	0.880	11.91	0	11.91	1.361	0.879
SCAD	15	9.22	24.22	0.951	0.916	15	9.22	24.22	0.951	0.916
Dant	12.14	0	12.14	1.519	0.866	12.14	0	12.14	1.519	0.866
RelaxL	15	0.77	15.77	1.064	0.906	15	0.76	15.76	1.064	0.906
SqrtL	15	2.55	17.55	1.025	0.910	15	2.55	17.55	1.025	0.910
ScalL	15	3.60	18.60	1.011	0.911	15	3.63	18.63	1.010	0.911
DC.VS	14.93	3.27	18.20	1.022	0.910	14.93	3.27	18.20	1.022	0.910
	Scenario 2.b									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (0.348)	% Dev (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	MSE (0.348)	% Dev (0.9)
LASSO.min	4.97	25.30	30.27	0.280	0.919	4.95	25.20	30.15	0.280	0.919
LASSO.1se	4.08	12.24	16.32	0.312	0.910	4.06	12.20	16.26	0.312	0.910
LASSO.BIC	3.13	7.58	10.71	0.331	0.905	3.12	7.55	10.66	0.331	0.905
AdapL.min	10	10.05	20.05	0.294	0.915	10	10.08	20.08	0.294	0.915
AdapL.1se	10	0	10	0.335	0.903	10	0	10	0.335	0.903
SCAD	10	1.73	11.73	0.325	0.906	10	1.73	11.73	0.325	0.906
Dant	9.78	0.28	10.06	0.375	0.892	9.78	0.28	10.06	0.375	0.892
RelaxL	10	1.45	11.45	0.329	0.905	10	1.39	11.39	0.330	0.905
SqrtL	10	5.54	15.54	0.315	0.909	10	5.54	15.54	0.315	0.909
ScalL	10	6.64	16.64	0.311	0.910	10	6.66	16.66	0.311	0.910
DC.VS	10	3.04	13.04	0.318	0.908	10	3.04	13.04	0.318	0.908

Table B.9: Comparison of all proposed algorithms for $p = 100$, $n = 300$ and $\rho = 0.5$ using different standardization techniques in Scenario 2. Oracle values are in brackets.

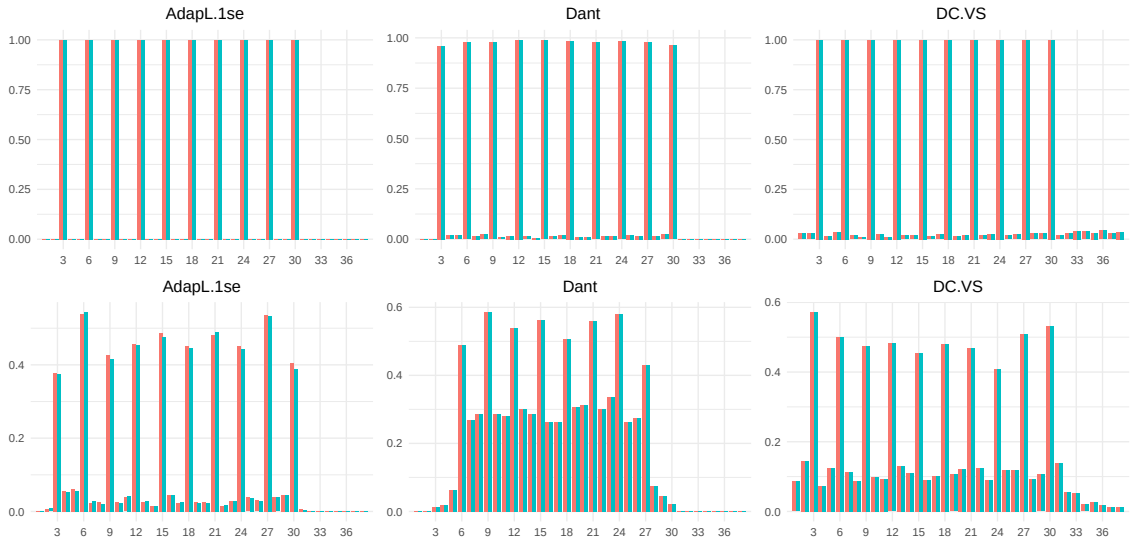


Figure B.12: Percentage of times each of the first 38 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 2.b taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

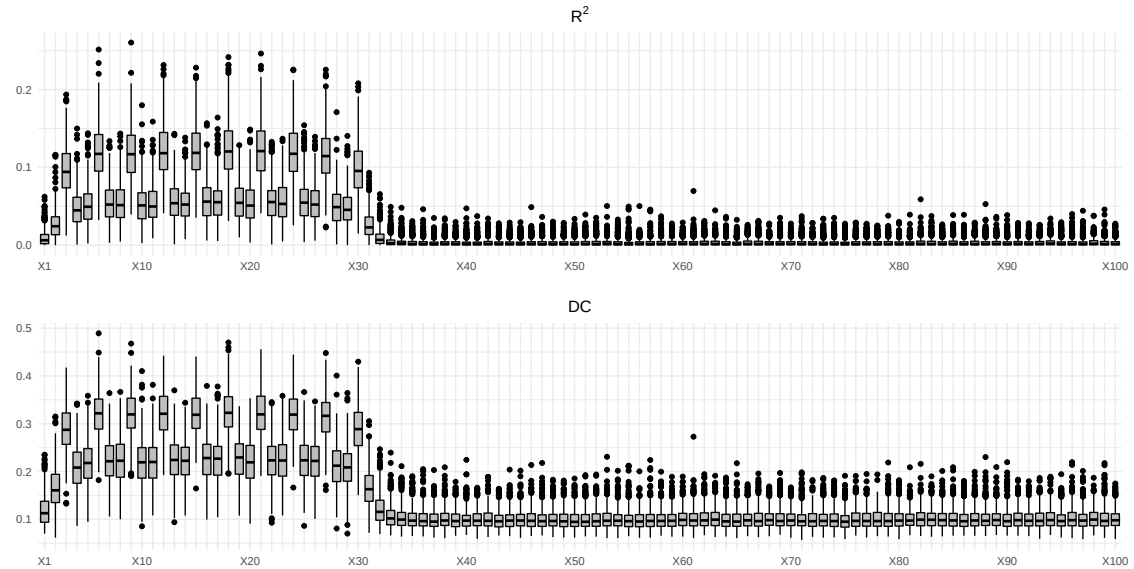


Figure B.13: Boxplots of the covariates loadings in terms of R^2 and DC for $n = 300$ in Scenario 2.b taking $\rho = 0.5$.

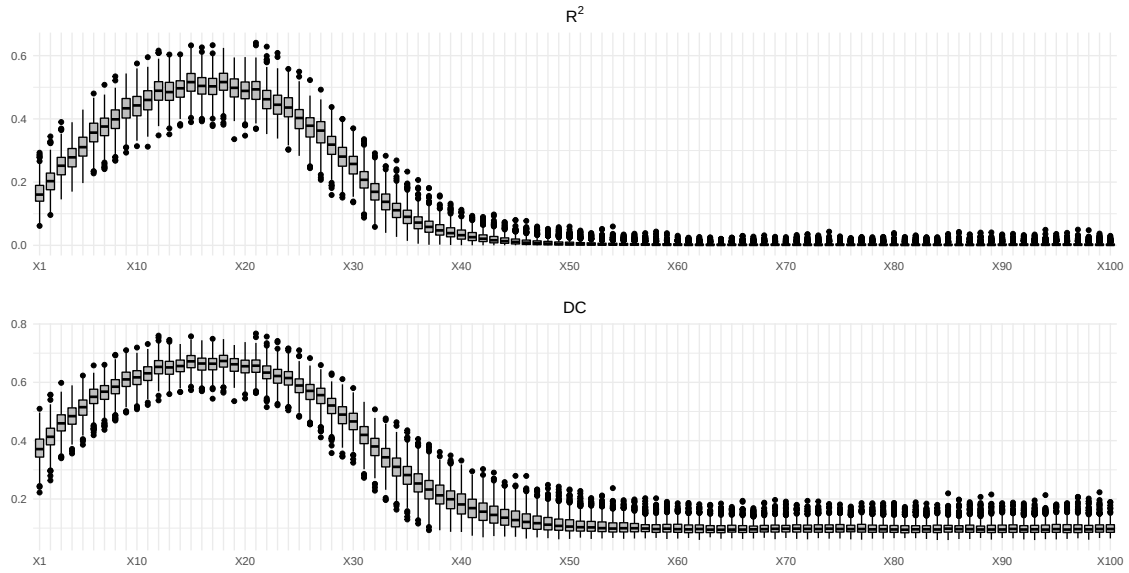


Figure B.14: Boxplots of the covariates loadings in terms of R^2 and DC for $n = 300$ in Scenario 2.b taking $\rho = 0.9$.

B.3.3 Scenarios 3.a y 3.b

		$\rho = 0.5$									
		Scenario 3.a					**Scenario 3.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	25	6.11	7.39	13.50	0.687	0.971	5.48	9.15	14.63	0.706	0.970
	50	7.13	8.23	15.35	1.355	0.946	6.79	11.36	18.15	1.243	0.951
	100	8.32	11.14	19.46	1.685	0.935	8.23	14.07	22.30	1.670	0.936
	150	9.00	13.86	22.86	1.870	0.929	9.08	18.16	27.24	1.799	0.932
	300	9.77	15.29	25.07	2.221	0.916	9.81	18.80	28.61	2.193	0.917
LASSO.1se	25	5.33	2.41	7.73	1.843	0.922	4.76	4.75	9.51	1.785	0.923
	50	6.10	1.79	7.89	2.483	0.901	5.87	4.82	10.68	2.267	0.911
	100	7.20	1.96	9.15	2.498	0.903	6.98	5.49	12.47	2.447	0.906
	150	7.87	2.38	10.25	2.497	0.905	7.77	6.08	13.85	2.456	0.907
	300	8.90	2.87	11.77	2.551	0.904	8.99	6.48	15.47	2.513	0.905
LASSO.BIC	25	6.43	11.82	18.25	0.070	0.997	5.87	13.26	19.13	0.062	0.998
	50	7.50	14.14	21.64	0.696	0.973	7.24	17.07	24.30	0.600	0.978
	100	9.52	73.98	83.50	0.328	0.988	9.57	76.97	86.55	0.269	0.990
	150	8.23	2.93	11.16	2.345	0.911	8.11	6.02	14.14	2.347	0.911
	300	9.08	2.28	11.36	2.539	0.904	9.02	5.16	14.18	2.529	0.905
		$\rho = 0.9$									
		Scenario 3.a					**Scenario 3.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (7.818)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (7.818)	**% Dev** (0.9)
LASSO.min	25	5.26	3.93	9.18	3.324	0.952	3.75	7.12	10.88	2.792	0.961
	50	6	4.42	10.41	5.060	0.932	4.88	7.77	12.66	4.828	0.935
	100	6.67	5.06	11.73	6.206	0.918	5.86	8.80	14.66	6.070	0.921
	150	6.96	7.59	14.55	6.450	0.915	6.41	11.82	18.23	6.278	0.918
	300	7.59	8.09	15.68	7.100	0.908	7.22	12.41	19.63	7.022	0.909
LASSO.1se	25	4.64	0.95	5.58	5.684	0.917	3.08	4.11	7.19	5.148	0.926
	50	5.40	0.81	6.21	6.927	0.907	4.07	4.62	8.69	6.500	0.913
	100	6.02	1.09	7.11	7.284	0.904	4.97	5.35	10.32	7.106	0.907
	150	6.34	1.57	7.91	7.351	0.904	5.39	6.12	11.51	7.230	0.906
	300	6.84	2.10	8.94	7.561	0.902	5.87	6.75	12.62	7.563	0.902
LASSO.BIC	25	5.40	6.65	12.05	1.544	0.980	3.90	9.05	12.95	1.485	0.981
	50	5.66	2.96	8.62	5.258	0.929	4.25	6.05	10.30	5.265	0.930
	100	7.04	25.88	32.92	4.830	0.937	6.43	31.75	38.18	4.452	0.942
	150	6.13	1.33	7.46	7.273	0.905	5.25	5.78	11.03	7.151	0.907
	300	6.42	1.25	7.67	7.643	0.901	5.74	6.30	12.04	7.546	0.903

Table B.10: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ without using standardization in Scenarios 3.a and 3.b. Oracle values are in brackets.

		$\rho = 0.5$									
		Scenario 3.a					**Scenario 3.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	25	5.45	16.27	21.72	0.210	0.989	5.47	16.34	21.81	0.280	0.984
	50	7.73	19.99	27.73	0.626	0.975	7.77	20.94	28.70	0.562	0.978
	100	9.10	20.79	29.89	1.292	0.950	9.09	19.98	29.07	1.346	0.948
	150	9.57	20.79	30.37	1.676	0.936	9.65	21.40	31.05	1.659	0.937
	300	9.96	20.20	30.17	2.159	0.919	9.96	19.56	29.52	2.165	0.919
LASSO.1se	25	4.62	10.11	14.74	1.357	0.933	4.69	10.49	15.18	1.251	0.937
	50	6.87	10.08	16.95	1.362	0.945	6.86	10.50	17.36	1.310	0.949
	100	8.44	8.77	17.21	1.858	0.928	8.43	8.46	16.90	1.896	0.927
	150	9.08	7.66	16.74	2.114	0.920	9.18	8.24	17.42	2.096	0.921
	300	9.82	6.37	16.19	2.404	0.909	9.84	5.94	15.78	2.409	0.909
LASSO.BIC	25	5.68	18.10	23.78	0	1	5.66	18.09	23.75	0	1
	50	8.22	36.73	44.95	0.009	1	8.25	36.65	44.90	0.010	1
	100	9.94	88.24	98.18	0.013	1	9.94	88.17	98.11	0.018	0.999
	150	8.45	3.14	11.59	2.371	0.910	8.48	3.34	11.81	2.378	0.910
	300	9.39	1.77	11.16	2.547	0.904	9.46	1.83	11.29	2.536	0.905
		$\rho = 0.9$									
		Scenario 3.a					**Scenario 3.b**				
	n	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (7.818)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (7.818)	**% Dev** (0.9)
LASSO.min	25	4.09	12.84	16.93	1.172	0.983	4.02	12.89	16.91	1.067	0.984
	50	5.30	12.43	17.73	3.577	0.952	5.36	12.62	17.98	3.570	0.952
	100	6.77	12.78	19.55	5.391	0.929	6.80	12.54	19.34	5.408	0.929
	150	7.34	13.53	20.87	6.065	0.920	7.43	13.25	20.68	6.112	0.920
	300	8.36	13.66	22.02	6.920	0.910	8.40	13.53	21.93	6.926	0.911
LASSO.1se	25	3.81	8.04	11.85	2.482	0.963	3.79	7.96	11.75	2.407	0.964
	50	4.92	7.21	12.13	4.871	0.934	4.96	7.27	12.23	4.856	0.935
	100	6.38	7.09	13.47	6.275	0.917	6.33	6.87	13.20	6.316	0.917
	150	6.93	7.05	13.98	6.769	0.911	7.03	7.04	14.07	6.770	0.912
	300	7.92	7.20	15.12	7.302	0.906	8.08	7.10	15.19	7.302	0.906
LASSO.BIC	25	4.25	18.14	22.38	0.019	1	4.19	18.09	22.28	0.022	1
	50	5.17	22.73	27.90	1.596	0.980	5.22	23.00	28.22	1.505	0.981
	100	9.71	85.83	95.55	0.241	0.997	9.68	85.06	94.74	0.298	0.996
	150	5.70	4.56	10.26	7.170	0.906	5.80	4.39	10.19	7.198	0.906
	300	6.76	4.77	11.53	7.527	0.903	6.81	4.59	11.40	7.526	0.903

Table B.11: Results of LASSO.min, LASSO.1se, LASSO.BIC for $p = 100$ using univariate standardization in Scenarios 3.a and 3.b. Oracle values are in brackets.

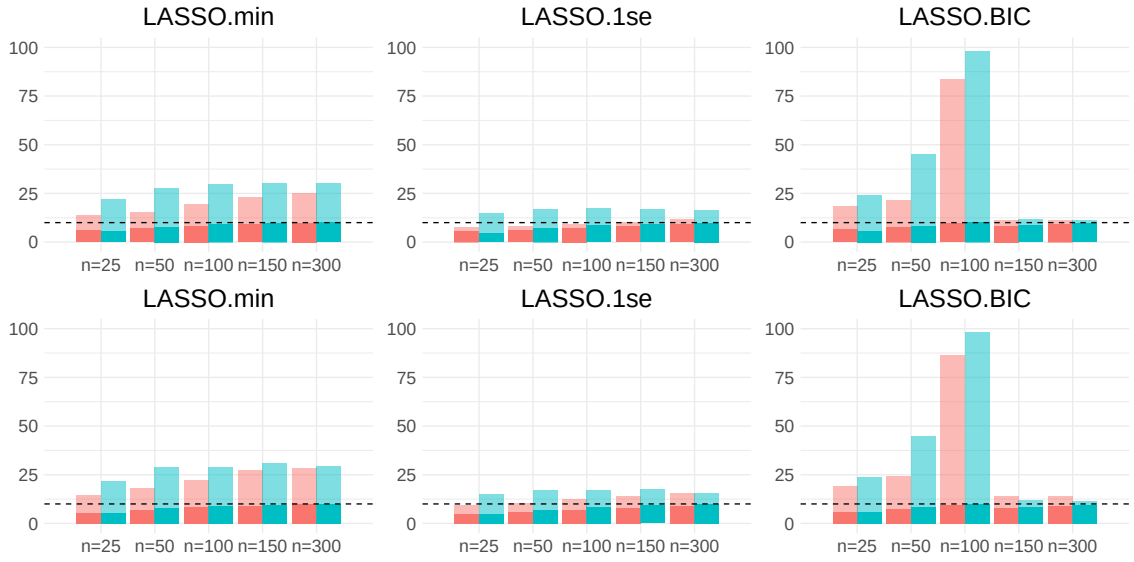


Figure B.15: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for $p = 100$ and $\rho = 0.5$ selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row). The dashed line marks the $s = 10$ value.

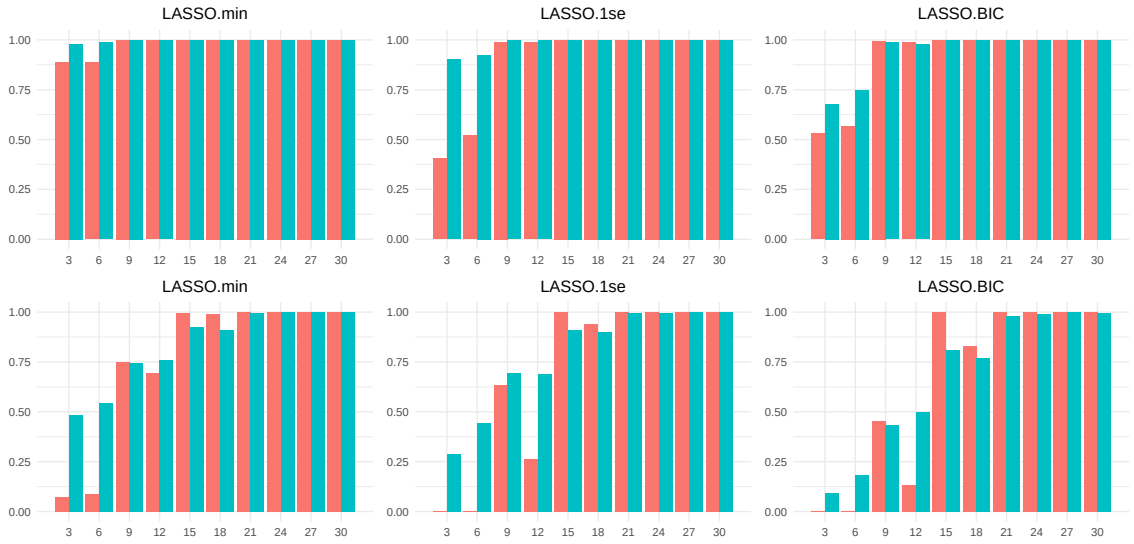


Figure B.16: Percentage of times each of the $s = 10$ relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 3.a taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

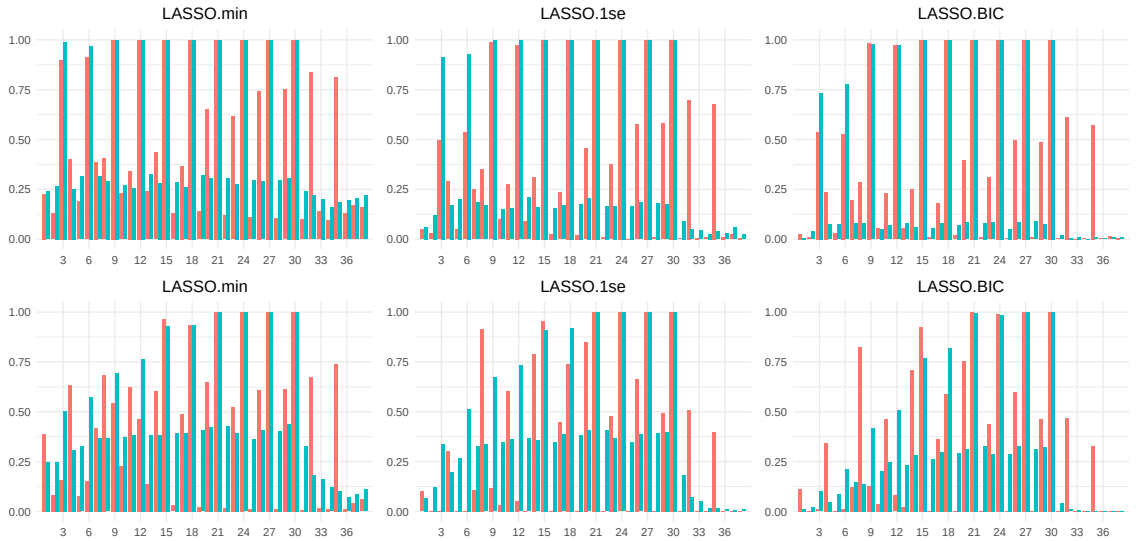


Figure B.17: Percentage of times each of the first 38 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ in Scenario 3.b taking $\rho = 0.5$ (the first row) and $\rho = 0.9$ (the second row).

Figure B.18: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row) for $\rho = 0.5$ and $n = 50$. The dashed line marks the $s = 10$ value.

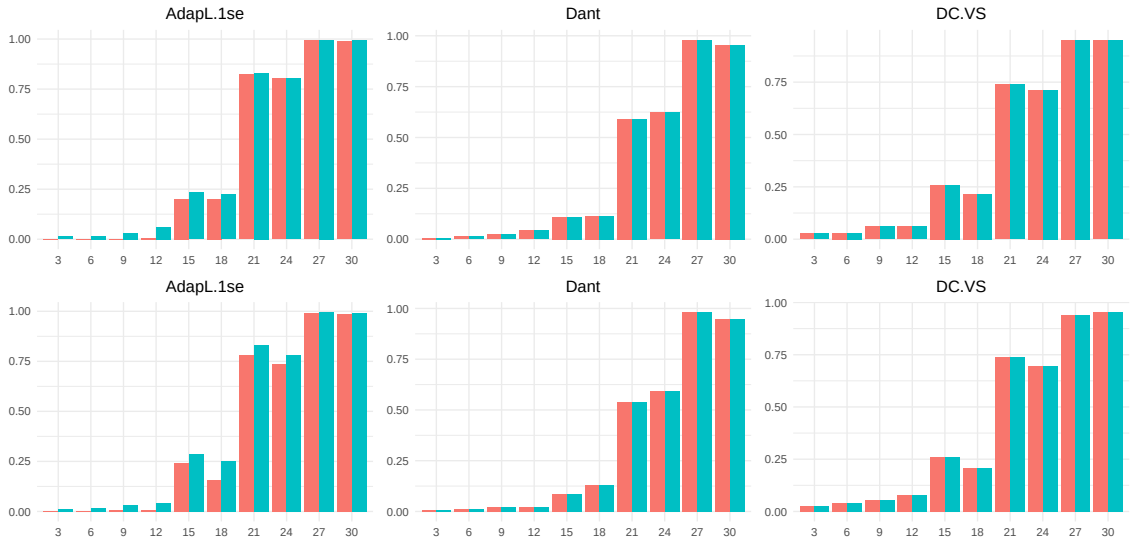


Figure B.19: Percentage of times each of the 10 relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 50$ taking $\rho = 0.5$ in Scenario 3.a (the first row) and Scenario 3.b (the second row).

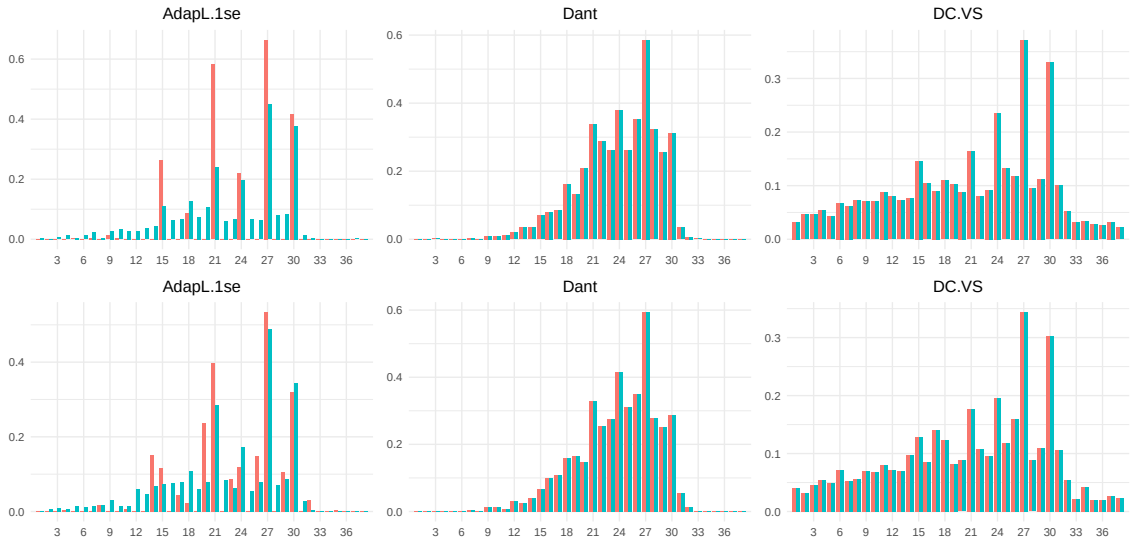


Figure B.20: Percentage of times each of the first 38 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 50$ taking $\rho = 0.9$ in Scenario 3.a (the first row) and Scenario 3.b (the second row).

	Scenario 3.a									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	7.13	8.23	15.35	1.355	0.946	7.73	19.99	27.73	0.626	0.975
LASSO.1se	6.10	1.79	7.89	2.483	0.901	6.87	10.08	16.95	1.362	0.945
LASSO.BIC	7.50	14.14	21.64	0.696	0.973	8.22	36.73	44.95	0.009	1
AdapL.min	5.22	0.12	5.34	3.339	0.869	5.67	2.07	7.74	2.194	0.914
AdapL.1se	4.02	0.01	4.02	4.631	0.817	4.20	0.21	4.41	4.187	0.835
SCAD	6.94	6.77	13.71	1.301	0.948	6.94	6.77	13.71	1.301	0.948
Dant	3.45	0.85	4.31	5.512	0.784	3.45	0.85	4.31	5.512	0.784
RelaxL	6.61	9.35	15.96	1.621	0.935	6.49	8.77	15.26	1.740	0.931
SqrtL	5.97	4.38	10.35	2.177	0.912	5.97	4.38	10.35	2.177	0.912
ScalL	6.91	4.21	11.12	1.653	0.934	6.28	5.44	11.72	1.881	0.924
DC.VS	4	1	5	4.198	0.834	4	1	5	4.198	0.834
	Scenario 3.b									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	6.79	11.36	18.15	1.243	0.951	7.77	20.94	28.70	0.562	0.978
LASSO.1se	5.87	4.82	10.68	2.267	0.911	6.86	10.50	17.36	1.310	0.949
LASSO.BIC	7.24	17.07	24.30	0.600	0.978	8.25	36.65	44.90	0.010	1
AdapL.min	5.05	0.56	5.62	3.186	0.877	5.62	1.99	7.62	2.210	0.915
AdapL.1se	3.91	0.06	3.97	4.708	0.816	4.24	0.22	4.46	4.073	0.840
SCAD	6.87	6.94	13.81	1.278	0.950	6.87	6.94	13.81	1.278	0.950
Dant	3.35	0.88	4.23	5.616	0.783	3.35	0.88	4.23	5.616	0.783
RelaxL	6.52	8.98	15.50	1.621	0.936	6.40	8.44	14.83	1.730	0.932
SqrtL	5.90	4.32	10.22	2.154	0.914	5.90	4.32	10.22	2.154	0.914
ScalL	6.68	8.32	14.99	1.405	0.945	6.27	5.58	11.85	1.845	0.926
DC.VS	4	1	5	4.137	0.837	4	1	5	4.137	0.837

Table B.12: Comparison of all proposed algorithms for $p = 100$, $n = 50$ and $\rho = 0.5$ using different standardization techniques in Scenario 3. Oracle values are in brackets.

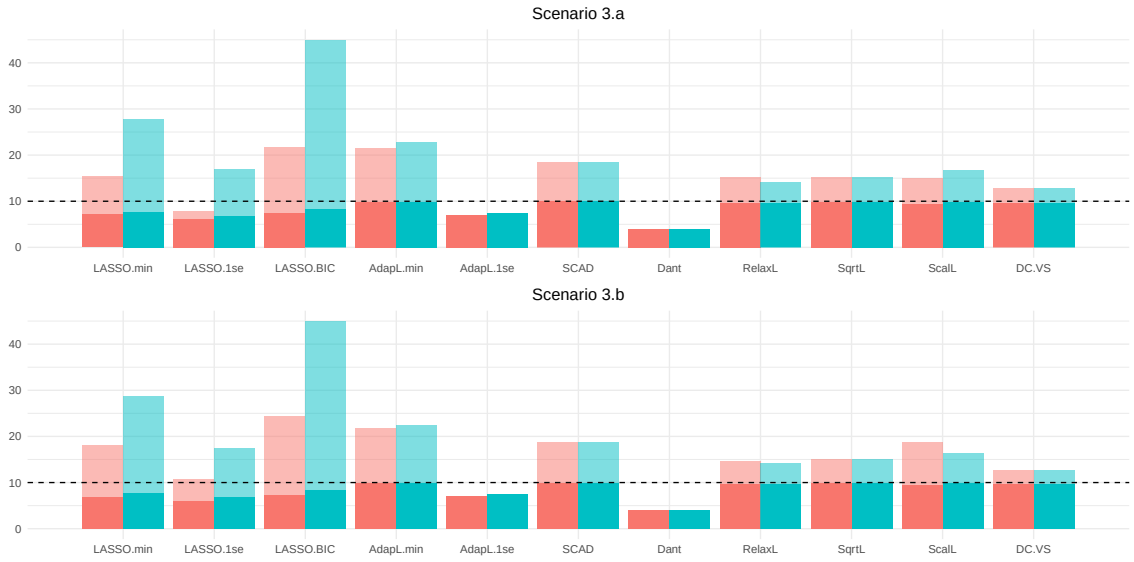


Figure B.21: Number of important covariates (dark pink/blue area) and noisy ones (soft pink/blue area) for proposed algorithms taking $p = 100$ and selected in terms of the without/univariate standardization in Scenarios 3.a (the first row) and 3.b (the second row) for $\rho = 0.5$ and $n = 300$. The dashed line marks the $s = 10$ value.

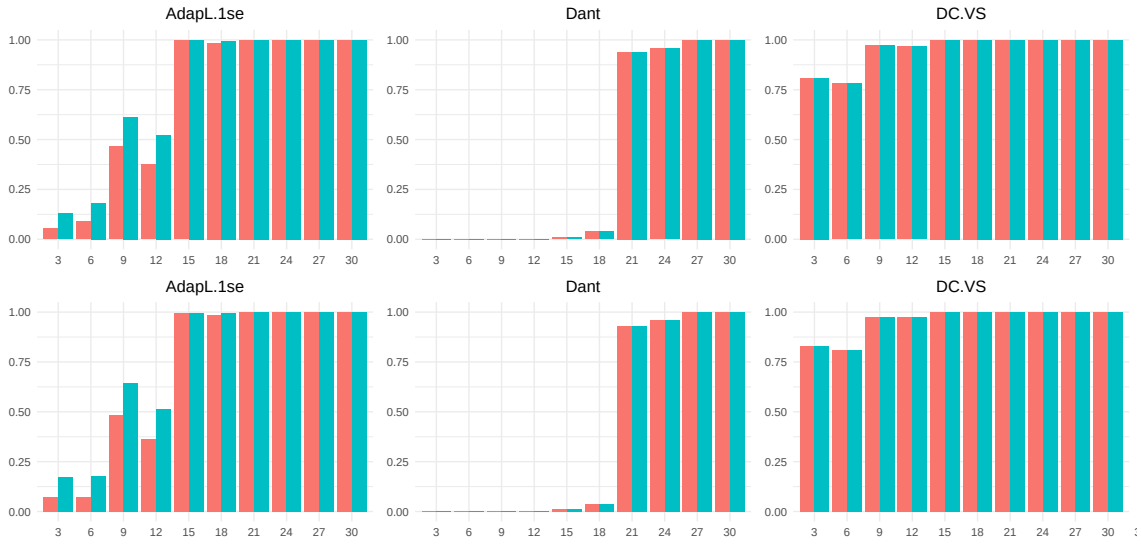


Figure B.22: Percentage of times each of the 10 relevant covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ taking $\rho = 0.5$ in Scenario 3.a (the first row) and Scenario 3.b (the second row).

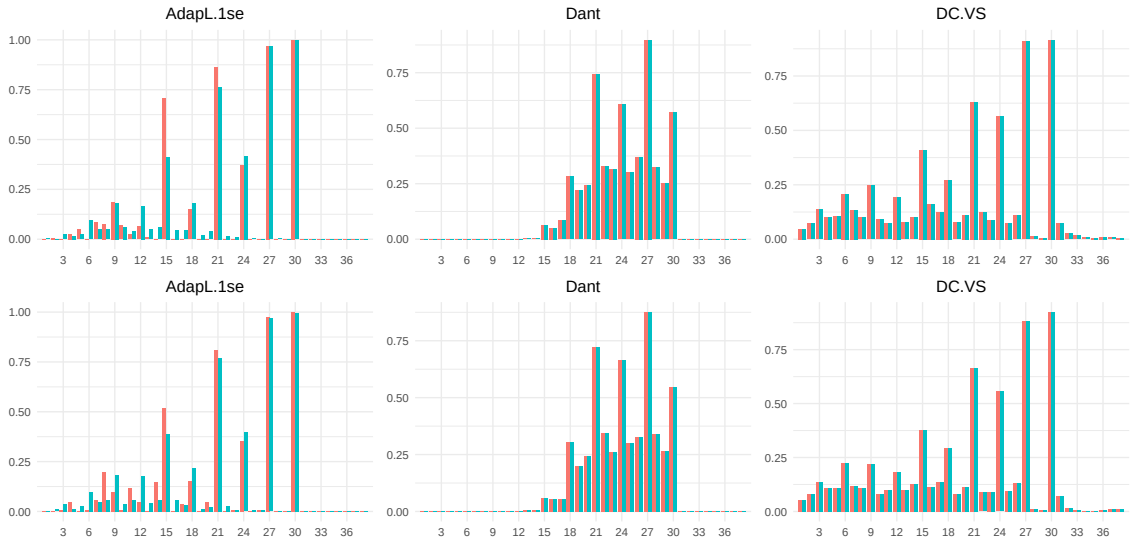


Figure B.23: Percentage of times each of the first 38 covariates enters the model for without/univariate standardization (pink/blue area) for $n = 300$ taking $\rho = 0.9$ in Scenario 3.a (the first row) and Scenario 3.b (the second row).

	Scenario 3.a									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	9.77	15.29	25.07	2.221	0.916	9.96	20.20	30.17	2.159	0.919
LASSO.1se	8.90	2.87	11.77	2.551	0.904	9.82	6.37	16.19	2.404	0.909
LASSO.BIC	9.08	2.28	11.36	2.539	0.904	9.39	1.77	11.16	2.547	0.904
AdapL.min	9.79	11.72	21.51	2.227	0.916	9.85	12.97	22.81	2.200	0.917
AdapL.1se	6.96	0.02	6.98	3.072	0.884	7.44	0.03	7.47	2.941	0.889
SCAD	9.93	8.54	18.47	2.301	0.913	9.93	8.54	18.47	2.301	0.913
Dant	3.94	0.01	3.95	5.444	0.795	3.94	0.01	3.95	5.444	0.795
RelaxL	9.63	5.51	15.14	2.449	0.908	9.53	4.67	14.19	2.480	0.906
SqrtL	9.83	5.40	15.23	2.427	0.908	9.83	5.40	15.23	2.427	0.908
ScalL	9.46	5.55	15.01	2.417	0.909	9.86	6.85	16.71	2.389	0.910
DC.VS	9.53	3.27	12.80	2.455	0.907	9.53	3.27	12.80	2.455	0.907
	Scenario 3.b									
	WITHOUT					**UNIVARIATE**				
METHOD	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)	$\lvert\hat{S} \cap S\rvert$ (10)	$\lvert\hat{S} \setminus S\rvert$	$\lvert\hat{S}\rvert$	**MSE** (2.679)	**% Dev** (0.9)
LASSO.min	9.81	18.80	28.61	2.193	0.917	9.96	19.56	29.52	2.165	0.919
LASSO.1se	8.99	6.48	15.47	2.513	0.905	9.84	5.94	15.78	2.409	0.909
LASSO.BIC	9.02	5.16	14.18	2.529	0.905	9.46	1.83	11.29	2.536	0.905
AdapL.min	9.80	12.00	21.80	2.222	0.916	9.84	12.66	22.50	2.204	0.917
AdapL.1se	6.98	0.01	6.98	3.067	0.885	7.50	0.01	7.51	2.928	0.890
SCAD	9.90	8.89	18.79	2.292	0.914	9.90	8.89	18.79	2.292	0.914
Dant	3.94	0.00	3.94	5.439	0.796	3.94	0.00	3.94	5.439	0.796
RelaxL	9.64	5.08	14.72	2.454	0.908	9.56	4.59	14.14	2.475	0.907
SqrtL	9.85	5.26	15.11	2.427	0.909	9.85	5.26	15.11	2.427	0.909
ScalL	9.54	9.32	18.86	2.384	0.910	9.87	6.43	16.30	2.396	0.910
DC.VS	9.59	3.12	12.71	2.458	0.907	9.59	3.12	12.71	2.458	0.907

Table B.13: Comparison of all proposed algorithms for $p = 100$, $n = 300$ and $\rho = 0.5$ using different standardization techniques in Scenario 3. Oracle values are in brackets.

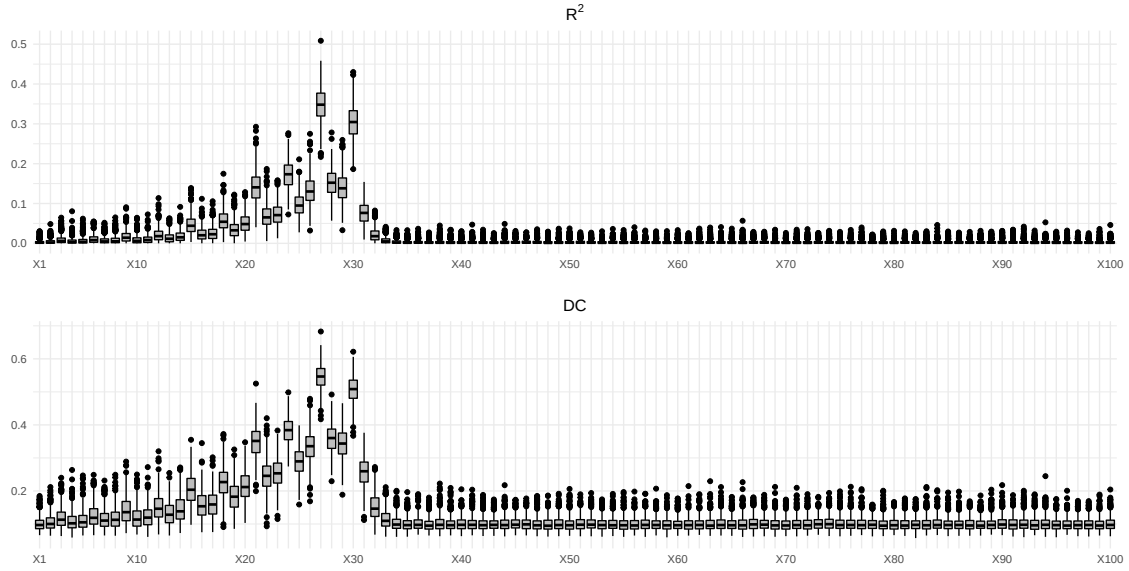


Figure B.24: Boxplots of the covariates loadings in terms of R^2 and DC for $n = 300$ in Scenario 3.b taking $\rho = 0.5$.

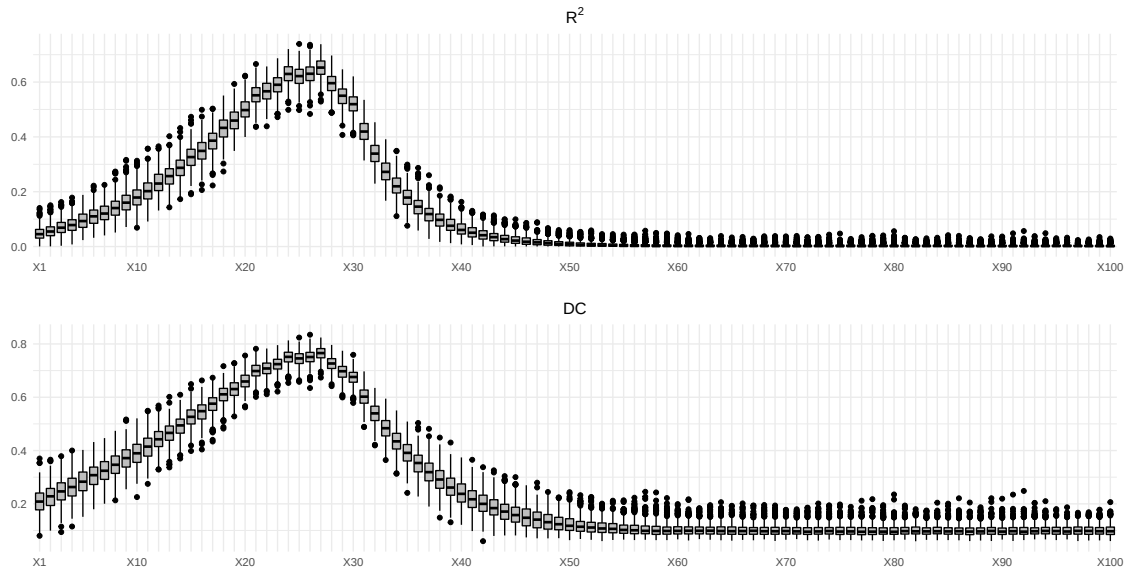


Figure B.25: Boxplots of the covariates loadings in terms of R^2 and DC for $n = 300$ in Scenario 3.b taking $\rho = 0.9$.

Appendix C

Extra results for MDD significant tests

C.1 Simulation details for considered competitors

In this section, we remind the scenario A structure of the FLCM algorithm implemented in Ghosal and Maity (2022a) as well as the form of scenario (B) for the ANFCM testing performance in Kim et al. (2018). Moreover, we explain how these algorithms are implemented to replicate their results.

C.1.1 Implementation details for FLCM algorithm

Under the linear assumption, data for $i = 1, \ldots, n$ is generated as

$$Y_i(t) = \beta_0(t) + X_i(t)\beta_1(t) + \varepsilon_i(t)$$

where $\beta_0(t) = 1 + 2t + t^2$ and $\beta_1(t) = d \cdot t/8$, for $d \geq 0$. The original covariate samples $X_i(\cdot)$ are i.i.d. copies of $X(\cdot)$, where $X(t) = a + b\sqrt{2}\sin(\pi t) + c\sqrt{2}\cos(\pi t)$, with $a \sim N(0, 1)$, $b \sim N(0, 0.85^2)$ and $c \sim N(0, 0.70^2)$ independent. It is assumed that the covariate $X_i(t)$ is observed with error, i.e. $W_i(t) = X_i(t) + \delta_{it}$ is getting instead, where $\delta_{it} \sim N(0, 0.6^2)$ and changes with every i and t. The error process is considered as

$$\varepsilon_i(t) = \xi_{i1}\sqrt{2}\cos(\pi t) + \xi_{i2}\sqrt{2}\sin(\pi t) + \xi_{i3t}$$

where $\xi_{i1} \overset{iid}{\sim} N(0, 2)$, $\xi_{i2} \overset{iid}{\sim} N(0, 0.75^2)$ and $\xi_{i3t} \overset{iid}{\sim} N_{\mathcal{T}}(0, 0.9^2 I_{\mathcal{T}})$, with ξ_{i3t} being generated as a multivariate normal of dimension $\mathcal{T}$ and these values change with i and t.

We consider the dense design, taking a total of $\mathcal{T} = 81$ equidistant time points in $[0, 1]$, being $t_1 = 0$ and $t_{81} = 1$. A Monte Carlo study is carried out using $M = 1000$ replicates to measure calibration and power, and p-values are calculated using $B = 100000$ samples generated under the null hypothesis of no effect ($H_0 : \beta_1(t) = 0$ for all t). Following the authors' guidelines, the number of basis components considered is $Q = 7$. To measure calibration and power we consider $d = 0$ and $d = 3, 7$, respectively. Besides, we take $n = 60, 100$ to compare their results with the MDD-based test ones. To implement this algorithm we have used the public code which can be found in `10.1016/j.ecosta.2021.05.003`. In particular, we generate the data and use the `FLCM.test1` function of the `test.R` script to implement the test.

C.1.2 Implementation details for ANFCM algorithm

In the case of the ANFCM approach, we perform Algorithm 1 of Kim et al. (2018) in hypothesis testing, which translates into testing the nullity of the second additive effect by

$$H_0 : \mathbb{E}\left[Y(t)|_{X_1(t)=x_1}\right] = F_0(t).$$

In this scenario, samples are generated verifying the additive assumption of

$$Y_i(t) = F_0(t) + F_1(X_{1i}(t), t) \quad \text{for} \quad i = 1, \ldots, n$$

where $F_0(t) = 2t + t^2$ and $F_1(X_{1i}(t), t) = d\{2\cos(X_1(t)t)\}$ for $d \geq 0$. The covariate $X_1(t)$ is given by $X_1(t) = a_0 + a_1\sqrt{2}\sin(\pi t) + a_2\sqrt{2}\cos(\pi t)$, where $a_0 \sim N\left(0, \{2^{-0.5}\}^2\right)$, $a_{j1} \sim N\left(0, \{0.85 \times 2^{-0.5}\}^2\right)$ and $a_2 \sim N\left(0, \{0.7 \times 2^{-0.5}\}^2\right)$. However, it is assumed that this covariate is observed with error. In particular, we get $W_{1i} = X_{1i}(t) + \delta_{it}$ where $\delta_{it} \sim N(0, 0.6^2)$ varies with respect to i and t. The considered error process is

$$\varepsilon_i(t) = \xi_{i1}\sqrt{2}\cos(\pi t) + \xi_{i2}\sqrt{2}\sin(\pi t) + \xi_{i3t}$$

where $\xi_{i1} \overset{iid}{\sim} N(0, 2)$, $\xi_{i2} \overset{iid}{\sim} N(0, 0.75^2)$ and $\xi_{i3t} \overset{iid}{\sim} N_{\mathcal{T}}(0, 0.9^2 I_{\mathcal{T}})$, being ξ_{i3t} generated as a multivariate normal of dimension $\mathcal{T}$. All these values are simulated changing with i and t.

Here, the dense design scenario is considered with $\mathcal{T} = 81$ equidistant time points in $[0, 1]$, being $t_1 = 0$ and $t_{81} = 1$. To study its calibration and power behavior a Monte Carlo study is carried out. We employ a total of $M = 1000$ replicates to study both. In this case, p-values are calculated by means of $B = 200$ bootstrap samples in all cases. Besides, following Kim et al. (2018) parameters selection, the number of the basis components taken is $Q = 7$. In order to measure calibration and power we test with $d = 0$ and $d = 3, 7$, we simulate under null and alternative hypotheses, respectively. Besides, we take $n = 60, 100$ to compare their results with the MDD-based test ones. We have found the code available in `https://www4.stat.ncsu.edu/~maity/software.html` and we borrowed it to reproduce the ANFCM simulations. Specifically, we make use of the `anova.datagen` function of the `datagenALL.R`[1] script to generate the data and apply `test.anova` function of the `test.R` script to implement the algorithm, using now `list(null.data.dn$Weval[[2]])`.

[1]We have adapted the code to correctly generate $X1(t)$ and $Y(t)$. In particular, in the `X` function, we have changed `X.list[[q]] =2^(1-q)*(a0%*%t(ones)+a1%*%t(phi1)+a2%*%t(phi2))` for the expression `X.list[[q]] = (a0%*%t(ones)+a1%*%t(phi1)+a2%*%t(phi2))/sqrt(2^(q-1))` to correct a typo. Besides, it is needed to change `F.anova2 = function(x1,x2,t,d)2*t+t^2+x1*sin(pi*t)/4+d*2*cos(x2*t)` by `F.anova2 = function(x2,t,d)2*t+t^2+d*2*cos(x2*t)` as well as `Fanova = F(Xeval[[1]],Xeval[[2]],trep,d)` by `Fanova = F(Xeval[[2]],trep,d)` in function `anova.datagen` to correctly define the modified version.

C.2 Graphics

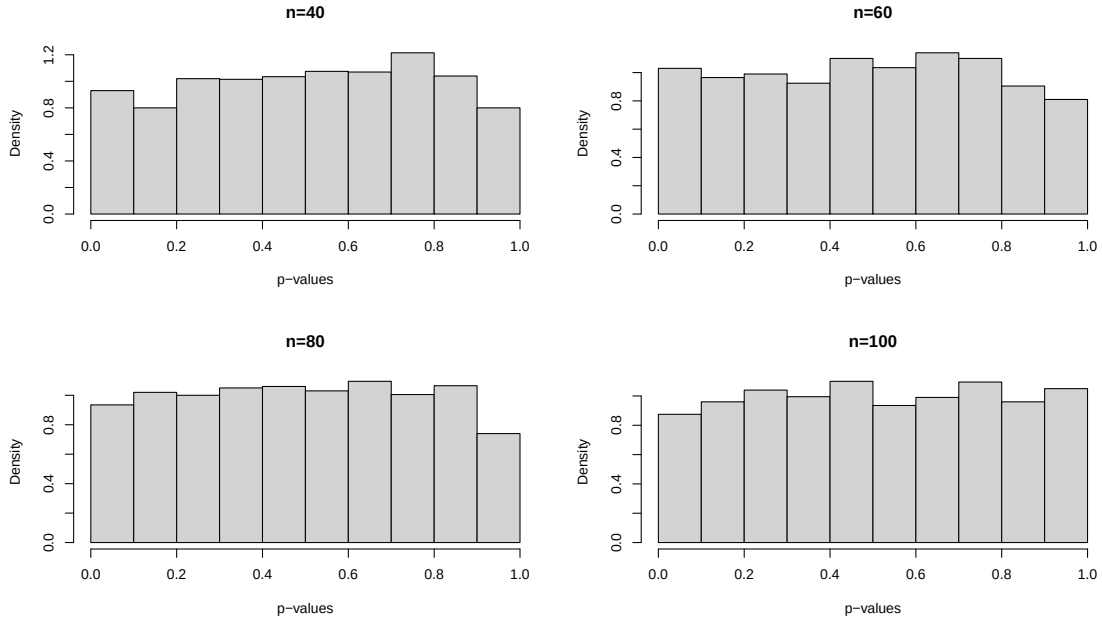


Figure C.1: Histograms of the p-values of the test statistics under H_0 using the wild bootstrap critical value for some values of n in Scenario A.

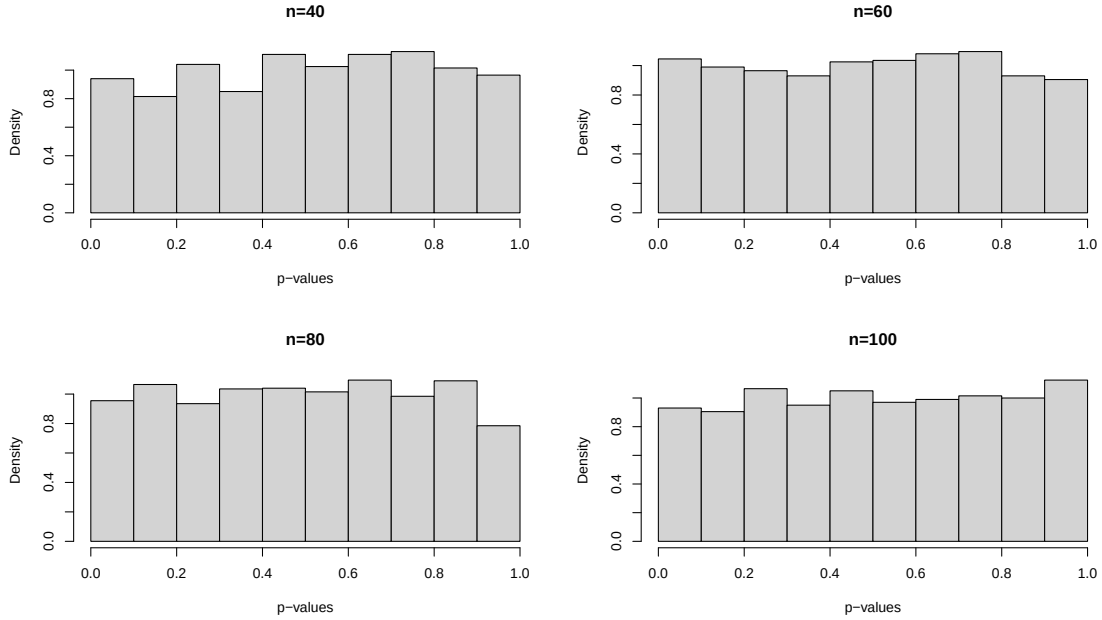


Figure C.2: Histograms of the p-values of the test statistics under H_0 using the wild bootstrap critical value for some values of n in Scenario B.

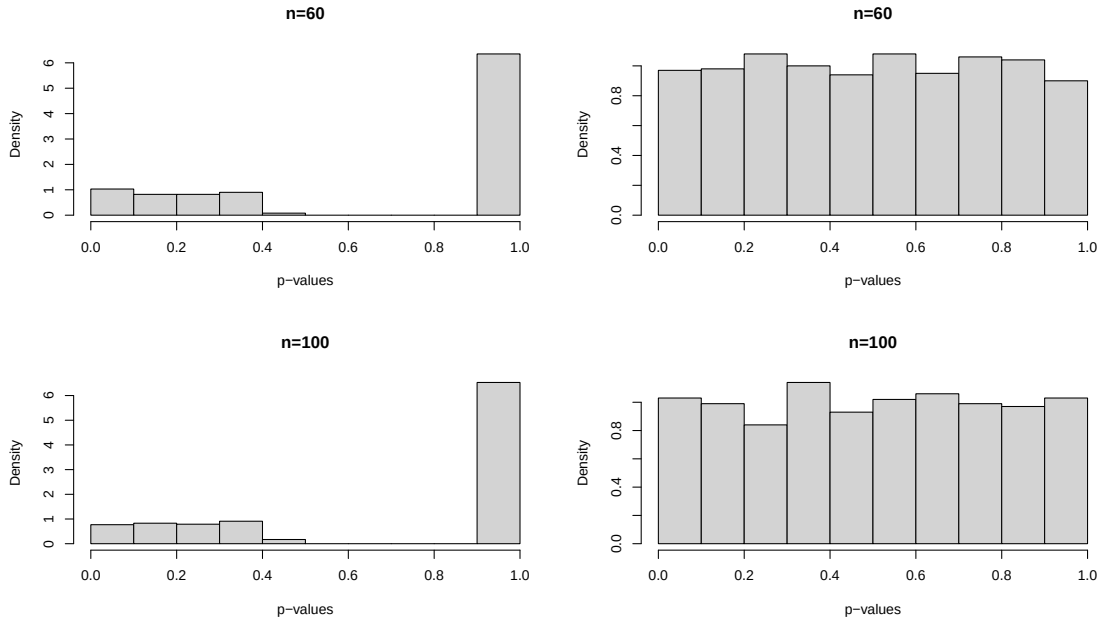


Figure C.3: Histograms of the test statistics p-values under H_0 for the FLCM (left column) and MDD (right column) methods in scenario A of Ghosal and Maity (2022a).

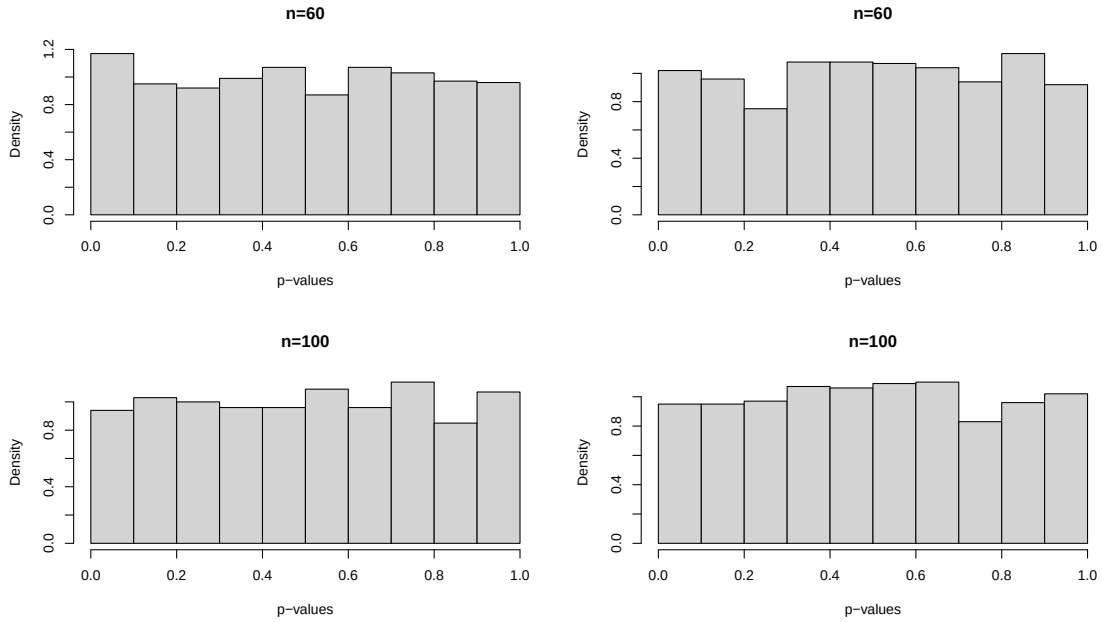


Figure C.4: Histograms of the test statistics p-values under H_0 for the ANFCM (left column) and MDD (right column) methods in modified scenario B of Kim et al. (2018).

C.3 Proofs of results for MDD significant tests

This document collects the proofs of the main results. In Section C.3.1 the unbiasedness of $\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$ is proved. Section C.3.2 displays the Hoeffding decomposition results and their implications under the null as well as local alternative hypothesis. Eventually, in Section C.3.3 asymptotic distribution for the statistic is obtained under the null and local alternatives.

We remind that operators $\mathbb{E}[\cdot]$, $\mathbb{V}[\cdot]$, $\mathbb{C}[\cdot,\cdot]$ and $\mathrm{tr}[\cdot]$ apply for the expectation, variance, covariance, and matrix trace, respectively. Terms denoted as $\tilde{A}$ represents an integrated version in the $\mathcal{D}$ domain, i.e. $\tilde{A} = \int_{\mathcal{D}} A(t)dt$. Symbols $\overline{A}$ and $\overline{\overline{A}}$ are the simple and double $\mathcal{U}$-centering versions introduced in Section 2 of the main document. When both situations arise together, for example, $\widetilde{\overline{\overline{B}}}$, this means the integrated version of the double centered (in this example) term. We keep notation $\dot{U}(X(t), X'(t))$ to say that this term changes now with the sample size n and with d, which is the dimension of the considered covariates in $D \subset \{1,\ldots,p\}$, $\#D = d \leq p$.

C.3.1 Unbiasedness of $\widetilde{MDD}_n^2(\boldsymbol{Y_n(t)}|_{\boldsymbol{X_{nj}(t)}})$

First of all, making use of expression (6) of the article, we remind that

$$\begin{aligned}\widetilde{MDD}^2(Y(t)|_{X_j(t)}) =& \mathbb{E}\left[J\left(X_j(t), \widetilde{X_j'(t)}\right) L(Y(t), Y'(t))\right] + \mathbb{E}\left[J\left(\widetilde{X_j(t), X_j'(t)}\right)\right]\mathbb{E}\left[L(\widetilde{Y(t), Y'(t)})\right] \\ & - 2\mathbb{E}\left[J\left(X_j(t), \widetilde{X_j'(t)}\right) L(Y(t), Y''(t))\right]\end{aligned}$$

with $J\left(\widetilde{X_j(t), X_j'(t)}\right) = \int_{\mathcal{D}} |X_j(t) - X_j'(t)|dt$ and $L(\widetilde{Y(t), Y'(t)}) = 1/2 \int_{\mathcal{D}} \|Y(t) - Y'(t)\|_q^2 dt$.

Now, applying $\mathcal{U}$-centering properties, it is verified that

$$\begin{aligned}&\sum_{i\neq l} \int_{\mathcal{D}} (A_{il}(t))_j \, \overline{\overline{B}}_{il}(t)dt \\ &= \sum_{i\neq l} \int_{\mathcal{D}} (A_{il}(t))_j \left(B_{il}(t) - \frac{1}{n-2}\sum_{q=1}^n B_{iq}(t) - \frac{1}{n-2}\sum_{r=1}^n B_{rl}(t) + \frac{1}{(n-1)(n-2)}\sum_{q,r=1}^n B_{qr}(t)\right) dt \\ &= \sum_{i\neq l} \int_{\mathcal{D}} (A_{il}(t))_j \, B_{il}(t)dt - \frac{1}{n-2}\sum_{i\neq l}\sum_{q=1}^n \int_{\mathcal{D}} (A_{il}(t))_j \, B_{iq}(t)dt \\ &\quad - \frac{1}{n-2}\sum_{i\neq l}\sum_{r=1}^n \int_{\mathcal{D}} (A_{il}(t))_j \, B_{rl}(t)dt + \frac{1}{(n-1)(n-2)}\sum_{i,l=1}^n\sum_{q,r=1}^n \int_{\mathcal{D}} (A_{il}(t))_j \, B_{qr}(t)dt \\ &= \mathrm{tr}\left[\left(\widetilde{AB}\right)_j\right] + \frac{\mathbf{1}_n^\top \left(\tilde{A}\right)_j \mathbf{1}_n \mathbf{1}_n^\top \tilde{B} \mathbf{1}_n}{(n-1)(n-2)} - \frac{2\mathbf{1}_n^\top \left(\widetilde{AB}\right)_j \mathbf{1}_n}{(n-2)}\end{aligned}$$

where $\mathbf{1}_n \in \mathbb{R}^n$ is a vector of ones and $(\tilde{A})_j = \int_{\mathcal{D}} (A(t))_j \, dt$.

Using Lemma 1 of Park et al. (2015), $n(n-3)\left((\overline{A})_j \cdot \overline{B}\right) = \sum_{i\neq l}(\overline{A}_{il})_j \overline{B}_{il} = \sum_{i\neq l}(A_{il})_j \overline{\overline{B}}_{il}$. Then, using the fact that $\tilde{B}_{ii} = 0$ it is verified that $\overline{\overline{B}}_{il} = \widetilde{\overline{B}}_{il}$ and we have

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = \frac{1}{n(n-3)}\left(\text{tr}\left[\left(\widetilde{A\overline{B}}\right)_j\right] + \frac{\mathbf{1}_n^\top\left(\tilde{A}\right)_j\mathbf{1}_n\mathbf{1}_n^\top\overline{B}\mathbf{1}_n}{(n-1)(n-2)} - \frac{2\mathbf{1}_n^\top\left(\widetilde{A\overline{B}}\right)_j\mathbf{1}_n}{(n-2)}\right)$$

Denote by $(n)_k = n!/(n-k)!$ and I_k^n the k-tuples of indices $\{1,\ldots,n\}$ without replacement. Then, it can be seen that

$$(n)_2^{-1}\mathbb{E}\left[\sum_{(i,l)\in I_2^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{il}(t)dt\right] = (n)_2^{-1}\mathbb{E}\left[\text{tr}\left[\left(\widetilde{A\overline{B}}\right)_j\right]\right] = \mathbb{E}\left[J\left(X_j(t),\widetilde{X_j'(t)}\right)L(Y(t),Y'(t))\right]$$

$$(n)_4^{-1}\mathbb{E}\left[\sum_{(i,l,q,r)\in I_4^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{qr}(t)dt\right] = (n)_4^{-1}\mathbb{E}\left[\mathbf{1}_n^\top\left(\tilde{A}\right)_j\mathbf{1}_n\mathbf{1}_n^\top\overline{B}\mathbf{1}_n - 4\mathbf{1}_n^\top\left(\widetilde{A\overline{B}}\right)_j\mathbf{1}_n\right.$$
$$\left.+2\text{tr}\left[\left(\widetilde{A\overline{B}}\right)_j\right]\right] = \mathbb{E}\left[J\left(\widetilde{X_j(t),X_j'(t)}\right)\right]\mathbb{E}\left[L(\widetilde{Y(t),Y'(t)})\right]$$

$$(n)_3^{-1}\mathbb{E}\left[\sum_{(i,l,q)\in I_3^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{iq}(t)dt\right] = (n)_3^{-1}\mathbb{E}\left[\mathbf{1}_n^\top\left(\widetilde{A\overline{B}}\right)_j\mathbf{1}_n - \text{tr}\left[\left(\widetilde{A\overline{B}}\right)_j\right]\right]$$
$$= \mathbb{E}\left[J\left(X_j(t),\widetilde{X_j'(t)}\right)L(Y(t),Y''(t))\right]$$

As a consequence, seeing that

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) = (n)_2^{-1}\sum_{(i,l)\in I_2^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{il}(t)dt + (n)_4^{-1}\sum_{(i,l,q,r)\in I_4^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{qr}(t)dt$$
$$-2(n)_3^{-1}\sum_{(i,l,q)\in I_3^n}\int_{\mathcal{D}}(A_{il}(t))_j\,B_{iq}(t)dt$$

this is clearly an unbiased estimator of $\widetilde{MDD}^2(Y(t)|_{X_j(t)})$.

C.3.2 Hoeffding decomposition

Let $\widetilde{\Psi_c(w_1,\ldots,w_c)} = \mathbb{E}\left[\int_{\mathcal{D}}\Psi(w_1,\ldots,w_c,Z_{c+1}(t),\ldots,Z_4(t))dt\right]$ for values of $c = 1,2,3,4$ and $Z_i(t) = (X_i(t),Y_i(t)) \overset{d}{=} (X(t),Y(t))$, where $\Psi(\cdot)$ is defined in Section 2 of the main manuscript and $\overset{d}{=}$ means equal in distribution. Let $w = (x,y)$, $w' = (x',y')$, $w'' = (x'',y'')$ and $w''' = (x''',y''')$, where $x,x',x'',x''' \in \mathbb{R}^p$ and $y,y',y'',y''' \in \mathbb{R}$. Besides let $Z'(t) = (X'(t),Y'(t))$, $Z''(t) = (X''(t),Y''(t))$ and $Z'''(t) = (X'''(t),Y'''(t))$ independent copies of $Z(t) = (X(t),Y(t))$. We define $\widetilde{U(x,x')} = \int_{\mathcal{D}}\mathbb{E}[J(x,X'(t))]dt + \int_{\mathcal{D}}\mathbb{E}[J(X(t),x')]dt - \int_{\mathcal{D}}J(x,x')dt - \int_{\mathcal{D}}\mathbb{E}[J(X(t),X'(t))]dt$ and $\widetilde{V(y,y')} = \int_{\mathcal{D}}(y-\mu_Y)(y'-\mu_Y)dt$ taking

$\mu_Y = \mathbb{E}[Y(t)]$. Then, we obtain that

$$\widetilde{\Psi_1(w)} = \frac{1}{2}\left\{\mathbb{E}\left[\widetilde{U(x,X(t))V(y,Y(t))}\right] + \widetilde{MDD}^2\left(Y(t)|_{X(t)}\right)\right\}$$

and

$$\begin{aligned}
&\widetilde{\Psi_2(w,w')} \\
&= \frac{1}{6}\left\{\widetilde{U(x,x')}\widetilde{V(y,y')} + \widetilde{MDD}^2(Y(t)|_{X(t)}) + \mathbb{E}\left[\widetilde{U(x,X(t))V(y,Y(t))}\right] + \mathbb{E}\left[\widetilde{U(x',X(t))V(y',Y(t))}\right]\right. \\
&\quad \left. + \mathbb{E}\left[\left(\widetilde{U(x,X(t))} - \widetilde{U(x',X(t))}\right)\left(\widetilde{V(y,Y(t))} - \widetilde{V(y',Y(t))}\right)\right]\right\}
\end{aligned}$$

Besides, we have

$$\begin{aligned}
\widetilde{\Psi_3(w,w',w'')} = \frac{1}{12}\Bigg\{&\left(2\widetilde{U(x,x')} - \widetilde{U(x',x'')} - \widetilde{U(x,x'')}\right)\widetilde{V(y,y')} \\
&+ \left(2\widetilde{U(x,x'')} - \widetilde{U(x,x')} - \widetilde{U(x',x'')}\right)\widetilde{V(y,y'')} \\
&+ \left(2\widetilde{U(x',x'')} - \widetilde{U(x,x')} - \widetilde{U(x,x'')}\right)\widetilde{V(y',y'')} \\
&+ \mathbb{E}\left[\left(2\widetilde{U(x,X(t))} - \widetilde{U(x',X(t))} - \widetilde{U(x'',X(t))}\right)\widetilde{V(y,Y(t))}\right] \\
&+ \mathbb{E}\left[\left(2\widetilde{U(x',X(t))} - \widetilde{U(x,X(t))} - \widetilde{U(x'',X(t))}\right)\widetilde{V(y',Y(t))}\right] \\
&+ \mathbb{E}\left[\left(2\widetilde{U(x'',X(t))} - \widetilde{U(x,X(t))} - \widetilde{U(x',X(t))}\right)\widetilde{V(y'',Y(t))}\right]\Bigg\}
\end{aligned}$$

and

$$\begin{aligned}
&\widetilde{\Psi_4(w,w',w'',w''')} \\
&= \frac{1}{12}\left\{\left(2\widetilde{U(x,x')} + 2\widetilde{U(x'',x''')} - \widetilde{U(x,x'')} - \widetilde{U(x,x''')} - \widetilde{U(x',x'')} - \widetilde{U(x',x''')}\right)\left(\widetilde{V(y,y')} + \widetilde{V(y'',y''')}\right)\right. \\
&\quad + \left(2\widetilde{U(x,x'')} + 2\widetilde{U(x',x''')} - \widetilde{U(x,x')} - \widetilde{U(x,x''')} - \widetilde{U(x'',x')} - \widetilde{U(x'',x''')}\right)\left(\widetilde{V(y,y'')} + \widetilde{V(y',y''')}\right) \\
&\quad \left. + \left(2\widetilde{U(x,x''')} + 2\widetilde{U(x'',x')} - \widetilde{U(x,x'')} - \widetilde{U(x,x')} - \widetilde{U(x''',x'')} - \widetilde{U(x''',x')}\right)\left(\widetilde{V(y,y''')} + \widetilde{V(y',y'')}\right)\right\}
\end{aligned}$$

Analysis under the null hypothesis

Under the null hypothesis we have that $\mathbb{E}\left[Y(t)|_{X_j(t)}\right] = \mathbb{E}\left[Y(t)\right]$ almost surely $\forall t \in \mathcal{D}$ and every $j = 1, \ldots, p$, which also translates into $\widetilde{MDD}^2(Y(t)|_{X(t)}) = 0$. Then, it is verified

that $\widetilde{\Psi_1(w)} = 0$, $\widetilde{\Psi_2(w,w')} = \widetilde{U(x,x')}\widetilde{V(y,y')}/6$ and

$$\begin{aligned}\widetilde{\Psi_3(w,w',w'')} =& \frac{1}{12}\left\{\left(2\widetilde{U(x,x')} - \widetilde{U(x',x'')} - \widetilde{U(x,x'')}\right)\widetilde{V(y,y')}\right.\\ &+ \left(2\widetilde{U(x,x'')} - \widetilde{U(x,x')} - \widetilde{U(x',x'')}\right)\widetilde{V(y,y'')}\\ &\left.+ \left(2\widetilde{U(x',x'')} - \widetilde{U(x,x')} - \widetilde{U(x,x'')}\right)\widetilde{V(y',y'')}\right\}\end{aligned}$$

Furthermore, under the null, we can verify that

$$\mathbb{V}[\widetilde{\Psi_2(Z(t),Z'(t))}] = \frac{1}{36}\mathbb{E}\left[\widetilde{U(X(t),X'(t))}^2\widetilde{V(Y(t),Y'(t))}^2\right] = \frac{1}{36}\tilde{\xi}^2$$

and

$$\begin{aligned}&\mathbb{V}[\widetilde{\Psi_3(Z(t),Z'(t),Z''(t))}]\\ &= \frac{3}{144}\mathrm{Var}\left[\left(2\widetilde{U(X(t),X'(t))} - \widetilde{U(X'(t),X''(t))} - \widetilde{U(X(t),X''(t))}\right)\widetilde{V(Y(t),Y'(t))}\right]\\ &= \frac{3}{144}\left\{4\tilde{\xi}^2 + 2\mathbb{E}\left[\widetilde{U(X(t),X''(t))}^2\widetilde{V(Y(t),Y'(t))}^2\right]\right.\\ &\quad\left.+2\mathbb{E}\left[\widetilde{U(X(t),X''(t))U(X'(t),X''(t))}\widetilde{V(Y(t),Y'(t))}^2\right]\right\}.\end{aligned}$$

Moreover,

$$\begin{aligned}&\mathbb{V}[\widetilde{\Psi_4(Z(t),Z'(t),Z''(t),Z'''(t))}]\\ &= \frac{6}{144}\mathbb{E}\left[\widetilde{V(Y(t),Y'(t))}^2\left(\widetilde{U(X(t),X''(t))} + \widetilde{U(X'(t),X'''(t))} + \widetilde{U(X'(t),X''(t))}\right.\right.\\ &\quad\left.\left.+\ \widetilde{U(X(t),X'''(t))} - 2\widetilde{U(X(t),X'(t))} - 2\widetilde{U(X''(t),X'''(t))}\right)^2\right]\\ &= \frac{1}{6}\left\{\mathbb{E}\left[\widetilde{V(Y(t),Y'(t))}^2\widetilde{U(X(t),X''(t))U(X'(t),X''(t))}\right] + \tilde{\xi}^2\right.\\ &\quad\left.+\ \mathbb{E}\left[\widetilde{V(Y(t),Y'(t))}^2\widetilde{U(X(t),X''(t))}^2\right] + \mathbb{E}\left[\widetilde{V(Y(t),Y'(t))}^2\right]\mathbb{E}\left[\widetilde{U(X(t),X''(t))}^2\right]\right\}\end{aligned}$$

Using the Cauchy-Schwarz inequality we can obtain that

$$\begin{aligned}&\mathbb{E}\left[\widetilde{U(X(t),X''(t))U(X'(t),X''(t))V(Y(t),Y'(t))}^2\right]\\ &\leq \left\{\mathbb{E}\left[\widetilde{U(X(t),X''(t))}^2\widetilde{V(Y(t),Y'(t))}^2\right]\right\}^{1/2}\left\{\mathbb{E}\left[\widetilde{U(X'(t),X''(t))}^2\widetilde{V(Y(t),Y'(t))}^2\right]\right\}^{1/2}\\ &= \mathbb{E}\left[\widetilde{U(X(t),X''(t))}^2\widetilde{V(Y(t),Y'(t))}^2\right].\end{aligned}$$

Besides, under the assumption that

$$\frac{\mathbb{E}\left[U(\widetilde{X(t),X''(t)})^2 V(\widetilde{Y(t),Y'(t)})^2\right]}{\tilde{\xi}^2} = o(n),$$
$$\frac{\mathbb{E}\left[V(\widetilde{Y(t),Y'(t)})^2\right]\mathbb{E}\left[U(\widetilde{X(t),X'(t)})^2\right]}{\tilde{\xi}^2} = o(n^2),$$

we have

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_n(t)}}) = \frac{1}{\binom{n}{2}} \sum_{1\le i<l\le n} U(\widetilde{X_i(t),X_l(t)})V(\widetilde{Y_i(t),Y_l(t)}) + \mathcal{R}_n,$$

where $\mathcal{R}_n$ is the remainder term which is asymptotically negligible (see Serfling (1980)).

Analysis under local alternatives

We consider the case where $\widetilde{MDD}^2(Y(t)|_{X(t)})$ is nonzero, i.e., the conditional mean of $Y(t)$ may depend on $X(t)$. Recall that $\widetilde{L(x,y)} = \mathbb{E}\left[U(\widetilde{x,X(t)})V(\widetilde{y,Y(t)})\right]$. Under the assumption that

$$\mathbb{V}\left[L(\widetilde{X(t),Y(t)})\right] = o(n^{-1}\tilde{\xi}^2), \quad \mathbb{V}\left[L(\widetilde{X(t),Y'(t)})\right] = o(\tilde{\xi}^2), \tag{C.1}$$

we get

$$\mathbb{V}\left[\widetilde{\Psi_1}\right] = o(n^{-1}\tilde{\xi}^2), \quad \mathbb{V}\left[\widetilde{\Psi_2}\right] = \frac{\tilde{\xi}^2}{36}(1+o(1))$$

which means that $\mathbb{V}\left[\widetilde{\Psi_1}\right]$ tends to zero as n increases and $\mathbb{V}\left[\widetilde{\Psi_2}\right]$ is always positive and nonnull.

Moreover,

$$\mathbb{V}\left[\Psi_3(Z(t),\widetilde{Z'(t)},Z''(t))\right] \le C\left\{\tilde{\xi}^2 + \mathbb{E}\left[U(\widetilde{X(t),X''(t)})^2 V(\widetilde{Y(t),Y'(t)})^2\right]\right\}$$

and

$$\mathbb{V}\left[\Psi_4(Z(t),Z'\widetilde{(t),Z''}(t),Z'''(t))\right]$$
$$\le C'\left\{\mathbb{E}\left[V(\widetilde{Y(t),Y'(t)})^2 U(\widetilde{X(t),X''(t)})^2\right] + \mathbb{E}\left[V(\widetilde{Y(t),Y'(t)})^2\right]\mathbb{E}\left[U(\widetilde{X(t),X'(t)})^2\right] + \tilde{\xi}^2\right\}$$

for some constants $C, C' \ge 0$.

Then, under assumption (C.1),

$$\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_n(t)}}) - \widetilde{MDD}^2(Y(t)|_{X(t)}) = \frac{1}{\binom{n}{2}} \sum_{1\le i<l\le n} U(\widetilde{X_i(t),X_l(t)})V(\widetilde{Y_i(t),Y_l(t)}) + \mathcal{R}_n.$$

Applying the above arguments to $\sum_{j\in D}\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}})$, it can be seen that

$$\sum_{j\in D}\left\{\widetilde{MDD}_n^2(\mathbf{Y_n(t)}|_{\mathbf{X_{nj}(t)}}) - \widetilde{MDD}^2(Y(t)|_{X_j(t)})\right\} = \frac{1}{\binom{n}{2}}\sum_{1\le i<l\le n}\dot{U}(\widetilde{X_i(t), X_l(t)})V(\widetilde{Y_i(t), Y_l(t)}) + \sum_{j\in D}(\mathcal{R}_n)_j.$$

where the kernel $\dot{U}(\widetilde{X(t), X'(t)}) = \sum_{j\in D} U_j(\widetilde{x_j(t), x_j'(t)})$ is changing now with (n, d). Next, to provide that the remainder term $\sum_{j\in D}(\mathcal{R}_n)_j$ is asymptotically negligible we assume that

$$\frac{\mathbb{E}\left[\dot{U}(\widetilde{X(t), X''(t)})^2 V(\widetilde{Y(t), Y'(t)})^2\right]}{\tilde{S}^2} = o(n),$$

$$\frac{\mathbb{E}\left[\dot{U}(\widetilde{X(t), X'(t)})^2\right]\mathbb{E}\left[V(\widetilde{Y(t), Y'(t)})^2\right]}{\tilde{S}^2} = o(n^2),$$

$$\mathrm{Var}\left[\dot{L}(\widetilde{X(t), Y(t)})\right] = o(n^{-1}\tilde{S}^2), \quad \mathrm{Var}\left[\dot{L}(\widetilde{X(t), Y'(t)})\right] = o(\tilde{S}^2)$$

with $\widetilde{\dot{L}(x, y)} = \mathbb{E}\left[\dot{U}(\widetilde{x, X(t)})V(\widetilde{y, Y(t)})\right]$.

C.3.3 Asymptotic normality under the null and alternatives

Here, we prove the asymptotic normality of T_D making use of the Central Limit Theorem for martingale difference sequences. For this purpose, we pay attention to the results of the Hoeffding decomposition of the above section under both, the null hypothesis and local alternatives to define an adequate mean-zero martingale sequence. In particular, results of Corollary 3.1 of Hall and Heyde (1980) are employed over this process.

Analysis under H_0

Define

$$S_r := \sum_{l=2}^{r}\sum_{i=1}^{r-1}\dot{U}(\widetilde{X_i(t), X_l(t)})V(\widetilde{Y_i(t), Y_l(t)}) = \sum_{l=2}^{r}\sum_{i=1}^{r-1}H(\widetilde{Z_i(t), Z_l(t)})$$

and the filtration $\mathcal{F}_r = \sigma\{Z_1(t), Z_2(t), \ldots, Z_r(t)\}$ with $Z_i(t) = (X_i(t), Y_i(t))$. Then, S_r is adaptive to $\mathcal{F}_r$ and is a mean-zero martingale sequence verifying $\mathbb{E}[S_r] = 0$ and

$$\mathbb{E}[S_{r'}|_{\mathcal{F}_r}] = S_r + \sum_{l=r+1}^{r'}\sum_{i=1}^{l-1}\mathbb{E}\left[\mathbb{E}\left[\dot{U}(\widetilde{X_i(t), X_l(t)})V(\widetilde{Y_i(t), Y_l(t)})|_{\mathcal{F}_r, X_i(t), X_l(t)}\right]|_{\mathcal{F}_r}\right] = S_r$$

for $r' \ge r$. Thus, by Corollary 3.1 of Hall and Heyde (1980) we can guarantee the asymptotic normality of T_D if the conditions (C.2) and (C.3) are verified. Specifically,

defining $\mathcal{W}_l = \sum_{i=1}^{l-1} H(\widetilde{Z_i(t), Z_l(t)})$, it is sufficient to see that

$$\sum_{l=1}^{n} B^{-2}\mathbb{E}\left[\mathcal{W}_l^2 \mathbb{I}(|\mathcal{W}_l| > \varepsilon B)|_{\mathcal{F}_{l-1}}\right] \xrightarrow{p} 0 \tag{C.2}$$

for B such that

$$\sum_{l=1}^{n} \mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]/B^2 \xrightarrow{p} C > 0. \tag{C.3}$$

We start proving that (C.3) is verified taking $B^2 = n(n-1)\tilde{S}^2/2$ and $C = 1$. This translates into proving that

$$\frac{2}{n(n-1)\tilde{S}^2}\sum_{l=1}^{n} \mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right] \xrightarrow{p} 1. \tag{C.4}$$

For this purpose, note that

$$\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}] = \mathbb{E}\left[\sum_{i,k=1}^{l-1} H(\widetilde{Z_i(t), Z_l(t)})H(\widetilde{Z_k(t), Z_l(t)})|_{\mathcal{F}_{l-1}}\right] = \sum_{i,k=1}^{l-1} G(\widetilde{Z_i(t), Z_k(t)}),$$

and

$$\begin{aligned}
&\frac{2}{n(n-1)}\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2] \\
&= \frac{2}{n(n-1)}\sum_{l=2}^{n}\mathbb{E}\left[\sum_{i,k=1}^{l-1}\int_{\mathcal{D}}(Y_i(t)-\mu(t))(Y_k(t)-\mu(t))(Y_l(t)-\mu(t))^2\dot{U}(X_i(t),X_l(t))\dot{U}(X_k(t),X_l(t))dt\right] \\
&= \frac{2}{n(n-1)}\sum_{l=2}^{n}\mathbb{E}\left[\sum_{i,k=1}^{l-1}\int_{\mathcal{D}}(Y_i(t)-\mu(t))^2(Y_l(t)-\mu(t))^2\dot{U}(X_i(t),X_l(t))^2dt\right] \\
&= \mathbb{E}\left[V(\widetilde{Y(t), Y'(t)})^2\dot{U}(\widetilde{X(t), X'(t)})^2\right] = \mathbb{E}\left[H(\widetilde{Z(t), Z'(t)})^2\right] = \tilde{S}^2.
\end{aligned} \tag{C.5}$$

Then, defining

$$\begin{aligned}
\mathcal{D}_1 &= \mathbb{E}\left[H(\widetilde{Z(t), Z''(t)})^2 H(\widetilde{Z'(t), Z''(t)})^2\right] - \left(\mathbb{E}\left[H(\widetilde{Z(t), Z'(t)})^2\right]\right)^2 = \mathbb{V}\left[G(\widetilde{Z(t), Z(t)})\right] \\
\mathcal{D}_2 &= \mathbb{E}\left[H(\widetilde{Z(t), Z'(t)}H(\widetilde{Z'(t), Z''(t)}H(\widetilde{Z''(t), Z'''(t)}H(\widetilde{Z'''(t), Z(t)}\right] = \mathbb{E}\left[G(\widetilde{Z(t), Z'(t)})^2\right]
\end{aligned}$$

we have for $l \geq l'$ that

$$\begin{aligned}
\mathbb{C}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right], \mathbb{E}\left[\mathcal{W}_{l'}^2|_{\mathcal{F}_{l'-1}}\right]\right] &= \sum_{i,k=1}^{l-1}\sum_{i',k'=1}^{l'-1}\mathbb{C}\left[G(\widetilde{Z_i(t), Z_k(t)}), G(\widetilde{Z_{i'}(t), Z_{k'}(t)})\right] \\
&= (l'-1)\mathcal{D}_1 + 2(l'-1)(l'-2)\mathcal{D}_2.
\end{aligned}$$

As a result, under the assumption that

$$\frac{\mathbb{E}\left[G(\widetilde{Z(t),Z'(t)})^2\right]}{\left\{\mathbb{E}\left[H(\widetilde{Z(t),Z'(t)})^2\right]\right\}^2}\longrightarrow 0,\qquad \frac{\mathbb{E}\left[H(\widetilde{Z(t),Z''(t)})^2 H(\widetilde{Z'(t),Z''(t)})^2\right]}{n\left\{\mathbb{E}\left[H(\widetilde{Z(t),Z'(t)})^2\right]\right\}^2}\longrightarrow 0$$

we have

$$\frac{4}{n^2(n-1)^2}\sum_{l,l'=2}^{n}\mathbb{C}\left[\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}],\mathbb{E}[\mathcal{W}_{l'}^2|_{\mathcal{F}_{l'-1}}]\right]=O(\mathcal{D}_1/n+\mathcal{D}_2)=o(S^4). \tag{C.6}$$

Next, we make use of result (C.6) to prove (C.4). We notice that convergence in r-mean for $r\geq 1$ implies convergence in probability. Then, proving

$$\lim_{n\to\infty}\mathbb{E}\left[\left(\frac{2}{n(n-1)\tilde{S}^2}\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]-1\right)^2\right]=0.$$

implies that condition (C.4) is verified.

Given that

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{2}{n(n-1)\tilde{S}^2}\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]-1\right)^2\right]\\
&=\mathbb{E}\left[\frac{4}{n^2(n-1)^2\tilde{S}^4}\left(\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right)^2+1-\frac{4}{n(n-1)\tilde{S}^2}\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right]\\
&\overset{(a)}{=}\frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\left(\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right)^2\right]+1-2\\
&=\frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\left(\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right)^2\right]-1
\end{aligned}$$

using in (a) that $\mathbb{E}\left[\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right]=\mathbb{E}[\mathcal{W}_l^2]$ and (C.5). Thus, it is enough to see that

$$\frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\left(\sum_{l=2}^{n}\mathbb{E}[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}]\right)^2\right]-1\longrightarrow 0. \tag{C.7}$$

Seeing that

$$\begin{aligned}
&\frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\left(\sum_{l=2}^{n}\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\right)^2\right] \\
&\qquad = \frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\sum_{l=2}^{n}\left(\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\right)^2 + 2\sum_{l=2}^{n-1}\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\sum_{k=l+1}^{n}\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] \\
&\qquad = \frac{4}{n^2(n-1)^2\tilde{S}^4}\mathbb{E}\left[\sum_{l,k=2}^{n}\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] \\
&\qquad = \frac{4}{n^2(n-1)^2\tilde{S}^4}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right]
\end{aligned}$$

and

$$\begin{aligned}
&\frac{4}{n^2(n-1)^2}\sum_{l,k=2}^{n}\mathbb{C}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right],\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] \\
&= \frac{4}{n^2(n-1)^2}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\cdot\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] - \frac{4}{n^2(n-1)^2}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\right]\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] \\
&= \frac{4}{n^2(n-1)^2}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\cdot\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] - \frac{4}{n^2(n-1)^2}\sum_{l=2}^{n}\left(\mathbb{E}[\mathcal{W}_l^2]\right)^2 \\
&\overset{(a)}{=} \frac{4}{n^2(n-1)^2}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\cdot\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] - \tilde{S}^4 \\
&\overset{(b)}{=} o(\tilde{S}^4)
\end{aligned}$$

where we apply (C.5) in (a) and (C.6) in (b), we have guaranteed that

$$\frac{4}{n^2(n-1)^2\tilde{S}^4}\sum_{l,k=2}^{n}\mathbb{E}\left[\mathbb{E}\left[\mathcal{W}_l^2|_{\mathcal{F}_{l-1}}\right]\cdot\mathbb{E}\left[\mathcal{W}_k^2|_{\mathcal{F}_{k-1}}\right]\right] - \frac{\tilde{S}^4}{\tilde{S}^4} \longrightarrow 0,$$

which implies (C.7). Then, the convergence in r-mean is verified taking $r = 2$. This proves condition (C.4) which ensures condition (C.3) for $B^2 = n(n-1)\tilde{S}^2/2$ and $C = 1$.

Next, we prove the remainder condition (C.2) for this value of B. For this aim, we notice that

$$\begin{aligned}
0 &\leq \sum_{l=1}^{n}B^{-2}\mathbb{E}\left[\mathcal{W}_l^2\mathbb{I}(\|\mathcal{W}_l\|_1 > \varepsilon B)|_{\mathcal{F}_{l-1}}\right] \\
&\leq B^{-2-s}\varepsilon^{-s}\sum_{l=1}^{n}\mathbb{E}\left[\|\mathcal{W}_l\|_1^{2+s}|_{\mathcal{F}_{l-1}}\right] \leq B^{-2-s}\sum_{l=1}^{n}\mathbb{E}\left[\|\mathcal{W}_l\|_1^{2+s}|_{\mathcal{F}_{l-1}}\right]
\end{aligned}$$

for some $s > 0$. We prove that taking $s = 2$

$$B^{-4}\sum_{l=1}^{n}\mathbb{E}\left[\|\mathcal{W}_l\|_1^4|_{\mathcal{F}_{l-1}}\right] \xrightarrow{p} 0$$

with $B^2 = n(n-1)\tilde{S}^2/2$. It this case it is sufficient to show that

$$B^{-4}\sum_{l=1}^{n}\mathbb{E}\left[\|\mathcal{W}_l\|_1^4\right] \longrightarrow 0.$$

Under the assumption that

$$\frac{\mathbb{E}\left[\widetilde{H(Z(t),Z'(t))}^4\right]/n + \mathbb{E}\left[\widetilde{H(Z(t),Z''(t))}^2\widetilde{H(Z'(t),Z''(t))}^2\right]}{n\left\{\mathbb{E}\left[\widetilde{H(Z(t),Z'(t))}^2\right]\right\}^2} \longrightarrow 0,$$

we have

$$\begin{aligned}
\sum_{l=2}^{n}\mathbb{E}\left[\|\mathcal{W}_l\|_1^4\right] &= \sum_{l=2}^{n}\sum_{i_1,i_2,i_3,i_4=1}^{l-1}\mathbb{E}\left[\widetilde{H(Z_{i_1}(t),Z_l(t))}\widetilde{H(Z_{i_2}(t),Z_l(t))}\widetilde{H(Z_{i_3}(t),Z_l(t))}\widetilde{H(Z_{i_4}(t),Z_l(t))}\right] \\
&\overset{(a)}{=} \frac{n(n-1)}{2}\mathbb{E}\left[\widetilde{H(Z(t),Z'(t))}^4\right] + 3\sum_{l=2}^{n}\sum_{i_1\neq i_2}\mathbb{E}\left[\widetilde{H(Z_{i_1}(t),Z_l(t))}^2\widetilde{H(Z_{i_2}(t),Z_l(t))}^2\right] \\
&= \frac{n(n-1)}{2}\mathbb{E}\left[\widetilde{H(Z(t),Z'(t))}^4\right] + 3\sum_{l=2}^{n}(l-1)(l-2)\mathbb{E}\left[\widetilde{H(Z(t),Z''(t))}^2\widetilde{H(Z'(t),Z''(t))}^2\right] \\
&= o(B^4)
\end{aligned}$$

where in (a) we apply $\mathbb{E}\left[\widetilde{H(Z(t),Z'(t))}\right] = 0$. As a result, this last implies the condition (C.2).

Analysis under local alternatives

Now, we extend the asymptotic normality results to local alternatives. We refer to Section 1.7 of the supplementary material of Zhang et al. (2018) for some discussion about the local alternative model. Its arguments can be easily extended to our context considering the corresponding integrated versions.

Thus, under the assumption that $\mathbb{V}\left[\widetilde{\dot{L}(X(t),Y(t))}\right] = o(n^{-1}\tilde{S}^2)$, we have

$$\begin{aligned}
&\frac{1}{\sqrt{\binom{n}{2}}\tilde{S}}\sum_{1\leq i<l\leq n}\left\{\widetilde{\dot{U}(X_i(t),X_l(t))V(Y_i(t),Y_l(t))} - \mathbb{E}\left[\widetilde{\dot{U}(X_i(t),X_l(t))V(Y_i(t),Y_l(t))}\right]\right\} \\
&= \frac{1}{\sqrt{\binom{n}{2}}\tilde{S}}\sum_{1\leq i<l\leq n}\widetilde{H^*(Z_i(t),Z_l(t))} \\
&\quad + \frac{1}{\sqrt{\binom{n}{2}}\tilde{S}}\sum_{1\leq i<l\leq n}\left\{\widetilde{\dot{L}(X_i(t),Y_i(t))} + \widetilde{\dot{L}(X_l(t),Y_l(t))} - 2\mathbb{E}\left[\widetilde{\dot{U}(X_i(t),X_l(t))V(Y_i(t),Y_l(t))}\right]\right\} \\
&= \frac{1}{\sqrt{\binom{n}{2}}\tilde{S}}\sum_{1\leq i<l\leq n}\widetilde{H^*(Z_i(t),Z_l(t))} + o_p(1).
\end{aligned}$$

Using similar arguments by replacing H with H^*, we can show that

$$\frac{1}{\sqrt{\binom{n}{2}}\tilde{S}} \sum_{1\leq i<l\leq n} H^*(\widetilde{Z_i(t), Z_l(t)}) \xrightarrow{d} N(0,1),$$

which implies that, using Slutsky theorem,

$$\frac{1}{\sqrt{\binom{n}{2}}\tilde{S}} \sum_{1\leq i<l\leq n} \left\{ \dot{U}(\widetilde{X_i(t), X_l(t)}) V(\widetilde{Y_i(t), Y_l(t)}) - \mathbb{E}\left[\dot{U}(\widetilde{X_i(t), X_l(t)}) V(\widetilde{Y_i(t), Y_l(t)}) \right] \right\} \xrightarrow{d} N(0,1).$$

Appendix D

Extra results for CDC significant tests

D.1 Graphic results for local bootstrap

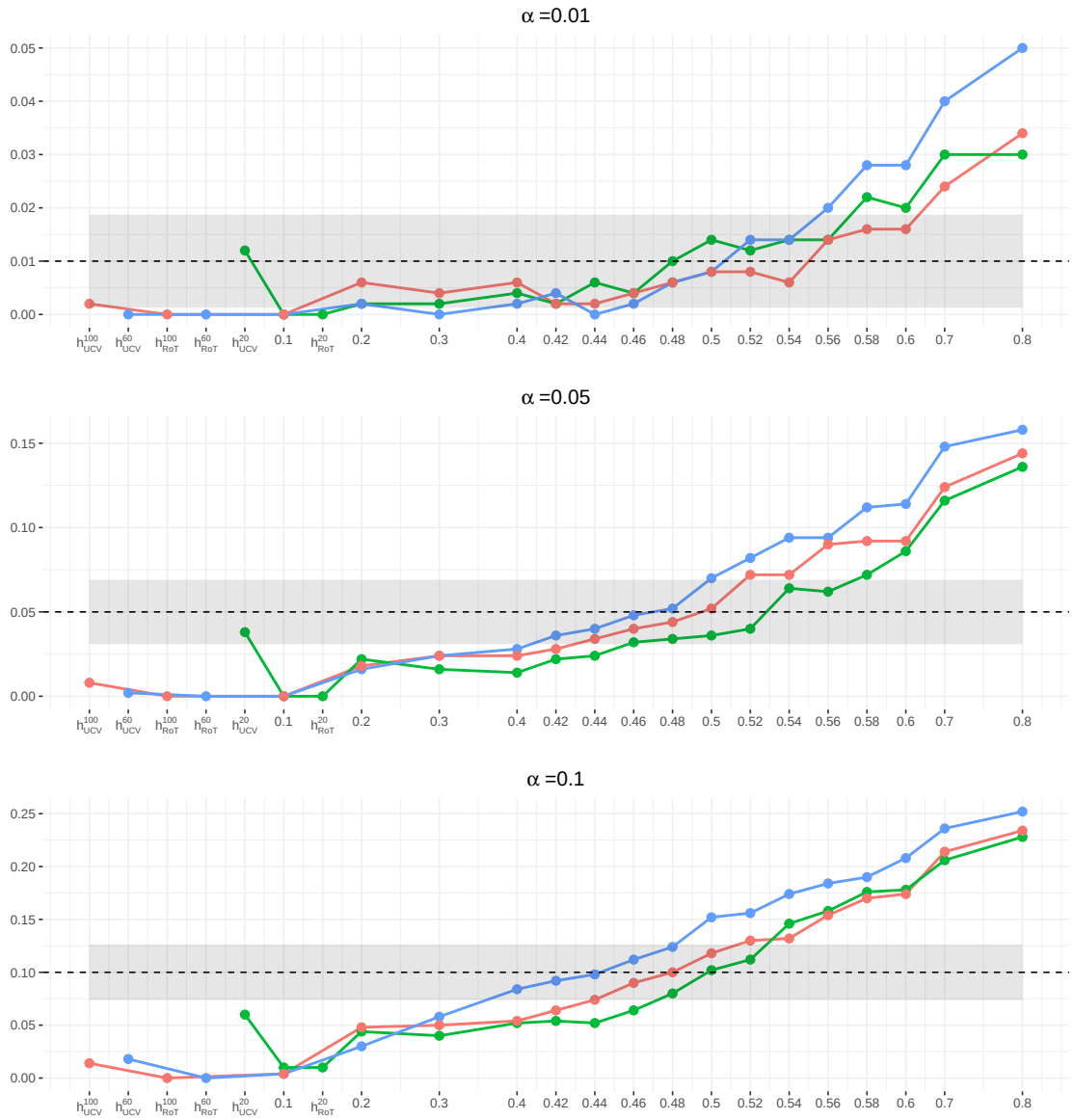


Figure D.1: Percentage of rejections under the null hypothesis (y axis) using the local bootstrap for different bandwidth parameters (x axis) taking levels $\alpha = 0.01, 0.05, 0.1$ and $n = 20$ (—●—), $n = 60$ (—●—) and $n = 100$ (—●—) in Scenario A. The bands mark the α confidence intervals at 95%.

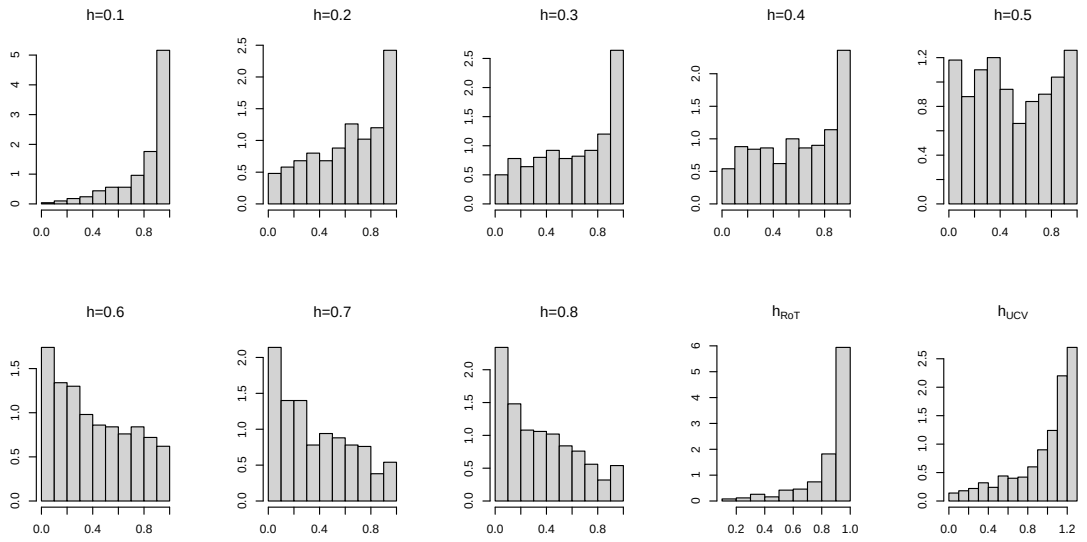


Figure D.2: Histograms of the obtained p-values using the local bootstrap in Scenario A for different bandwidth values taking $n = 100$ and simulating under the null hypothesis H_0.

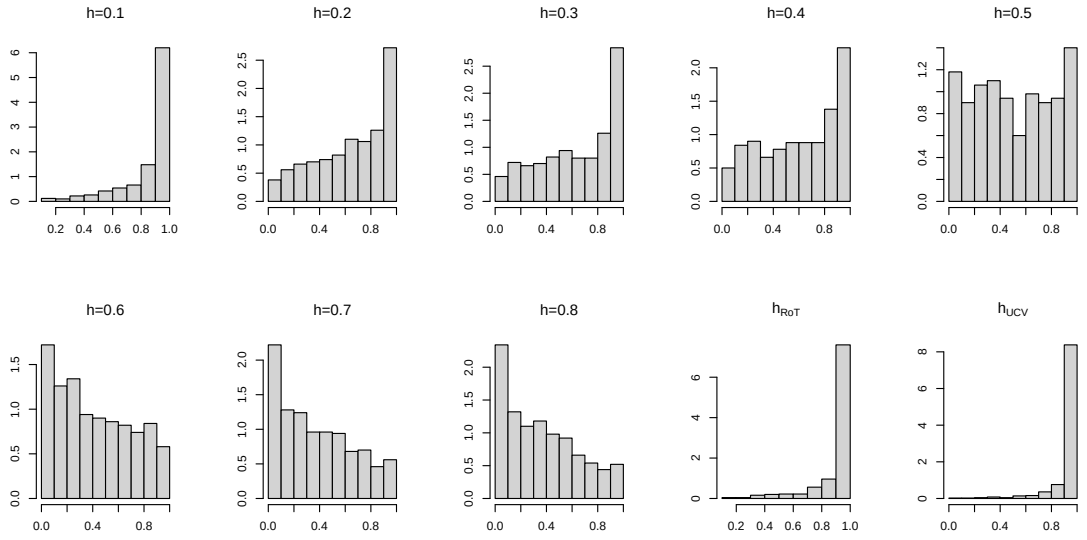


Figure D.3: Histograms of the obtained p-values using the local bootstrap in Scenario B for different bandwidth values taking $n = 100$ and simulating under the null hypothesis H_0.

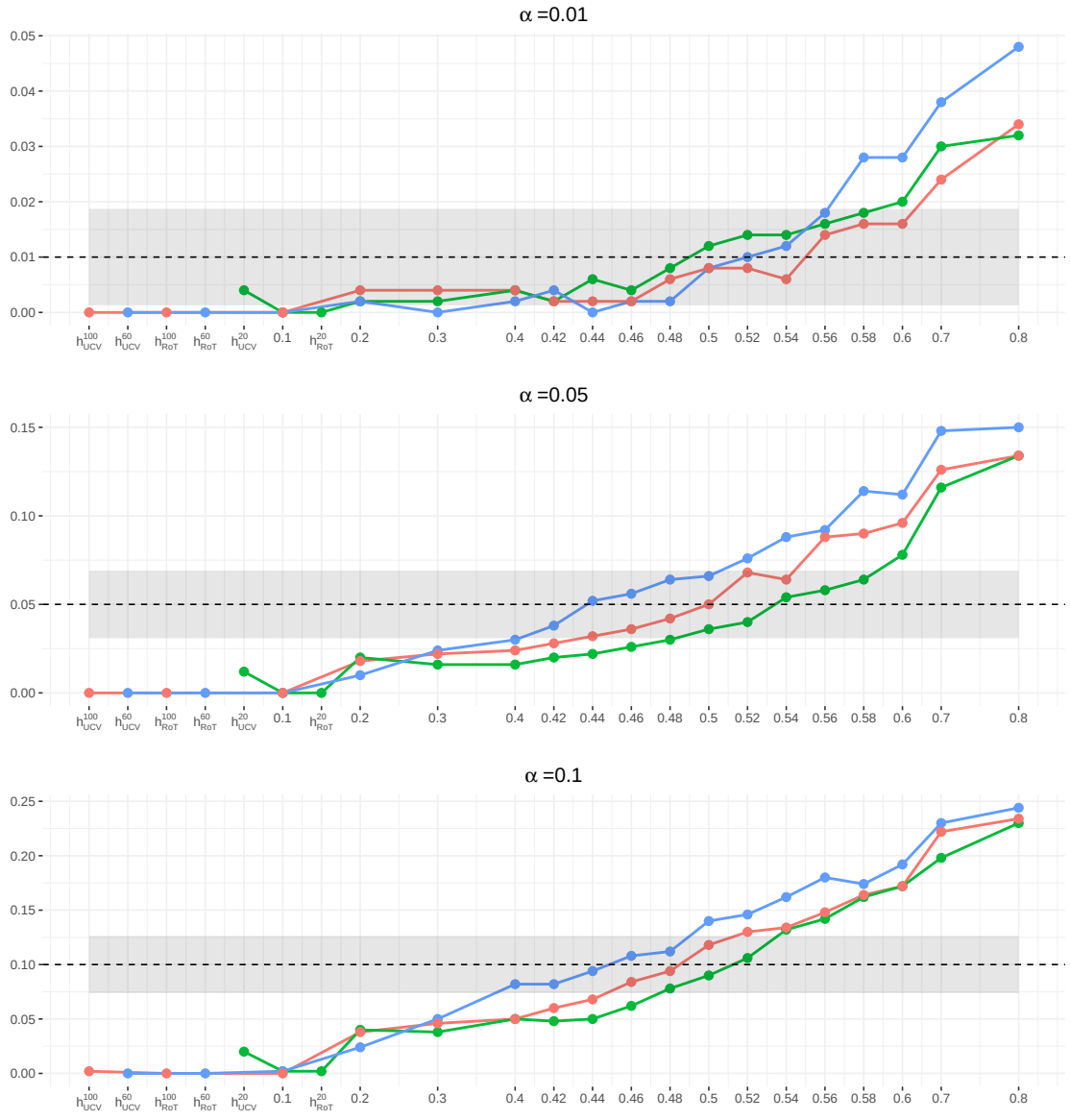


Figure D.4: Percentage of rejections under the null hypothesis (y axis) using the local bootstrap for different bandwidth parameters (x axis) taking levels $\alpha = 0.01, 0.05, 0.1$ and $n = 20$ (—●—), $n = 60$ (—●—) and $n = 100$ (—●—) in Scenario B. The bands mark the α confidence intervals at 95%.

D.2 Graphic results for local bootstrap only on Y

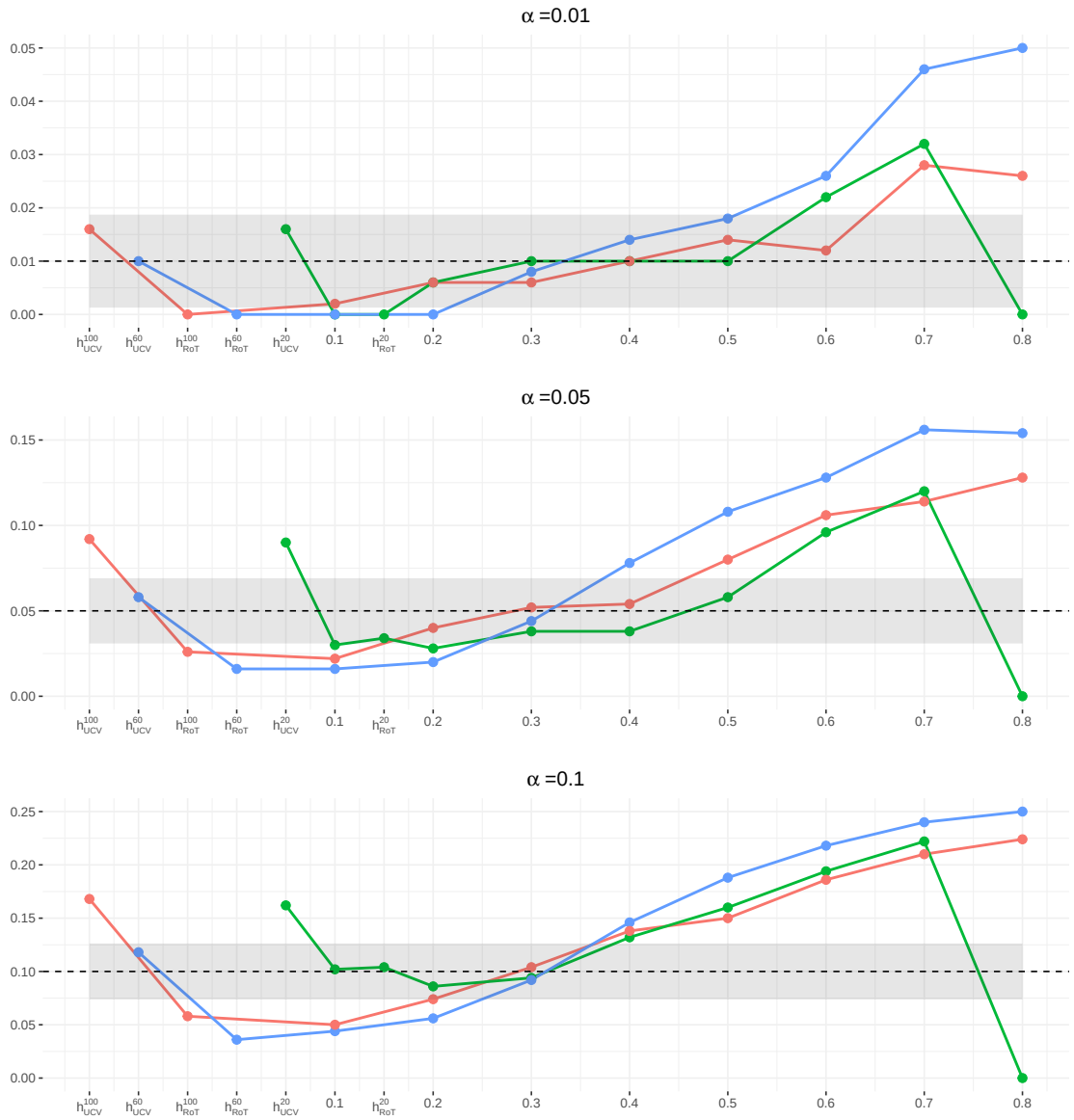


Figure D.5: Percentage of rejections under the null hypothesis (y axis) using the local bootstrap only resampling on Y for different bandwidth parameters (x axis) taking levels $\alpha = 0.01, 0.05, 0.1$ and $n = 20$ (—●—), $n = 60$ (—●—) and $n = 100$ (—●—) in Scenario A. The bands mark the α confidence intervals at 95%.

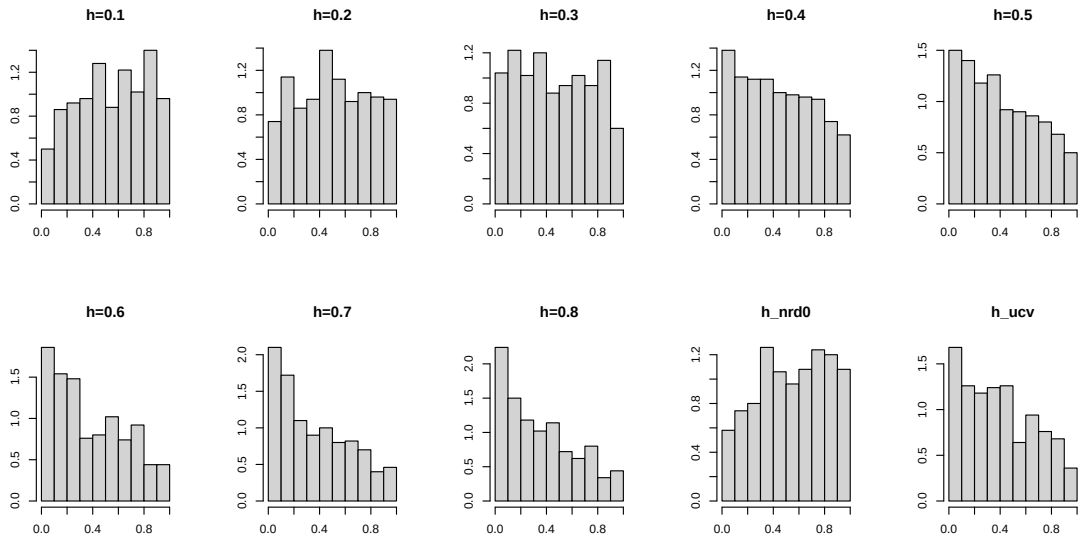


Figure D.6: Histograms of the obtained p-values using the local bootstrap only resampling on Y in Scenario A for different bandwidth values taking $n = 100$ and simulating under the null hypothesis H_0.

D.3 Calibration using permutations

Instead of the local bootstrap proposed in Section 6.2 and implemented in Algorithm 6.1, the test can be calibrated using permutations. This results in the scheme displayed in Algorithm D.1.

Algorithm D.1 (Calibration by means of permutations for significance tests using CDC)**.** Given a kernel function $K(\cdot)$ and some proper bandwidth parameter h:

1. For $i = 1 \ldots, N$ estimate $CDC^2\left(Y(t), X(t)|_{t=t_i}\right)$ by means of $\mathcal{V}_N(t_1), \ldots, \mathcal{V}_N(t_N)$ as defined in expression (6.3).

2. Approximate the sample statistic $E = \int_{\mathcal{D}\setminus\mathcal{N}} CDC^2(Y(t), X(t)|_t)\omega(t)f(t)dt$ by means of numerical techniques using $\{\mathcal{V}_N(t_1), \ldots, \mathcal{V}_N(t_N)\}$.

3. For $i = 1 \ldots, N$, draw Y_i^* and X_i^* from the estimators of the distribution functions given by

$$\hat{F}_{Y|_{t=t_i}}(y) = \frac{\mathbb{I}\{t^y \in [t_i - h, t_i + h]\}}{\sum_{l=1}^{N} \mathbb{I}\{t_l \in [t_i - h, t_i + h]\}} \quad \text{and} \quad \hat{F}_{X|_{t=t_i}}(x) = \frac{\mathbb{I}\{t^x \in [t_i - h, t_i + h]\}}{\sum_{l=1}^{N} \mathbb{I}\{t_l \in [t_i - h, t_i + h]\}},$$

respectively, where $\mathbb{I}\{\cdot\}$ is the indicator function and t^y, as well as t^x, denote the time points associated with y and x values. Roughly speaking, each observed value Y_l or X_l, with $l = 1, \ldots, N$, has a probability $1/\#\{t_l \in [t_i - h, t_i + h]\}$ to be chosen as the i-th bootstrap sample if $|t_l - t_i| \leq h$ and zero otherwise.

4. For $i = 1, \ldots, N$ obtain $\mathcal{V}_N^*(t_1), \ldots, \mathcal{V}_N^*(t_N)$ by expression (6.3) using the resamples $\mathbf{W_N^*} = \{(\mathbf{Y_1^*}, \mathbf{X_1^*}, \mathbf{t_1}), \ldots, (\mathbf{Y_N^*}, \mathbf{X_N^*}, \mathbf{t_N})\}$.

5. Approximate the permutated statistic $E^* = \int_{\mathcal{D}\setminus\mathcal{N}} CDC^{2*}(Y(t), X(t)|_t)\omega(t)f(t)dt$ making use of $\{\mathcal{V}_N^*(t_1), \ldots, \mathcal{V}_N^*(t_N)\}$.

6. Repeat steps 3-5 a number B of times obtaining $\{(E^*)^{(1)}, \ldots, (E^*)^{(B)}\}$.

7. Compute the resampling p-value as $\frac{1}{1+B}\left(1 + \sum_{b=1}^{B} \mathbb{I}\{(E^*)^{(b)} \geq E\}\right)$.

In this case, the same bandwidth value keeps been employed for both, statistic estimation and permutations algorithm as for the local bootstrap case.

Next, the good behavior of this implementation is proved. For this purpose, following guidelines of Section 6.3, Scenario A is employed. We refer the reader to Section 6.3.1 for more details about implementation and the employed bandwidth selection procedure. Then, an optimal bandwidth h is searched between some quantities in the $[0, 1]$ domain, considering a wide grid of values in this interval.

Figure D.8 collects the percentage of rejections, simulating under the null hypothesis when the test is calibrated using permutations for some h values. It seems that optimal options for the bandwidth parameter are around the $h = 0.46$ value. In this area, all percentages of rejections are between the confidence intervals. As a result, some bandwidth values guarantee that the test is well-calibrated. As expected, the optimal value for h can change relative to the sample size n, as for the local bootstrap. So, some optimal values for a given n may not be optimal for another. An example of the distribution of the p-values for $n = 100$ is displayed in Figure D.7. This illustration also proves that optimal values are close to $h = 0.5$.

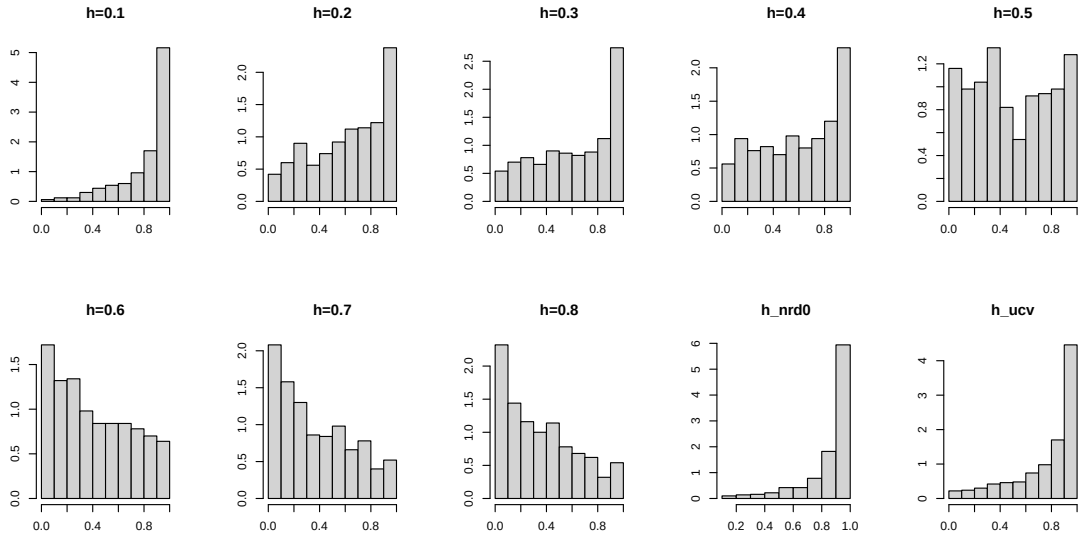


Figure D.7: Histograms of the obtained p-values using permutations in Scenario A for different bandwidth values taking $n = 100$ and simulating under the null hypothesis H_0.

Besides, given the results displayed in Figures D.7 and D.8, the rule-of-thumb (h_{RoT}) and the unbiased cross-validation criterion for density estimation (h_{UCV}) perform poorly for permutations calibration as well. Again, these automatic criteria select very small values, even fewer than 0.1, and obtain a percentage of rejections out of the confidence intervals. Hence, another approach is needed for proper calibration.

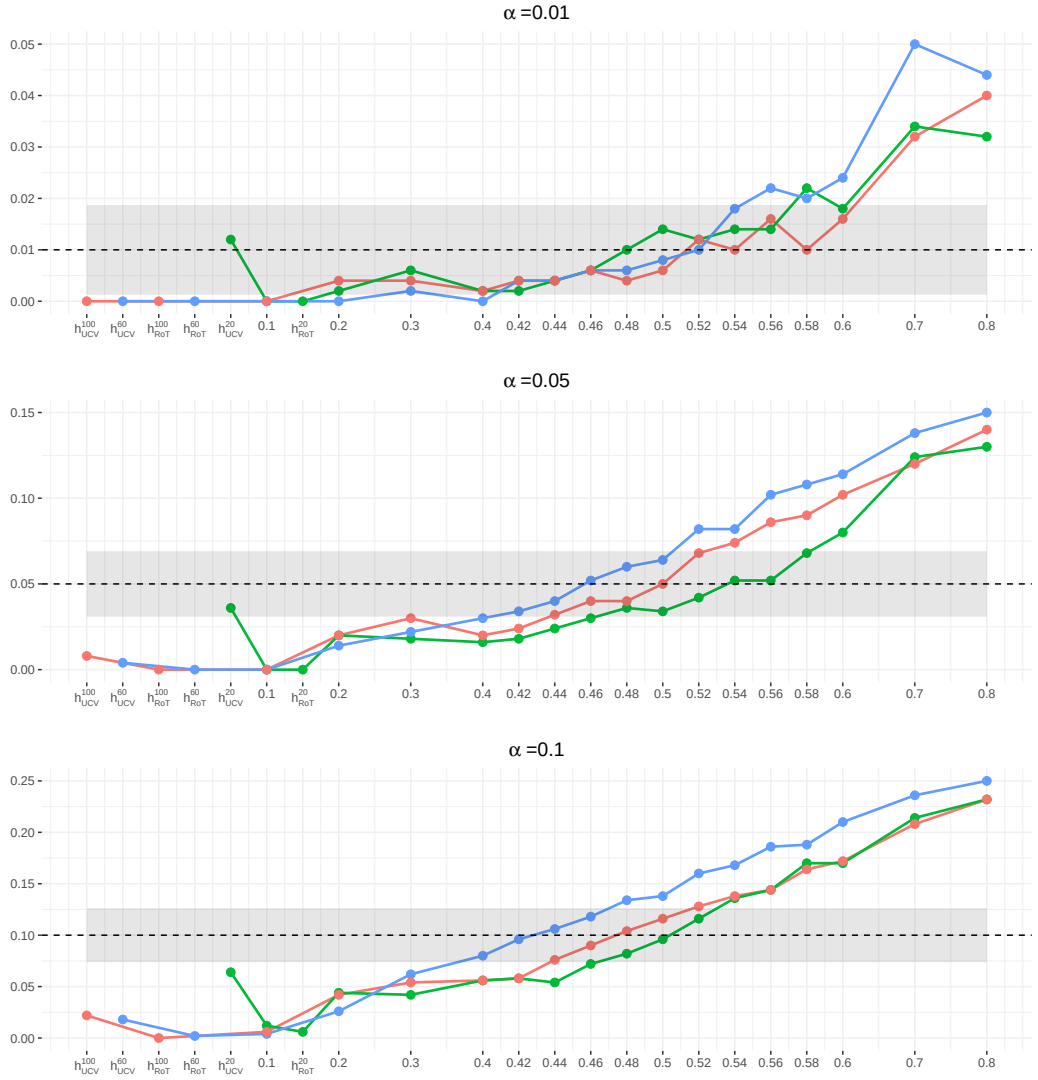


Figure D.8: Percentage of rejections under the null hypothesis (y axis) using permutations for different bandwidth parameters (x axis) taking levels $\alpha = 0.01, 0.05, 0.1$ and $n = 20$ (–●–), $n = 60$ (–●–) and $n = 100$ (–●–) in Scenario A. The bands mark the α confidence intervals at 95%.

Summing up, permutations can be employed to calibrate the CDC-based test introduced in (6.2). This is an alternative totally valid. Nevertheless, this option faces the same problems as the ones commented on for the local bootstrap in Section 6.2. A proper selection of the bandwidth value is tricky in practice, and the "automatic" procedures

as h_{RoT} (employed in Wang et al. (2015)) or h_{UCV} for density estimation have displayed misbehavior. In conclusion, a search through a grid of values in the domain of t is also necessary for this procedure.

Resumo en galego

Nun contexto de alta dimensión, o número de covariables empregadas para explicar unha variable de interese, p, é moi probable que sexa grande, incluso maior co número de mostras dispoñibles ($p > n$). É nesta situación onde os procedementos ordinarios para axustar os modelos de regresión comezan a funcionar mal. Polo tanto, novas alternativas son necesarias. En particular, un primeiro paso de selección de covariables é de interese para considerar unicamente aqueles termos relevantes e reducir a dimensión do problema. Para este fin existen dúas posibilidades. A primeira baséase na consideración dunha certa estrutura no modelo. Baixo esta suposición, as técnicas de penalización son unha alternativa amplamente empregada para estimar o modelo e seleccionar covariables, simultaneamente. Pola contra, se non se quere asumir unha estrutura no modelo, as medidas de dependencia de última xeración baseadas en distancias son unha opción atractiva para a selección de covariables. Estas técnicas son capaces de detectar os termos importantes para calquera estrutura subxacente, pero, pola contra, non se obtén unha estimación do modelo.

O obxectivo desta tese é o estudo e desenrolo de técnicas de selección de covariables para modelos de regresión en contextos actuais de alta dimensión ou de datos funcionais de interese. Para este fin, téñense en conta as dúas vertentes comentadas anteriormente. Comézase motivando a necesidade da utilización de procedementos para a selección de covariables no caso da alta dimensión no Capítulo 1. Neste, expóñense os problemas que teñen que afrontar os modelos de regresión neste contexto. A continuación, no Capítulo 2, realízase unha revisión extensa e crítica das técnicas de penalización para a selección de covariables. Esta desenvólvese para o modelo lineal de alta dimensión no marco vectorial e está centrada no método LASSO. No Capítulo 3 estúdase o funcionamento da regresión LASSO como selector de covariables baixo o suposto de linealidade. En particular, distínguese entre un estudo baixo distintas estruturas de dependencia e un segundo que, ademais, considera covariables en distintas escalas. Os seus resultados son comparados con modificacións e alternativas do mesmo mediante estudos de simulación. Finalmente, extráense conclusións baseadas nos resultados obtidos para cada caso. Así mesmo, a comparativa esténdese a catro bases de datos reais. Seguidamente, preséntanse os coeficientes de correlación baseados en distancias no Capítulo 4. Estes permiten a selección de covariables sen ningunha suposición previa sobre a estrutura do modelo. Isto tradúcese en ferramentas útiles que permiten a selección de covariables en modelos complexos. Particularmente, faise uso da "martingale difference divergence" (MDD) e da "conditional distance covariance" (CDC) no modelo funcional concorrente (MFC). No Capítulo 5, novos tests de especificación baseados na MDD son propostos para a versión síncrona do MFC co fin de seleccionar covariables baixo o suposto de aditividade. Próbase o bo funcionamento do test mediante un estudo de simulación e unha comparativa con competidores existentes na literatura. Ademais, aplícase este procedemento a tres bases de datos reais. Para rematar, trátase a versión asíncrona do MFC no Capítulo 6. Neste caso, proponse un novo test de especificación empregando a CDC. Coméntanse os problemas

que xorden neste novo contexto e como solucionalos. O documento de tese remata cunha sección de resultados, conclusións e traballo futuro comentando os resultado obtidos e posibles liñas de investigación a tratar no posterior.

A continuación, danse máis detalles dos contidos de cada capítulo.

Capítulo 1. Os problemas dos modelos de regresión no contexto da alta dimensión: a necesidade da redución da dimensión

Nun modelo de regresión búscase explicar unha variable de interese ou resposta, Y, mediante o uso de $p \geq 1$ covariables explicativas $X = (X_1, \ldots, X_p)^\top$. Con esa finalidade, asúmese que Y e X veñen relacionadas por unha función regresora $m(\cdot)$, a cal é tipicamente descoñecida. Isto resulta no modelo de regresión dado por

$$Y = m(X) + \varepsilon,$$

onde ε é o erro do modelo, o cal non é directamente observado na práctica. Este erro téndese a asumir que é condicionalmente independente de X en termos de $m(\cdot)$.

Neste capítulo introdúcense os problemas que teñen que enfrontar os modelos de regresión nun contexto de alta dimensión. Estes motivan o contido dos seguintes capítulos do documento da tese.

Primeiro, na Sección 1.1, introdúcense algúns conceptos sobre diferentes estruturas de modelos de regresión que serán empregadas no resto do manuscrito. O modelo de regresión lineal, onde $m(X) = X^\top \beta$ e empregado nos Capítulos 2 e 3, introdúcese na Sección 1.1.1. Posteriormente, a formulación aditiva, dada por $m(X) = \sum_{j=1}^{p} f_j(X_j)$ e que se estudará no Capítulo 5, preséntase na Sección 1.1.2. Finalmente, a regresión local analízase na Sección 1.1.3. Esta asume unha estrutura totalmente xeral para $m(X)$ e emprega técnicas de carácter non paramétrico para a súa estimación. Dita regresión conecta coa formulación xeral coa que se traballa no Capítulo 6. En todos os casos, expóñense os procedementos usuais para a súa estimación. Ademais, en todos estes modelos, arguméntanse os problemas que aparecen no contexto da alta dimensión, centrándonos no caso onde se considera un gran número de covariables, p, ou incluso que $p > n$.

A continuación, todos os problemas comentados anteriormente recóllense na Sección 1.2, onde se explican máis detalladamente as súas consecuencias e implicacións. Podemos clasificalos en tres grupos: a maldición da dimensionalidade (Sección 1.2.1), inconsistencias na estimación do modelo (Sección 1.2.2) e efectos de colinealidade ou concurvidade (Sección 1.2.3). A maldición da dimensionalidade aparece ante valores elevados de p e implica a perda do carácter local e da interpretabilidade dos resultados. Afecta, principalmente, a técnicas non paramétricas como é a regresión local. En termos de aparición de inconsistencias na estimación dos modelos, na Sección 1.2.2 explícase como non é posible estimar os modelos polos procedementos usuais cando $p > n$. Proporciónase unha explicación para cada unha das formulacións propostas. No relativo aos efectos de colinealidade ou concurvidade, estes tradúcense na aparición de malos condicionamentos nas matrices que recollen a información das covariables. Como se explica na Sección 1.2.3, conforme aumenta a dimensión p, tamén

o fan as probabilidades de que se dea algún destes fenómenos. Estes problemas motivan a necesidade de redución da dimensión.

Finalmente, na Sección 1.3, a través dunha análise dos problemas observados, motívase a necesidade de empregar técnicas que permitan reducir a dimensionalidade. Esta discusión da pé ao uso de técnicas de selección de covariables no marco da alta dimensión, motivando o estudo que se desenvolve nos seguintes capítulos do documento. En particular, hai dúas posibles vías: seleccionar covariables tras asumir unha estrutura no modelo, como pode ser o uso de técnicas de penalización, estimando dito modelo ao mesmo tempo, ou seleccionar termos sen supoñer ningunha estrutura, empregando novos coeficientes de dependencia. O primeiro caso, usando técnicas de penalización, trátase nos Capítulos 2 e 3. En relación a segunda opción, no Capítulo 4 analízanse novos coeficientes de dependencia baseados en distancias que serán empregados posteriormente nos Capítulos 5 e 6.

Capítulo 2. O "least absolute shrinkage and selection operator" (LASSO)

Unha vez motivada a necesidade de reducir o número de covariables en alta dimensión no Capítulo 1, e con especial interese no caso de $p > n$, propóñense técnicas de selección de covariables. Neste capítulo, comezamos dando solucións para o caso máis sinxelo: asumindo linealidade nun modelo de regresión vectorial. Isto dá como resultado a formulación vista en (1.2). Neste contexto, na literatura, fíxose un gran esforzo mediante o estudo e implantación de técnicas de regularización. O enfoque máis coñecido e aínda máis empregado é o "least absolute shrinkage and selection operator" ou LASSO, proposto por Tibshirani (1996). Dito procedemento propón a inclusión dunha penalización de tipo L_1 no proceso de estimación, regulada por un parámetro $\lambda > 0$. Este resulta no estimador

$$\hat{\beta}^{LASSO} = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\},$$

Este procedemento introdúcese en detalle na Sección 2.1.

Porén, a regresión LASSO presenta varios requisitos e inconvenientes como selector de variables na práctica. Na Sección 2.2 recóllese unha análise destas características. Estas están relacionadas co nesgo do estimador LASSO (Sección 2.2.1), co feito de que o LASSO require da verificación de fortes condicións teóricas para garantir a súa consistencia (Sección 2.2.2), con que aparecen moitos falsos positivos na súa selección de variables (Sección 2.2.3) e con que é moi complicado facer unha selección óptima do valor da penalización λ na práctica (Sección 2.2.4). En relación o nesgo, o estimador LASSO é sempre nesgado. Este é o prezo a pagar como compensación por poder estimar o modelo cando $p > n$. No caso da consistencia, precísanse que se verifiquen certas condición sobre a matriz de covariables, o vector β e o tamaño mostral. A maioría destas condicións non se poden comprobar na práctica sen coñecer o conxunto real de covariables importantes, o cal dificulta o poder garantir o seu bo comportamento na práctica. Ademais, o LASSO non pode, simultaneamente, reducir o número de falsos positivos e aumentar o de verdadeiros

positivos. Isto tradúcese en que é necesario permitir o LASSO engadir ruído no modelo para garantir que se recupera a maior parte das covariables relevantes. Por último, unha selección adecuado de parámetro λ precisa coñecer a varianza do erro do modelo, o cal tampouco é posible na práctica. Para solucionar este problema, o que se adoita facer é empregar técnicas de validación cruzada para a súa estimación.

Para dar solucións aos problemas que ten que facer fronte o LASSO, na literatura propóñense novas modificacións deste procedemento. Un estudo das mesmas lévase a cabo na Sección 2.3. Unha clasificación modesta destas modificacións, segundo a súa natureza, podería facerse en catro categorías: versións ponderadas (Sección 2.3.1), o LASSO mesturado con remuestreo (Sección 2.3.2), versións empregando "thresholds" (Sección 2.3.3) ou outras alternativas distintas ao enfoque LASSO (Sección 2.4). Ademais, o LASSO tamén presenta versións deseñadas para certas estruturas especiais, como cando se pode establecer un certo orden entre as covariables ou cando o carácter "sparsity" búscase en termos de grupos de covariables. Estas formulacións preséntanse na Sección 2.3.4. Aínda que difiren na súa estrutura, metodoloxía e características, todas as opcións propostas nas seccións citadas perseguen o mesmo obxectivo: a selección de covariables.

A continuación, na Sección 2.5, motívase o estudo de técnicas de selección de covariables cando p é grande con catro bases de datos reais onde este fenómeno acontece. A primeira base de datos é un estudo xénico que busca explicar a produción de riboflavina (Sección 2.5.1). A segunda fai referencia a un estudo de pacientes con cancro de próstata (Sección 2.5.2) e na terceira trátase de modelar a graxa corporal usando distintas medidas fisiolóxicas (Sección 2.5.3). No último caso, estúdase unha base de datos referente o vinho verde de Portugal (Sección 2.5.4). Todas elas son exemplos de bases de datos cun número grande de covariables, onde se observan distintas estruturas de dependencia entre as covariables e que estas están en distintas escalas. Motivando así o estudo que se realiza no Capítulo 3.

Finalmente, realízase unha análise sobre as características, beneficios, desvantaxes e evolución da regresión LASSO, así como do seu impacto, na Sección 2.6. Ademais, tamén se comenta a súa posible extensión a outro tipo de modelos de regresión.

Parte do contido deste capítulo recóllese no traballo Freijeiro-González et al. (2022a).

Capítulo 3. A regresión LASSO como selector de covariables. Comportamento baixo estruturas de dependencia e covariables en diferentes escalas

Como vimos no capítulo anterior a través de catro exemplos, na práctica, é moi común atopar que os datos reais teñen distintas estruturas de dependencia entre as súas covariables e que estas están en distintas escalas. Este feito motiva o estudo levado a cabo nesta parte do documento. Neste Capítulo 3 analízase o funcionamento da regresión LASSO como selector de covariables mediante estudos intensivos de simulación, asumindo distintas estruturas de dependencia e configuracións de escalas para as covariables. Ademais, compárase o seu funcionamento con modificacións e alternativas da mesma.

Na Sección 3.1 comezamos analizando como é o comportamento do LASSO baixo distintos marcos de dependencia onde todas as covariables están en escala unitaria. Para este fin, realízase un amplo estudo de simulación considerando os escenarios presentados na

Sección 3.1.1. En vista dos malos resultados da regresión LASSO (Sección 3.1.2), incluso baixo o caso de independencia, o seu comportamento compárase co de algúns derivados e competidores axeitadamente escollidos (Sección 3.1.3). Como resultado deste estudo, observamos que unha elección adecuada do procedemento a empregar depende do tipo de escenario de dependencia que se estea considerando. Na Sección 3.1.4 analízanse os resultados obtidos e dáse unha orientación facendo uso das mesmas. Toda a Sección 3.1 está recollida en Freijeiro-González et al. (2022a).

A continuación, na Sección 3.2, considéranse, aparte de diferentes estruturas de dependencia, covariables en distintas escalas. Neste caso, compróbase como o LASSO e os seus competidores se comportan nestes contextos. Ademais, desenvólvese unha comparativa entre o caso sen estandarización ou empregando unha estandarización univariante para todos os escenarios considerados. Aquí, empregaremos os escenarios de simulación presentados na Sección 3.2.1. De novo, analízase o comportamento do LASSO (Sección 3.2.2), compárase co dos competidores propostos (Sección 3.2.3) e extráense conclusións en base os resultados observados na Sección 3.2.4. Igual que no caso do estudo de dependencia, non todos os procedemento serán axeitados, senón que dependerá da natureza dos datos.

Posteriormente, na Sección 3.3, analízase o efecto de aplicar un primeiro paso tipo "screening" para reducir a dimensionalidade, considerando distintos coeficientes de dependencia. Este enfoque é tamén un dos máis empregados para a selección de covariables. En particular, testarase o funcionamento do coeficiente de determinación (R^2), da "distance covariance" (DC) e dos "partial least squares" (PLS).

Finalmente, os catro conxuntos de datos reais introducidos na Sección 2.5 do Capítulo 2 que motivan este estudo son analizados tendo en conta as directrices observadas nas anteriores seccións. Esta análise ponse en práctica na Sección 3.4.

Capítulo 4. Novas medidas de dependencia baseadas en distancias para datos complexos

Ata o de agora, mostrouse como realizar selección de covariables nun marco de alta dimensión cando se asume unha estrutura no modelo, como se amosa para a suposición de linealidade nos capítulos 2 e 3. Non obstante, a estrutura do modelo non sempre se pode coñecer de antemán. Polo tanto, teñen especial interese as técnicas de selección de covariables que non precisen de supostos na función regresora. Para este fin, empréganse novas medidas de dependencia baseadas en distancias para construír estatísticos adecuados que permitan realizar tests de significación. En concreto, estas ideas serán empregadas para seleccionar covariables en modelos complexos, onde estimar unha función de regresión o suficientemente flexible é un problema difícil. Estas medidas son modificacións da innovadora "distance covariance" ou DC introducida en Székely et al. (2007). O coeficiente DC permite testar a independencia entre dous vectores aleatorios $X \in \mathbb{R}^p$ e $Y \in \mathbb{R}^q$ formulada como

$$H_0 : X \perp Y \quad \text{vs.} \quad H_1 : X \not\perp Y,$$

e vén dado pola expresión

$$DC^2(X,Y) = \|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)\|^2 = \frac{1}{c_p c_q}\int_{\mathbb{R}^{p+q}} \frac{|\varphi_{X,Y}(t,s) - \varphi_X(t)\varphi_Y(s)|^2}{\|t\|_p^{p+1}\|s\|_q^{q+1}} dt\, ds,$$

onde $\varphi_{X,Y}$ é a función característica conxunta e φ_X, máis φ_Y, son as versións marxinais de X e Y, respectivamente.

A necesidade destas novas métricas de dependencia está motivada na Sección 4.1, revisando os coeficientes existentes para detectar patróns de dependencia e os seus inconvenientes. En particular, as medidas usuais só son capaces de recoller relacións lineais ou monótonas. Este problema soluciónase co uso da DC e derivados, os cales recollen todo tipo de estruturas de dependencia entre dous vectores aleatorios. Ademais, estas novas medidas de dependencia seguen funcionando adecuadamente no caso de ter máis covariables co número de mostras dispoñibles, $p > n$, o cal non acontece cos procedementos clásicos.

Algúns dos coeficientes de distancia resultantes, os cales proban diferentes tipos de dependencia, introdúcense ao longo da Sección 4.2. Estes son a DC (Sección 4.2.1), a "martingale difference divergence" ou MDD (Sección 4.2.2) e a "conditional distance covariance" ou CDC (Sección 4.2.3). Para todos eles revísase a súa formulación, as súas boas propiedades teóricas e a construción de estimadores axeitados para cada caso. Ademais, coméntase o seu uso como selector de covariables, así como a súa adaptación a outros contextos, como é a súa utilización nos contrastes de bondade de axuste.

Finalmente, na Sección 4.3, lévase a cabo unha discusión sobre a súa aplicación e as súas vantaxes en modelos complexos. En particular, coméntase a súa posible extensión a outros contextos como a espazos métricos, ao marco funcional, a regresión cuantil, series de tempo ou modelos de curación, como algúns exemplos. Como comentabamos ao principio, a utilización destes coeficientes permite non asumir ningunha estrutura no modelo de regresión e realizar selección de covariables mediante a aplicación de tests de independencia. Facendo uso destas ideas, estendemos a estrutura lineal considerada nos Capítulos 2 e 3 a modelos máis complexos. En particular, centrámonos no modelo funcional concorrente (MFC) que será estudado nos seguintes capítulos do documentos. No Capítulo 5 proponse un novo procedemento de selección de covariables para a súa versión síncrona, mentres que no Capítulo 6 trabállase co caso asíncrono.

Capítulo 5. Novos tests de significación baseados no coeficiente MDD para a versión síncrona do modelo funcional concorrente

No Capítulo 4 propóñense novos coeficientes de dependencia baseados en distancias para probar a significación das covariables en modelos complexos, sen necesidade de estimación previa da función regresora. Neste capítulo, propoñemos un novo procedemento para deseñar tests de significación para a versión síncrona dun modelo funcional concorrente (MFC) aditivo, facendo uso da MDD introducida no Capítulo 4.

O MFC é un modelo de regresión onde a resposta $Y = (Y_1, \ldots, Y_q) \in \mathbb{R}^q$ e as covariables $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$, con $q, p \geq 1$, son todas funcións dun mesmo argumento $t \in \mathcal{D}$, e a

relación é concorrente, simultánea ou punto a punto. Este modelo resulta en

$$Y(t) = m(t, X(t)) + \varepsilon(t),$$

onde $m(\cdot)$ é a función regresora e $\varepsilon(t)$ é o erro do modelo. Agora, o erro é un proceso que se asume de media nula, independente de X e con función de covarianza $\Omega(s,t) = \mathbb{C}\left[\varepsilon(s), \varepsilon(t)\right]$, onde $\mathbb{C}[\cdot,\cdot]$ é o operador de covarianza.

A versión síncrona do MFC asume que todas as curvas son observadas nos mesmos instantes temporais. Mais información sobre dito modelo atópase na Sección 5.1, xunto cunha motivación da necesidade dun primeiro paso de selección de variables para reducir a dimensión do problema.

O noso novo procedemento propón unha forma innovadora de seleccionar covariables na versión síncrona dun MFC aditivo. Este baséase no uso dunha versión innesgada do coeficiente MDD, que se presenta na Sección 5.2. En particular, a selección de covariables faise mediante tests de significación que son reescritas como tests de independencia usando a MDD. Desta forma, na Sección 5.3, propóñense os novos tests de dependencia. Dáse unha xustificación teórica do seu bo funcionamento e tamén se propón un esquema de remuestreo para calcular os p-valores na práctica. Este procedemento conta coas vantaxes de que non é necesario estimar a función regresora para seleccionar os termos importantes e non é preciso empregar ningún tipo de "tuning parameter". Estas características contrastan cos métodos que existen na literatura actual.

Un estudo de simulación lévase a cabo na Sección 5.4 para verificar o seu bo funcionamento, xunto cunha comparativa con dous competidores, propostos e estudados por Ghosal and Maity (2022a) e Kim et al. (2018). Demostrando así, que o noso procedemento é moi competitivo. Logo, os novos tests propostos aplícanse a tres conxuntos de datos reais na Sección 5.5. Estes datos relaciónanse cun estudo da marcha en nenos con problemas locomotores (Sección 5.5.1), o avance da gripe en Estados Unidos (Sección 5.5.2) e o modelaxe do alugueiro casual de bicicletas na cidade de Washington D.C. explicado mediante as condicións climatolóxicas (Sección 5.5.3). Para rematar, desenvólvese unha discusión dos resultados na Sección 5.6.

Os contidos deste capítulo están recollidos en Freijeiro-González et al. (2022b).

Capítulo 6. Novos tests de significación baseados no coeficiente CDC para a versión asíncrona do modelo funcional concorrente

No capítulo 5 propuxéronse novos test de significación para a versión síncrona do MFC aditivo. Non obstante, estes só funcionan no caso de considerar observacións temporais síncronas. Neste capítulo propoñemos novas ideas de técnicas de selección de covariables para a versión xeral asíncrona do MFC. É dicir, agora permítese que as curvas sexan observadas en distintos instantes temporais e que, ademais, o número de observacións varíe ducha curva a outra. Unha introdución do modelo MFC dáse na Sección 6.1. Tamén flexibilizamos os supostos sobre a función regresora, permitindo que teña calquera estrutura e non nos restrinximos ao caso dos efectos aditivos. Estas ideas para aplicar selección de

variables lévanse a cabo mediante tests de independencia condicionada, facendo uso do coeficiente CDC, introducido anteriormente na Sección 4.2.3.

Na Sección 6.2, introdúcense as novas ideas para os tests de significación. Similar ao Capítulo 5, estes tests de significación son reescritos como tests de independencia condicional empregando o coeficiente da CDC. Agora, é necesario recorrer a técnicas de carácter non paramétrico para empregar a información local proporcionada polos datos achegados, xa que é de esperar que non se dispoña de moitas observacións para un instante dado. Desta forma, a estimación da CDC require dunha elección adecuada dun parámetro de suavizado e dunha función núcleo. Isto discútese nas Seccións 6.2.1 e 6.2.2. Unha vez formulado o test, xustifícase o seu bo comportamento teórico e tamén se propón un esquema de remuestreo local para calibralo na práctica. Posteriormente, desenvólvese un estudo de simulación na Sección 6.3 para comprobar o seu bo funcionamento. Finalmente, as conclusións obtidas coméntase na Sección 6.4.

Apéndices A, B, C e D

Os apéndices A, B, C e D conteñen resultados suplementarios e demostracións en relación ao contido da Sección 3.1 do capítulo 3, da Sección 3.2 do Capítulo 3, do Capítulo 5 e do Capítulo 6, respectivamente.

Further information

Objectives

The main objective of this thesis project is the study and development of covariates selection techniques for regression models in recent high dimensional or functional data contexts. This project is carried out in several statements essential, in consideration of the proposed objectives:

a) Review of covariates selection techniques for linear models in the high dimensional vectorial framework. Here, it is assumed that the number of covariates can be equal or greater than the available sample size. Intensive analysis of usual techniques as the L1 or LASSO penalization (Tibshirani (1996)), the SCAD penalty (Fan (1997)), the Dantzig selector (Candes and Tao (2007)) and more innovative alternatives, like the adaptive LASSO (Zou (2006), Huang et al. (2008)) or the use of distance covariance or DC coefficient (Székely et al. (2007)) for variable selection (Febrero-Bande et al. (2019)), among others. Study of their behavior under different dependence structures and standardization techniques. Guidelines about what can be expected from every algorithm and what is the best option for each considered scenario, paying attention to the data nature. Application to some real data sets.

b) Analysis of existing covariates selection procedures in functional concurrent models. Development of new significance tests focused on the selection of covariates using derivations of the DC coefficient. For this aim, state-of-the-art dependence measures, such as the martingale difference divergence or MDD (Shao and Zhang (2014)) for synchronous observations or the conditional distance covariance or CDC (Wang et al. (2015)) for asynchronous design, are employed. In the first context, an additive structure is assumed for the regression model, which provides quite a flexibility. Conversely, in the asynchronous case, a totally general formulation is taken into consideration. In both contexts, it is not necessary to know in advance or estimate the actual form of the regressor function. This fact is a new advantage compared to the existing literature. Besides, smoothing parameters are avoided in the synchronous context employing the MDD techniques. A novel and specific alternative using the CDC coefficient is proposed concerning the asynchronous framework. Validation of the proposed new tests through simulation studies, respectively. In addition, comparison with existing competitors and application to some real data sets for the synchronous case.

c) Collection of algorithms studied and programmed in a public repository, jointly with the real data sets analyzed.

Block a) is carried out along Chapters 2 and 3. The review and dependence study portion has resulted in the published article Freijeiro-González et al. (2022a). Chapter 4 is

devoted to a preliminary review of distance-based correlation coefficients for implementing specification tests on functional concurrent models. This study leads to block b). Some of this information, jointly with the novel significance test for the synchronous functional concurrent model carried out in Chapter 5, resulted in a new paper. This work is submitted to the TEST journal and has passed the first review. A preliminary version is Freijeiro-González et al. (2022b). Eventually, Chapter 4, along with Chapter 6, will complete block b) with the development of new significance tests for the asynchronous version of the functional concurrent model. Eventually, block c) is covered by the development of a public repository in GitHub. We refer below for more details about articles and journals, jointly with the simulations code.

Methodology

This thesis follows the classical research methodology in the field of statistics. In the first place, an in-depth study of the topic of interest is performed, carrying out an extensive review of the existing literature. Next, new points of view or the implementation of novel covariates selection techniques are proposed. A theoretical and practical analysis is performed, analyzing the advantages and drawbacks of all procedures in each case. Eventually, some conclusions arise based on their study.

The R software (R Core Team (2019)), jointly with C++ code, has been employed for all simulations and novel procedures implementation.

Some details on the methodology are given below in terms of the chapters. Chapter 1 is introductory, so this is excluded.

Chapter 2: The least absolute shrinkage and selection operator (LASSO).

- An exhaustive bibliographical review of the LASSO algorithm as a variable selector for linear regression models.
- Introduction of the LASSO requirements and inconveniences as a variable selector from a theoretical point of view. Analysis of the required conditions and drawbacks implications in practice.
- Review of the existing LASSO derivatives. Comparison with widely employed or innovative competitive procedures. Analysis of their advantages and drawbacks.
- Introduction of some examples of real data sets that motivate the use of LASSO.

Chapter 3: LASSO regression as a variable selector. Performance under dependence structures and different scales on covariates.

- Motivation of the LASSO problems under dependence structures. Intensive simulation study to compare LASSO and competitors' performance. Some guidance about the best options under dependence.

- Problems of the LASSO regression facing covariates with different scales. Analysis under different dependence frameworks by means of an extensive simulation study. Conclusions about a proper recovery for these contexts.
- Study of the effect of a first screening step to reduce the problem dimensionality. Application to dependence contexts with covariates having different scales. Advantages and limitations observed in simulation analysis.
- Application to real data. Analysis of covariates selection techniques applied over some real data sets with dependence structures and covariates with different scales.

Chapter 4: Novel distance-based dependence measures for complex data.

- Analysis of the problems of the classical correlation coefficients and motivation of the advantages of the new distance-based ones.
- An exhaustive bibliographical review of novel distance-based measures. Examples of their use in covariates selection procedures.
- Application of these different dependence measures in complex data. Advantages of this approach over other covariates selection procedures.

Chapter 5: New significance tests for the synchronous functional concurrent model based on the martingale difference divergence coefficient.

- Introduction of the synchronous functional concurrent model (FCM) and the importance of covariates selection.
- Development of new significance tests for the synchronous FCM based on the martingale difference divergence coefficient. Study of their performance by means of a simulation study.
- Comparison of the new approach with existing competitors in literature. Display of its advantages.
- Application in real data sets.

Chapter 6: New significance tests for the asynchronous functional concurrent model based on the conditional distance covariance coefficient.

- Introduction to the asynchronous version of the FCM.
- Development of new global significance tests for the asynchronous FCM using the conditional distance covariance coefficient. Study of their performance by means of a simulation study.

Simulations code

All simulation code developed in this document has been collected in the public GitHub repository `https://github.com/LauraFreiG/Covariates_selection.git` along with the

real data sets examples. Thus, all the results are reproducible. Then, the code can be employed or modified in case of interest.

In particular, code related to Sections 3.1, 3.2 and real data sets examples introduced in Section 2.5, and studied later in Section 3.4, are collected in the "Linear Regression" folder. This covers Chapter 3 and contains the code for Freijeiro-González et al. (2022a).

Results for the concurrent model formulation are collected in the folder "Functional Concurrent Model". Inside this, the code of Chapter 5 is summarized in the folder "Synchronous FCM". This code is the one employed in Freijeiro-González et al. (2022b). Simulations corresponding with Chapter 6 will be added once a preprint will be available.

Articles and journals

A Critical Review of LASSO and Its Derivatives for Variable Selection Under Dependence Among Covariates (Freijeiro-González et al. (2022a))

TITLE: A Critical Review of LASSO and Its Derivatives for Variable Selection Under Dependence Among Covariates.

AUTHORS: L. Freijeiro-González[1], M. Febrero-Bande[1] and W. González-Manteiga[1].

YEAR: 2022.

AFFILIATIONS: [1]Departamento de Estatística, Análise Matemática e Optimización, Universidade de Santiago de Compostela.

JOURNAL: International Statistical Review.

STATUS: Published.

ISSN: 0306-7734.

PUBLISHER: John Wiley & Sons, Ltd.

LINK: `https://onlinelibrary.wiley.com/doi/full/10.1111/insr.12469`.

JCR IMPACT FACTOR: 1.946 (Q2 in Statistics and Probability (47/125)).

COPYRIGHT AND USE: Related to permission, the Wiley publishing company claims that *"If you are the author of a published Wiley article, you have the right to reuse the full text of your published article as part of your thesis or dissertation. In this situation, you do not need to request permission from Wiley for this use"*. This exempts the author to ask for permission. This information can be found in Wiley.com.

CONTENTS OF THE THESIS BASED ON THIS ARTICLE: Chapter 2 and the first part of Chapter 3.

CONTRIBUTIONS OF THE PHD CANDIDATE: The candidate contributed to the conceptualization of the works, methodology, software implementation, formal analysis of the results, and the manuscript preparation.

Novel specification tests for additive concurrent model formulation based on martingale difference divergence (Freijeiro-González et al. (2022b))

TITLE: Novel specification tests for additive concurrent model formulation based on martingale difference divergence.

AUTHORS: L. Freijeiro-González[1,2], M. Febrero-Bande[1,2] and W. González-Manteiga[1,2].

YEAR: 2022.

AFFILIATIONS: [1]Centro de Investigación y Tecnología Matemática de Galicia (CITMAga); [2]Departamento de Estatística, Análise Matemática e Optimización, Universidade de Santiago de Compostela.

STATUS: Submitted manuscript to the TEST journal (`https://www.springer.com/journal/11749`). This has passed the first review process.

LINK: A preliminary version can be found in the arXiv repository `https://arxiv.org/abs/2208.00701`.

COPYRIGHT AND USE: This repository claims that *"If you are the copyright holder of the work, you do not need arXiv's permission to reuse the full text"*. This information can be found in arxiv.org.

CONTENTS OF THE THESIS BASED ON THIS ARTICLE: part of Chapter 4 and complete Chapter 5.

CONTRIBUTIONS OF THE PHD CANDIDATE: The candidate contributed to the conceptualization of the works, methodology, proofs of the theoretical results, software implementation, formal analysis of the results, and the manuscript preparation.

Funding information

This research has been supported by the Consellería de Cultura, Educación e Ordenación Universitaria along with the Conselleríaa de Economía, Emprego e Industria of the Xunta de Galicia (project ED481A-2018/264). This work has also been supported by Project PID2020-116587GB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe" and the Competitive Reference Groups 2021–2024 (ED431C 2021/24) from the Xunta de Galicia through the ERDF. The Supercomputing Center of Galicia (CESGA) is also acknowledged for providing computational resources.

Bibliography

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer.

Aneiros, G., Novo, S., and Vieu, P. (2022). Variable selection in functional regression models: A review. *Journal of Multivariate Analysis*, 188:104871. 50th Anniversary Jubilee Edition.

Bach, F. R. (2008). Bolasso: model consistent Lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine Learning*, pages 33–40. ACM.

Barber, R. F. and Candès, E. J. (2019). A knockoff filter for high-dimensional selective inference. *The Annals of Statistics*, 47(5):2504–2537.

Barber, R. F. and Candès, E. J. (2015). Controlling the false discovery rate via knockoffs. *The Annals of Statistics*, 43(5):2055–2085.

Beale, E. M. L., Kendall, M. G., and Mann, D. W. (1967). The discarding of variables in multivariate analysis. *Biometrika*, 54(3-4):357–366.

Belloni, A. and Chernozhukov, V. (2013). Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521 – 547.

Belloni, A., Chernozhukov, V., and Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806.

Belloni, A., Chernozhukov, V., and Wang, L. (2014). Pivotal estimation via square-root lasso in nonparametric regression. *The Annals of Statistics*, 42.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300.

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *Ann. Stat.*, 29.

Bergersen, L. C., Glad, I. K., and Lyng, H. (2011). Weighted lasso with data integration. *Statistical Applications in Genetics and Molecular Biology*, 10(1).

Bhadra, A., Datta, J., Polson, N. G., and Willard, B. (2019). Lasso meets horseshoe: A survey. *Statistical Science*, 34(3):405–427.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009). Simultaneous analysis of LASSO and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732.

Bogdan, M., Van Den Berg, E., Sabatti, C., Su, W., and Candès, E. J. (2015). SLOPE-adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103.

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232–253.

Bühlmann, P. et al. (2013). Statistical significance in high-dimensional linear models. *Bernoulli*, 19(4):1212–1242.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media.

Bunea, F. (2008). Honest variable selection in linear and logistic regression models via l1 and l1 + l2 penalization. *Electronic Journal of Statistics*, 2.

Bunea, F., Tsybakov, A. B., and Wegkamp, M. H. (2007). Aggregation for gaussian regression. *The Annals of Statistics*, 35(4):1674–1697.

Bühlmann, P., Kalisch, M., and Meier, L. (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1(1):255–278.

Candes, E., Fan, Y., Janson, L., and Lv, J. (2018). Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577.

Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351.

Candes, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223.

Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986 – 2018.

Chaudhuri, A. and Hu, W. (2019). A fast algorithm for computing distance correlation. *Computational Statistics and Data Analysis*, 135:15–24.

Chen, L.-P. (2021). Feature screening based on distance correlation for ultrahigh-dimensional censored data with covariate measurement error. *Computational Statistics*, 36(2):857–884.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553. Smart Business Networks: Concepts and Empirical Evidence.

Cuesta-Albertos, J. A., García-Portugués, E., Febrero-Bande, M., and González-Manteiga, W. (2019). Goodness-of-fit tests for the functional linear model based on randomly projected empirical processes. *The Annals of Statistics*, 47(1):439–467.

Dalalyan, A. S., Hebiri, M., and Lederer, J. (2017). On the prediction performance of the Lasso. *Bernoulli*, 23(1):552 – 581.

Davis, R. A., Matsui, M., Mikosch, T., and Wan, P. (2018). Applications of distance correlation to time series. *Bernoulli*, 24(4A):3087 – 3116.

Dehling, H., Matsui, M., Mikosch, T., Samorodnitsky, G., and Tafakori, L. (2020). Distance covariance for discretized stochastic processes. *Bernoulli*, 26(4):2758 – 2789.

Descloux, P. and Sardy, S. (2021). Model Selection With Lasso-Zero: Adding Straw to the Haystack to Better Find Needles. *Journal of Computational and Graphical Statistics*, 30(3):530–543.

Dezeure, R., Bühlmann, P., Meier, L., and Meinshausen, N. (2015). High-dimensional inference: Confidence intervals, p-values and R-software hdi. *Statistical Science*, 30(4):533–558.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2005). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18.

Dunn, O. J. (1958). Estimation of the means of dependent variables. *The Annals of Mathematical Statistics*, pages 1095–1111.

Edelmann, D., Fokianos, K., and Pitsillou, M. (2019). An Updated Literature Review of Distance Correlation and Its Applications to Time Series. *International Statistical Review*, 87(2):237–262.

Edelmann, D. and Goeman, J. (2022). A Regression Perspective on Generalized Distance Covariance and the Hilbert–Schmidt Independence Criterion. *Statistical Science*, 37(4):562 – 579.

Edelmann, D., Welchowski, T., and Benner, A. (2022). A consistent version of distance covariance for right-censored survival data and its application in hypothesis testing. *Biometrics*, 78(3):867–879.

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

Fan, J. (1997). Comments on «wavelets in statistics: A review» by A. Antoniadis. *Journal of the Italian Statistical Society*, 6(2):131.

Fan, J., Guo, S., and Hao, N. (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):37–65.

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.

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

Fan, J., Peng, H., et al. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961.

Fanaee-T, H. and Gama, J. (2014). Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127.

Febrero-Bande, M., González-Manteiga, W., and Oviedo de la Fuente, M. (2019). Variable selection in functional additive regression models. *Computational Statistics*, 34(2):469–487.

Febrero-Bande, M. and Oviedo de la Fuente, M. (2012). Statistical computing in functional data analysis: The R package fda.usc. *Journal of Statistical Software*, 51(4):1–28.

Freijeiro-González, L., Febrero-Bande, M., and González-Manteiga, W. (2022a). A Critical Review of LASSO and Its Derivatives for Variable Selection Under Dependence Among Covariates. *International Statistical Review*, 90(1):118–145.

Freijeiro-González, L., Febrero-Bande, M., and González-Manteiga, W. (2022b). Novel specification tests for additive concurrent model formulation based on martingale difference divergence. https://arxiv.org/abs/2208.00701. Submitted manuscript.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Fu, W. J. (1998). Penalized Regressions: The Bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416.

Ghosal, R. and Maity, A. (2022a). A score based test for functional linear concurrent regression. *Econometrics and Statistics*, 21:114–130.

Ghosal, R. and Maity, A. (2022b). Variable selection in nonparametric functional concurrent regression. *Canadian Journal of Statistics*, 50(1):142–161.

Ghosal, R., Maity, A., Clark, T., and Longo, S. B. (2020). Variable selection in functional linear concurrent regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 69(3):565–587.

Giacobino, C., Sardy, S., Diaz-Rodriguez, J., Hengartner, N., et al. (2017). Quantile universal threshold. *Electronic Journal of Statistics*, 11(2):4701–4722.

Giraud, C. (2014). *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC.

Giraud, C., Huet, S., Verzelen, N., et al. (2012). High-dimensional regression with unknown variance. *Statistical Science*, 27(4):500–518.

Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Di, C., Gellar, J., Harezlak, J., McLean, M. W., Swihart, B., Xiao, L., Crainiceanu, C., and Reiss, P. T. (2021). *refund: Regression with Functional Data*. R package version 0.1-24.

Goldsmith, J. and Schwartz, J. E. (2017). Variable selection in the functional linear concurrent model. *Statistics in medicine*, 36(14):2237–2250.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. In Jain, S., Simon, H. U., and Tomita, E., editors, *Algorithmic Learning Theory*, pages 63–77, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hall, P. and Heyde, C. C. (1980). *Martingale limit theory and its application*. Academic press.

Haris, A., Simon, N., and Shojaie, A. (2022). Generalized sparse additive models. *Journal of Machine Learning Research*, 23(70):1–56.

Hastie, T. and Efron, B. (2013). *lars: Least Angle Regression, Lasso and Forward Stagewise*. R package version 1.2.

Hastie, T. and Tibshirani, R. (1990). *Generalized additive models*. Wiley Online Library.

Hastie, T. and Tibshirani, R. (1993). Varying-coefficient models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(4):757–796.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Tibshirani, R. J. (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC press.

Hocking, R. R. and Leslie, R. N. (1967). Selection of the best subset in regression analysis. *Technometrics*, 9(4):531–540.

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

Hofner, B., Boccuto, L., and Göker, M. (2015). Controlling false discoveries in high-dimensional situations: Boosting with stability selection. *BMC Bioinformatics*, 16:144.

Homrighausen, D. and McDonald, D. J. (2018). A study on tuning parameter selection for the high-dimensional lasso. *Journal of Statistical Computation and Simulation*, 88(15):2865–2892.

Hu, W., Lin, N., and Zhang, B. (2020). Nonparametric testing of lack of dependence in functional linear models. *PLOS ONE*, 15(6):1–24.

Hua, W.-Y. and Ghosh, D. (2015). Equivalence of kernel machine regression and kernel distance covariance for multidimensional phenotype association studies. *Biometrics*, 71(3):812–820.

Huang, J., Ma, S., and Zhang, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618.

Huo, X. and Székely, G. J. (2016). Fast computing for distance covariance. *Technometrics*, 58(4):435–447.

Jansen, S. (2021). On distance covariance in metric and Hilbert spaces. *ALEA*, 18:1353–1393.

Javanmard, A., Javadi, H., et al. (2019). False discovery rate control via debiased lasso. *Electronic Journal of Statistics*, 13(1):1212–1253.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909.

Javanmard, A. and Montanari, A. (2018). Debiasing the lasso: Optimal sample size for Gaussian designs. *The Annals of Statistics*, 46(6A):2593–2622.

Jiang, B., Ding, C., and Luo, B. (2014). Covariate-correlated lasso for feature selection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 595–606. Springer.

Jiang, C.-R., Wang, J.-L., et al. (2011). Functional single index models for longitudinal data. *The Annals of Statistics*, 39(1):362–388.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Kim, J., Maity, A., and Staicu, A.-M. (2018). Additive nonlinear functional concurrent model. *Statistics and its Interface*, 11:669–685.

Lahiri, S. N. (2021). Necessary and sufficient conditions for variable selection consistency of the LASSO in high dimensions. *The Annals of Statistics*, 49(2):820 – 844.

Lai, T., Zhang, Z., and Wang, Y. (2020). Testing independence and goodness-of-fit jointly for functional linear models. *Journal of the Korean Statistical Society*, 50.

Lee, C. E. and Shao, X. (2018). Martingale Difference Divergence Matrix and Its Application to Dimension Reduction for Stationary Multivariate Time Series. *Journal of the American Statistical Association*, 113(521):216–229.

Lee, C. E., Zhang, X., and Shao, X. (2020). Testing conditional mean independence for functional data. *Biometrika*, 107(2):331–346.

Lee, E. R., Mammen, E., et al. (2016). Local linear smoothing for sparse high dimensional varying coefficient models. *Electronic Journal of Statistics*, 10(1):855–894.

Leng, C., Lin, Y., and Wahba, G. (2006). A note on the Lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273–1284.

Li, R., Zhong, W., and Zhu, L. (2012). Feature screening via distance correlation learning. *Journal of the American Statistical Association*, 107(499):1129–1139.

Li, X., Zhao, T., Wang, L., Yuan, X., and Liu, H. (2019). *flare: Family of Lasso Regression*. R package version 1.6.0.2.

Liu, J., Li, R., and Wu, R. (2014). Feature selection for varying coefficient models with ultrahigh-dimensional covariates. *Journal of the American Statistical Association*, 109(505):266–274.

Lounici, K. (2008). Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electronic Journal of Statistics*, 2.

Lu, J. and Lin, L. (2020). Model-free conditional screening via conditional distance correlation. *Statistical Papers*, 61:225–244.

Lyons, R. (2013). Distance covariance in metric spaces. *The Annals of Probability*, 41(5):3284–3305.

Maity, A. (2017). Nonparametric functional concurrent regression models. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(2):e1394.

McCullagh, P. and Nelder, J. A. (2019). *Generalized linear models*. London: Chapman and Hall, 2 edition.

Meinshausen, N. (2007). Relaxed Lasso. *Computational Statistics & Data Analysis*, 52(1):374–393.

Meinshausen, N. (2012). *relaxo: Relaxed Lasso*. R package version 0.1-2.

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the LASSO. *The Annals of Statistics*, 34(3):1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Meinshausen, N. and Yu, B. (2009). Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246–270.

Miller, A. (2002). *Subset Selection in Regression*. Chapman and Hall/CRC, 2nd edition edition.

Nan, Y. and Yang, Y. (2014). Variable selection diagnostics measures for high-dimensional regression. *Journal of Computational and Graphical Statistics*, 23(3):636–656.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234.

Olshen, R. A., Biden, E. N., Wyatt, M. P., and Sutherland, D. H. (1989). Gait analysis and the bootstrap. *The Annals of Statistics*, 17(4):1419–1440.

Ospína-Galindez, J., Giraldo, R., and Andrade-Bejarano, M. (2019). Functional regression concurrent model with spatially correlated errors: application to rainfall ground validation. *Journal of Applied Statistics*, 46(8):1350–1363.

Paparoditis, E. and Politis, D. (2000). The local bootstrap for kernel estimators under general dependence conditions. *Annals of the Institute of Statistical Mathematics*, 52:139–159.

Park, T., Shao, X., and Yao, S. (2015). Partial martingale difference correlation. *Electronic Journal of Statistics*, 9(1):1492 – 1517.

Pearson, K. (1920). Notes on the history of correlation. *Biometrika*, 13(1):25–45.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ramsay, J. O., Graves, S., and Hooker, G. (2020). *fda: Functional Data Analysis*. R package version 5.1.9.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer, New York.

Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030.

Reid, S., Tibshirani, R., and Friedman, J. (2016). A study of error variance estimation in Lasso regression. *Statistica Sinica*, 26:35–67.

Ročková, V. and George, E. I. (2018). The Spike-and-Slab LASSO. *Journal of the American Statistical Association*, 113(521):431–444.

Saligrama, V. and Zhao, M. (2011). Thresholded basis pursuit: LP algorithm for order-wise optimal support recovery for sparse and approximately sparse signals from noisy random measurements. *IEEE Transactions on Information Theory*, 57(3):1567–1586.

Schick, A. (1997). On U-statistics with random kernels. *Statistics & Probability Letters*, 34(3):275–283.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. (2013). Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263 – 2291.

Sen, A. and Sen, B. (2014). Testing independence and goodness-of-fit in linear models. *Biometrika*, 101(4):927–942.

Serfling, R. J. (1980). *Approximation theorems of mathematical statistics*. John Wiley & Sons.

Shao, X. and Zhang, J. (2014). Martingale difference correlation and its use in high-dimensional variable screening. *Journal of the American Statistical Association*, 109(507):1302–1318.

Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. (2013). A sparse-group Lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245.

Siri, W. E. (1956). The gross composition of the body. In *Advances in Biological and Medical Physics*, volume 4, pages 239–280. Elsevier.

Song, F., Chen, Y., and Lai, P. (2020). Conditional distance correlation screening for sparse ultrahigh-dimensional models. *Applied Mathematical Modelling*, 81:232–252.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. (2012). Feature selection via dependence maximization. *J. Mach. Learn. Res.*, 13(1):1393–1434.

Städler, N., Bühlmann, P., and Van De Geer, S. (2010). l_1-penalization for mixture regression models. *TEST*, 19(2):209–256.

Stamey, T. A., Kabalin, J. N., and Ferrari, M. (1989). Prostate Specific Antigen in the Diagnosis and Treatment of Adenocarcinoma of the Prostate. II. Radiation Treated Patients. *The Journal of Urology*, 141(5):1084–1087.

Su, L. and Zheng, X. (2017). A martingale-difference-divergence-based test for specification. *Economics Letters*, 156:162–167.

Su, W., Bogdan, M., and Candès, E. (2017). False discoveries occur early on the Lasso path. *The Annals of statistics*, 45(5):2133–2150.

Sun, T. (2019). *scalreg: Scaled Sparse Linear Regression*. R package version 1.0.1.

Sun, T. and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika*, 99(4):879–898.

Sun, T. and Zhang, C.-H. (2013). Sparse matrix inversion with scaled Lasso. *The Journal of Machine Learning Research*, 14(1):3385–3418.

Székely, G. J. and Rizzo, M. L. (2014). Partial distance correlation with methods for dissimilarities. *The Annals of Statistics*, 42(6):2382 – 2412.

Szekely, G. J. and Rizzo, M. L. (2017). The energy of data. *Annual Review of Statistics and Its Application*, 4:447–479.

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.

Székely, G. J. and Rizzo, M. L. (2013). The distance correlation t-test of independence in high dimension. *Journal of Multivariate Analysis*, 117:193–213.

Tardivel, P. J. and Bogdan, M. (2022). On the sign recovery by least absolute shrinkage and selection operator, thresholded least absolute shrinkage and selection operator, and thresholded basis pursuit denoising. *Scand. J. Stat.*, 49(4):1636–1668.

Teran Hidalgo, S., Wu, M., Engel, S., and Kosorok, M. (2018). Goodness-Of-Fit Test for Nonparametric Regression Models: Smoothing Spline ANOVA Models as Example. *Computational Statistics & Data Analysis*, 122.

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. (2011). Regression shrinkage and selection via the LASSO: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282.

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. (2013). The lasso problem and uniqueness. *Electronic Journal of statistics*, 7:1456–1490.

Tibshirani, R. J. and Efron, B. (1993). An introduction to the bootstrap. *Monographs on Statistics and Applied Probability*, 57:1–436.

Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R., et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202.

Van de Geer, S., Bühlmann, P., and Zhou, S. (2011). The adaptive and the thresholded Lasso for potentially misspecified models (and a lower bound for the Lasso). *Electronic Journal of Statistics*, 5:688–749.

Van De Geer, S. A. and Bühlmann, P. (2009). On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392.

Vidaurre, D., Bielza, C., and Larrañaga, P. (2012). Lazy lasso for local regression. *Computational Statistics*, 27(3):531–550.

Vidaurre, D., Bielza, C., and Larranaga, P. (2013). A Survey of l_1 Regression. *International Statistical Review*, 81.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using l_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202.

Wang, H., Zhong, P.-S., Cui, Y., and Li, Y. (2017). Unified empirical likelihood ratio tests for functional concurrent linear models and the phase transition from sparse to dense functional data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80.

Wang, L. (2013). The l_1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151.

Wang, S., Nan, B., Rosset, S., and Zhu, J. (2011). Random lasso. *The Annals of Applied Statistics*, 5(1):468.

Wang, X., Pan, W., Hu, W., Tian, Y., and Zhang, H. (2015). Conditional distance correlation. *Journal of the American Statistical Association*, 110(512):1726–1734.

Wasserman, L. and Roeder, K. (2009). High dimensional variable selection. *The Annals of Statistics*, 37(5A):2178.

Weinstein, A., Barber, R., and Candès, E. (2017). A power and prediction analysis for knockoffs with lasso statistics. *arXiv preprint arXiv:1712.06465*.

Wissler, C. (1905). The spearman correlation formula. *Science*, 22(558):309–311.

Xu, K. and Chen, F. (2020). Martingale-difference-divergence-based tests for goodness-of-fit in quantile models. *Journal of Statistical Planning and Inference*, 207:138–154.

Xu, K. and He, D. (2021). Omnibus model checks of linear assumptions through distance covariance. *Statistica Sinica*, 31:1055–1079.

Xue, L. and Zhu, L. (2007). Empirical likelihood for a varying coefficient model with longitudinal data. *Journal of the American Statistical Association*, 102(478):642–654.

Yang, E., Lozano, A., and Ravikumar, P. (2014). Elementary estimators for high-dimensional linear regression. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 388–396, Bejing, China. PMLR.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation. *Biometrika*, 92:937–950.

Yao, F., Müller, H., and Wang, J. (2005). Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590.

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.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35.

Zhang, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942.

Zhang, J., Clayton, M., and Townsend, P. (2011). Functional concurrent linear regression model for spatial images. *Journal of Agricultural, Biological, and Environmental Statistics*, 16:105–130.

Zhang, J., L. Y. and Cui, H. (2021). Model-free feature screening via distance correlation for ultrahigh dimensional survival data. *Statistical Papers*, 62(6):2711–2738.

Zhang, X., Yao, S., and Shao, X. (2018). Conditional mean and quantile dependence testing in high dimension. *The Annals of Statistics*, 46(1):219 – 246.

Zhao, F., Lin, N., Hu, W., and Zhang, B. (2022). A faster U-statistic for testing independence in the functional linear models. *Journal of Statistical Planning and Inference*, 217:188–203.

Zhao, P. and Yu, B. (2006). On model selection consistency of LASSO. *Journal of Machine Learning Research*, 7:2541–2563.

Zhou, S. (2010). Thresholded Lasso for high dimensional variable selection and statistical estimation. *arXiv preprint arXiv:1002.1583*.

Zhu, C., Zhang, X., Yao, S., and Shao, X. (2020). Distance-based and RKHS-based dependence metrics in high dimension. *The Annals of Statistics*, 48(6):3366 – 3394.

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.

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509 – 1533.

In a Big Data context, the number of covariates used to explain a variable of interest, p, is likely to be high, sometimes even higher than the available sample size ($p > n$). Ordinary procedures for fitting regression models start to perform wrongly in this situation. As a result, other approaches are needed. A first covariates selection step is of interest to take into consideration only the relevant terms and to reduce the problem dimensionality. The purpose of this thesis is the study and development of covariates selection techniques for regression models in complex settings. In particular, we focus on recent high dimensional or functional data contexts of interest. We resort to regularization techniques when some model structure can be assumed or to novel dependence coefficients based on distances when not.