Functional data regression with prediction and interpretability : property inference in chemometrics with sparse Partial Least Squares (PLS)

Louna Alsouki

THESE de DOCTORAT DE
L'UNIVERSITE CLAUDE BERNARD LYON 1

en cotutelle avec l'Université Saint-Joseph de Beyrouth (Liban)

Ecole Doctorale N° 512
Ecole doctorale Informatique, mathématiques de Lyon (Infomaths)

Discipline : Mathématiques

Soutenue publiquement le 15/06/2023, par

Louna Alsouki

Régression de données fonctionnelles avec prédiction et interprétabilité : inférence de propriétés en chimiométrie avec des moindres carrés partiels sparse (PLS)

Devant le jury composé de :

Lambert-Lacroix, Sophie	Professeur HDR, Université Grenoble Alpes	Présidente et rapporteure
Bro, Rasmus,	Professeur, Copenhagen University	Rapporteur
Cardet, Hervé,	Professeur HDR, Université de Bourgogne	Examinateur
Ciuperca, Gabriela,	Maître de conférence HDR, Université Lyon 1	Examinatrice
Chaux, Caroline,	Directrice de Recherche, CNRS	Examinatrice
Marteau, Clément	Professeur HDR, Université Lyon 1	Directeur de thèse
Duval, Laurent	Ingénieur de recherche, IFPEN	Co-directeur de thèse
El-Haddad, Rami	Professeur associé, Université Saint Joseph de Beyrouth	Co-directeur de thèse
Helbert, Céline	Maître de conférence, Ecole centrale de Lyon	Invitée

”All things must pass”
To my sister Carine
and my mentor François

Abstract

Analytical chemistry plays a crucial role in various fields as it it covers identification, quantification, and characterization of chemical substances. It is essential for understanding the composition and behavior of matter and for developing new materials and technologies. It handles specially complex mixtures notably oil, an integral part of modern life, used as a source of energy and raw material for a range of products. It is a collection of hydrocarbons, must be refined using efficient technologies to produce lighter products and reduce impurities. Their physicochemical properties are essential for refining, transportation, and storage processes and can influence the quality of the deriving products. However, extracting them are pricy and resource-intensive. Spectroscopy is a widely used rapid analysis method that exploits the physical properties of products, using a signal profile represented by functional data. However, the accuracy of spectroscopy may be lower, and results may not be as comprehensive as standardized methods. Chemometrics techniques can create a predictive model for each property using rapid analysis spectra. It achieves two main objectives: predicting physico-chemical properties of new mixtures from reference mixtures and providing additional insights into the most related parts of the signal.

Multivariate calibration techniques establishes a mathematical relationship between functional data obtained from physico-chemical measurements $\mathbf{X}$ and traditional numerical macroscopic properties $\mathbf{y}$. Linear regression is used to establish the relationship between the two. As $\mathbf{X}$ is high dimensional, classical OLS regression is inapplicable and reduction techniques are used. A trade-off between accuracy and simplification is thus necessary to handle high-dimensional data problems.

This thesis uses real data to predict the density of oil cuts using infrared spectra. It was provided by IFPEN and made public for further scientific study. This manuscript also includes simulated data generated using Gaussian mixtures and sparse linear relationships to test hypotheses and assess the accuracy and interpretability of predictions.

An evaluation procedure was established, including new calibration-validation algorithm called CalValXy, which selects calibration observations using both $\mathbf{X}$ and $\mathbf{y}$ information. Metrics were used to evaluate the similarity of the calibration to the overall data and the prediction accuracy. The thesis also focuses on the interpretability of the results by detecting information using parsimony indicators, which refers to the presence of a relatively small number of non-zero coefficients in the model.

Dimension reduction techniques in data analysis includes projection (like PLS) and penalized (like lasso) methods. A new approach called Dual sparse Partial Least Squares was developed, which combines the advantages of both techniques for improved interpretability and accuracy in prediction models. The method uses a dual norm of selected penalties and our studies suggest four types of norms. A comparative benchmark test showed that the approach provided better interpretation of the trained prediction model with accurate prediction. It was also implemented in an R package called dual.spls, which also includes real data, a data simulation algorithm, a calibration and validation method, and evaluation tools.

Keywords— Partial least squares, lasso, ridge, regression, sparsity, dual norm, chemometrics, machine learning

Résumé

La chimie analytique joue un rôle crucial dans divers domaines car elle couvre l'identification, la quantification et la caractérisation des substances chimiques. Elle est essentielle pour comprendre la composition et le comportement de la matière et pour développer de nouveaux matériaux et technologies. Elle traite spécialement des mélanges complexes formés par un ensemble de molécules différentes, notamment les mélanges pétroliers, la source d'énergie la plus utilisée dans le monde depuis la révolution industrielle. Ils sont utilisés pour produire de l'essence, du diesel et d'autres combustibles pour les voitures, les camions, les avions et les bateaux. Leur caractérisation peut être établie à l'aide des propriétés physico-chimiques globales telles que la densité, la viscosité, le point d'éclair, le point d'écoulement, etc.. Ces méthodes sont normalisées et peuvent avoir un impact significatif sur les processus de raffinage, de transport et de stockage du pétrole. Elles peuvent également influencer la composition et la qualité des produits qui en dérivent. Cependant, les analyses de référence nécessitent des ressources importantes et sont coûteuses, limitant ainsi le nombre d'analyses pour le suivi des processus. De fait, il est donc nécessaire de disposer d'analyses rapides.

Les méthodes d'analyse rapide utilisent principalement la spectroscopie, qui exploite les propriétés physiques des produits. Cette approche présente plusieurs avantages, tels que la miniaturisation, les faibles coûts de fonctionnement et la rapidité. La spectroscopie comprend plusieurs types comme l'infrarouge, la résonance magnétique nucléaire, etc. Chacun diffère par la plage de longueurs d'onde utilisée, le type d'interaction impliquée ou le type de substance étudiée. Toutefois, ils ont tous en commun d'utiliser un profil de signal représenté par des données fonctionnelles. Cependant, leur précision est relativement inférieure et les résultats peuvent ne pas être aussi exhaustifs que ceux obtenus à partir de méthodes plus standardisées. Pour cela, les techniques de la chimiométrie peuvent résoudre ce problème en créant un modèle prédictif pour chaque propriété. De ce fait, cette thèse a deux objectifs principaux : le premier consiste à prédire les propriétés physico-chimique de nouveaux mélanges à partir de mélanges de référence, tandis que le deuxième objectif est d'apporter des éclairages supplémentaires sur les parties du signal qui est relié le plus la propriété d'intérêt.

Les N spectres obtenus à partir de mesures physico-chimiques sont considérés comme des données fonctionnelles. Ces dernières décrivent des phénomènes ou des variables qui varient continuellement dans le temps ou dans l'espace. Les spectres peuvent être discrétisés en un nombre fini de point P et stockés dans une matrice $\mathbf{X} \in \mathbb{R}^{N \times P}$. Les propriétés macroscopiques à prédire sont des données traditionnelles qui sont des valeurs numériques regroupées dans un vecteur $\mathbf{y} \in \mathbb{R}^{N}$. Pour résoudre ce problème, on utilise généralement des techniques de calibration

multivariée pour l'analyse prédictive, et la modélisation de régression pour établir une relation mathématique entre les données des spectres $\mathbf{X}$ et les propriétés macroscopiques $\mathbf{y}$. Cette thèse se place dans un contexte de régression linéaire où la relation entre les deux parties sont représentées par un vecteur de coefficient de régression $\boldsymbol{\beta}$. Les spectres de mesures physico-chimiques ont généralement un nombre d'observations N dans $\mathbf{X}$ plus petit que le nombre de variables P représentant la longueur d'onde. Ainsi, nous traitons principalement des cas où $P \geq N$. La méthode OLS classique utilisée généralement en regression linéaire minimise l'erreur quadratique de prédiction. Cependant, avec les problèmes de grande dimension, l'estimation OLS n'est plus applicable. ll est important de réduire le nombre de variables afin de pouvoir les visualiser et les analyser plus facilement. Toutefois, les techniques de réduction de dimension peuvent entraîner une perte de précision. La solution consiste à accepter une certaine perte de précision en échange d'une simplification des données.

IFPEN a fourni des données réelles pour appuyer cette étude. Chaque donnée comprend des propriétés physico-chimiques standardisées pour une variété de coupes de pétrole ainsi qu'un ou plusieurs spectres d'analyse rapide associés. Dans le chapitre 2, on présente les données réelles utilisées dans ce manuscript et rendues publicques dans un article pour encourager le développement d'autres études scientifiques. Elles sont composées de spectres infrarouges qu'on utilise pour prédire la densité des coupes considérées. En outre, nous avons proposé un générateur de données qui permet de reproduire des ensembles de données similaires aus données réelles. Les données simulées sontgénéralement importantes dans les projets de data science car elles permettent aux scientifiques de tester des hypothèses et de mener des expériences sans avoir à se limiter aux données réelles disponibles. Nous avons simulé des spectres fonctionnels représentés par une matrice $\mathbf{X}$ explicative en utilisant des mélanges de Gaussiennes et avons lié linéairement la réponse $\mathbf{y}$ à un nombre petit de variable de $\mathbf{X}$. De cette façon, nous avons pu évaluer la précision des prédictions en utilisant des modèles linéaires et évaluer l'interprétabilité lors de la création de régressions parcimonieuses.

Avant de commencer l'analyse et la modélisation des données, une procédure d'évaluation a été mise en place. L'un des objectifs principaux de la thèse est d'obtenir une précision de prédiction élevée. Dans un contexte de regression, la prédiction est souvent évaluée en séparant les données en ensembles de calibration et de validation. Le choix d'observation de calibration peut être fait aléatoirement sans grande connaissance préalable de la population. Dans cette procédure, chaque observation a une chance égale d'être retenue. Cependant, il n'est jamais garanti d'obtenir une calibration représentative de l'ensemble de l'échantillon. L'algorithme Kennard et Stone est moins aléatoire et largement utilisé en chimiométrie. Il sélectionne de manière séquentielle des observations de calibration uniformément espacées par rapport au valeurs dans $\mathbf{X}$. Or en regression, la réponse $\mathbf{y}$ porte aussi des informations importantes. D'où la deuxième contribution de cette thèse: un algorithme de calibration-validation nommé CalValXy détaillé dans le chapitre 3. Il sélectionne les observations de calibration en utilisant à la fois les informations de $\mathbf{X}$ et de $\mathbf{y}$. Pour évaluer la performance de la calibration, nous avons utilisé des mesures telles que la distance euclidienne et la distance Φ_2 qui quantifie la similitude entre l'ensemble de calibration et les données, ainsi que des métriques telles que l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE) et le coefficient de détermination (R^2) pour évaluer la précision de

prédiction. Les résultats obtenus ont montré que la calibration construite avec cet algorithme couvre de manière uniforme l'espace expérimental et fournit des prédictions en régression précises par rapport aux méthodes de référence. Nous avons également cherché à fournir des informations sur le signal en fonction de la prédiction des propriétés, c'est-à-dire en considérant d'évaluer l'interprétabilité des résultats. Nous avons choisi de relever ce défi en détectant des informations à l'aide d'indicateurs de parcimonie. La parcimonie dans un modèle de régression fait référence à la présence d'un nombre relativement faible de coefficients non nuls dans le modèle. Les modèles de régression parcimonieux sont plus faciles à interpréter car ils ne comprennent que les variables les plus importantes dans le modèle, ce qui permet de localiser plus facilement les facteurs les plus pertinents qui permettent une bonne prédiction. Ils nécessitent également moins de calculs pour construire le modèle de régression. Cela peut être particulièrement bénéfique lorsqu'on traite des données de grandes dimension. Nous proposons d'évaluer la parcimonie en calculant la mesure de comptage ℓ_0 des coefficients de régression β et en les comparant graphiquement aux spectres originaux pour évaluer la localisation.

Les techniques de réduction de dimension impliquent la transformation de données d'un espace de grande dimension vers un espace de faible dimension tout en préservant les informations clés des données d'origine. Deux catégories d'approches sont couramment utilisées : les méthodes projectives et les méthodes pénalisées. Les méthodes de projection sont basées sur la synthèse de la matrice de données $\mathbf{X}$ originale dans un espace de dimension inférieure, utilisant souvent des techniques telles que la régression PLS (moindres carrées partiels), couramment utilisées en chimiométrie. Le principe de la PLS consiste a résumer $\mathbf{X}$ en une matrice de score $\mathbf{T}$ en maximisant la covariance entre $\mathbf{T}$ et $\mathbf{y}$. La PLS a montré son efficacité grâce a la simplicité de son algorithme et la précision dans ses prédictions. Cependant, les résultats manquent d'interprétabilité. Les méthodes de pénalisation, quant à elles, consistent à régulariser les coefficients de régression pour une meilleur interprétabilité. Le lasso est une technique souvent utilisée. Sa régularisation de type ℓ_1 permet de produire des résultas parcimonieux visant une bonne interprétabilité. Néanmoins, le lasso possède des désavantages dans des situations de grande dimension comme sa saturation quand N variables sont selectionnées. Pour bénéficier des avantages des méthodes de projection et de pénalisation, leur combinaison a été proposée dans la litérature nommée sparse PLS. Comme troisième contribution de cette thèse détaillée dans le chapitre 4, une nouvelle approche généralisée appelée Dual sparse PLS a été développée. Cette méthode est inspirée par les sPLSs et applique une réduction adaptative. Elle est basée sur une norme duale de la pénalité sélectionnée. Nous avons proposé quatre types de normes inspirés des techniques connues: lasso, groupe lasso, moindres carrés et ridge, qui présentent des performances de calibration et de validation quasiment équivalentes au modèles de référence, avec moins de composantes formant $\mathbf{T}$. Un test de référence comparatif a été réalisé à l'aide des données simulés et de données réelles de spectroscopie proche infrarouge. Il a été constaté que les coefficients obtenus indiquent l'emplacement exact des zones de données influentes. Cela fournit une meilleure interprétation du modèle de prédiction formé.

Un package R appelé dual.spls a été développé pour mettre en œuvre la regression Dual-sPLS et la rendre accessible à la communauté scientifique pour résoudre des problèmes réels. En plus de la modélisation, ce package propose des fonctions supplémentaires pour faciliter son

utilisation autonome, notamment des données réelles, un algorithme de simulation de données, une méthode de calibration et validation CalValXy et des outils d'évaluation. Le chapitre 5 propose un tutoriel d'utilisation de dual.spls avec des exemples graphiques et les lignes de codes.

Ce travail de thèse a donc permis de gérer un projet de data science dans le domaine de la chimiométrie. Avant de proposer des solutions, nous nous somme mis des objectifs selon la problématique de caractérization du pétrole à l'aide de données fonctionelles de grande dimension. Ensuite, dans un premier lieu, nous avons regrouper les données qui peuvent nous être utile et surtout, nous avons contribué à aider la communauté scientifique en publiant des spectres infrarouges réels avec leurs densités associées en tant que source ouverte pour d'autres travaux. Dans un deuxième lieu, nous avons proposée une méthode de calibration validation qui permet de crée des modèle à partir d'un sous ensemble représentatif des observations selon l'espace de $\mathbf{X}$ et $\mathbf{y}$. Enfin, nous avons concue une méthode de régression qui présente de nombreux avantages:

- les prédictions correspondent ou dépassent les méthodes de reférence ou comparables,
- dans les différentes propositions de normes que nous avons examinées, elles produisent en outre des représentations parcimonieuses des données simulées et réelles,
- elles offrent une localisation interprétable des caractéristiques du point de vue des données fonctionnelles,
- elles permettent enfin le regroupement des variables : la possibilité de rassembler les variables explicatives en sous-ensembles plus significatifs (échantillons contigus autour d'un pic, bandes spectrales disjointes associées à un composé) pour pouvoir combiner différentes modalités physico-chimiques.

Mots clés— Régression moindre carrées, lasso, ridge, régression, parcimonie, norme duale, chimiométrie, machine learning

Acronyms

API American Petroleum Institute.

ASTM American Society of Testing Materials.

AWLS Automatic Weighted Least Squares.

CV Cross validation.

Dual-sPLS Dual sparse Partial Least Squares.

Icoshift interval correlation optimization shifting.

IR Infrared.

KS Kennard and Stone algorithm.

lasso Least absolute shrinkage and selection operator.

MAE Mean Absolute Error.

NIPALS Non- linear iterative partial least squares.

NIR Near-Infrared.

NMR Nuclear Magnetic Resonance.

OLS Ordinary Least Square.

PC Principal Component.

PCA Principal Component Analysis.

PCR Principal Component Regression.

PLS Partial Least Squares.

$\mathbf{R}^2$ Determination coefficient.

RMSE Root Mean Square Error.

simdist Simulated distillation.

SNV Standard Normal Variate.

sPLS sparse Partial Least Squares.

SRS Simple Random Sampling.

TMS Tetramethylsylan.

Contents

1 Introduction

1.0 Data science general workflow

This thesis is the outcome of a work pleasantly completed in a tripartite setting: in a joint agreement with the Claude Bernard University of Lyon 1 and the Saint Joseph University of Beirut and in collaboration with the French Institute of Petroleum New Energy (IFPEN). With this multi-background context, we were able to raise a wide range of questions and gather several contributions, each presented in a separate chapter. As the latter are component of a conventional data analysis scheme, we have chosen to start this manuscript with a form of generalization of what a data science workflow may be. We will roughly detail how, during these last three years, each step was completed and the questions we have specifically asked ourselves. A straight-through read of this manuscript will give the impression of a certain redundancy in the presentation of some concepts. We have preferred to retain the repeats as they serve to keep each chapter's integrity.

Data science makes it possible to model problems using multiple data and algorithms to produce efficient decisions. As illustrated in Figure 1.1, this practice is at the crossroads of skills in statistics and data analysis, computer sceince and business. In particular, machine learning can be used, hence allowing the ability to recognize structures in masses of data.

This thesis takes place in the field of chemometrics where chemical data are analyzed using mathematical tools to provide maximum relevant chemical information. In our project, we exploit two types of data: functional data that allows to discover a fine characterization of chemical mixtures, and macroscopic data providing generic physico-chemical properties and contributions at various levels (pre-treatement, inference...)were proposed in the manuscript. The core stages of this thesis are compatible with those found in other data science projects. For this reason, it seemed important to address the following steps (also illustrated in Figure 1.2), extensively developed throughout the manuscript:

(S1) **Defining questions and setting up context:** before embarking on a data science project, it is essential to understand the corresponding environment. Asking a set of measurable, clear, and concise questions helps in identifying problems and specifying main objectives. Therefore, building a roadmap will greatly clarify the goals of the project for all team members. In chemometrics, cores area of study are classification, pattern recognition, experimental design, signal processing, etc. For this project, we focused on multivariate calibration where we build models to forecast properties of complex petroleum

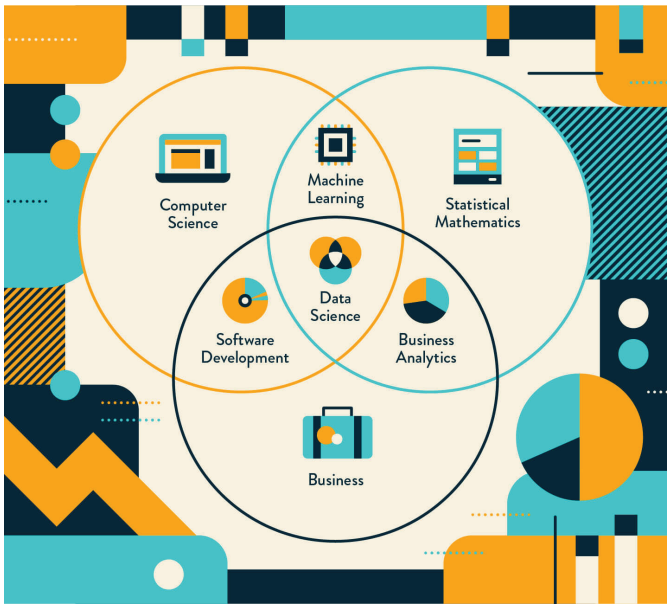

Figure 1.1 ~ *Data science: fusion of computer science, math, and business (extracted from* `https://clevertap.com/blog/data-science/`*).*

mixtures based on physico-chemical measurements. Section 1.1 details context, motivation and objectives.

(S2) **Acquiring and collecting data:** after setting up a solid idea of the context, aggregating the appropriate data and conceiving an appropriate experimental design come next. Data can be collected from internal or external sources. Regardless of the topic of the study, accurate data collection is imperative to maintain the virtue of research. Quality control helps ensuring the latter. Databases related to chemometrics are specifically designed to store chemical information. During the preparation of this PhD, IFPEN supplied this work with several data sets of physico-chemical measurements (Nuclear magnetic resonance, Near infrared and distillation) and associated macroscopic properties (cetane number and density). Data provided are presented in Section 1.2.

(S3) **Cleaning and pre-treating data:** data from different sources may have different formats, types and specific features. To prevent skewing of the project, anomalies must be identified and duplicates purged. Overseeing this can lead to errors regarding hypothesis or model biases for example. In chemical analysis, data are often large and common issues can emerge like outliers that can affect analysis and missing values. For cleaning, we usually aim at detecting and eliminating them. Moreover, popular pre-treatement procedures are smoothing techniques, interpolation procedure, reducing noise, baseline-correction processes etc. For example, in this work, we turned to a Savitzky–Golay filter for a derivative pre-processing to the NIR data (see Table 1.1) to increase its precision without distorting the signal tendency. Scatter-corrective dispersion correction methods were also applied on

other datasets. Details are covered in Section 1.3.

(S4) **Designing application and evaluation procedures:** this is a preliminary step before analyzing data. It depends on the initial objectives and type of analysis specified in step (S5). Procedure of applications of adequate methods are set up for a clear representation of results. An organized evaluation strategy is required based on project goals. For multivariate calibration objective set in (S1), one can choose metrics, reference methods, boundaries to improve etc. In this thesis, several dimensionality issues are associated to the considered data. Hence, we mainly resort to dimension reduction techniques like Partial Least Squares (PLS) as reference method. We consider different prediction errors metrics like Root mean squares error (RMSE), Mean absolute error (MAE) and determination coefficient (R^2) to improve. Another challenge is to look to interpret regression coefficients for information extraction. Strategies for evaluation are conceived and detailed in Section 1.4.

(S5) **Analyzing data:** This step allows manipulating data to extract meaningful insights. There are four main categories of data analysis:

- descriptive: identifies what has already happened,
- diagnostic: understands why something has happened,
- predictive: estimates future bearings based on historical data,
- prescriptive: provides future guidance.

Chemometrics covers all types. However, according to step (S1), we focus here on a predictive data analysis. More specifically, we turn to predictive modeling with regression procedures. Classical regression and dimension reduction methods are benchmarked for improvement. The considered approaches are presented in Section 1.5.

(S6) **Improving, evaluating and sharing results:** when all the previous steps are completed, outcomes must be interpreted and analysis should be evaluated. This is crucial since it measures gain from the overall work. Contributions of our thesis have been shared to users through an R package and several conferences and submitted articles. All of them summarized in Section 1.6

1.1 Defining questions and setting up context (S1)

The purpose of this chapter is to clearly define our questions and objectives. The petroleum industry, to which the thesis is applied, will be familiarized along with certain terminology specific to this sector. We will then go through the mathematical background that is typically taken into account in such a context such as ours.

1.1.1 Chemical aspect

Humans use energy from the environment and transform it into useful forms that fulfill their needs. Despite their concerns for the environment, currently, the primary sources of energy used

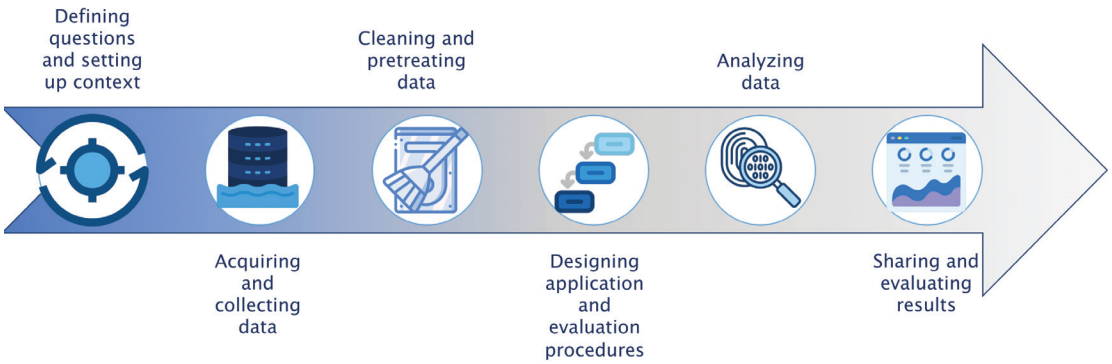

Figure 1.2 ~ *Six fundamental steps to complete a data analytics project.*

still include oil and represents approximately one third of human needs in energy [18]. It is essential for transportation, industry, pharmacy, etc.

Oil is a continuum of molecules, the majority of which are hydrocarbons. In terms of structure, these hydrocarbons are quite heterogeneous. Distillation allows crude oil to be separated into petroleum cuts (gasoline, kerosene, diesel, etc.) according to the boiling temperature of the molecules that compose it. The compounds with boiling temperatures higher than 350°C are referred to as heavy products, most abundant in the manufacture of oil. These heavy products can only be operative in conjunction with the development of effective refining technologies can. Petroleum refining turns these heavy goods into light products (fuels) and lower their metal content and other impurities. It aim to make quality products available to the consumer, in compliance with precise standards, particularly environmental ones, and with the quantities required by the market, which makes analyzing these sub-products an important issue. Heavy products are hard to be analyzed due to their complex, polydisperse composition and physical nature (opaque and viscous).

The analysis carried out for the characterization of heavy petroleum products can be grouped into three main types: global physico-chemical properties, mass repartition by chemical families and elemental analyses. Note that they are standardized. The first form of analysis surrounds this thesis. The most commonly used macroscopic properties are density, refractive index and viscosity. Section 1.2 and [87] both have further details on this topic. However, the reference analysis that are currently employed for characterizing them are pricy and resource-intensive. The number of analysis available for process monitoring is thus limited by cost and time and

rapid analysis is therefore needed.

The rapid analysis methods based on the exploitation of the physical properties of products are essentially spectroscopic methods. The advantages of this approach includes miniaturization, inexpensive running costs, and rapidity [9]. There are five types of spectroscopy: Infrared (IR), Ultraviolet-Visible (UV/Vis), Nuclear Magnetic Resonance (NMR), Raman, and X-Ray. They differ by wavelength range chosen, the type of interaction involved, or the type of substance under study. Their common thread is their signal profile represented by functional data. Examples of the latter can be found in Section 1.2.

However, rapid analysis is an indirect method. Indeed it is rarely possible to directly relate band intensity to property values. It is therefore necessary to resort to chemometrics techniques and develop a predictive model for each property in order to extract the relevant spectral information for its description. Therefore the focal point of this thesis relies on two main objectives:

(O1) predict properties of novel mixtures having previous knowledge of reference ones

(O2) provide additional insights on the signal parts that drives the majority of the property

We describe which statistical approaches allow achieving these objectives in the following section.

1.1.2 Predictive modeling

Resulting spectra from physico-chemical measurements are classified as functional data as they represent intensities according to a continuum observed in a largely finite set of points. This problem is usually solved by using multivariate calibration techniques [19] for predictive analysis where regression modeling [42] is mostly used. It operates by conceiving a mathematical relationship between explanatory data (spectra) and response data (macroscopic properties).

We establish below some formal notation and definitions that will be used throughout the manuscript.

Matrices, vectors and scalars are denoted by boldface uppercase letters, boldface lowercase and light lowercase letters respectively, e.g. $\mathbf{X}$, $\mathbf{y}$ and λ. The transpose of a given matrix $\mathbf{X}$ is $\mathbf{X}^T$. The identity matrix of size P is represented by $\mathbf{I}_P$. The ℓ_1-norm and the ℓ_2-norm of vector a $\mathbf{w} \in \mathbb{R}^P$ are

$$\|\mathbf{w}\|_1 = \sum_{p=1}^{P} |w_p| \qquad \text{and} \qquad \|\mathbf{w}\|_2 = \sqrt{\sum_{p=1}^{P} |w_p|^2}\,. \tag{1.1}$$

The vector of signs of any vector $\mathbf{w} \in \mathbb{R}^P$ is noted $\text{sign}(\mathbf{w})$, and $(\mathbf{w})_+$ is the vector composed of $\mathbf{w}_p$ if $\mathbf{w}_p \geq 0$ and 0 if $\mathbf{w}_p < 0$[(1)].

[(1)] It corresponds to the Rectified Linear Unit (ReLU), a popular activation function for neural networks.

We observe $N \in \mathbb{N}^*$ spectra stored in a matrix $\mathbf{X} \in \mathbb{R}^{N\times P}$ where for any $p \in \{1,\dots,P\}$ and $n \in \{1,\dots,N\}$, columns p and row n denotes scalar x_{np}, the $n^{\texttt{th}}$ spectra (observation) denotes row $\mathbf{x}_n \in \mathbb{R}^P$ and $p^{\texttt{th}}$ wavelength (variable) denotes column $\mathbf{x}^{(p)} \in \mathbb{R}^N$. The chemical property is represented by a response vector $\mathbf{y} \in \mathbb{R}^N$. In the following, we assume — without loss of generality— that both explanatory matrix $\mathbf{X}$ and response vector $\mathbf{y} \in \mathbb{R}^N$ are mean-centered.

According to the discussion displayed in Section 1.1.1, our aim is to build a mathematical link between response variable $\mathbf{y}$ and corresponding spectra $\mathbf{X}$. We focus in this manuscript on a linear relationship[(2)] between the two compounds, transcribed by the following equation:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\,, \tag{1.2}$$

where $\boldsymbol{\epsilon}$ is the measurement error depending on the variability within the sample. It is expected to be independent of $\mathbf{X}$, with zero mean. Parameter $\boldsymbol{\beta}$ describes the relationship between $\mathbf{X}$ and $\mathbf{y}$ and should be estimated, especially to understand the effect of each covariates on response $\mathbf{y}$. Equation (1.2) also allows to predict new responses related to external observations .

Ordinary Least Squares (OLS) is a common technique for estimating coefficients $\boldsymbol{\beta}$ from the model (1.2). It aims to minimize the sum of square differences between the observed and predicted values. More formally, it estimates $\boldsymbol{\beta}$ by

$$\hat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta}\in\mathbb{R}^P}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\,. \tag{1.3}$$

When $P \leq N$ and provided $\mathbf{X}$ has full column rank, the solution of optimization problem (1.3) is:

$$\hat{\boldsymbol{\beta}}^{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\mathbf{y}\,. \tag{1.4}$$

The ordinary least squares (OLS) prediction is $\hat{\mathbf{y}}_{LS} = \mathbf{X}\hat{\boldsymbol{\beta}}^{\text{OLS}} = \mathcal{P}_{[\mathbf{X}]}\mathbf{y}$, where $\mathcal{P}_{[\mathbf{X}]}$ is the orthogonal projection onto the space spanned by the columns of $\mathbf{X}$.

In our context, we recall that the number of lines (observations) in $\mathbf{X}$ corresponding to the spectra is smaller than the number of variables representing the wavelengths. Thus, we mainly deal with cases where $P \geq N$ i.e. data is composed of a large number of variables and relatively few observations. In this context, OLS estimation fails due to Gram matrix $\mathbf{X}^T\mathbf{X}$ shape. Although it is a squared matrix of P dimension, its rank is lower than N which makes the optimization problem (1.3) ill-conditioned (most of the eigenvalues are null) and singular.

As high-dimensional data structures are rapidly becoming recurrent, reducing the number of variables is crucial for visualization and analysis. Simple data also helps algorithm to be less time consuming. However dimension reduction techniques comes at the expense of accuracy, thus finding a trade off between accuracy and simplicity can be helpful. Most used methods will be detailed in Section 1.5.

[(2)] All regression methods can be applied for a multidimensional response. However, we restrict our study to one-dimensional $\mathbf{y}$.

Data	Explanatory matrix $\mathbf{X}$	Response vector $\mathbf{y}$	Number of observations N	Number of variables P
D_{NIR}	Near-Infrared	Density	208	~1500
$D_{\text{NIR}}^{\text{NMR}}$	Nuclear Magnetic Resonance	Cetane number	93	~14 000
	Near-Infrared			~2300
$D_{\text{simdist}}^{\text{NMR}}$	Nuclear Magnetic Resonance	Density	243	~65 000
	Simulated distillation			101

Table 1.1 ~ *Three internal real data sets provided by IFPEN.*

1.2 Acquiring and collecting data (S2)

This section describes briefly data sets manipulated during this thesis. Chapter 2 will provide further information.

Two types of data were handled. On the one hand, data from internal source were provided by IFPEN. Table 1.1 list down their nature and dimensions and this section offers an extensive description. On the other hand, we also resorted to simulated data to empower data-driven decision making. Integrating simulated data provides richer insights due to all the different scenarios that can be generated.

Real data ~ As shown in Table 1.1, three internal data sets were made available by IFPEN. Each provides one microscopic property, stored in the response vector $\mathbf{y} \in \mathbb{R}^N$, that we wish to predict and one or two spectral physico-chemical measurement, represented by the matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$, that we wish to analyze for forecasting.

In the following, we will give a brief description of each kind of measurement. As macroscopic densities, D_{NIR} and $D_{\text{simdist}}^{\text{NMR}}$ consider sample's densities and $D_{\text{NIR}}^{\text{NMR}}$ focus on Cetane number index. .

Density is the physico-chemical characteristic that is most frequently used to describe oil. The American Petroleum Institute (API) gravity is typically used to indicate the density of crude oils. It is related to specific density in such a fashion that an increase in API gravity corresponds to a decrease in specific density. The oil industry makes extensive use of density measurement because it provides a rapid, accurate, and repeatable indicator of the quality of an oil cut. It is strongly correlated to a large number of properties such as unsaturated carbon content, hydrogen content, ... It is measured depending on the product's state (liquid or solid) at 70°C. The lower

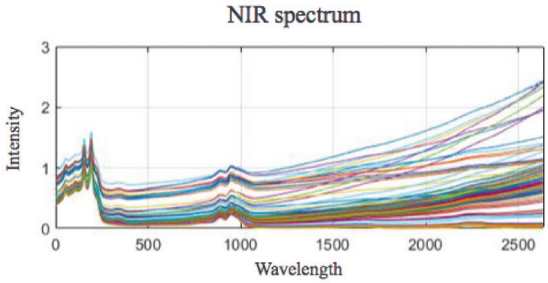

Figure 1.3 ~ *Near-Infrared (NIR) spectra of 208 heavy oil samples from* D_{NIR} *data.*

the density, the lighter the cut and it is described as more paraffinic. Conversely, the higher the density, the more the aromatic character will be predominant.

Cetane is a chemical molecule easily ignited under pressure and naturally contained in diesel. It is used as the industry-standard measurement for assessing the quality of fuel combustion due to its high flammability. Specifically, this measure is referred to as cetane number. A fuel that ignites fast has a high cetane number. The scale for measuring cetane ranges from 0 to 100. Cetane number is an inverse function of a fuel's ignition delay, the time period between the start of ignition and the first identifiable pressure increase during combustion of the fuel.

For physico-chemical measurments of rapid analysis, data offers Near-Infrared (NIR), Nuclear Magnetique Resonance (NMR) and simulated distillation (simdist) spectra. More detailed information of these approaches are found in [65]. These three families of spectra are described below.

Infrared spectroscopy (IR) is an analytical technique based on the principle of absorption of radiation (infrared) by matter. Infrared is the radiation corresponding to wavelengths directly greater than those of the visible light spectrum. Infrared radiation falls into three areas: the near-infrared from 800 nm to 3000 nm, the mid-infrared from 3000 nm to 25 000 nm and the far-infrared from 25 000 nm to 10×10^7 nm. In literature [21] [26], NIR spectroscopy is the most frequently used approach to characterize heavy oil products. Radiation absorption by oil samples depends on their composition, more precisely it is linked to chemical bonds. The experiment is based on illuminating the sample at different frequencies (or wavelengths) and measuring the sample's light absorption. The absorption of light at each of these wavelengths compose the spectrum. The units are typically expressed in wavenumbers, $\bar{\nu}$ in cm^{-1}: $\bar{\nu} = \dfrac{\nu}{c} = \dfrac{1}{\lambda}$ where ν is the frequency in Hz, c the velocity of light and λ the wavelength. Numerous publications [20, 13, 48] have been written about determining macroscopic physico-chemical properties using NIR spectroscopies. Figure 1.3 represent an example of near 200 overlayed spectra.

NMR spectroscopy is a widely used tool in organic chemistry to identify the chemical bonds of complex molecular structures in the form of a spectrum. The experiment is based on the

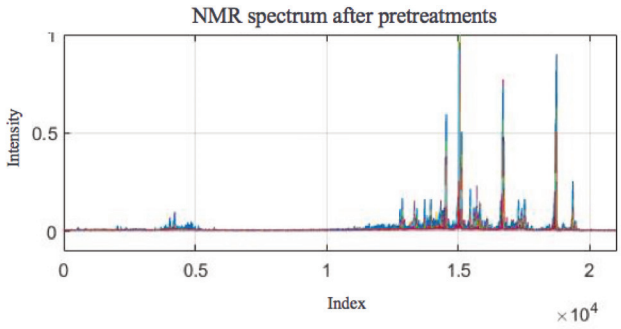

Figure 1.4 *~ Nuclear magnetic resonance (NMR) spectra of 243 heavy oil samples from* $D^{\text{NMR}}_{\text{simdist}}$ *data.*

excitation of the nuclei sample with radio waves into nuclear magnetic resonance [17]. The changes in the resonance frequency gives access to details of the electronic structure of a molecule and its individual functional groups. An NMR spectrum consists of peaks or series of peaks called signals corresponding to the resonance of the different protons present in the molecule. Figure 1.4 overlay approximately 240 NMR spectra where peaks are easily spotted. These signals are placed on a horizontal axis indicating a value called "chemical shift" noted δ and expressed in part per million (ppm). The chemical shift reflects the shift between the resonance frequency of the protons of the molecule studied and a resonance frequency taken as a reference. In general, the reference frequency is the proton resonance frequency of the Tetramethylsylan molecule (TMS). NMR is also popularly used to predict oil properties [74, 11, 99].

Simulated distillation provides fast and reliable data distribution of boiling points for petroleum fuels and finished products [100]. Simulated distillation (simdist) is measured by introducing the study sample into a column and separating the hydrocarbons according to their boiling point; then the correspondence between the retention time and the boiling point is established and the curve percentage weight-boiling point of hydrocracking slices is obtained. 243 curves are illustrated in Figure 1.5. Examples of petroleum characterization using simulated distillation are found in [22, 109].

Simulated data ~ During this work, an algorithm simulating data that mimics the real chemical data shapes or conditions was developed. This provides the ability to test millions of scenarios by altering parameters and thus saving a great deal of time and energy. It allows us to ensure reaching objectives (O1) and (O2) (see Section 1.1). The algorithm is based on mixing K Gaussians peaks with the same preset scale σ^2 and randomly picked amplitudes A_{ik} and locations μ_k, for $i \in \{1, \dots, N\}$ and $k \in \{1, \dots, K\}$. They are summed as follows, uniformly sampled in P variables and regrouped in an explanatory matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$. More formally, each row $\mathbf{x}_i$ for

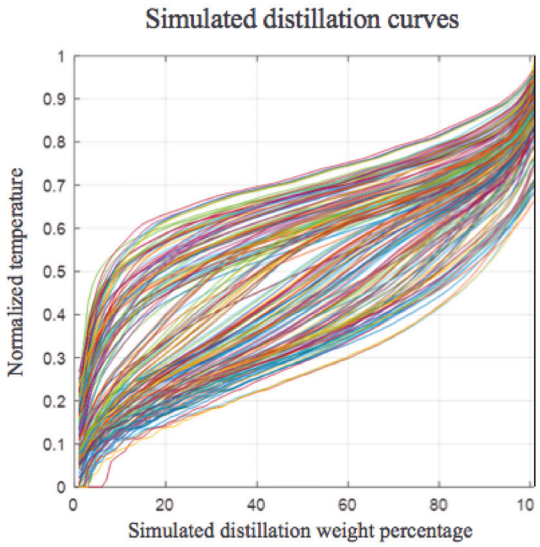

Figure 1.5 *~ Simulated distillation (simdist) curves of 243 heavy oil samples from* $D_{\text{simdist}}^{\text{NMR}}$ *data.*

$i \in \{1, \ldots, N\}$ of matrix $\mathbf{X}$ corresponds to a curve:

$$\sum_{k=1}^{K} A_{ik} \exp(-\frac{x - \mu_k}{2\sigma^2}) \,. \tag{1.5}$$

evaluated on a regular grid from 1 to P variables i.e. each grid point is associated to one variable.

Objective (O1) focus on linking variables. Thus, the response vector $\mathbf{y}$ is defined by an explicit linear model composed of weighted sums of $\mathbf{X}$ values. Weights can be random or fixed quantities by range of indices. Objective (O2) targets localizing variables influential features. Therefore, we mainly create sparse additive model. More precisely, we set $S \ll P$ positive weights and $P - S$ null weights i.e. only S variables are accountable for computing response $\mathbf{y}$.

When tackling step (S5), several methods popularly considered when dealing with these kinds of data will be introduced. For clear and complete understanding, we will use simulated data D_{SIM} to illustrate each approach. D_{SIM} (described in Chapter 2) is built with N =300 mixtures of K =30 Gaussians represented by P =1000 variables. In Figure 1.6, highlighted red areas denote S variables involved in the computation of the response $\mathbf{y}$.

1.3 Cleaning and pre-treating data (S3)

The goal of cleaning and pre-treating data is to remove spectral variation unrelated to the property desired to predict. These spectral variation may be random (noise), instrument-related,

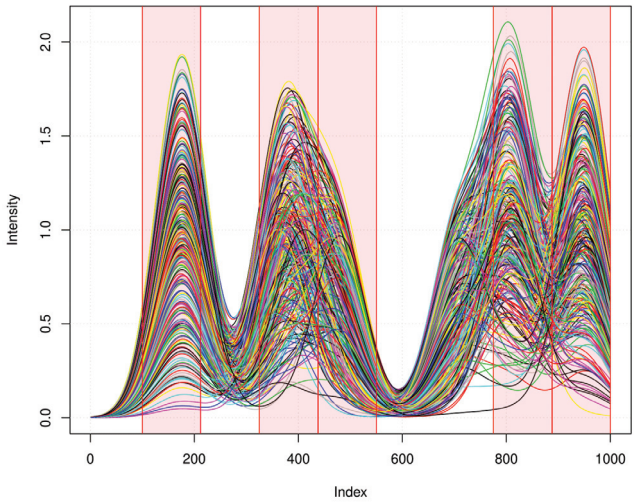

Figure 1.6 ~ *Sparse simulated data D_{SIM} represented by 300 curves. Each curve is stored in a row in the explanatory matrix $\boldsymbol{X}$ used to predict a simulated vector $\boldsymbol{y}$. Highlighted red areas represents the only variables of $\boldsymbol{X}$ linked to $\boldsymbol{y}$.*

or caused by physico-chemical interferences (diffusion). Removing them offers improvement of predictive models [79]. According to the noise's primary cause, pre-treatment techniques can be split into two broad categories: scatter-corrective dispersion correction methods and spectral derivatives.

Due to confidentiality restrictions placed on $D_{\text{NIR}}^{\text{NMR}}$ and $D_{\text{simdist}}^{\text{NMR}}$, only applications on D_{NIR} are offered for real data in this manuscript. As a supplement, we will describe how each were pre-treated and briefly go through some procedures.

NIR spectra from D_{NIR} and $D_{\text{NIR}}^{\text{NMR}}$ were both derivated using Savitzky Golay smoothing (see Section 1.3.1) for a window of 15 variables and from a polynomial of order 2. Additionally, to lessen the scatter effects between samples, Standard Normal Variate (SNV) [10], a popular scatter correction technique, is applied to NIR spectra from data $D_{\text{NIR}}^{\text{NMR}}$.

NMR spectra from $D_{\text{NIR}}^{\text{NMR}}$ data were first aligned using the interval correlation optimization shifting (Icoshift) [82]. Afterwards, Automatic Weighted Least Squares of order 2 (AWLS), Savitzky Golay smoothing and Normalization over the area of each spectrum were applied as pre-treatment.

We choose to describe briefly in this section Savitzky-Golay filter and Standard Normal Variate (SNV). The cited references above contain descriptions and illustrations of the remaining approaches.

1.3.1 Savitzky-Golay filter

Both additive (vertical baseline offset) and multiplicative (vertical baseline shifts as a function of wavelength) effects may appear in the spectra. To get round of such perturbations, spectra derivatives are usually considered. In this context, savitzky-Golay algorithm [83] is the most widely used technique in chemometrics for the derivation of spectra. The principle is the following: for a given spectra and a given width v (odd scalar in $\mathbb{N}^+$), it calculates a polynomial fit of order $o \in \mathbb{N}^+$ in each filter window as the filter is moved across the signal. Mathematically, for a given window of width v, it operates as follows

$$x_p^* = \frac{1}{R} \sum_{h=-H}^{H} c_h x_p + h\,, \quad for p \in \{\frac{v+1}{2}, \ldots, P - \frac{v+1}{2}\} \tag{1.6}$$

where

- x_p^* is the new value at variable p,
- R is a normalizing coefficient,
- H is the gap size on each side of variable p,
- c_h are the pre-computed coefficients, that depends on the chosen polynomial order and degree.

Figure 1.7 borrowed from [39] illustrates an example where the blue lines represents a signal spectrum and the filled dots the measurements. The signal is a spectrum that has been discretely measured (blue line with measurements at the filled dots). On the bottom left, black dotted lines indicates three filter windows for an application of Savitzky Golay smoothing for $v = 7$. The subplot in the upper right corner displays an example for the window 22 to 28 The polynomial fit at the center point determines the filter estimate at the center of each window, thus v is always an odd integer. The "X" sign in the subplot provides the filtered signal at point 25 which is the center point. The smoothing is complete when the filter run through each variable.

The Savitzky-Golay filter especially is interesting because it is designed to preserve specific features of the signal, such as peaks or slopes, by adjusting the order of the polynomial and the size of the window used in the filter. The parameters used in the Savitzky-Golay filter include the order of the polynomial used to fit the data, the size of the window used for each point, and the degree of smoothing desired. Generally, higher order polynomials and larger window sizes will result in better smoothing, but may also lead to more distortion of the signal. The degree of smoothing can be controlled by adjusting the number of points used in the filter.

To compute derivative, Savitzky Golay simply estimates it for each filter window for a given order d. Figure 1.8 shows NIR spectra from data D_{NIR} after a first derivative filtering using Savitzky Golay smoothing for a window of 15 variables and from a polynomial of order 2.

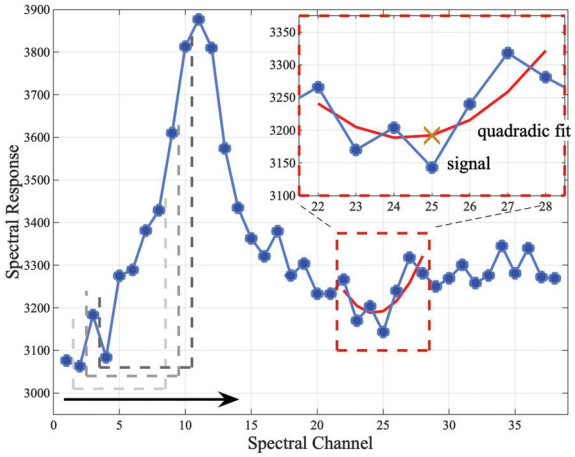

Figure 1.7 ~ *Spectrum measured at discrete points (blue line with dots). Filter windows, $w = 7$, are shown in the bottom left. A quadradic fit is shown in the top right for windows 22 to 28 with corresponding filter value at point 25 given as X. Figure borrowed from [39].*

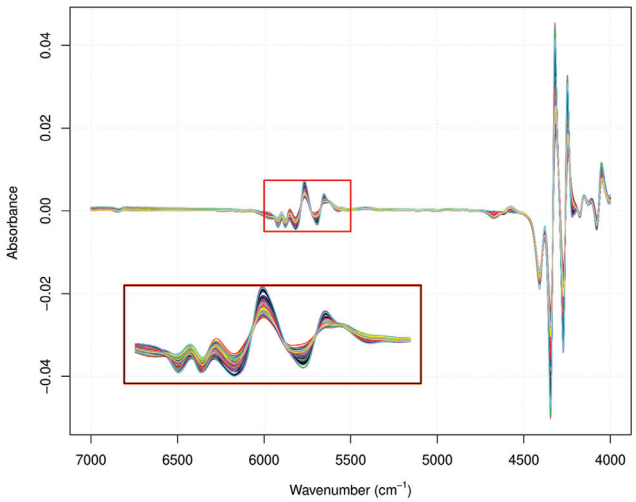

Figure 1.8 ~ *First derivative of 208 NIR data spectra from data set D_{NIR}. Right bottom subplot provide a clearer representation of overlaid spectra in wavenumber range from 7000 to 4000 cm^{-1}*

1.3.2 Standard Normal Variate (SNV)

SNV is a straightforward methodology and has efficient scattering correction powers. It is frequently applied to spectra where discrepancies across otherwise identical spectra are caused by baseline and pathlength changes. Therefore, it normalizes the data and operates row-wise, for $n \in \{1, \ldots, N\}$:

$$SVN_n = \frac{\mathbf{x}_n - \bar{x}_n}{s_n}, \tag{1.7}$$

where

- SVN_n Corrected value,
- $\bar{x}_n$ mean value of the uncorrected n^{th} spectrum $\mathbf{x}_n$,
- s_n standard deviation of the values of the n^{th} spectrum $\mathbf{x}_n$.

The resultant SVN_n spectra are independent of the initial absorbance values as they consistently have a zero mean value and a variance of one. [35] provides a clear illustration of the impact of SVN changes.

1.4 Designing application and evaluation procedures (S4)

Clear objectives were stated[(3)]. Before analyzing data and building the most adequate predictive model, evaluation procedures must be specified. Therefore, in order to evaluate whether each goal was reached, this section will be divided in two, each dedicated respectively for (O1) and (O2).

1.4.1 Prediction evaluation (O1)

This objective focuses on the power of prediction of conceived model. In our work we resort to splitting data into two different subsets. This is essential to assess newly built models performance, otherwise, the prediction will be overly optimistic and biased. First,a representative set, named calibration, is needed to set up the model by estimating parameters. Second, a smaller similar set, named validation, is essential to determinate most appropriate parameters and estimate the quality of the final model. Several calibration-validation methods have been proposed in the literature. We evoke here the two most used.

Simple random sampling (SRS) is the most straightforward method. It is based on randomly selecting calibration with little advance knowledge about the population. In this procedure, each observation has equal chance to be retained. Due to randomization, it has a low risk of sampling bias. However, a representative calibration of the overall sample is never garanteed.

[(3)] see step (S1) in Section 1.1.1

Kennard and Stone(KS) algorithm ~ Kennard and Stone [54] splitting is less random. Its corresponding algorithm is popularly used in chemometrics [97, 70]. It sequentially selects calibration promoting the most uniform spatial distribution possible which means it is based on computation of distances between samples. Thus, the choice of distance is influential. Kennard and Stone generally uses either Euclidean d^E or Mahalanobis distance d^M defined for $i, j \in \{1, \ldots, N\}$:

$$d_{ij}^E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{p=1}^{P} (x_{ip} - x_{jp})^2}, \tag{1.8}$$

$$d_{ij}^M = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}, \tag{1.9}$$

where $\mathbf{\Sigma}$ is the sample covariance matrix.

Let $\mathbf{D} = \{o_1, \ldots, o_N\}$ be the set of the N where, for any $n \in \{1, \ldots, N\}, o_n = (\mathbf{x}_n, y_n) \in \mathbb{R}^{P+1}$ gathers $\mathbf{x}_n \in \mathbb{R}^P$ from the $\mathbf{X}$ space and the corresponding response $\mathbf{y}_n \in \mathbb{R}$. At each iteration, a candidate will be removed from $\mathbf{O}$ and added to the calibration set $\mathbf{C}$. At the end of the procedure, $\mathbf{C}$ will contain $N_c < N$ points, where N_c is a number set in prior, and the $N - N_c$ remaining data in $\mathbf{O}$ will form the validation set. With KS algorithm, computations are done using the projections of the observations on the $\mathbf{X}$-space.

Starting point: The first calibration point is the observation that is the farthest away from the centroid $\mathbf{G} \in \mathbb{R}^P$ with components $G_p = N^{-1} \sum_{n=1}^{N} x_{np}$ for $p \in \{1, \ldots, P\}$. This observation is removed from $\mathbf{D}$ and assigned to the calibration set $\mathbf{C}$.

Another way of initiating the algorithm is to select the first most distant couple of observations.

Iteration $n+1$: Assume that we have successively determined n $(1 \leq n < N_c)$ calibration points $c_1, \ldots, c_n$, we determine the next observation c_{n+1} along these operations.

1. Compute the distance $\Delta(o, \mathbf{C})$ between each point o of $\mathbf{D}$ and the calibration set $\mathbf{C} = \{c_1, \ldots, c_n\}$

$$\Delta(o, \mathbf{C}) = \min\{d_{\mathbf{X}}(o, c_1), \ldots, d_{\mathbf{X}}(o, c_n)\} \quad \forall o \in \mathbf{D}.$$

   Here $d_{\mathbf{X}}$ represents the distance chosen between the $\mathbf{X}$-values of the observations.

2. The observation $o \in \mathbf{D}$ that has the largest $\Delta(o, \mathbf{C})$ will be the $n+1$-th calibration point c_{n+1}.

Step 2 is repeated until the desired number N_c of calibration points is attained.

These steps are summarized in Algorithm 1.

Algorithm 1: Kennard and Stone

Input: $\mathbf{X}$, N_c
Set $\mathbf{O} = \{o_1, \ldots, o_N\}$, the set of the overall observations
Compute $G_p = N^{-1} \sum_{n=1}^{N} x_{np}$ for $p \in \{1, \ldots, P\}$ (centroid)
Determine $c_1 = \text{argmax}_{o \in \mathbf{O}} d(G, o)$
Move c_1 from $\mathbf{O}$ to calibration set $\mathbf{C}$
while number of observations in $\mathbf{C}$ is less than N_c **do**
  $o = \text{argmax}_{o \in \mathbf{O}} d(o, \mathbf{C})$
  Move o from $\mathbf{O}$ to $\mathbf{C}$.
end while

Comparing SRS to Kennard and Stone splitting ~ Figures 1.9 and 1.10 illustrate the splitting of D_{SIM} after applying PCA (see Section 1.5.1 for more details) with a 80-20 calibration and validation percentage. We notice that with KS, each validation observation (in green) is represented by one or more calibration (in red), unlike with SRS. With the latter, loads of validation points seem to be dispersed, hence the model cannot be calibrated to suit the majority of samples. Here, KS provides a more adequate calibration. However, the splitting is employed in a prediction context that involves two sets of variables: $\mathbf{X}$ and $\mathbf{y}$. The KS algorithm ignores response vector $\mathbf{y}$ and solely uses explanatory factors to calculate distances. In Chapter 3, we present a new splitting technique called CalValXy that takes into account both sets.

After sample splitting, models are built and predictions are done over the two subsets. To assess models prediction accuracy, it is only logical to consider prediction errors metrics. We choose the following three error measures: the root mean squares error (RMSE), the mean absolute error (MAE) and the determination coefficient (R^2) defined respectively as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2} = \frac{1}{\sqrt{N}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2 \,, \tag{1.10}$$

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n| = \frac{1}{N} \|\mathbf{y} - \hat{\mathbf{y}}\|_1 \,, \tag{1.11}$$

$$R^2 = \frac{\sum_{n=1}^{N} (y_n - \bar{y})^2}{\sum_{n=1}^{N} (y_n - \hat{y}_n)^2} \qquad \text{with} \qquad \bar{y} = \frac{\sum_{n=1}^{N} y_n}{N} \,, \tag{1.12}$$

where in each definition, $\mathbf{y} \in \mathbb{R}^N$ denotes the response vector and $\hat{\mathbf{y}}$ its estimate.

1.4.2 Coefficient interpretation (O2)

This task requires additional understandings regarding the relationship between $\mathbf{X}$ and $\mathbf{y}$. More precisely, it requires strict information concerning localization of most influential variables according to response $\mathbf{y}$. Regression coefficients $\hat{\boldsymbol{\beta}}$ quantify the link between predictor variables and response by providing estimates of unknown vector $\boldsymbol{\beta}$ from Equation (1.3). The coefficient value represents the mean change in the response given a one unit change in the predictor while

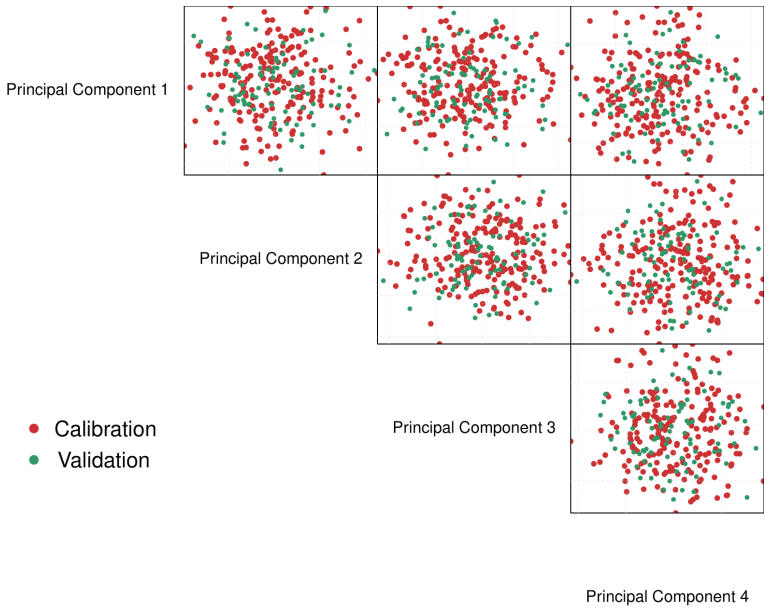

Figure 1.9 ~ *Simple random splitting of PCA transformation of D_{SIM} with 80-20 percentage splitting according to first four principal components.*

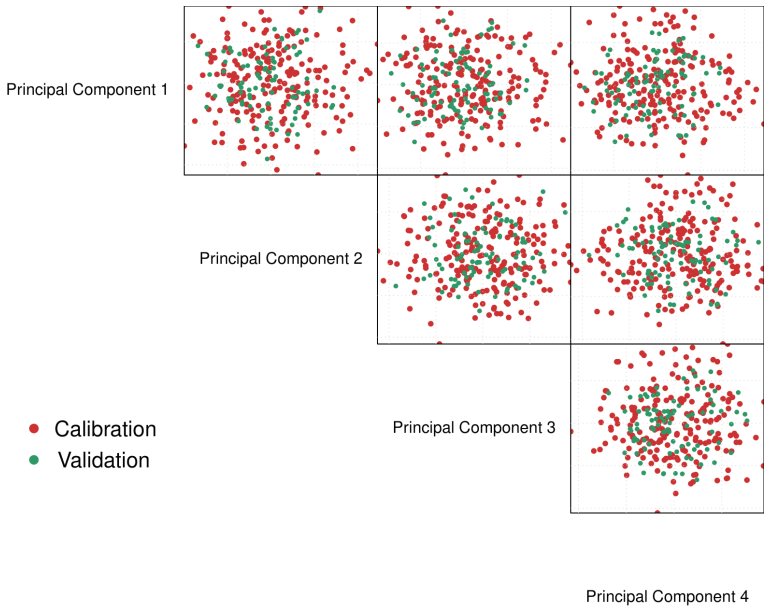

Figure 1.10 ~ *Kennard and Stone splitting of PCA transformation of* D_{SIM} *with 80-20 percentage splitting according to first four principal components.*

holding other variables in the model constant. A simple way to grasp regression coefficient is to picture them as linear slopes. Comparing $\hat{\boldsymbol{\beta}}$ to original spectra is a straightforward technique to interpret coefficients and try to extract information.

Model selection in statistics is the process of choosing the pertinent predictors to include in the model. A possible way to select a small number of relevant features from a training set is to fit a sparse linear model that only depends on a subset of $S \ll P$ predictors. Resulting sparse estimator $\hat{\boldsymbol{\beta}}$ will give insights about the location of the most relevant variables. Sparse regression is frequently used with high-dimensional data (see [14, 12] for recent publications).

To evaluate interpretation of models, $\hat{\boldsymbol{\beta}}$ will be plotted against original data spectra like in Section 1.5.1 for example. The model gives further insights when it is more selective and has accurate localization. To emphasize on sparsity performance, we introduce an objective criteria that quantifies the sparsity amount. In the following, $\ell_0(\mathbf{w})$, the sparsity index or count measure [24] will denote the non-zero coordinates of $\mathbf{w}$ and $\ell_0^c(\mathbf{w})$ its complement i.e. $\ell_0^c(\mathbf{w}) = P - \ell_0(\mathbf{w})$. A relatively small $\ell_0(\mathbf{w})$ reflects a stronger sparsity.

1.5 Analyzing data (S5)

Objectives prompt the use of a predictive analysis. With high-dimensional data like ours, dimension reduction procedures are aided. They consist of converting data from a high-dimensional space to a low-dimensional space while preserving most significant information of the original data. Some impressing benefits [76] of these reductions are:

- less processing power and training time, hence improving thus machine learning performances,
- allowing data visualization,
- avoiding overfitting problems,
- fixing multicollinearity issues.

Different existing approaches can be classified as either projection or penalized methods. This section focuses on presenting each type and describing which specific ones were tested for data analysis. To illustrate the latter, each method is applied to D_{SIM} data and corresponding graphs will be presented to support the theoretical explanation.

1.5.1 Projection methods

Projection methods summarize original matrix $\mathbf{X}$ by building a new space of dimension $M < N$ with $M \in \mathbb{N}^*$. The idea is to build an orthogonal matrix of weights, $\mathbf{W} \in \mathbb{R}^{P \times D}$ such as $\mathbf{T} = \mathbf{XW}$ is of full rank. Matrices $\mathbf{W}$ and $\mathbf{T}$ are denoted respectively loadings and scores. Scores $\mathbf{T}$ are used instead of original $\mathbf{X}$, and linear model (1.2) is reformulated as:

$$\mathbf{y} = \mathbf{T}\boldsymbol{\beta}^{\mathbf{W}} + \boldsymbol{\epsilon} = \mathbf{XW}\boldsymbol{\beta}^{\mathbf{W}} + \boldsymbol{\epsilon}\,. \tag{1.13}$$

With this reduction, OLS is applied and $\boldsymbol{\beta}^{\mathbf{W}}$ is estimated using Equation (1.4):

$$\tilde{\boldsymbol{\beta}}^{\mathbf{W}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}\mathbf{y}\,. \tag{1.14}$$

We denote by $\hat{\boldsymbol{\beta}}^{\mathbf{W}}$ the estimated regression coefficient associated to Equation (1.2) via dimension reduction. Thus,

$$\hat{\boldsymbol{\beta}}^{\mathbf{W}} = \mathbf{W}\tilde{\boldsymbol{\beta}}^{\mathbf{W}} = \mathbf{W}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}\mathbf{y}\,. \tag{1.15}$$

In this last equation, the singularity problem of OLS estimator in high dimensional situations is solved. Indeed, the inversion of matrix $\mathbf{T}^T\mathbf{T}$ is required and conceivable in projection methods where dimension is reduced.

The above equations show that this type of methods projects response $\mathbf{y}$ in the new lower dimensional space to estimate them with $\hat{\mathbf{y}}$ i.e. $\hat{\mathbf{y}} = \mathcal{P}_{\mathbf{T}}\mathbf{y}$, where $\mathcal{P}_{.}$ denotes the orthogonal projection onto the space spanned by the subscript.

Projection methods differ by the way where the lower-dimensional $M < N$ sub-space is built. We focus in the following on two specific construction strategies leading to the so-called Principal Component Regression (PCR) and Partial Least Squares (PLS) regressions.

Principal Component Regression ~ PCR is based on Principal Component Analysis (PCA) [107]. Also known as Karhunen-Loève transform (KLT), it is widly used for several purposes: dimension reduction, data compression, feature extraction, data visualization, among others.

The PCA algorithm is commonly used in several fields that deals with high-dimensional data like neuroimaging [36], chemometrics [30], bioinformatics [66], voice recognition [44], etc. It based on an orthogonal projection of $\mathbf{X}$ onto a smaller subspace, called principal sub-space [105]. The latter is built while maximizing the variance between new covariates, i.e. solving optimization problem

$$\max_{\mathbf{W}\in\mathbb{R}^{P\times M}} \mathbf{W}^T\Sigma\mathbf{W} \quad \text{s.t.} \quad \mathbf{W}^T\mathbf{W} = I_M\,, \tag{1.16}$$

where $\Sigma = n^{-1}\mathbf{X}^T\mathbf{X}$ is the empirical covariance matrix of $\mathbf{X}$.

The PCA loadings $\mathbf{W}$ are the eigenvectors of the symmetric covariance matrix Σ associated to its eigenvalues sorted from largest to smallest. They allow building the set of scores $\mathbf{T}$ denoted as the principal components (PCs). The first PC offers the maximum potential variance, which represents a variation as large as that in the input characteristics, followed by later PCs. PCA is thus essentially interesting when dealing with highly correlated variables. Indeed, when the information is redundant, i.e. variables are linearly dependent, PCA excludes components associated with small eigenvalues to build a decorellated set of score vectors. They are built to also be orthogonal due to the constraint in Problem (1.16).

In a conventional linear regression model, PCR is especially used to estimate the unknown regression coefficients in order to represent the link between predictors and responses. Instead of using $\mathbf{X}$, a PCA transformation is applied and the new set of PCs are used as regressors. PCR deals with the problem of dimensionality by building a new sub-space solely depending on variation between predictors. However, in a regression problem, two sets of variables are considered: the explanatory variables in $\mathbf{X}$ and the response variable in $\mathbf{y}$. The latter is not taken into consideration, potentially affecting the prediction accuracy, unlike the PLS described in the next section [41].

Partial Least Squares ~ PLS methods (also called *projection to latent structures*) were first developped in the late 1960s to the 1980s by economist Herman Ole Andreas Wold [68]. PLS was mainly developped for chemometrics [104, 106]. When the goal is to predict while reducing dimension, the technique is called partial least square (PLS) regression [16, 1]. Unique to PLS, and unlike PCR, the extracted factors $\mathbf{W}$ account for both predictor and response variation. It is considered a flexible technique as it it can be applied to wide data with low sample size [43].

The fundamental idea of PLS avatars is to project data onto a lower-dimensional space like mentioned in Section 1.5.1 by building linear combination of original variables. Its basic idea is to compress the predictor matrix $\mathbf{X}$ by maximizing covariance between $\mathbf{X}$ and $\mathbf{y}$. The corresponding optimization problem for the first loading $\mathbf{w}_1$ is:

$$\max_{\mathbf{w}_1}(\mathbf{y}^T\mathbf{X}\mathbf{w}_1) \quad \text{s.t.} \quad \|\mathbf{w}_1\|_2 = 1\,. \tag{1.17}$$

The convex Problem (1.17) can be solved with Lagrange multipliers leading to a closed form solution: $\mathbf{w}_1 = \mathbf{X}^T\mathbf{y}$. PLS uses the weight vector $\mathbf{w}_1$ to compress regressor $\mathbf{X}$ into the first score vector $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$.

Following components are built iteratively and a new lower dimensional score matrix $\mathbf{T} \in \mathbb{R}^{N\times M}$ is produced. Several approaches can be considered. Two popular algorithm are NIPALS [102] and SIMPLS [32]. While NIPALS uses deflation to iteratively compute components, SIMPLS is more direct [63]. NIPALS iteratively computes weight vectors by deflation while SIMPLS is more direct. NIPALS considers the part of $\mathbf{X}$ that is orthogonal to $\mathbf{t}_k, k < m$. Thus, iteratively, for the m^{th} component, $\mathbf{X}$ is replaced by $\mathbf{X}_m$ such that:

$$\mathbf{X}_m = \mathbf{X} - \mathcal{P}_{\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}}\mathbf{X} = \mathbf{X}_{m-1} - \mathcal{P}_{\mathbf{t}_{m-1}}\mathbf{X}_{m-1}\,, \tag{1.18}$$

where $\mathcal{P}_{\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}}$ denotes the orthogonal projection onto the space spanned by components $\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}$.

Based on Proposition 1 from [57], the regression coefficients for M components are computed as:

$$\hat{\boldsymbol{\beta}}_M^{PLS} = \mathbf{W}(\mathbf{T}^T\mathbf{X}\mathbf{W})^{-1}\mathbf{T}^T\mathbf{y}\,. \tag{1.19}$$

The vector of regression fitted values $\hat{\mathbf{y}}$ for M components is the projection of response vector $\mathbf{y}$ onto the space spanned by scores columns of $\mathbf{T}$.

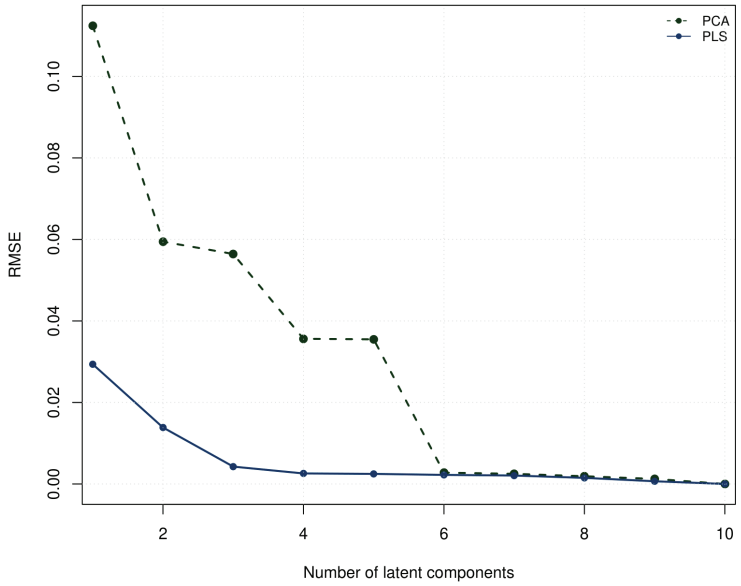

Figure 1.11 *~ RMSE values of PCA and PLS regressions on D_{SIM} with respect of the number of latent components.*

According to [31], PLS is known for

1. its algorithm simplicity,
2. its accuracy in predictions,
3. its good performance when dealing with highly-correlated data etc.

However, in order to get valuable information about the variables, its regression coefficients are challenging to interpret.

Comparing PCR to PLSR ~ The RMSE values for PCA and PLS regressions applied on D_{SIM} data, are compared in Figure 1.11 while increasing the number of latent components. Lower RMSE indicates more accurate forecasts. The RMSE values of the two methods decreases as the number of components rises. Indeed, the more components are used the more variance is preserved, thus more information. A significant disparity between the two curves for the first six components hints that PLSR is more accurate at making predictions than PCR. For this dataset, this suggests that PLS will be used as the primary reference technique for improvement to reach this thesis prediction objective (O1).

We examine the PCA and PLS regression coefficients in Figure 1.12 by comparing them to the original data D_{SIM}. It is obvious that both curves share characteristics and exhibit variation

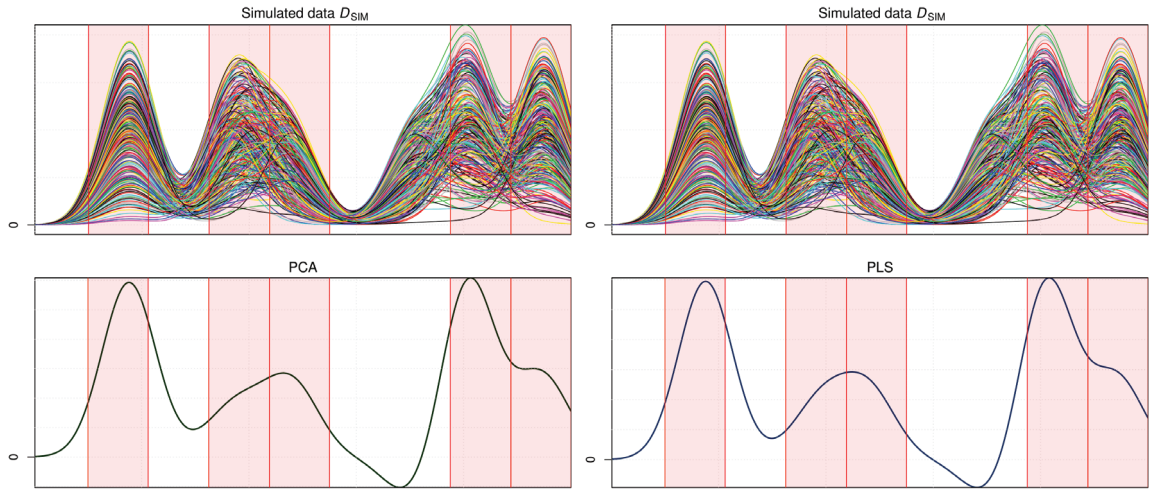

Figure 1.12 ~ *Original data D_{SIM} (top) compared to PCA (bottom left) and PLS (bottom right) regression coefficients for six components.*

in the same places. Additionally, the curves resemblance to the original data spectrum can be easily seen. However, we recall that D_{SIM} is built to be sparsely linearly linked to response $\mathbf{y}$, thus we no in prior the location of influential variables. The interpretation of $\hat{\boldsymbol{\beta}}$ with both regressions is impractical as the curves do not hint to where the information is located. Thus, (O2) is hard to be achieved using these projection methods.

1.5.2 Penalized methods

Another way to perform dimension reduction is to resort to penalization. Popular penalized methods adds to Problem 1.3 a penalty function *pen* as in the following:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^P}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + pen(\boldsymbol{\beta})\,. \tag{1.20}$$

Penalized regression methods constrain regression coefficients by shrinking them towards zero. If the amount of shrinkage is large enough, they can also perform variable selection by zeroing some coefficients. This latter results in a less complex model by choosing the most crucial features. It offers the benefit of being more straightforward to control than the entire collection of variables.

The choice of the penalty function *pen* depends on study objectives. Table 1.2 explicits some popularly used penalization.

All these penalties are associated with hyper-parameters. For example, the Lasso strength of shrinkage depends on λ. Hyper-parameters are all positive values. They control the amount of regularization and choosing the appropriate ones is crucial and remains a challenge. Cross-validation is a way to tune the hyperparameters using only the training data [45]. In the following applications, regularization parameters are chosen by using CV.

Lasso [47]	$\lambda\|\boldsymbol{\beta}\|_1$	acheives variable selection and handles high dimesion problems
Ridge [49]	$\lambda\|\boldsymbol{\beta}\|_2^2$	limits instability of predictions due to correlated variables
Elastic-net [112, 47]	$\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2$	combine above penalties to improve the approaches
Fused-lasso [96]	$\lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2 \sum_{p=2}^{P} \|\beta_j - \beta_{j-1}\|$	ensures that spatially close variables are activated together
Group lasso [111]	$\lambda \sum_{g=1}^{G} v_g \|\boldsymbol{\beta}_g\|_2$	where G groups of variables are associated v_g weight for a sparse selection of groups.

Table 1.2 ~ *Popular penalties.*

Least absolute shrinkage and selection operator **~** Least absolute shrinkage and selection operator, also known as lasso [95], is a popular regression analysis technique that allows the interpretation of the resulting statistical model by performing variable selection. The corresponding penalty is based on an ℓ_1 regularization technique that may build sparse models with few coefficients. In particular, certain coefficients can be reduced to zero and be dropped from the model. Equation (1.20) in this case is formulated as:

$$\arg\min_{\boldsymbol{\beta}\in\mathbb{R}} \left[\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \right]. \tag{1.21}$$

Problem (1.21) is reformulated as:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_1 \leq t\,, \tag{1.22}$$

with a one-on-one correspondance between parameters λ and t.

The level of shrinkage applied to the estimate is controlled by threshold t, which must be greater than 0. To acquire outcomes that can be interpreted, a suitable parameter is essential. When $t \geq \|\hat{\boldsymbol{\beta}}^{\mathrm{LS}}\|_1$, the lasso estimate is equivalent to the standard least squares solution $\hat{\boldsymbol{\beta}}^{\mathrm{LS}}$, if existing, as stated in [95]. Additionally, it chooses, on average, half of the variables when $t = \frac{\|\hat{\boldsymbol{\beta}}^{\mathrm{LS}}\|_1}{2}$. In the orthonormal design case, i.e. $\mathbf{X}^T\mathbf{X} = I_P$, there exists a closed form solution $\hat{\beta}^{\mathrm{l}}$ called *soft thresholding*. It zeros coefficients with small magnitudes and reduces the others relatively to the threshold. It verifies:

$$\hat{\beta}_p^{\mathrm{l}} = \mathrm{sign}(\hat{\beta}_p^{\mathrm{LS}})(|\hat{\beta}_p^{\mathrm{LS}}| - \lambda)_+ \quad \forall p \in \{1, \ldots, P\}\,. \tag{1.23}$$

Despite having been successful in several applications, there are some recognized limitations [112, 47] as:

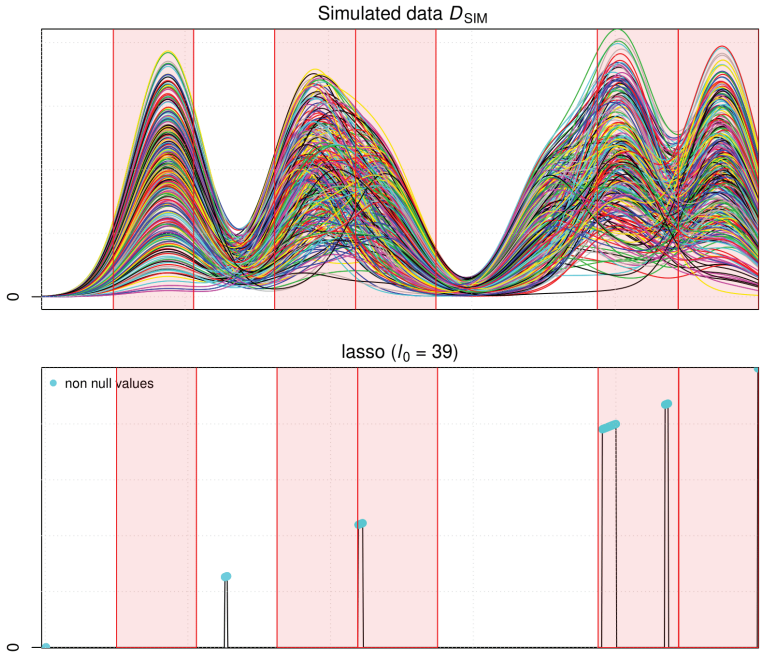

Figure 1.13 ~ *Original data D_{SIM} compared to lasso regression coefficients.*

1. non-strict convexity of Problem (1.21) in high dimensional cases,
2. algorithm saturation when N variables have been chosen
3. tendency to select moderately representative variables when using strongly correlated variables

Lasso application on simulated data D_{SIM} is illustrated in Figure 1.13. We can notice that variable selection is evident with sharp peaks. Lasso only selects 39 variables out of 1000 and shrinks the rest to zero. However, some selected variables do not appear to be in the red areas i.e. lasso selected irrelevant variables. This hints that in some cases, lasso interpretation can be inaccurate.

Ridge penalization ~ Another popular penalization procedure is the ridge regression [49]. Compared to the lasso optimization Problem 1.21, the penalty function is replaced with an ℓ_2 constraint equivalent to square of the magnitude of the coefficients. Equation (1.20) is re-written as:

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2 . \tag{1.24}$$

With ridge regression, the singularity problem of matrix $(\mathbf{X}^T\mathbf{X})$ is solved by adding a variation λ to the matrix spectrum. Therefore, the solution always exists, expressed as:

$$\hat{\boldsymbol{\beta}}^{\text{r}} = (\mathbf{X}^T\mathbf{X} + t\lambda I_P)^{-1}\mathbf{X}^T\mathbf{y} . \tag{1.25}$$

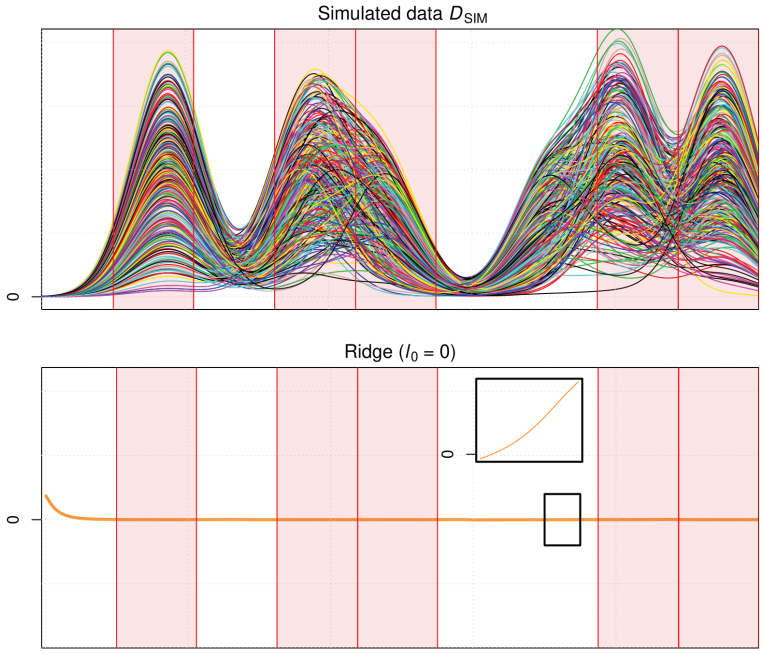

Figure 1.14 ~ *Original data D_{SIM} compared to ridge regression coefficients. Hyper-parameter λ is selected using cross validation. Subplot in orange provides a detailed curve with smaller y-axis scale proving that coefficients are not shrunk to zero.*

This regularization causes shrinkage of coefficients aiding in reducing multicollinearity and model complexity. We can notice that when parameter λ is close to zero, Equation (1.24) is closer to standard linear regression Problem (1.3).

Ridge regression shrinks regression coefficients corresponding to variables with minor contribution to the outcome close to zero. Compared to the lasso, it uses an ℓ_2-norm instead of the ℓ_1 penalization but retains most variables by design.

Regression coefficients of ridge application on simulated data D_{SIM} are portrayed in Figure 1.14 and compared to original data. We can notice that variables are shrunked close to zero and thus model is less complex. However, analyzing coefficients for information about variables influence on response $\mathbf{y}$ is practically impossible. In fact, as we know where most influential variables are located, we can not deduct their locations by relying on the ridge regression coefficient of this application.

1.5.3 Blending methods: sparse Partial Least Squares (sPLS)

PLS projection method and lasso variable selection approach have important advantages that allows reaching respectively (O1) and (O2). Sparse Partial Least Squares (sPLS) denotes a body of works that combines both strategies in order to balance precision of prediction and meaningful

interpretation. They add lasso inspired penalties to the PLS framework by integrating an ℓ_1-norm to optimization problem (1.17). For $\lambda_s > 0$ and with an orthogonality constraint on components, we get, for the first component:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min}\{-\mathbf{y}^T\mathbf{X}\mathbf{w} + \lambda_s\|\mathbf{w}\|_1\}, \qquad \text{for } \mathbf{w}^T\mathbf{w} = 1\,, \tag{1.26}$$

Problem (1.26) is hard to handle and was tackled throughout the years with several reformulations each introducing the penalty differently. We mention three important contributions and denote them after their first author. With $\mathbf{z} = \mathbf{X}^T\mathbf{y} = N\widehat{\mathrm{Cov}}(\mathbf{X}, \mathbf{y})$:

1. $sPLS_{LeCao}$ — In 2008, Lê Cao and its co-authors [61] penalized a sparse Singular Value Decomposition (SVD) proposed in [84]. Their optimization problem is the following:

$$\underset{\mathbf{u},\mathbf{v}\in\mathbb{R}^P}{\arg\min} = \|\mathbf{z} - \mathbf{u}\mathbf{v}^T\|_2^2 + \lambda\|\mathbf{u}\|_1\,, \tag{1.27}$$

   where $\lambda > 0$ is the sparsity parameter. Their algorithm iteratively penalizes the SVD decomposition of the product of the deflated versions of $\mathbf{X}$ and $\mathbf{y}$.

2. $sPLS_{Chun}$ — In 2010, [25] used a surrogate direction $\mathbf{c}$ close to the original vector $\mathbf{w}$. Problem (1.26) is reformulated by imposing the ℓ_1 penalty on $\mathbf{c}$, providing an approximate solution. Their optimization problem is the following:

$$\underset{\mathbf{w},\mathbf{c}\in\mathbb{R}^P}{\arg\min} = \{-K\mathbf{w}^T\mathbf{z}\mathbf{w}(1-K)(\mathbf{c}-\mathbf{w})^T\mathbf{z}(\mathbf{c}-\mathbf{w}) + \lambda_1\|\mathbf{c}\|_1 + \lambda_2\|\mathbf{c}\|_2^2\}\,, \tag{1.28}$$

   where $\lambda_1, \lambda_2 > 0$ are the sparsity parameters and $K > 0$ is fixed. Solving (1.28) is done by alternatively iterating between solving for $\mathbf{w}$ for fixed $\mathbf{c}$ and solving for $\mathbf{c}$ after fixing $\mathbf{w}$.

3. $sPLS_{Durif}$ — In 2018, [37] consider another reformulation of the following optimization problem:

$$\underset{\mathbf{w}\in\mathbb{R}^P}{\arg\min}\{\frac{1}{2}\|\mathbf{z} - \mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_1\}\,, \tag{1.29}$$

   under the constraints $\|\mathbf{w}\|_2 = 1$ and orthogonality between components and where $\lambda > 0$ controls sparsity. Resolution of Problem 1.29 uses recent notions from proximal optimization [8]. Additionally, the authors provide an adaptive method for computing sPLS weights vectors relating them to classical PLS ones.

Optimization problem of each SPLS variants presented above are solved using different mathematical tools. However, variable selection is their common objective. It is achieved by using the soft threshold operator

$$g_\lambda(\mathbf{u}) = \mathrm{sign}(\mathbf{u})(|\mathbf{z}| - \eta)_+ \tag{1.30}$$

where $u \in \mathbb{R}^P$ and $\eta \in \mathbb{R}^+$. The thresholding is based on a value η that is used to be compare with all the coefficients of $\mathbf{u}$. Soft thresholding first sets to zero coefficients whose absolute values are lower than the threshold η and then shrinks the nonzero coefficients toward zero. It is known to provide smoother results [40].

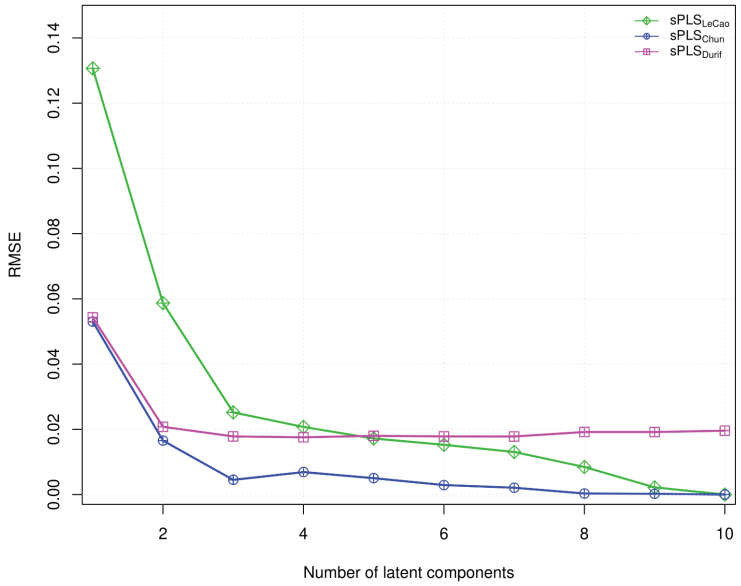

Figure 1.15 ~ *RMSE values of sPLSs variants $sPLS_{LeCao}$, $sPLS_{Chun}$ and $sPLS_{Durif}$ regressions on D_{SIM} with respect to the number of latent components. Corresponding hyper-parameters are selected usin cross validation.*

Figure 1.15 represents RMSE values of each sPLS regression avatar applied on D_{SIM} data for 1 to 10 latent components. Naturally, values decrease with the increasing number of latent component similarly to results for PLS and PCA application in Figure 1.11. Starting with four components, decreasing speed is much slower and each curve is found to plateau. The curves represent close values to the PLS RMSE (see Figure 1.11) indicating accuracy in their predictions.

In Figure 1.16, regression coefficients of the three variants are compared to the original data D_{SIM}. We can notice that the most informative results are those from $\mathrm{sPLS}_{\mathrm{LeCao}}$ since it only selects 60 features. However, in some important areas (highlighted in red), all variables are shrunk to zero. Several peaks also appear localizing the wrong information in clear ranges. $\mathrm{sPLS}_{\mathrm{Chun}}$ and $\mathrm{sPLS}_{\mathrm{Durif}}$ almost retain all variables providing unsatisfactory results.

1.6 Improving, evaluating and sharing results (S6)

This section presents a summary of the contributions achieved in this thesis and provides a description of the organization of the manuscript. This work was involved in improving several aspects in the steps listed earlier which makes our contributions covering a wide range of themes.

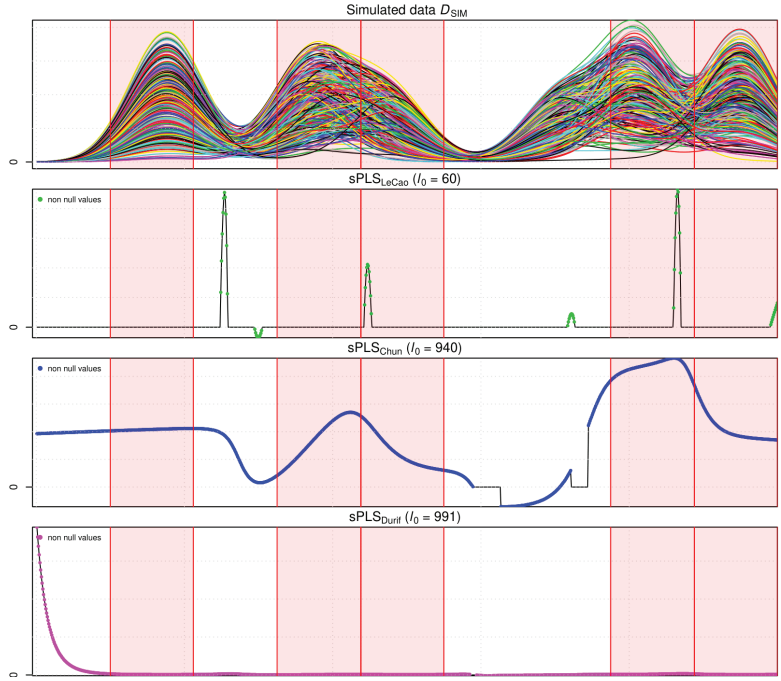

Figure 1.16 ~ *Original data D_{SIM} compared to sPLS variants $sPLS_{LeCao}$, $sPLS_{Chun}$ and $sPLS_{Durif}$ regression coefficients for six components.*

1.6.1 Real and simulated data *(Chapter 2)*

Every study requires solid supporting real data. However, it is hard to access interesting and useful databases. Sometimes, open source data are incomplete or unreliable. The first contribution of this thesis was setting up the D_{NIR} real data base as open source for it to be be used and exploited with benchmarking in other studies.

Two types of data were used in this thesis: real and simulated. On the one hand, real data ensures that study is valid, reliable, and applicable to real-world situations. Thus, we rigorously detail the development and description of real NIR data D_{NIR} in Chapter 2. On the other hand, simulated data allows testing hypotheses and models in a controlled environment, conduct experiments or simulations that would be difficult or impossible to do in real life, explore different scenarios, and identify patterns that may not be immediately apparent in real-world data. Therefore, we also provide a simulation approach for synthetic data (like D_{SIM} mentioned in Section 1.2). By altering particular parameters, it is intended to deliver data that is comparable to real spectra like NIR or NMR. Proposed simulated data algorithm also allows generating several complementary explanatory data linked to the same response. Additionally, as we focus on sparse regressions, it has the advantage of providing data that verifies the hypothesis that certain explanatory factors may be more strongly associated with response $\mathbf{y}$ than others. In this manner, variable selection technique may be efficiently investigated for feature localization.

Related scientific production: ~ Paper entitled "MLnir IFPEN near-infrared spectroscopy dataset for property prediction: 208 NIR hydro- carbon spectra and density response", to be submitted in June 2023 in *Data in Brief*.

1.6.2 CalValXy splitting *(Chapter 3)*

When designing evaluation procedure in (S4), we raise an issue in calibration and validation splitting. As this work is placed in a regression context, two data are involved: independent $\mathbf{X}$ and response $\mathbf{y}$. As mentioned in Section 1.4.1, Kennard and Stone algorithm does not take into account information brought by response $\mathbf{y}$. Therefore, the second contribution solves this dilemma by proposing a novel experimental design called CalValXy.

In a few words, CalValXy mainly stratifies response $\mathbf{y}$ and applies the Kennard Stone algorithm to predictor matrix $\mathbf{X}$ in the concerned strata. Through numerical simulations we evaluated similarities between the —smaller— calibration set and the original data. We showed that the new approach offered a decent representation of the initial database. By testing CalValXy splitting against alternative splitting methods, we were also able to evaluate the prediction performance. By computing RMSE values, we found that our new method outperforms the others in prediction. Overall, CalValXy was judged to be simple and reliable. Chapter 3 provides a full description of the new algorithm and associated outcomes.

Related scientific production ~ Paper entitled "CalValXy: well-balanced and stratified calibration/validation splitting using both predictors $\mathbf{X}$ and response $\mathbf{y}$", to be submitted in June 2023 in *Technometrics*.

1.6.3 Dual sparse Partial Least Squares *(Chapter 4 and 5)*

In step (S5), we applied and evaluated methods already documented in literature. As presented in Section 1.5, each approach have some downsides to improve and advantages to preserve. Thus a mix of these regression methods was conceived in one generalized sparse regression called Dual sparse Partial Least Squares (Dual-sPLS). It is the third contribution of this thesis. Dual-sPLS is based on PLS1 algorithm [57]. It relies on the dual norm of a chosen penalty norm [8] where shrinkage is applied adaptively.

Definition 1.6.1 *Let $\Omega(\cdot)$ be a norm on $\mathbb{R}^P$. For any $\mathbf{z} \in \mathbb{R}^P$, the associated dual norm, denoted $\Omega^*(\cdot)$, is defined as*

$$\Omega^*(\mathbf{z}) = \max_{\mathbf{w}} (\mathbf{z}^T \mathbf{w}) \quad \text{s.t.} \quad \Omega(\mathbf{w}) = 1\,. \tag{1.31}$$

Comparing (1.17) and (1.31), we find that the optimization of the PLS objective function amounts to finding the vector $\mathbf{w}_1$ that fits the dual norm of the ℓ_2-norm of $\mathbf{z}$, where $\mathbf{z} = \mathbf{X}^T\mathbf{y}$. This motivates us to evaluate different norm expressions that could be used as domain-related penalizations. Thus, for any norm $\Omega(.)$ used, the first loading vector will be:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \{-\mathbf{z}^T \mathbf{w}\}, \qquad \text{s.t. } \Omega(\mathbf{w}) = 1\,. \tag{1.32}$$

NIPALS deflation scheme introduced in Section 1.5.1 is adopted to iteratively compute the rest of the components. Although formulation is generic, we emphasize four types of norms that aim at

1. combining heterogeneous and high-dimensional data sources,
2. providing accurate predictions,
3. extracting pertinent knowledge for better localization.

Chapter 4 provide all important details concerning this new approach. We also implemented corresponding package dual.spls in R. It includes the following main functions, each of them being associated to specific penalty:

1. **Dual-sPLS$_\mathbf{l}$** *(pseudo-lasso norm, d.spls.lasso()).* Similar to the sPLS Problem (1.26), an intuitive norm combines ℓ_2 and ℓ_1:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 + \|\mathbf{w}\|_2\,. \tag{1.33}$$

2. **Dual-sPLS$_\mathbf{gl}$** *(pseudo-group lasso norm, d.spls.GL()).* Inspired by group lasso [86], it combines groups of measurements. It applies pseudo-lasso to each group individually while constraining the total set. For G groups, $\mathbf{w}_g$ represents the variables of the loading vector $\mathbf{w}$ that belongs to group g. The corresponding norm is formulated as:

$$\Omega(\mathbf{w}) = \sum_{g=1}^{G} \alpha_g \|\mathbf{w}_g\|_2 + \lambda_g \|\mathbf{w}_g\|_1\,, \tag{1.34}$$

where $\alpha_g \geq 0, \forall g \in \{1,\ldots,G\}$ and $\sum_{g\in 1,\ldots,G} \alpha_g = 1$.

3. **Dual-sPLS$_{\mathbf{LS}}$** *(pseudo-least squares norm, d.spls.LS()).* It introduces $\mathbf{N}_1$, a matrix of p columns, and applies when $\mathbf{X}$ is not singular:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{N_1 w}\|_1 + \|\mathbf{Xw}\|_2 \,. \tag{1.35}$$

   The classical least squares solution is recovered for $\lambda = 0$.

4. **Dual-sPLS$_{\mathbf{r}}$** *(pseudo-ridge norm, d.spls.ridge()).* It deals with cases where $\mathbf{X}$ is singular and resorts to a ridge-like penalization:

$$\Omega(\mathbf{w}) = \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{Xw}\|_2 + \|\mathbf{w}\|_2 \,. \tag{1.36}$$

A tutorial of R package dual.spls is detailed in Chapter 5 and regroups a set of functions used for Dual-sPLS fitting listed above. It also provides functions for sparse data simulations and real NIR data presented in Chapter 2. CalValXy splitting algorithm is also included in this package with extra functions for error computing, data visualization, etc.

The construction of weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_M$ differs in each of the four options but follows similar steps. Each alternative has a closed form solution that can be expressed with the soft thresholding operator introduced in Equation (1.30).

A common issue in shrinkage methods is the choice of the appropriate regularization parameter. Thereby, we conceived an adaptive algorithm that computes it according to the number of variables that we would like to keep in the active set at each iteration also detailed in Chapter 4.

Results of applications of each norm case, on simulated and real data are illustrated in Chapter 4 and 5. They show how Dual-sPLS reaches objectives (O1) and (O2) by finding balance between accurate prediction and satisfactory interpretation.

Related scientific production ~ Paper entitled "Dual-sPLS: a family of Dual Sparse Partial Least Squares regressions for feature selection and prediction with tunable sparsity; evaluation on simulated and near-infrared (NIR) data", published in April 2023 in *Chemometrics and Intelligent Laboratory Systems*, R package " dual.spls: Dual sparse partial least squares regression" published in October 2022 in *CRAN* and paper entitled "Sparse PLS with a group lasso inspired penalty: inside the dual.spls package", to be submitted in June 2023 in *Journal of Statistical Software*.

2 Data

This chapter is dedicated to the presentation of the data processed throughout this thesis. They chiefly consists in a database of real experiments and a variety of simulated observations. In Section 2.2, we present MLnir, a collection of 208 near-infrared spectra of complex mixtures with their density feature (D_{NIR}), provided by IFPEN. They are subject to an open data publication [38]. In Section 2.3, we simulate different sets (D_{SIM} and alike) of explanatory matrices $\boldsymbol{X}$ based on mixtures of Gaussians. They emulate typical analytical chemistry measurements. They provide us with a precise control on how the influental spectral variates influence the (potentially non-linear) modeled response $\boldsymbol{y}$. They allow a more thorougful assemement of the proposed calibration/validation and regression algorithms (resp. CalValXy and dual-sPLS), especially in the evaluation of the impact of sparsity on coefficient localization.

Contents

2.1 Introduction

The inspiration of our work resides in the analysis of chemical compounds, as can be found in products related to refined crude oil, a longlasting major energy resource. They appear as complex mixtures of numerous hydrocarbons, sulfur, oxygen, nitrogen, and metal-containing organic species with a wide range of molecular weights, contents, and structures [90]. Their characterization is crucial to their rationalized use, either in terms of efficiency or safety. It consists for instance in determining global properties of a sample, as well as identifying more precise their chemical components [71, 80, 108].

Characterization can be performed using physico-chemical standardized methods [89]. The American Society of Testing Materials (ASTM) and the International Organization for Standardization (ISO) methodologies are commonly used analytical techniques in the oil industry. Those techniques can additionally be customized to meet specific requirements. Some of them however require a certain volume of the analyzed mixture sample (in the order of liters), which is not compatible with present high-throughput experiments (producing millimeters of products). Furthermore, standardized methods might be lengthy and costly. Alternative characterizations of complex mixtures as thus desirable.

Combining physico-chemical analysis and chemometric techniques has become a promising approach to evaluate properties and composition of chemical mixtures [67, 73]. Among many spectroscopic techniques, we highlight infrared (IR) measurements. For the given oil sample, it results in a spectrum which may benefit from being represented, and treated, as an instance of functional data. Figure 2.1 illustrates an example of a single near-infrared spectrum.

In chemometrics, it may be rewarding to explicit links between analytical physico-chemical measurements (e.g. IR spectra) $\mathbf{X}$ to properties $\mathbf{y}$ deriving from standardized methods (e.g. sample density) [55]. Regression models can be built for this purpose. The resulting model, through well-chosen calibration or learning, would be used to predict — within a certain precision — properties faster, from small-sized chemical samples, in partial replacement of standardized approaches. In this thesis, our main objectives are to both efficiently construct predictive models, and detect (sparse) areas of spectra that are likely to be most related to the considered property. Assessment qualitatively and quantitatively those objective requires different sorts of representative datasets.

We recall that IFPEN provided different sets of real data to support this study: Nuclear Magnetic Resonance, Near-Infrared spectrometry, Simulated distillation (see Table 1.1). They were acquired along the main problematic dealt with this thesis. Each dataset is obtained from a given quantity of chemical samples (here petroleum cuts). For each sample, data is composed of one or several macroscopic chemical property, and one or a couple of associated (spectral) physico-chemical measurements. The latter are mainly represented by a monodimensional signal (or function) with varying intensities along a specific ordinal axis. They are discretized with a relatively fine resolution, yielding high dimensional sets of data. For confidentiality purposes, only a D_{NIR} near-infrared spectroscopy dataset is discussed in this manuscript and made available. We therefore mostly use the terms spectrum/spectra.

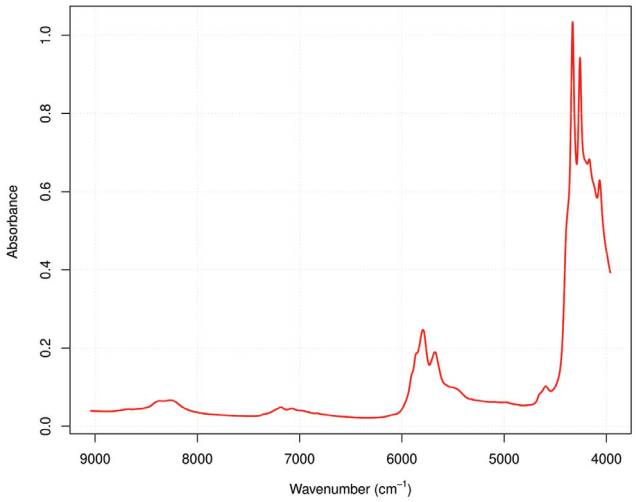

Figure 2.1 ~ *Example of a near-infrared spectrum.*

In addition, we resort to data simulation, to more accurately assess — and compare — the performance of different statistical models. It aims at mimicing real analytical data with a known generative model for the associated property. A key strength of simulated data is the ability to explore different hypotheses and settings, while knowing presicely the ground truth, which is not often accessible in actual chemical experiments. It also allows to produce variability in data generation.

Section 2.2 first provides background information on near-infrared spectroscopy before going into depth regarding the composition of D_{NIR} real data. Then, three examples of the simulation situations are described in Section 2.3 along with the associated algorithm. We conclude in Section 2.4.

2.2 Real data

The set D_{NIR} contains a two sets of information. First, it provides the density of 208 petroleum cuts, analyzed in [60]. This physico-chemical property plays a critical role in identifying the mixture functionality. It is defined as the mass per unit volume and measured at specified pressure and temperature as it varies when cuts are fluids. API (American Petroleum Institute) gravity is a commonly used index of the density of a crude oil or refined products. API gravity is thus an inverse measure of a petroleum liquid's density relative to that of water. A crude with a higher API is lighter (lower density). A crude that is heavier and/or denser has a lower API. It is commonly considered that lighter (high API) crudes are more desirable, as they allow refining to produce more valuable products.

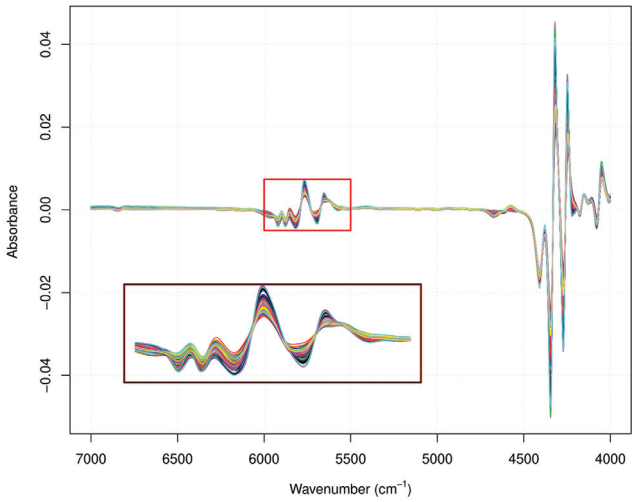

Figure 2.2 *~ First derivative of 208 NIR data spectra from dataset D_{NIR}. Right bottom subplot provides a clearer representation of overlaid spectra in wavenumber range from 7000 to 4000 cm^{-1}.*

Second, near-infrared spectra of the same 208 cuts are also included in D_{NIR}. It measures how much near-infrared radiation a matter absorbs. The ordinal variable region typically ranges from 800 nm to 3000 nm. NIR spectroscopy provides insight about molecule atom bonds, with variations in length and strength. Hence the frequency at which a given bond absorbs infrared radiation will differ over a range of bonds and modes of vibration. An infrared spectrometer examines a sample by exposing it to infrared light at a variety of frequencies, and detecting the absorptions caused by each type of bond in the complex. This produces a discrete spectrum representing the transmission intensity against the wavenumber [6]. Two organic compounds ought to exhibit different spectra.

As spectra span a relatively large range of intensity values, and may be affected by trends and noise that seem unrelated to the property of interest. It is customary to first remove artifacts and apply a diversity enhancement operator. Smoothed derivatives — such as obtained with Savitzky-Golay filters — may serve both purposes. The 208 different pre-processed spectra in D_{NIR} are depicted in Figure 2.2. Each is represented with a vector containing the transmission intensity of each wavenumber of the infrared range. In D_{NIR}, the latter browses 1557 wavenumbers.

To build a predictive model, all vectors are regrouped one matrix where each column covers one wavenumber (variable) and each row represents one selected portion of a spectrum (observation).

2.3 Simulated data

Chemometric high-dimensional functional data, like D_{NIR}, are generaly complicated and require an understanding of the underlying chemical processes. As sharable data is seldom, and the ground truth relating the dependent variables to the response is unknown, assessing and comparing the performance of statistical models is an uneasy task.

We therefore developed a tunable generative model allowing us to mimic different behaviors observed in chemometric high-dimensional functional data. It follows a standard sum-of-peak-like description, plus noise. There, random positive peaks are expected to represent quantities, somehow related to constituents (we intentionally keep this imprecise) of a given chemical sample. From that data (explanatory matrix $\mathbf{X}$), we may select a sparser support composed of non-contiguous segments. Such segments are highlighted in red in Figure 2.3. The contributions of values of $\mathbf{X}$ restricted to the limited support are then combined in a appropriate way (linearly with positive weights or through a monotonous function), to yield response $\mathbf{y}$. The motivation lies in the evaluation of the proper localization of coefficients for a given prediction method. Since the model is overdetermined, we expect feature selection to pick coefficients mostly inside the limited support. From a chemometrics point-of-view, this may help the interpretation of whose variables in spectra are most related to the macroscopic property, and possibly enhance the chimical insight.

On the one hand, for each row (observation) n of independent matrix $\mathbf{X}$, K Gaussians are mixed. We affect them the same scale σ, while their locations μ_k and amplitudes A_{nk} are randomly generated for $n \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$. The resulting mixture of N Gaussians are discretized, using a uniform sampling in P variables. In other words, each curve represents one row $\mathbf{x}_n$ for $i \in \{1, \ldots, N\}$ of matrix $\mathbf{X}$:

$$\sum_{k=1}^{K} A_{nk} \exp\left(-\frac{x - \mu_k}{2\sigma^2}\right). \tag{2.1}$$

evaluated on a regular grid from 1 to P indices/variables.

On the other hand, response $\mathbf{y}$ is generated. First, as we consider linear regression context, we emulate the response link to matrix $\mathbf{X}$ as a linear combination with a preset uncertainty ϵ. This allows us to evaluate prediction performance of models. Second, our aim is to evaluate models beyond prediction accuracy. We therefore measure model interpretability in addition to accuracy, with the hope of finding a balance between them two. Model interpretability is closely linked to sparsity, particularly in the context of high-dimensional data ($N \ll P$). When a regression model contains a large number of predictors, it becomes increasingly difficult to understand which predictors are contributing the most towards the outcome of interest. Sparsity refers to situations where the data contains many variables (or predictors) that have little to no impact on the outcome of interest $\mathbf{y}$. Thus the latter is simulated by imposing only $S \ll P$ positive weights in the linear link between $\mathbf{X}$ and $\mathbf{y}$ and $P - S$ null weights. Thus, only S variables (the cardinal of the limited support) are accountable in the construction of $\mathbf{y}$.

Three pratical scenarios were considered in the evaluation process. Hence, three data were simulated, denoted D_{SIM}, $\overline{D}_{\text{SIM}}$ and D^2_{SIM} and illustrated in Figure 2.3. Highlighted bands indicate positively weighted variables i.e. influential variables locations.

- D_{SIM} represents $N = 300$ mixtures (spectra) of $K = 30$ Gaussians discretized into $P = 1000$ variables. We set $\sigma = 0.05$ and $\epsilon = 0.5$. This simulated data is characterized by being functional and high dimensional with some peaks that appear on the set of spectra. Since the corresponding matrix of explanatory variables $\mathbf{X}$ is most generally singular, which hamper certain methods, we also conceived a non-singular set.

- $\overline{D}_{\text{SIM}}$ consists in a non-singular matrix of independent variables $\mathbf{X}$. It possesses $N = 200$ rows and $P = 50$ variables. Spectra are composed using $K = 100$ Gaussians of $\sigma = 0.01$ standard deviation. Response $\mathbf{y}$ is constructed linearly linked to $\mathbf{X}$ with some noise $\epsilon = 0.5$.

- To evaluate the group-lasso property of dual-SPLS (Chapter 4), D^2_{SIM} contains two independent sets of explanatory variables denoted $\mathbf{X}_1$ and $\mathbf{X}_2$ of $N = 300$. Both are linearly linked to a same response $\mathbf{y}$ with an uncertainty set to $\epsilon = 0.5$. $\mathbf{X}_1$ and $\mathbf{X}_2$ use respectively $K_1 = 10$ and $K_2 = 4$ Gaussians of standard deviation $\sigma_1 = 0.03$ and $\sigma_2 = 0.2$ and discretized spectra into $P_1 = 5000$ and $P_2 = 2000$ variables.

Finally, to evaluate the proposed calibration/validation method CalValXy (Chapter 3), we also resort to non-linear response generation, namely using the square root of the amplitude of the positive Gaussian peaks.

2.4 Conclusion

In conclusion, this chapter presented high-dimensional real petroleum-related spectra D_{NIR} collected from IFPEN, as well as simulated data. These data sets are highly valuable for conducting predictive modeling and analysis, which can help to understand the composition of petroleum samples and optimize production processes. The datasets help ensuring the accuracy and interpretability of the results. In the following chapters, several machine learning algorithms were tested on the datasets, and their performance was evaluated using various metrics for prediction precision and sparsity efficiency. The availability of these data sets will enable researchers and industry professionals to perform comparative analyses of predictive models. D_{NIR} and the data simulation algorithm are already available in the R package dual.spls.

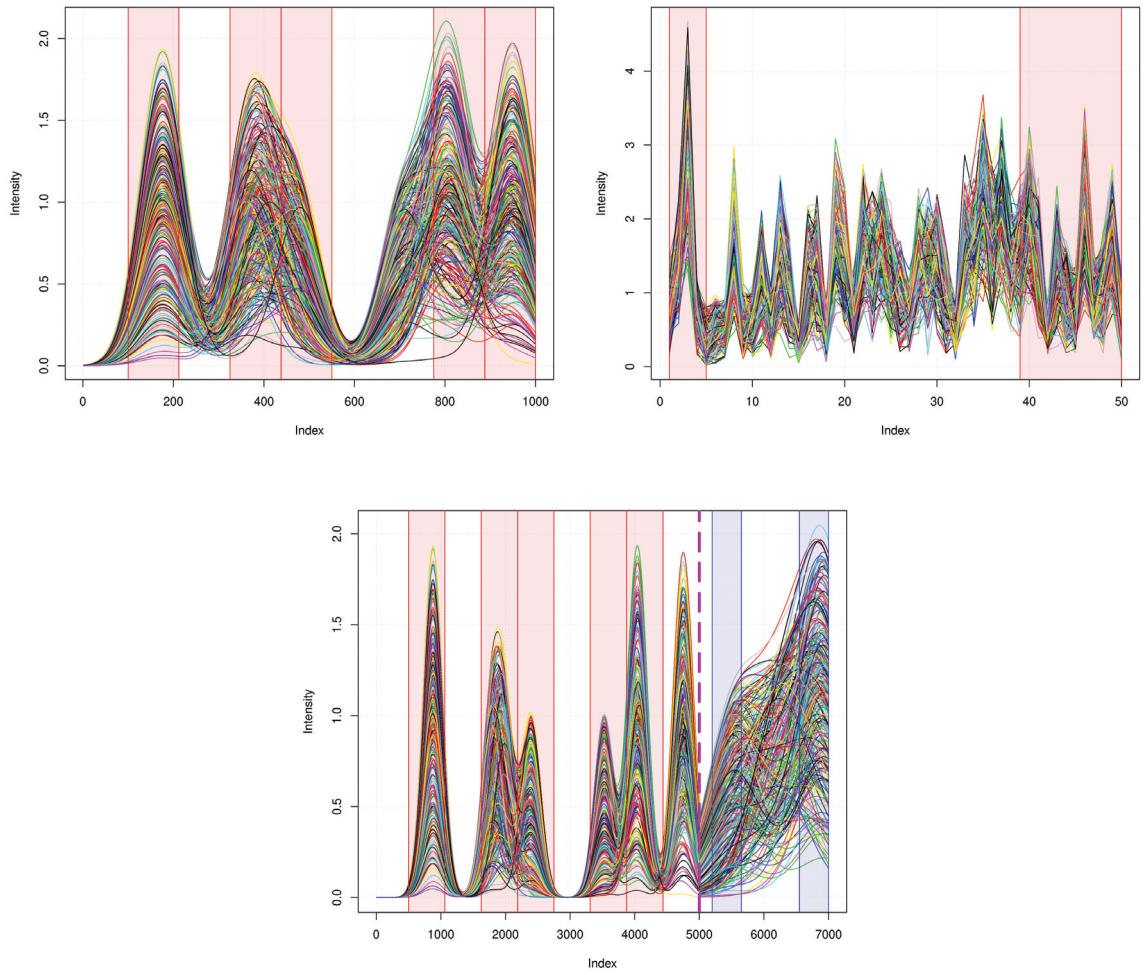

Figure 2.3 ~ *Simulated data. (Top left)* D_{SIM}*, (top right)* $\overline{D}_{\mathrm{SIM}}$ *and (bottom)* D^2_{SIM}*. Each curve is stored in a row in explanatory matrix* $\mathbf{X}$ *used to predict a simulated vector* $\mathbf{y}$*. Highlighted areas represent the only variables of X linked to y.*

3

Calibration and Validation with CalValXy

This Chapter is dedicated to the detailed presentation of CalValXy splitting method for which a preliminary version is available in [2]. Calibration and validation splitting helps in assessing models. In a regression context, disadvantage of state-of-the-art methods is ignoring response $\boldsymbol{y}$ *while partitioning data. CalValXy steps up and takes into account the overall information. Recent techniques are contrasted with numerical results obtained from simulated and and real benchmark datasets. This chapter is a preprint and is aimed to be submitted to Technometrics for June 2023.*

Contents

3.1 Introduction

In regression analysis, splitting a dataset into calibration and validation sets is a common practice to evaluate the performance of a model. First, the calibration (also called training or estimation) set is used to train the model and to estimate its coefficients. Then, the validation set is used to assess how well the model is performing. More formally, the trained model is evaluated on the validation data. The error measuring the difference between the predicted and actual values provides some hint on the accuracy performance of the trained model. The primary purpose of this procedure is to prevent overfitting, which occurs when a model is too complex and fits the noise in the data rather than the underlying trend or pattern. In this context, a real challenge in tuning a model is to properly select the validation points to be as representative as possible of the whole dataset.

In the present work, we consider a dataset of N points described by P factors and stored in a matrix $\mathbf{X} \in \mathbb{R}^{N\times P}$. For any $n \in \{1,\dots,N\}$, the n-th row of $\mathbf{X}$ represents an observation denoted by $\mathbf{x}_n = (x_{n1},\dots,x_{nP})$. The latter is associated to a response (or dependent) variable y_n displayed in a vector $\mathbf{y} \in \mathbb{R}^N$. Calibration and validation splitting can be used in various types of predictive models, including linear regression, logistic regression, neural networks, and many others. In this paper, we focus our attention on linear regression modeling of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\,, \tag{3.1}$$

that relates the response variable $\mathbf{y}$ to the matrix of independent variables $\mathbf{X}$. Here, $\boldsymbol{\beta}$ is the vector of regression coefficients or parameters, and $\boldsymbol{\epsilon}$ is the error vector that captures the unexplained variation in $\mathbf{y}$ that is not accounted for by the predictor variables. Numerous techniques are commonly employed to construct sets for calibration and validation purposes, as the simple random sampling [56, 46], the Kennard-Stone or CADEX algorithm [54] and the SPlit technique [53] (see Section 3.2 for more details on these procedures). Our objective is to provide a splitting method that improves their performance while providing an additional qualitative description of the dataset. In particular, our aim is to take into consideration not only the matrix $\mathbf{X}$, but also the response vector $\mathbf{y}$.

The remaining of this paper is organized as follows. In Section 3.2, we briefly recall some splitting techniques that will be useful to describe the new design in Section 3.3. In Section 3.4, we illustrate its efficiency through simulations on high dimensional databases. Conclusions are drawn in Section 3.6. The algorithm is available in an R package that we present at the end of the paper.

3.2 Review of splitting techniques

In this section, we succinctly overview some splitting techniques available in the literature.

The easiest way to split a dataset into calibration and validation sets is the simple random sampling technique (SRS) [56, 64, 62, 27]. In SRS, the selection of the calibration set is done at random and each individual in the population has an equal chance of being picked. This procedure is considered as a fair and unbiased method of sampling which makes it commonly used in scientific research and opinion polls. However, its implementation may not ensure that

all subgroups in the population are represented equally in the sample retained for calibration, especially when these subgroups represent small percentages of the whole population.
To improve the representativeness and accuracy of the calibation set, a stratified sampling [46] can be perfomed. This technique first divides the population into non-overlapping subgroups, called strata, based on certain characteristics of the candidates. Then, a random sample is taken from each stratum, which usually is proportional to the size of the stratum in the overall population. Stratified sampling is more efficient than simple random sampling, especially when the population is heterogeneous with high variability in the variable of interest. Nevertheless, it can be challenging to identify the right characteristics to use and that reflect the underlying population accurately. Clearly, stratified sampling requires additional effort to implement than SRS especially when the strata are small and/or numerous.
More sophisticated splitting procedures have been developped, of which the Kennard and Stone (also called CADEX for Computer-Aided Design of Experiment) algorithm [54]. It is based on distance computation between observations in the $\mathbf{X}$-space. It begins by selecting the two points that are the farthest apart, and then adds new points one at a time until the desired subset size is reached. At each step, the new added candidate is the one that is farthest from all of the previously selected ones. The main advantage of CADEX is that it is relatively easy to implement and can be applied to a wide range of data types. However, it is sensitive to outliers. This can result in the selection of outliers as representative samples, which can negatively impact the accuracy of the model or analysis. An extension to CADEX, called DUPLEX algorithm, was poposed in [88]. It alternates the constructions of both the calibration and validation sets as follows. First, the two points which are farthest apart are assigned to the calibration set while the two points in the remaining list which are farthest apart are assigned to the validation set. At the second step, the point which is farthest from the two points in the calibration set is added to the calibration set and the point which is farthest from the two points in the validation set is included in this set, and so on until all points in the list have been assigned to one of the two sets. Unlike the previous technique, DUPLEX distributes the extreme points between both sets.
A totally different procedure was recently poposed in [53] and named SPlit (for Support Points-based split). Computations are done using both $\mathbf{X}$ and $\mathbf{y}$ by merging them in one big matrix. It creates the set with the smallest cardinality between the training and the testing sets in two steps. The first one consists in finding the values of a set of variables that minimize an energy distance defined in [92]. The second step is a sequential nearest neighbor search to select the representative points from the dataset. Nonetheless, SPlit algorithm can focus on constructing the validation set and thus neglects the calibration set. This may deliver a model that might not be well calibrated. A faster version of SPlit which enables its application to big data problems was proposed in [52].
The common thread between these splitting techniques and the algorithm we present in the upcoming section, is to select calibration sets that cover the experimental space as uniformly as possible and provide accurate regression predictions. To check if these objectives are met, two criteria will be investigated in the experiments of Section 3.4. First, we will calculate Euclidean distances and ϕ_2 distances to evaluate the similarity between the calibration set and the whole dataset. Then, we will compute the root mean squared error when performing two types of regression that are well known in the chemometrics field: the Partial Leaset Squares [104] and

least absolute shrinkage and selection operator [95].

3.3 CalValXy: algorihm description and example

CalValXy splits datasets using both the predictors in $\mathbf{X}$ and their outcomes in $\mathbf{y}$. The principale (see Section 3.3.1 for a formal definition) is the following. Observations are initially stratified according to $\mathbf{y}$ values[(1)]. Then, strata are sampled dynamically, along the lines of the CADEX algorithm, using the $\mathbf{X}$'s values. In particular, we aim to select a calibration set that efficiently represents the overall observations in the $\mathbf{X}$-space, while using underlying patterns in the response vector for more accurate predictions.

3.3.1 CalValXy description

Let $\mathbf{V} = \{v_1, \ldots, v_N\}$ initially denote the set of the N "values" combining $\mathbf{X}$ and $\mathbf{y}$. For any $n \in \{1, \ldots, N\}$, $v_n = (\mathbf{x}_n, y_n) \in \mathbb{R}^P \times \mathbb{R}$. This set will be deflated iteratively by picking appropriate calibration points. They will be assigned to a growing calibration set $\mathbf{C}$, initialized as the empty set $\varnothing$. The final deflated $\mathbf{V}$ will contain the validation points. Free parameters are $N_c < N$, the final cardinal of $\mathbf{C}$ and a number $K \in \mathbb{N}^*$ of (expertise-based or) data-based classes. Let $d_{\mathbf{X}}$ represents a distance function rescricted to the space of $\mathbf{X}$, i.e. between the $\mathbf{X}$-values of observations. The appropriate distance may be selected inside a quantity of candidates [34], provided that it is adapted to the data structure (for instance a Riemannian metric). We define $\mu(.)$, a measure of the central tendency of a finite set of points. It may belong to generalized means, medoids (for robustness to outliers) or be derived from $d_{\mathbf{X}}$. Note that $\mathbf{X}$ may represents either the original data, or a transformation thereof; the latter may be a standard distance-preserving decomposition — Principal Component Analysis (PCA), Fourier or wavelet representations —, a feature selection or a dimension reduction method. It may prove useful for initial attribute extraction or computation speedups.
We chose here for simplicity the Euclidean distance: or ℓ_2-norm for points $\mathbf{u}_i$ and $\mathbf{u}_j$ in $(\mathbb{R}^P \times \mathbb{R})$, their "restricted distance" is defined wit respect to their first P dimensions, as $d_{\mathbf{X}}(u_i, u_j) = \sqrt{\sum_{p=1}^{P}(u_{i,p} - u_{j,p})^2} = \|\mathbf{u}_i - \mathbf{u}_j\|_2$. Consequently, we choose $\mu(.)$ as the standard centroid, a common choice in chemometrics and quite natural for the Euclidean distance.

Step 0: response vector $\mathbf{y}$ *stratification in* K *classes* ~ This step aims to segment the observations into K non-overlapping classes, according to the response pattern. The classes are meant to use explicit or implicit domain-related information. For instance, observations may be labeled, offering a meaningful segmentation. As for $\mathbf{X}$, in a problem-dependent manner, one may also transform the response $\mathbf{y}$, or focus on gaps/distances between their values. This results in a collection of mutually disjoint $\mathbf{V}_{k,\,1\leq k\leq K}$ classes whose union span all observations. Such specific categorie are not alxways available. However, the range of a scalar response $\mathbf{y}$ is generally easily characterized by natural description based on the location of values in the range. One often resorts to a number of named characterizations with a semantic gradient (FIND THE PROPER): low/mid/high, cold/tepid/warm/hot, etc. When the boundaries are not known, we

(1) Strata may also incorporate additional a priori labelling knowledge on the data.

suppose at least that their approximate quantity K can be provided. We therefore stratify as follows. Values of $\mathbf{y}$ are first sorted (in increasing order). We then split the interval $[y_{\min}, y_{\max}]$ into K contiguous subintervals with endpoints $y_{\min} = r_0 < r_1 < \cdots < r_{K-1} < r_K = y_{\max}$. We define $\mathbf{R}_k = [r_{k-1}, r_k[$ for $1 \leq k < K$ and $\mathbf{R}_K = [r_{K-1}, r_K]$. Therefore, $[y_{\min}, y_{\max}] = \cup_{k=1}^{K} \mathbf{R}_k$ and we define $\mathbf{V}_k = \{v = (\mathbf{x}, y) \in \mathbf{V} | y \in R_k\}$.
This is illustrated on a uniform partition with $K = 10$ in Figure 3.1 for dataset D_{NIR} described in Section 3.4.

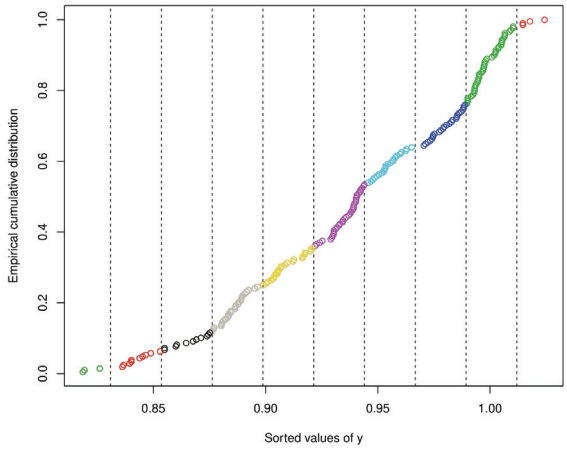

Figure 3.1 *~ Splitting a response vector* $\mathbf{y}$ *uniformly into* $K = 10$ *intervals with different colors. Dotted vertical lines denote subinterval boundaries according to the sorted values of* $\mathbf{y}$ *(horizontal axis).*

The next steps use restrictions of the observations on the $\mathbf{X}$-space.

Step 1: starting calibration point selection ~ We first select a seed c_0 as a reference location for dataset $\mathbf{X}$. It can be chosen at random in the convex hull of the dataset. As it is common is calibration/validation selection, we somehow look for extreme dispersions with resepct to the dataset. We suggest to choose it as a measure of the central tendency $c_0 = (\mu(\mathbf{X}), \mu(\mathbf{y}))$ (for dimension consistency: the last coordinate does not play an active role). Then, the first calibration point c_1 is chosen among the farthest observations from c_0, according to distance $d_{\mathbf{X}}$. For multiple candidates, we pick by convention the one with the lowest index. It is removed from the appropriate $\mathbf{D}_{k(1)}$ and added to the (empty) calibration set $\mathbf{C}$.

Step 2: iterative calibration point selection ~ We assume that we have successively determined the first $n < N_c$ calibration points $c_1, \ldots, c_n$, c_n formerly in $\mathbf{V}_{k(n)}$ before being appened to $\mathbf{C}$. Remember that $\mathbf{V}_k$ subsets are dynamically shrunk, and may become empty. We define the following search rule: the successor $\mathbf{V}_{k(n+1)}$ of $\mathbf{D}_{k(n)}$ is the next non-empty subset, assuming a cyclic indexing: V_1 "follows" V_K.

1. Compute the compound distance $\Delta(v, \mathbf{C})$ between each point v of $\mathbf{V}_{k(n+1)}$ and the calibration set $\mathbf{C}$ (equal to $\{c_1, \ldots, c_n\}$ at this time)

$$\Delta(v, \mathbf{C}) = \min\{d_{\mathbf{X}}(v, c_1), \ldots, d_{\mathbf{X}}(v, c_n)\} \quad \forall v \in \mathbf{V}_{k(n+1)}.$$

2. The observation $v \in \mathbf{V}_{k(n+1)}$ with the largest value $\Delta(v, \mathbf{C})$ (and the least index) is removed from this set and added to $\mathbf{C}$ as the $n+1$-th calibration point c_{n+1}.

Step 2 is iterated until the desired number N_c of calibration points in $\mathbf{C}$ is reached. The validation set is therefore its complement with respect to the whole dataset, or equivalently the union of the residual shrunk $\mathbf{V}_k$s. The whole CalValXy procedure is summarized as pseudo-code in Algorithm 2. Along the lines of [54], the first iterations are given for a simple illustrative example, presented in Section 3.3.2.

Algorithm 2: CalValXy: Constructing calibration and validation sets from a predictor matrix $\mathbf{X}$ and a response vector $\mathbf{y}$

Input: $\mathbf{X}$, $\mathbf{y}$, $N_{\mathbf{C}}$, K, $d_{\mathbf{X}}$ (metric), $\mu(.)$
Split $(\mathbf{X}, \mathbf{y})$ into K classes $\mathbf{V}_k$ (e.g. using $\mathbf{y}$)
$c_0 = \mu(\mathbf{X})$
$\mathbf{C} = \varnothing$ and $\mathbf{V} = (\mathbf{X}, \mathbf{y})$
Determine $c_1 = \text{argmax}_{v \in \mathbf{V}} d(v, c_0)$
Find $k(1) | c_1 \in \mathbf{V}_{k(1)}$
for $n \leftarrow 1$ to N **do**
  $\mathbf{V}_{k(n)} \leftarrow \mathbf{V}_{k(n)} \setminus c_n$
  $\mathbf{C} \leftarrow \mathbf{C} \cup c_{n-1}$
  Find next none empty $\mathbf{V}_{k(n+1)}$ set (cycling indices)
  $c_{n+1} = \text{argmax}_{v \in \mathbf{V}_{k(n+1)}} d(v, \mathbf{C})$
end for

3.3.2 Simple example of the design procedure

To better understand how CalValXy operates, we apply it to an example drawn from [54]. Consider a factorial structure of the form 5^2 where the dataset of independent variables, stored in $\mathbf{X} \in \mathbb{R}^{25 \times 2}$, is represented in Figure 3.2. To apply CalValXy algorithm, a response vector $\mathbf{y}$ is required. For the sake of the example simplicity, a pattern of three groups of relatively close values is set. Figure 3.2-(right) shows the response values of each observation, in coherence with the factorial design represented in Figure 3.2-(left).
According to the pattern of the response, we set the number of strata to be $K = 3$. Often, the percentage of calibration is set to 80 % hence, here we would like to select namely $N_c = 20$.
After sorting $\mathbf{y}$, observations are consequently rearranged and listed as follows

$$\begin{aligned}\mathbf{V} = (&v_{19}, v_4, v_{16}, v_9, v_1, v_{14}, v_{21}, v_{11}, v_6, v_{24}, v_{20}, v_7,\\ &v_5, v_{15}, v_{10}, v_{12}, v_{22}, v_{25}, v_2, v_{17}, v_3, v_8, v_{18}, v_{23}, v_{13}).\end{aligned}$$

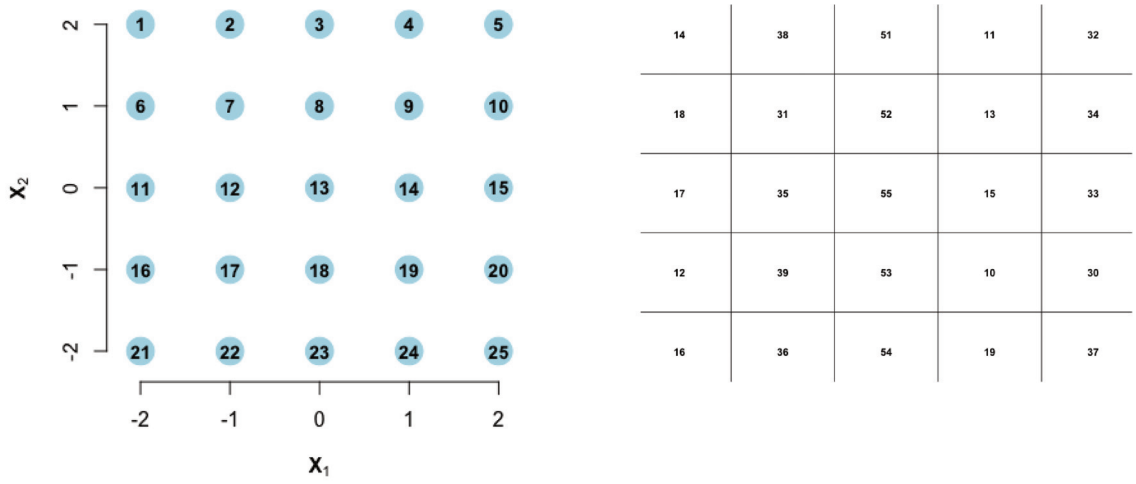

Figure 3.2 ~ *Data used for the example: factorial structure of the form* 5^2 *with two variables (left); associated response values* $\mathbf{y}$ *(right).*

The values in $\mathbf{y}$ vary between 10 and 59. The three classes composed are $[10, 26[$, $[26, 42[$ and $[43, 59]$. Thus, the three subsets as represented in Figure 3.3 are:

$$\begin{aligned}\mathbf{V}_1 &= (v_{19}, v_4, v_{16}, v_9, v_1, v_{14}, v_{21}, v_{11}, v_6, v_{24}),\\ \mathbf{V}_2 &= (v_{20}, v_7, v_5, v_{15}, v_{10}, v_{12}, v_{22}, v_{25}, v_2, v_{17}),\\ \mathbf{V}_3 &= (v_3, v_8, v_{18}, v_{23}, v_{13}).\end{aligned}$$

Due to the strong symmetry of the data, it is natural to choose $\mu(.)$ to be the centroid. In the $\mathbf{X}$-space, the centroid $\mathbf{G}$ corresponds exactly to observation o_{13}. The farthest points to $\mathbf{G}$ are o_1, o_5, o_{21} and o_{25}. Choosing the observation with smallest index, o_1 is selected as the starting point. We pursuit by using [54] notations, hence, the first design point is:

$$\texttt{Point 1} - v_1 - (-2, 2)$$

and belongs to subset $\mathbf{V}_1$. It is removed from its class and assigned to to $\mathbf{C}$. For the second calibration point, we consider the subset $\mathbf{D}_2$. By computing the distance between each point in $\mathbf{V}_2$ and v_1, the observation with the largest distance is identified to be v_{25}. This is detectable when looking at Figure 3.2. Thus, the second design point is

$$\texttt{Point 2} - v_{25} - (2, -2)$$

Now consider $\mathbf{V}_3$ that contains five candidates. We compute the distances between each of its points and the calibration ones, v_1 and v_{25}. and their corresponding Δ. The third design point is set to be

$$\texttt{Point 3} - v_{13} - (0, -2)$$

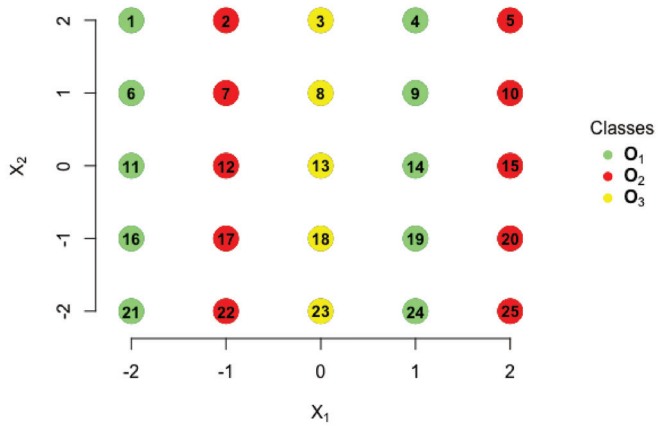

Figure 3.3 ~ *Classes partition of factorial structure* 5^2 *with two variables according to the response values.*

Iteratively, the above is repeated until N_c points have been selected. Figure 3.4 represents the final calibration and the validation sets.

3.4 Numerical experiments and discussion

A certain diversity in performance assessment matters for fair evaluation. We therefore evaluate CalValXy with respect to the standard SRS [56], CADEX [54] methods and the more recent SPlit [53] described in Section 3.2[(2)]. We study whether the resulting calibration/validation splitting can be beneficial. A first objective is the representativity of the calibration set $\mathbf{C}$, regarding the independent variables $\mathbf{X}$; we use two classical set comparison indices. A second objective is the prediction accuracy for the dependent variable $\mathbf{y}$; we employ two reference inference methods: PLS and lasso. The above methodology is deployed on two datasets. The first one D_{SIM} is a mixture of Gaussians, inspired by chemometrics, generated stochasticaly with dependent variable $\mathbf{y}$ related to $\mathbf{X}$ with a controlled non-linear model, to assess variability in assay realizations. The other, D_{NIR}, comes from a real analytical chemistry database, MLnir. The latter, to be published in [38], is already used in [3]. They are described respectively in Sections 3.4.2 and 3.4.3.

3.4.1 Methodology

As a large number of variables are involved, we have performed for each dataset an initial dimension reduction on $\mathbf{X}$ with PCA [107] for the calibration/validation sets selection only.

[(2)] We use their implemetation from R packages: split, prospectr.

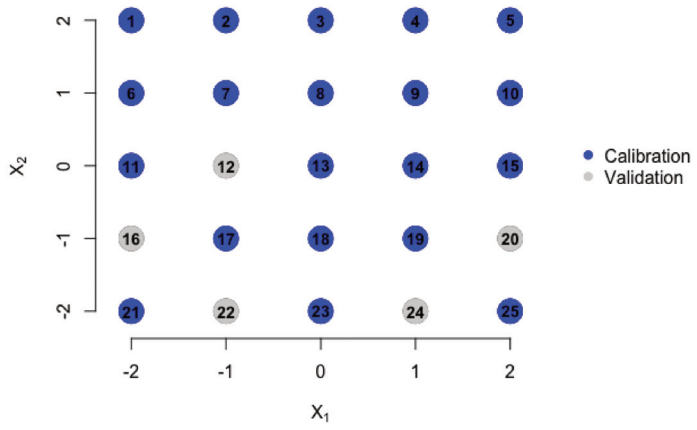

Figure 3.4 ~ *CalValXy splitting of factorial structure* 5^2 *with two variables with* 80% *calibration.*

This amounts to transformation eluded to in Section 3.3.1. Its roles are thrice: feature selection coupled with dimension redeuction through an orthogonal transformation. We chose to retain the components which account for 99.900 % of the variance of each dataset. The resulting dimension — typically about 2 to 5 — accelerates the extensive computations of distances.
On the one hand, we consider evaluating similarity between the calibration sets and the original datasets. The objective is to determine to what extend the –smaller– calibration set, denoted $\mathbf{C}$, is a good representative of the dataset. This is measured using the following two numerical indicators:

- The ratio of Euclidean distances:

$$r_d = \frac{\sum_{c,c' \in \mathbf{C}} d_{\mathbf{X}}(c,c')}{\sum_{o,o' \in \mathbf{X}} d_{\mathbf{X}}(o,o')}, \tag{3.2}$$

  A larger value of the ratio of the Euclidean distances shows a greater repulsion between the selected points.

- The ratio of the ϕ_2 distances:

$$r_{\phi_2} = \frac{\phi_2(\mathbf{C})}{\phi_2(\mathbf{X})}, \tag{3.3}$$

  where, for a dataset of N_A points, $\mathbf{A} = \{o_1, \ldots, o_{N_A}\}$ s.t. for any $a \in \{1, \ldots, N_A\}, o_a = (\mathbf{x}_a, y_a) \in \mathbb{R}^{P+1}$,

$$\phi_2(\mathbf{A}) = \left[\sum_{i=1}^{N_A} \sum_{j=1}^{i-1} d_{\mathbf{X}}(o_i, o_j)^{-2} \right]^{1/2},$$

Smaller ratio of the r_{ϕ_2} distances corresponds to a more uniformly spaced design over the set of candidates. This indicator is closely related to maximizing Euclidean distances between the points. In order to simplify the criterion comparison, we compute $1 - r_{\phi_2}$ and aim at a large value.

On the other hand, since calibration and validation splitting is specially useful in regression, we evaluate our algorithm in terms of prediction accuracy. We recall the linear model introduced in 3.1. We choose to apply Partial Least Squares regression (PLS) [104, 106] popularly used with functional chemometric data like D_{NIR}. It is a statistical technique that consists on projecting data into a lower dimensional space composed of latent components that explain the maximum covariance between the two sets. We also consider the least absolute shrinkage and selection operator (lasso) [95] algorithm which adds an ℓ_1 penalization to the original least squares optimization. It uses a regularization parameter that is usually selected by cross validation [91]. In both cases, Predictions are evaluating by computing the Root Mean Square Error (RMSE) of validation. For the validation set $\mathbf{V} = \{v_1, \dots, v_{N_V}\}$:

$$RMSE = \sqrt{\frac{1}{N_V} \sum_{n=1}^{N_V} (\hat{y}_{v_n} - y_{v_n})^2},$$

where y_{v_n} is the response value of validation observation v_n and $\hat{y}_{v_n}$ is its predicted value. Hereafter, we describe each database that are at the core of our simulations.

3.4.2 Simulated dataset $\mathcal{D}_{\text{SIM}}$: Gaussian mixtures

Simulated data are used to test the validity and reliability of models, and to explore how associated methods perform under different conditions and several scenarios. Thus, for a stable evaluation of CalValXy, $N = 300$ positively weighted mixture of $G = 10$ Gaussian peaks are generated with preset scale $\sigma^2 = 0.05$, randomly picked amplitudes A_{ig} and locations μ_g, for $n \in \{1, \dots, N\}$ and $g \in \{1, \dots, G\}$. Each row (observation) $\mathbf{x}_n \in \mathbb{R}^P$ for $n \in \{1, \dots, N\}$ of the associated $\mathbf{X}$ explanatory matrix is formulated as:

$$\mathbf{x}_n = \left(\sum_{g=1}^{G} A_{ng} \exp\left(-\frac{(x_p - \mu_g)^2}{2\sigma^2}\right) \right)_{p \in \{1,\dots,P\}}, \tag{3.4}$$

where $\{x_1, \dots, x_P\}$ are a uniform discretization of range $[0, 1]$. Response vector $\mathbf{y}$ is defined by an explicit linear model composed of weighted sums of the squared root of $\mathbf{X}$ values i.e. $\mathbf{y} = \sqrt{\mathbf{X}}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \frac{1}{2})$. Weights defining $\boldsymbol{\beta}$ are fixed quantities by range of indices[(3)]. We denote this kind of simulated data D_{SIM}. For more accurate and reliable evaluation, we use the Monte Carlo method that proposes to simulate multiple D_{SIM} data and assess the criteria values for each or take their average. We choose to simulate one hundred D_{SIM} where for each, $N = 300, P = 1000, G = 10, \sigma^2 = 0.05, \sigma_{\boldsymbol{\epsilon}}^2 = 0.5$ and $\boldsymbol{\beta}$ remains as specified in the supplementary material. Each simulation is then different by the Gaussians randomly chosen A_{ig} and locations μ_g for $n \in \{1, \dots, N\}$ and $g \in \{1, \dots, G\}$. An example of D_{SIM} data is represented in Figure 3.5.

[(3)] Associated code is provided as supplementary material

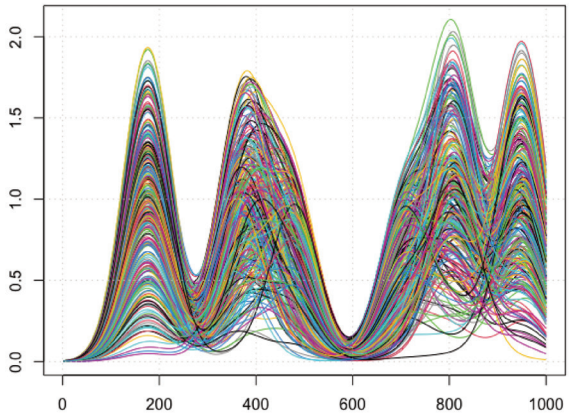

Figure 3.5 ~ *Simulated data $\mathcal{D}_{SIM}$. Each curve corresponds to a line of the matrix* $\mathbf{X}$.

The metrics r_d and $1 - r_{\phi_2}$ are displayed in Figure 3.6 using violin plots. This type of data visualization displays the distribution of numeric data by combining a box plot with a kernel density plot. It provides insights into the shape, central tendency, and spread of the data. For both criteria CalValXy outperforms considerably SPlit and random sampling in terms of average value worst-case performance. Althought CalValXy and Kennard and Stone violon plots are close, we notice that an improvement where each quartile represents the highest value with the new approach. This indicates that CalValXy selects a calibration set more uniformly spaced and that represents well the overall population by nominating observations that are as far away as possible from each others.

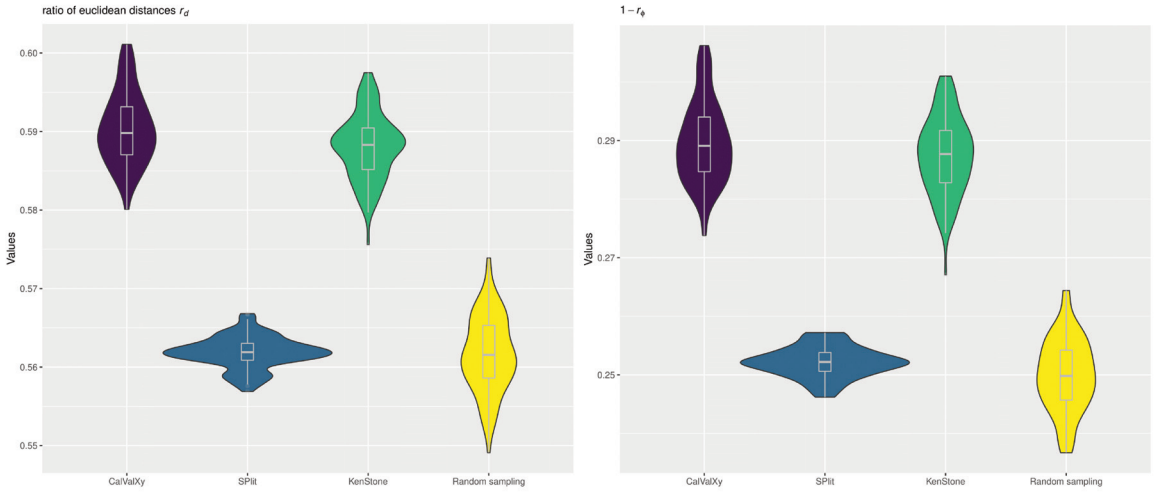

Figure 3.6 ~ *CalValXy evaluation on simulated data $\mathcal{D}_{SIM}$. Values of r_d (left) and $1 - r_{\phi_2}$ (right). Splitting methods from left to right: CalValXy, SPlit, Kennard and Stone, and random sampling.*

The validation RMSE values are shown in Figure 3.7. For PLS regression (left), average values are close for the four splitting methods. However, variability is much smaller for CalValXy compared to its challengers. This implies that the RMSE values of the new approach are close to each other, suggesting that they are more reliable and precise with higher degree of confidence in the accuracy of the measurements. For lasso regression (right), we can notice a large improvement on accuracy on the average level of the violon plots where RMSE for CalValXy is the lowest. We also note significant enhancement in the worst-case performance of CalValXy over other splitting methods. Thus, CalValXy produces calibration and validation sets that clearly improve prediction and model fitting on the simulated data D_{SIM}.

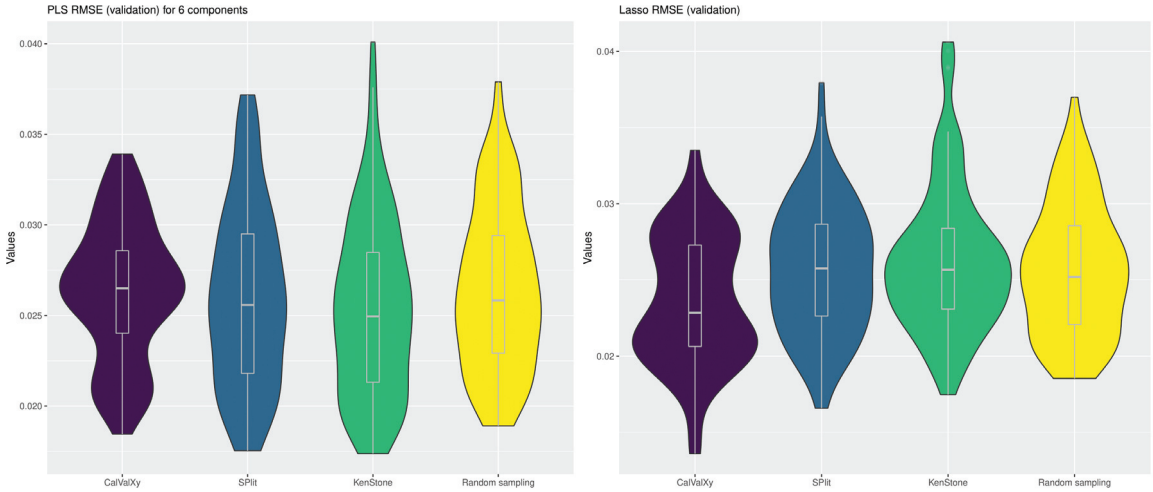

Figure 3.7 ~ *CalValXy evaluation on simulated data $\mathcal{D}_{SIM}$. Validation RMSE values of PLS for six components (left) and lasso regression (right). Splitting methods from left to right: CalValXy, SPlit, Kennard and Stone, and simple random sampling.*

3.4.3 Real data $\mathcal{D}_{\text{NIR}}$: Near-infrared spectroscopy

The real data considered here consists of near-infrared (NIR) spectra of hydrocarbon samples. It is based on the principle of absorption of radiation (infrared) by matter. NIR spectroscopy is the most frequently used approach to characterize heavy oil products. Radiation absorption by oil samples depends on their composition. More precisely it is linked to chemical bonds. An exhaustive literature on these techniques can be found in [23, 65]. The data set D_{NIR} considered in this Section was provided by IFPEN. It was partly exposed in [59], is available at `http://www.laurent-duval.eu` and will be subject to a forthcoming publication [38].
D_{NIR} is composed of 208 samples described by 1557 variables regrouped in $\mathbf{X}$. The latter actually represents pre-treated NIR raw spectra with a derivate using Savitzky Golay smoothing [83] to reduce additive and multiplicative effects. We aim to explain the density property $\mathbf{y}$, a physico-chemical characteristic that is most frequently used to describe oil. Density measurement is widely used to characterize oil cuts because its efficiency and precision. A graphical representation of $\mathcal{D}_{\text{NIR}}$ is plotted in Figure 3.8.

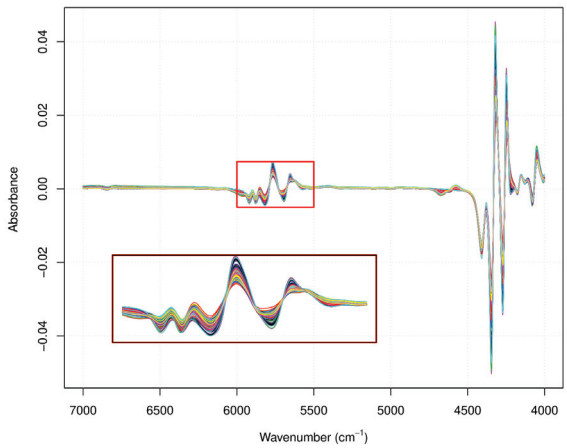

Figure 3.8 ~ *First derivative of 208 NIR data spectra from dataset $\mathcal{D}_{NIR}$. Each curve is stored in a row in explanatory matrix X. Right bottom subplot provide a clearer representation of overlaid spectra in wavenumber range from 6000 to 5500 cm^{-1}.*

Since we have a single data set, variability can not be evaluated like in Section 3.4.2, thus, the representation of the results will differ. Indeed, here, for RMSE computations, with PLS regression, we vary latent components from 1 to 10 and assess each case. Table 3.1 displays the computed values of r_d and $1-r_{\Phi_2}$. We can see that values for CalValXy and Kennard and Stone splitting are relatively close and both outperform SPlit and SRS. Furthermore, Table 3.1 shows the RMSE values obtained when performing a lasso fitting. We notice that CalValXy, SPlit and Kennard and Stone algorithm exhibit good performance with close values, much smaller than random sampling. However, for PLS regression, the improvement using the new approach is considerable. Figure 3.9 entails that the accuracy, measured by the RMSE, globally improves as the number of latent variables increases for all four splitting methods. From five to ten latent variables, all curves tend to plateau. CalValXy provides the best results (i.e. the lowest curve) which shows that the new approach seems more reliable in predicting density using chemical data.

	CalValXy	SPlit	CADEX	SRS
r_d	0.6224	0.5572	0.6059	0.5715
$1-r_{\phi_2}$	0.5490	0.2669	0.5875	0.3369
PLS RMSE	$[0.0056; 0.003]$	$[0.0047; 0.0043]$	$[0.0058; 0.0042]$	$[0.0062; 0.0061]$
Lasso RMSE	0.0041	0.004	0.0046	0.0073

Table 3.1 ~ *Computation of criteria r_d and $1-r_{\phi_2}$ and RMSE ranges of PLS and values of lasso regressions for the different splitting techniques using $\mathcal{D}_{NIR}$.*

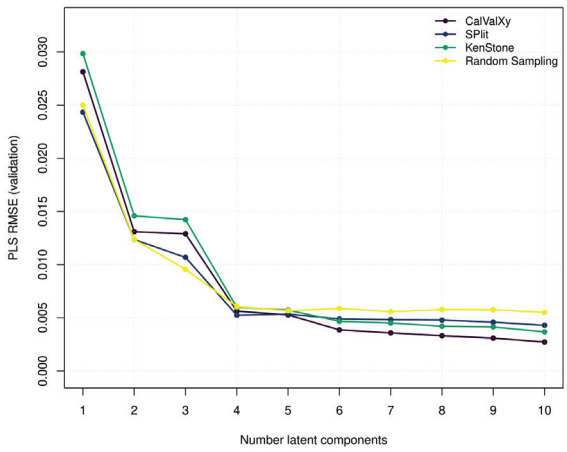

Figure 3.9 *~ RMSE values with respect to the number of latent components for the different splitting techniques using $\mathcal{D}_{NIR}$.*

3.5 R package for CalValXy

We developped an R-Studio package that splits a dataset into two samples using the CalValXy algorithm. One sample is used for calibration and the other one to perform PLS, Dual-PLS and other types of regression. The package [5, 4] also provides a funtion that simulates mixtures of Gaussians to build sets of predictors and responses like the $\mathcal{D}_{\text{SIM}}$ dataset.

3.6 Conclusion

We proposed a new algorithm called CalValXy that splits a dataset into calibration and validation sets. It stratifies the response vector and applies the main steps of the Kennard and Stone algorithm to the corresponding strata of the predictor matrix. We compared the new procedure to some usual splitting techniques through numerical simulations in which we computed several error estimates. Overall, our results showed that the derived combination gives very satisfactory results; in particular it provides calibration sets with better coverage of the initial database. We also tested the algorithm on regression models in which the Root Mean Square Error of predictions was computed to measure the expected uncertainty. The CalValXy approach outperforms the other splitting techniques, regardless the number of components.
Calibration and validation splitting requires to preset the number of calibration observation N_C. The latter can impact the accuracy and reliability of the process. If too few calibration samples are used, the model may not be able to capture the full range of variability in the data, resulting in underfitting. On the other hand, if too many calibration samples are used, the model may overfit the calibration data, leading to poor generalization to new data. Therefore, the choice of the number of calibration samples depends on the complexity of the model and

the size of the dataset. In general, the most common splitting ratio is to use 70% of the data for calibration and 30% for validation. However, this is not always the best choice, and other ratios may be more appropriate in some cases. CalValXy in particular divide the observation in prior into K classes according to the response $\mathbf{y}$. This number is also required to be chosen and depends on the response pattern. When $\mathbf{y} \in \mathbb{R}$, as in our cases, one may want to plot the empirical cumulative distribution of the $\mathbf{y}$ values in order to detect a certain pattern that can help in choosing the best K. Although CalValXy proposed a method to choose each class, the calibration and validation splitting can still be applied if user want to assign a category to each observation manually.
In chemometrics, regression techniques other than those considered in this paper have proven their effectiveness, in particular, [72, 7]. In future works, we plan to analyze the preformance of these regressions when using CalValXy to varied types of data.

3.7 Acknowledgements

This work was performed within the framework of the LABEX MILYON (ANR-10-LABX- 0070) of Universiteé de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR). We acknowledge the financial support of the Research Council of the Saint Joseph University of Beirut. Ghislain Durif contributed in the code validation procedure in R. IFPEN provided the real NIR data set used in the applications. Our mentor, colleague or friend Francçois Wahl passed away unexpectedly during the writing of this paper. He was the driving force of our team.

4 Dual sparse partial least squares

This chapter is dedicated to the development of a novel regression method called dual sparse partial least squares (Dual-sPLS), which aims to overcome the limitations of existing approaches in providing accurate predictions and pertinent interpretation of chemical data sources. The proposed method is a unified formulation for regression methods that blends dimension reduction and variable selection in a PLS formalism. It also allows for variable grouping and offers interpretable localization of features from a functional data or statistical point of view. The chapter provides detailed explanations of the principles behind the Dual-sPLS family, norm penalties, and algorithms, as well as benchmarking results of real and simulated data, concluding remarks, and supplementary material.

Contents

4.1 Introduction

Two main feats of chemometrics reside in first, providing reliable inference and second, offering interpretability of chemical data sources. On the one hand, one may expect to estimate, within a given precision, responses $\mathbf{Y} \in \mathbb{R}^{N\times Q}$ (e.g. hydrocarbon properties: viscosity, density, cetane number [78]) from spectra or variables represented by quantities $\mathbf{X} \in \mathbb{R}^{N\times P}$ (nuclear magnitude resonance or NMR, chromatography, infrared spectroscopy, etc. [101]). It aims at relating a target $\mathbf{Y}$ to $\mathbf{X}$ through a predictive model: for instance, NMR spectra can be linked to viscosity with predictive purposes. On the other hand, one also wishes to interpret how variables in $\mathbf{X}$ influence quantities $\mathbf{Y}$, i.e. which spectral features are most consistent with response prediction, a question related to wavelength selection. For instance, which spectral bands in NMR, in terms of continuous localization, could be related to the viscosity index estimation (see e.g. [98])? This can be transcribed by a regression model, often considered linear:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\,, \tag{4.1}$$

where $\boldsymbol{\epsilon}$ is expected to be independent of $\mathbf{X}$, with zero mean. With the growing size of consolidated analytical chemistry databases, chemometrics still require methodologies to 1) provide accurate predictions 2) extract pertinent knowledge or offer useful insights on measurements 3) combine heterogeneous or high-dimensional data sources. When the number P of variables (samples) is far greater than the number of observations (signals) N ($P \gg N$), naive statistical models risk overfitting. This notably happens in standard least squares optimizations. Dimension reduction techniques are generic approaches to deal with high dimensionality. They include projection methods or variable selection algorithms. Commonly used strategies start with PCA/PCR (principal component analysis/regression), performed only on explanatory variables in $\mathbf{X}$. They however do not incorporate information held by the response $\mathbf{Y}$. Partial least squares (PLS) [104, 106], also called projection onto latent structures, is therefore common in chemometrics, with better prediction-prone latent components. However, PLS sometimes lacks appropriate interpretability.

As for variable selection, one often resorts to the lasso algorithm (least absolute shrinkage and selection operator [95]). Shrinkage induces a form of sparsity, which amounts to selecting important variables. It is however known to be sensitive to data types. It does not always yield interpretable coefficients. Blends of the two above – dimension reduction and variable selection – have recent avatars called sparse PLS (sPLS). While they enforce lower dimensional decompositions, they do not always provide chemically pertinent feature localization for physico-analytical measurements. Thereby, we propose a dual sparse PLS family dedicated to one dimensional or univariate responses: $\mathbf{y} = \mathbf{Y}$, with $\mathbf{y} \in \mathbb{R}^N$. It generalizes the standard PLS1 algorithm by supplementing it with adequate penalties. This formally provides a unified formulation for regression methods in the spirit of the lasso mentioned above, and also least-squares or ridge, all blended in a PLS formalism. It also allows *variable grouping*: the possibility to gather explanatory variables into more meaningful subsets (contiguous samples around a peak, disjoint spectral bands associated to a compound). This can be used to combine different physico-chemical modalities. Resolution resorts to the dual norm of the chosen Dual-sPLS penalty. This new method has many advantages:

1. predictions match or outperform state-of-the-art or comparable methods,

2. in the different norm options we considered, they additionally yield sparse representations of both simulated and real chemical near infrared data, even singular, a frequent ill-conditioning issue in high dimension,

3. they finally offer a interpretable localization of features from a functional data or statistical point of view.

Those three properties combined offer alternative surrogates to classical approaches (PLS, lasso, least squares, ridge). It permits both accurate inference and pertinent domain-related interpretation.
The paper is structured as follows: setting notations, we briefly revise in Section 4.2 the background of the PLS, recall classical variable selection methods and evoke their blending in sparse PLS schemes, previously proposed. Then, in Section 4.3, we explain principles behind the DualsPLS family and detail the list of norm penalties and their algorithms in three main instances: the (group) lasso form —being the most important— and least squares and ridge forms. Thereafter Section 4.4 describes tested data (simulated and real) and the choices of model settings, calibration and validation. Each of the three penalties types are extensively benchmarked in Section 4.5. We finally draw concluding remarks with perspectives in Section 4.6 and supplementary material in the appendix.

Notation and definitions

Matrices, vectors and scalars are denoted by boldface uppercase letters, boldface lowercase and light lowercase letters respectively, e.g. $\mathbf{X}$, $\mathbf{y}$ and λ. The transpose of matrix $\mathbf{X}$ is $\mathbf{X}^T$. The identity matrix of size P is represented by I_P. The ℓ_1-norm and the ℓ_2-norm of vector a $\mathbf{w}$ of length P are

$$\|\mathbf{w}\|_1 = \sum_{p=1}^{P} |w_p| \qquad \text{and} \qquad \|\mathbf{w}\|_2 = \sqrt{\sum_{p=1}^{P} |w_p|^2}\,. \tag{4.2}$$

We denote by $\ell_0(\mathbf{w})$ the sparsity index or count measure [24] of the non-zero coordinates of $\mathbf{w}$ and $\ell_0^c(\mathbf{w})$ its complement i.e. $\ell_0^c(\mathbf{w}) = P - \ell_0(\mathbf{w})$. To choose the number of latent variables we rely on the mean squared error (MSE) expressed as

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2\,, \tag{4.3}$$

for a response vector $\mathbf{y}$ of N observations and a given estimate $\hat{\mathbf{y}}$. For performance evaluation, we choose the root mean squares error (RMSE), the mean absolute error (MAE) and the determination coefficient (R^2):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2} = \frac{1}{\sqrt{N}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2\,, \tag{4.4}$$

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|y_n - \hat{y}_n| = \frac{1}{N}\|\mathbf{y} - \hat{\mathbf{y}}\|_1\,, \tag{4.5}$$

$$\text{R}^2 = \frac{\sum_{n=1}^{N}(y_n - \bar{y})^2}{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2} \qquad \text{where} \qquad \bar{y} = \frac{\sum_{n=1}^{N} y_n}{N}\,. \tag{4.6}$$

The vector of signs of $\mathbf{w}$ is noted $\text{sign}(\mathbf{w})$, and $(\mathbf{w})_+$ is the vector composed of w_p if $w_p \geq 0$ and 0 if $w_p < 0$[(1)].

In the following, we assume that the matrix $\mathbf{X} \in \mathbb{R}^{N\times P}$ of independent variables and the response vector $\mathbf{y} \in \mathbb{R}^N$ are mean-centered. We use the convention where columns denote variables and rows observations.

4.2 Background

4.2.1 Partial Least Squares (PLS)

PLS originated from econometrics [68]. It was progressively and succesfully applied to other fields [69]: social and behavioral sciences, biosciences from bioinformatics [16] to neuroimaging [58], and chemometrics [104, 106]. It denotes a class of methods aimed at explaining the relationship between explanatory data and responses with the help of latent variables. They boast the management of both formative and reflective measurements, require low sample sizes and mild distributional assumptions.

PLS avatars root on projecting response onto a lower M-dimensional space spanned by new orthogonal directions $\{\mathbf{t}_1, \ldots, \mathbf{t}_M\}$ constructed as linear combinations of original variables. Its principle consists in compressing the predictor $\mathbf{X}$ into a smaller score matrix $\mathbf{T}$ of those $M < P$ variables. Thus, PLS computes M weights $\{\mathbf{w}_1, \ldots, \mathbf{w}_M\}$ forming the loading matrix $\mathbf{W}$ such that $\mathbf{T} = \mathbf{XW}$. As a result, loadings form an orthogonal basis. When Principal Component Analysis (PCA) [107] ought to best summarize $\mathbf{X}$ by taking into account only the correlation between the variables in $\mathbf{X}$, the PLS steps up and also consider the covariance between $\mathbf{X}$ and $\mathbf{y}$. Several algorithms have been proposed. NIPALS (nonlinear iterative partial least squares) [103] and SIMPLS [32] are most popular. When applied to a one-dimensional reponse, as in our case, both are shown to be equivalent. They solve the following optimization problem for the first component:

$$\max_{\mathbf{w}}(\mathbf{y}^T\mathbf{X}\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1\,. \tag{4.7}$$

The convex Problem (4.7) can be solved with Lagrange multipliers. For $\mu > 0$, it rewrites:

$$\min_{\mathbf{w}} L(\mathbf{w}) \quad \text{where } L(\mathbf{w}) = -\mathbf{z}^T\mathbf{w} + \mu(\|\mathbf{w}\|_2 - 1) \;\text{ and }\; \mathbf{z} = \mathbf{X}^T\mathbf{y}\,. \tag{4.8}$$

Solving (4.8) leads to

$$\mathbf{w} = \mathbf{X}^T\mathbf{y}\,. \tag{4.9}$$

The PLS algorithm uses the weight vector $\mathbf{w}$ to compress regressor $\mathbf{X}$ into score vector $\mathbf{t} = \mathbf{Xw}$. NIPALS iteratively computes weight vectors by deflation while SIMPLS is more direct.

(1) It corresponds to the Rectified Linear Unit (ReLU), a popular activation function for neural networks.

Let $\mathcal{P}_{\cdot}$ denotes the orthogonal projection onto the space spanned by components specified in subscript. For instance, scores $\{\mathbf{t}_1, \ldots, \mathbf{t}_{m-1}\}$ span the space corresponding to $\mathcal{P}_{\mathbf{t}_{m-1}}$. The algorithm considers the part of $\mathbf{X}$ that is orthogonal to $\mathbf{t}_k, k < m$. For the m^{th} component, $\mathbf{X}$ is replaced by $\mathbf{X}_m$ such that:

$$\mathbf{X}_m = \mathbf{X} - \mathcal{P}_{\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}}\mathbf{X} = \mathbf{X}_{m-1} - \mathcal{P}_{\mathbf{t}_{m-1}}\mathbf{X}_{m-1}\,. \tag{4.10}$$

The NIPALS variant PLS1 for an univariate reponse as given in [50] is described in Algorithm 3.

Algorithm 3: NIPALS PLS1

Input: $\mathbf{X}, \mathbf{y}, M$
$\mathbf{X}_1 = \mathbf{X}$
for $m = 1, \ldots, M$ **do**
  $\mathbf{w}_m = \mathbf{X}_m^T\mathbf{y}$ (weight vector computation)
  $\mathbf{t}_m = \mathbf{X}_m\mathbf{w}_m$ (component construction)
  $\mathbf{X}_{m+1} = \mathbf{X}_m - \mathcal{P}_{\mathbf{t}_m}\mathbf{X}_m$ (deflation)
end for

This algorithm produces a new lower dimensional score matrix $\mathbf{T} \in \mathbb{R}^{N\times M}$. Proposition 1 from [57] explicits the regression coefficients for M components as:

$$\hat{\boldsymbol{\beta}}_M^{PLS} = \mathbf{W}(\mathbf{T}^T\mathbf{X}\mathbf{W})^{-1}\mathbf{T}^T\mathbf{y}\,. \tag{4.11}$$

The vector of regression fitted values $\hat{\mathbf{y}}$ for M components is the projection of response vector $\mathbf{y}$ onto the space spanned by scores columns of $\mathbf{T}$.

4.2.2 Least absolute shrinkage and selection operator

By selecting the most important features, variable selection produces a less complicated model. It has the potential advantage of being easier to handle than the complete full set of variables. The optimization problem in standard linear regression is stated as:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2\,. \tag{4.12}$$

Provided $\mathbf{X}$ has full column rank, the ordinary least squares (LS) estimation is $\hat{\mathbf{y}}_{LS} = \mathcal{P}_{[\mathbf{X}]}\mathbf{y}$, where $[\mathbf{X}]$ is the space spanned by the columns of $\mathbf{X}$. In other terms, $\hat{\boldsymbol{\beta}}^{\text{LS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ A popular sparsity-based approach is the lasso developed by Tibshirani in 1996 [95]. It is reknown for its ℓ_1 penalty scheme that shrinks less relevant variables to zero. It is obtained by solving:

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_1 \leq \lambda\,. \tag{4.13}$$

Threshold parameter $\lambda > 0$ controls the extent of shrinkage applied to the estimate; that is, the number ℓ_0^c of coefficients set to zero. An appropriate λ is important to get interpretable results. If $\hat{\boldsymbol{\beta}}^{\mathrm{LS}}$ exists, as mentioned in [95], then for a $\lambda \geq \|\hat{\boldsymbol{\beta}}^{\mathrm{LS}}\|_1$, the lasso estimate $\hat{\boldsymbol{\beta}}^{\mathrm{l}}$ is equal to the ordinary least square solution. And for $\lambda = \dfrac{\|\hat{\boldsymbol{\beta}}^{\mathrm{LS}}\|_1}{2}$, it selects on average half of the variables. We can reformulate (4.13) as

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + t\|\boldsymbol{\beta}\|_1 \,. \tag{4.14}$$

Note that there is a (non-explicit) correspondence between parameters λ and t. In the orthonormal design case, i.e. $\mathbf{X}^T\mathbf{X} = I_P$, there exists $\hat{\beta}^{\mathrm{l}}$ closed form solution called *soft thresholding* verifying

$$\hat{\beta}_p^{\mathrm{l}} = \mathrm{sign}(\hat{\beta}_p^{\mathrm{LS}})(|\hat{\beta}_p^{\mathrm{LS}}| - \lambda)_+ \quad \forall p \in \{1, \ldots, P\} \,. \tag{4.15}$$

Coefficients whose magnitude is smaller than λ are set to zero. Amplitudes of the others are shrunk with respect of the threshold. While proved successful for numerous applications, some drawbacks are reported [112, 47]. Some are: 1) non strict convexity of the criterion when the number of predictors exceeds the number of observations ($P > N$) 2) algorithm saturation when N variables have been selected 3) with highly correlated variables, tendency to pick mildly representative ones.
Another shrinking method is ridge regression [49] with optimization problem:

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + t\|\boldsymbol{\beta}\|_2 \,. \tag{4.16}$$

Its trick is to add a diagonal matrix to $(\mathbf{X}^T\mathbf{X})$ in order to overcome the singularity problem. Therefore, the solution always exists, expressed as:

$$\hat{\boldsymbol{\beta}}^{\mathrm{r}} = (\mathbf{X}^T\mathbf{X} + tI_P)^{-1}\mathbf{X}^T\mathbf{y} \,. \tag{4.17}$$

Compared to the lasso, it uses an ℓ_2-norm instead of the ℓ_1 penalization but retains most variables by design.

4.2.3 Blending methods: sparse Partial Least Squares (sPLS)

Sparse Partial Least Squares (sPLS) denotes a body of works adding a variable selection flavor to the standard PLS framework. We focus here on ones specifically using lasso inspired penalties. An ℓ_1-norm can be incorporated in optimization problem (4.7). Noting

$$\widehat{\mathrm{Cov}}(\mathbf{Xw}, \mathbf{y}) = \frac{1}{N}\mathbf{w}^T\mathbf{z}, \quad \text{with} \quad \mathbf{z} = \mathbf{X}^T\mathbf{y} = N\widehat{\mathrm{Cov}}(\mathbf{X}, \mathbf{y}) \,, \tag{4.18}$$

adding the coupling parameter $\lambda_s > 0$ and orthogonality constraint on components $\{\mathbf{t}_1, \ldots, \mathbf{t}_M\}$, the sPLS optimization problem is stated as:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p}\{-\widehat{\mathrm{Cov}}(\mathbf{Xw}, \mathbf{y}) + \lambda_s\|\mathbf{w}\|_1\}, \qquad \text{for } \mathbf{w}^T\mathbf{w} = 1 \,. \tag{4.19}$$

Problem (4.19) is tackled in 2008 [61] using sparse PCA [84]. Then iterative PLS [93] is combined to singular value decomposition. We denote it as $\text{sPLS}_{\text{LeCao}}$ after the first author. In 2010 [25], Problem (4.19) is reframed by imposing the ℓ_1 penalty on a surrogate direction close to the original vector $\mathbf{w}$, providing an approximate solution with $\text{sPLS}_{\text{Chun}}$. In 2018, [37] reformulates Problem (4.19) using recent results from proximal optimization [8] with $\text{sPLS}_{\text{Durif}}$. In this last case, $\text{sPLS}_{\text{Durif}}$ provides an exact and closed-form solution reminiscing the soft threshold operator. Moreover, they suggest an adaptive method for computing the sPLS weight vectors using classical PLS ones.
Along the lines of methods presented above, Dual-sPLS aims at inference and interpretability: accurate predictions combined with sparse localization features for better chemometrics performance. Following [37], we also wish to provide a means to tuning the relative sparsity of the outcome. Finally, as different analytical chemistry modalities provide different insights on chemical mixtures, the Dual-sPLS is designed to naturally allow the combination of heterogeneous datasets as a byproduct of the versatile dual norm approach[(2)].

4.3 Dual Sparse Partial Least Squares (Dual-sPLS)

4.3.1 Motivation and purposes

In statistics and machine learning, it is quite standard to penalize a data fidelity ℓ_2-norm by a penalty involving a specific norm, e.g. ridge (ℓ_2), lasso ℓ_1, etc. (see previous Sections). These penalty options are crucial, they drive the obtention of admissible or reasonable solutions, for instance towards sparsity. The goal of this contribution is to provide a general paradigm for this kind of task. Arbitrary norm choices may not lead to trackable algorithms. However, the concept of dual norm is a means to formulate a unifying optimization framework, for which one can opt for penalties with practical algorithmic properties.

Definition 4.3.1 *Let $\Omega(\cdot)$ be a norm on $\mathbb{R}^P$. For any $\mathbf{z} \in \mathbb{R}^P$, the associated dual norm, denoted $\Omega^*(\cdot)$, is defined as*

$$\Omega^*(\mathbf{z}) = \max_{\mathbf{w}} (\mathbf{z}^T \mathbf{w}) \quad \text{s.t.} \quad \Omega(\mathbf{w}) = 1 \,. \tag{4.20}$$

Comparing (4.7) and (4.20), we find that the optimization of the PLS objective function amounts to finding the vector $\mathbf{w}_1$ that fits the dual norm of the ℓ_2-norm of $\mathbf{z}$, where $\mathbf{z} = \mathbf{X}^T\mathbf{y}$. This gives us the incentive to evaluate different norm expressions that could be used as domain-related penalizations. Thus, for any norm $\Omega(.)$ used, the first component will be:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathbb{R}^p} \{-\mathbf{z}^T \mathbf{w}\}, \quad \text{s.t. } \Omega(\mathbf{w}) = 1 \,. \tag{4.21}$$

Imposing a form of sparsity on the solution has inspired a quantity of research. The ℓ_1-norm is one of the earliest penalties, that encourages sparsity while remaining convex, and associated with efficient, tractable algorithmic implementations. Our study focuses on norms that 1) have been employed as penalties in previous works and 2) provide explicit, straightforward, and effective algorithms within the PLS framework. Namely, although formulation is generic, we

(2) Application of this extention is not performed here and is subject to a later work.

emphasize four types of norms. They make practical sense when dealing with measurements typically available in chemometrics, starting with the lasso analogue, a natural and intuitive approach. We provide the corresponding R [77] package dual.spls [5] with a complete description. It contains the following main functions, each of them being associated to specific penalty:

1. **Dual-sPLS$_\mathbf{l}$** *(pseudo-lasso norm, d.spls.lasso()).* Similar to the sPLS Problem (4.19), an intuitive norm combines ℓ_2 and ℓ_1:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 + \|\mathbf{w}\|_2 \,. \tag{4.22}$$

   Dual-sPLS$_\mathrm{l}$ is inspired by ℓ_1 lasso and implemented for situations where we seek selection of features with most impact on the response when dealing with large, highly-correlated data.

2. **Dual-sPLS$_\mathbf{gl}$** *(pseudo-group lasso norm, d.spls.GL()).* Inspired by group lasso [86], it combines groups of measurements. It applies pseudo-lasso to each group individually while constraining the total set. For G groups, $\mathbf{w}_g$ represents the variables of the loading vector $\mathbf{w}$ that belongs to group g. The corresponding norm is formulated as:

$$\Omega(\mathbf{w}) = \sum_{g=1}^{G} \alpha_g\|\mathbf{w}_g\|_2 + \lambda_g\|\mathbf{w}_g\|_1 \,, \tag{4.23}$$

   where $\alpha_g \geq 0, \forall g \in \{1,\ldots,G\}$ and $\sum_{g\in 1,\ldots,G} \alpha_g = 1$. Dual-sPLS$_\mathrm{gl}$ is mainly thought for the following not-exclusive cases, akin to multiblock PLS. First, for a single type of measurement $\mathbf{X}$, when G different subsets of scalar variables are expected to contribute jointly to the response, e.g. from wavelength selection or prior analytical chemistry knowledge. Second, when a single response $\mathbf{y}$ can be predited by G distincts sets of measurements $\mathbf{X}_1$, $\ldots$, $\mathbf{X}_G$, e.g. different physico-chemical modalities that could be complementary.

3. **Dual-sPLS$_\mathbf{LS}$** *(pseudo-least squares norm, d.spls.LS()).* It introduces $\mathbf{N}_1$, a matrix of p columns, and applies when $\mathbf{X}$ is not singular:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{N_1 w}\|_1 + \|\mathbf{Xw}\|_2 \,. \tag{4.24}$$

   The mild conditions on $\mathbf{N}_1$ are provided in 4.A.2. Dual-sPLS$_\mathrm{LS}$ adds a variable selection flavor to classical least-squares. Therefore, it can be employed when shrinking original LS regression parameters is desired. The classical least squares solution is recovered for $\lambda = 0$.

4. **Dual-sPLS$_\mathbf{r}$** *(pseudo-ridge norm, d.spls.ridge()).* It deals with cases where $\mathbf{X}$ is singular and resorts to a ridge-like penalization:

$$\Omega(\mathbf{w}) = \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{Xw}\|_2 + \|\mathbf{w}\|_2 \,. \tag{4.25}$$

The construction of weight vectors $\mathbf{w}_1,\ldots,\mathbf{w}_M$ differs in each of the four cases. It however follows similar steps as for the PLS. Starting with a reformulation of optimization Problem (4.20) and using Lagrange multipliers, we aim at iteratively minimizing the function $L(\mathbf{w}) = -\mathbf{z}^T\mathbf{w} + \mu(\Omega(\mathbf{w}) - 1)$, for $\mu > 0$. As some norms are not differentiable, we resort to the more

generic notion of subgradient $\nabla\Omega(\mathbf{w})$ [8]. It identifies to the classical differential when it is defined. The subgradient of L vanishes for

$$\nabla\Omega(\mathbf{w}) = \frac{\mathbf{z}}{\mu}\,. \tag{4.26}$$

It is then sufficient to substitute the gradient — when it exists— of the considered norm of $\Omega(\mathbf{w})$ in (4.26).
We provide in the following a detailed analysis for the pseudo-lasso case of Dual-sPLS (see (4.22)) and some remarks for the other norms. In all cases we impose that $\mathbf{w}$ and $\mathbf{z}$ lie in the same orthant; it generalizes, in n dimensions, the quadrant in the 2D plane or the octant in the 3D space. In other words, corresponding coordinates of $\mathbf{w}$ and $\mathbf{z}$ have the same sign.

4.3.2 Norm options (lasso, group lasso, least squares and ridge)

Pseudo-lasso ~ We rconsider Equation (4.22). Let $\boldsymbol{\delta}$ be the sign vector of $\mathbf{w}$ and $\mathbf{z}$. By differentiating $\Omega(\mathbf{w})$, we get

$$\nabla\Omega(\mathbf{w}) = \lambda\boldsymbol{\delta} + \frac{w}{\|w\|_2}\,, \tag{4.27}$$

and by substituting it in (4.26), we obtain

$$\frac{\mathbf{w}}{\|\mathbf{w}\|_2} = \frac{\mathbf{z}}{\mu} - \lambda\boldsymbol{\delta}\,. \tag{4.28}$$

The closed-form solution of the Dual-sPLS$_1$ optimization problem consists in zeroing coordinates whose magnitude is lower than the soft threshold λ and in reducing the others toward zero. Thus, for $\nu = \lambda\mu$ and $p \in \{1, \ldots, P\}$, it can be expressed as:

$$\frac{w_p}{\|\mathbf{w}\|_2} = \frac{1}{\mu}\delta_p(|z_p| - \nu)_+\,. \tag{4.29}$$

A common issue is the choice of the appropriate shrinking parameter. Cross-Validation [91], evoked in Section 4.4.3, is popularly adopted in sparse regressions. We choose a more intuitive option. We obtain it adaptively, according to the proportion of variables that we would like to keep in the active set at each iteration. The procedure is illustrated in Figure 4.1. It represents the empirical cumulative distribution of sorted magnitudes of $|\mathbf{X}^T\mathbf{y}|$ from the real data D_{NIR} described later in Section (4.4.2). Fixing a shriking ratio ς of expected zero coefficients (e.g. $\varsigma = 80\,\%$), we select the threshold ν at iteration m as depicted. As the cumulative destribution is non-decreasing, we choose the first x-axis value corresponding to ordinate 0.800.

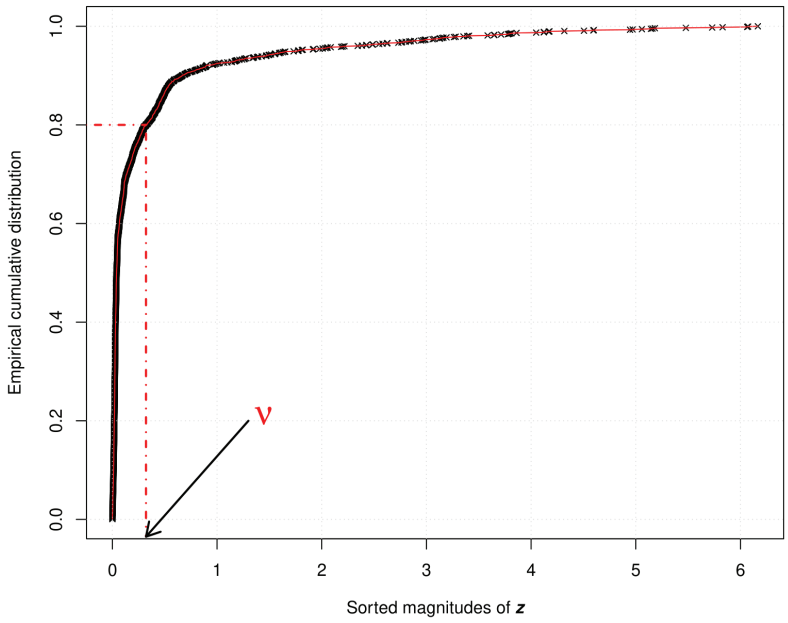

Figure 4.1 ~ *Empirical cumulative distribution of the sorted magnitude of* $\boldsymbol{z} = \mathbf{X}^T\mathbf{y}$ *(black crossed connected by solid red line) from real data* D_{NIR}*. Dotted red line illustrated the selection of appropriate* ν *for* 80 % *of sparsity.*

To guarantee the unit norm property for $\mathbf{w}$, we set $\mu = \|\mathbf{z}_\nu\|_2$ where $\mathbf{z}_\nu$ is the vector of coordinates $\delta_p(|z_p| - \nu)_+$ for $p \in \{1, \ldots, P\}$. Consequently,

$$\mathbf{w} = \frac{\mu}{\nu\|\mathbf{z}_\nu\|_1 + \|\mathbf{z}_\nu\|_2^2}\mathbf{z}_\nu.$$

The rationale behind constrainting the direction $\mathbf{w}$ instead of the regression coefficients $\hat{\boldsymbol{\beta}}$ is their collinearity. Indeed, the estimator writes $\hat{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y}$. Being collinear, soft-thresholding $\mathbf{w}$ performs a variable selection at the same location in $\hat{\boldsymbol{\beta}}$ coordinates. The pseudo-lasso DualsPLS is described in Algorithm 4.

Algorithm 4: Dual-sPLS$_1$

Input: $\mathbf{X}, \mathbf{y}, M$ (number of components desired), ς (shrinking ratio)
$\mathbf{X}_1 = \mathbf{X}$
for $m = 1, \dots, \mathrm{M}$ **do**
  $\mathbf{z}_m = \mathbf{X}_m^T \mathbf{y}$ (weight vector)
  Find ν adaptatively according to ς
  $\mathbf{z}_\nu = (\delta_p(|z_p| - \nu)_+)_p$ (applying the threshold), $p \in \{1, \dots, P\}$
  $\mu = \|\mathbf{z}_\nu\|_2$ and $\lambda = \dfrac{\nu}{\mu}$
  $\mathbf{w}_p = \dfrac{\|\mathbf{z}_\nu\|_2}{\nu\|\mathbf{z}_\nu\|_1 + \|\mathbf{z}_\nu\|_2^2}\mathbf{z}_\nu$ (loadings)
  $\mathbf{t}_m = \mathbf{X}_m \mathbf{w}_m$ (component)
  $\mathbf{X}_{m+1} = \mathbf{X}_m - \mathcal{P}_{\mathbf{t}_m}\mathbf{X}_m$ (deflation)
end for
$\hat{\boldsymbol{\beta}} = \mathbf{W}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y}$

Note that as long as $\mathbf{w}$ and $\mathbf{z}_\nu$ are collinear, the sparsity of the results only requires the computation of $\mathbf{w}$, up to a non-zero factor.

Pseudo-group lasso ~ Response $\mathbf{y}$ may be explained separately by explanatory variables of different nature with prediction models. Combining them appropriately is potentially beneficial both in predictive and interpretative powers. The same reasoning could be used to partition the dataset into groups.
Physico-chemical motivation resides in segmenting a spectrum into homogenous bands or combining complementary modalities (e.g. IR and NMR) to predict the same property (e.g. viscosity, density). We consider G groups, and $\mathbf{z}_g$ sub-vector of $\mathbf{z}$ denotes variables belonging to group g. The group lasso inspired norm is expressed as in Equation (4.23). The closed-form solution is collinear to the vector $\mathbf{z}_{\nu_g}$. It is given by

$$\mathbf{z}_{\nu_g} = \boldsymbol{\delta}_g(|\mathbf{z}_g| - \nu_g)_+ \quad \text{and } \mathbf{z}_\nu = \left(\mathbf{z}_{\nu_g}\right)_{g\in\{1,\dots,G\}}, \tag{4.30}$$

$\boldsymbol{\delta}_g$ being the vector of signs of $\mathbf{w}_g$ and $\nu_g = \lambda_g\mu$ for $g \in \{1, \dots, G\}$. Each group is driven by its own threshold ν_g. The latter can be obtained similarly as in Section 4.3.2. Note that this Dual-sPLS version reduces to the pseudo-lasso case when $G = 1$.

Pseudo-least squares and pseudo-ridge ~ The above can be generalized in many ways, by defining more versatile norm shapes, including notably weighted norms. One such possibility is $\forall \mathbf{w} \in \mathbb{R}^P$

$$\Omega(\mathbf{w}) = \lambda_1\|\mathbf{N}_1\mathbf{w}\|_1 + \|\mathbf{N}_2\mathbf{w}\|_2 + \lambda_2\|\mathbf{w}\|_2\,. \tag{4.31}$$

It is not easily solvable in general. However, an appropriate choice of matrices $\mathbf{N}_1$ and $\mathbf{N}_2$, and factors λ_1 and λ_2 allow us to recover the lasso and group lasso norms, but also several other already known concepts, like fused lasso, least squares or ridge. We focus here on two main situations whose optimization problem resolution can be obtained analytically. An obvious option heavily inspired by least squares regression sets $\mathbf{N}_2 = \mathbf{X}$ and $\lambda_2 = 0$. Its resolution

supplements the traditional least squares problem with a more selective shrinkage akin to that of our pseudo-lasso. Namely we first note

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{N_1 w}\|_1 + \|\mathbf{X w}\|_2 \,. \tag{4.32}$$

Then for $\nu = \mu\lambda$ and δ the vector of signs of $\mathbf{N_1 w}$ and $\mathbf{N_1 z}$,

$$\frac{\mathbf{w}}{\|\mathbf{X w}\|_2} = (\mathbf{X}^T\mathbf{X})^{-1}\frac{\mathbf{z}}{\boldsymbol{\mu}} - \lambda(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{N}_1^T\boldsymbol{\delta}\,, \tag{4.33}$$

where we have implicitly assumed that $\mathbf{X}$ has full rank. Consequently, we penalize $|\hat{\boldsymbol{\beta}}^{\mathrm{LS}}|$ instead of $|\mathbf{z}|$. For equation (4.33) to take a genuine pseudo-lasso form, it is sufficient that $\mathbf{N}_1$ verifies

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{N}_1^T\boldsymbol{\delta} = \mathrm{sign}\Big((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}\Big)\,. \tag{4.34}$$

However, as it does not play a role in loadings' computation, it does not need to be computed explicitly. Thus, the coordinates of the simplified closed-form solution is:

$$\frac{\mathbf{w}_p}{\|\mathbf{X w}\|_2} = \frac{1}{\mu}\mathrm{sign}(\hat{\beta}_p^{\mathrm{LS}})(|\hat{\beta}_p^{\mathrm{LS}}| - \nu)_+\,, \tag{4.35}$$

When $\mathbf{X}$ is singular, the above cannot hold. Meanwhile, this case can be addressed with a regularization inspired by the ridge [49]. By choosing $\mathbf{N}_1 = I_P$, $\mathbf{N}_2 = \lambda_2\mathbf{X}$ and $\lambda_2 = 1$, equation (4.31) writes

$$\Omega(\mathbf{w}) = \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{X w}\|_2 + \|\mathbf{w}\|_2\,. \tag{4.36}$$

It amounts to penalize $|\mathbf{z}_{\nu_2}|$ where $\mathbf{z}_{\nu_2} = \Big(\nu_2\mathbf{X}^T\mathbf{X} + I_P\Big)^{-1}$ and $\nu_2 = \lambda_2\dfrac{\mathbf{w}}{\|\mathbf{X w}\|_2}$, instead of $|\mathbf{z}|$ like in the pseudo-lasso. For $\nu_1 = \lambda_1\mu$, the closed-form solution is formulated as:

$$\frac{\mathbf{w}}{\|\mathbf{w}\|_2} = \frac{1}{\mu}\delta(|\mathbf{z}_{\nu_2}| - \nu_1)_+\,. \tag{4.37}$$

where $\delta = \mathrm{sign}(\mathbf{z}_{\nu_2}\mathbf{z})$. Adding the diagonal perturbation resolves the non-invertability of $\mathbf{X}^T\mathbf{X}$.

4.4 Simulated and real data, model settings, evaluation

4.4.1 Simulated sparse data: Gaussian mixtures D_{SIM} and $\overline{D}_{\mathrm{SIM}}$

For an in-depth analysis of machine learning algorithms, resorting to simulated data allows an unbiased access to ground truth. We thereby propose a parametrized model. It is thought to provide similarities with common analytical chemistry data, with all sparse parameters controlled. We choose a positively weighted mixture of K Gaussians peaks with preset identical scale σ and amplitudes A and locations μ are drawn from uniform distributions. They are summed as follows:

$$\sum_{k=1}^{K} A_k \exp(-\frac{(x-\mu_k)^2}{2\sigma^2})\,. \tag{4.38}$$

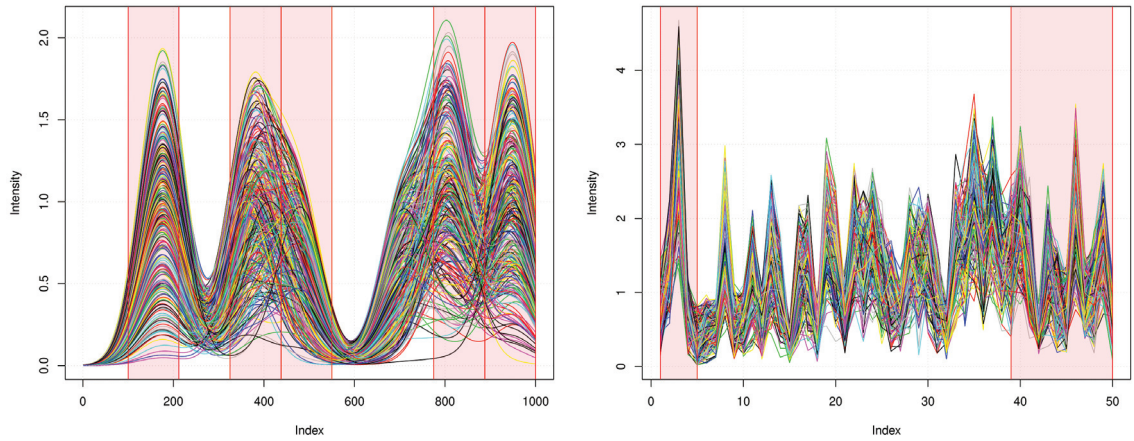

Figure 4.2 ~ *D_{SIM} (left) and $\overline{D}_{\mathrm{SIM}}$ (right) simulated data. Ranges of variables involved in the linear response model* $\mathbf{y}$ *are highlighted in red.*

and uniformly sampled. The response vector $\mathbf{y}$ is defined by an explicit linear model(affected by a stochastic Gaussian contamination) composed of weighted sums of $\mathbf{X}$ values. Weights can be random or fixed quantities by ranges of indices.
In this work, to evaluate Dual-sPLS in both precision and information location, we devise a sparse additive model with only $S \ll P$ positive weights and $P - S$ null weights. Namely, only S variables are responsible in the construction of response $\mathbf{y}$. This information is especially beneficial to demonstrate the strength of variable selection in sparse methods. Since we deal with high-dimensional situations, we simulated D_{SIM}: 300 mixtures of 30 Gaussians represented by 1000 variables (Figure 4.2 (left)). Highlighted red areas denote variables involved in the computation of response $\mathbf{y}$. The corresponding matrix of D_{SIM} is singular and used in the evaluation of Dual-sPLS$_{\mathrm{l}}$ and Dual-sPLS$_{\mathrm{r}}$. Since the Dual-sPLS$_{\mathrm{LS}}$ is only operational with invertible matrices, we also simulated non-singular data matrix $\overline{D}_{\mathrm{SIM}}$, 200 mixtures of 100 Gaussians represented by 50 variables. The response $\mathbf{y}$ corresponding to $\overline{D}_{\mathrm{SIM}}$ depends only on the first five and last twelve variables as shown in Figure 4.2 (right).

4.4.2 Real data: near-infrared (NIR) spectroscopy D_{NIR}

In chemistry, complex mixtures of molecules are analyzed with different physico-chemical methods. Besides, determining macroscopic properties is important to their use.
The evaluation on real data is done using NIR spectra of hydrocarbon samples. NIR is based on the principle of absorption of radiation (infrared) by matter [23]. Infrared radiations correspond to wavenumbers directly lesser than those of the visible light spectrum. The absorption of radiation depends on chemical bonds, therefore a NIR spectrum encodes information about the composition of the sample. We focus on the density property which is obtained by standardized methods. The IFPEN dataset D_{NIR} was partly exposed in [60, 59]. It is available

at http://www.laurent-duval.eu/opus-dual-spls-sparse-pls/ and subject to a forthcoming publication [38]. It is composed of 208 samples with 1557 variables. The corresponding matrix $\mathbf{X}$ is singular. Many chemical data require adequate preprocessing: normalization, baseline removal [75], deconvolution [24]. Here we simply apply a discrete derivative obtained with a Savitzky–Golay smoothing filter [83] of degree 2 and length 15. It serves as both a crude baseline filter and diversity enhancement operator [33]. The NIR preprocessed dataset D_{NIR} is represented in Figure 4.3.

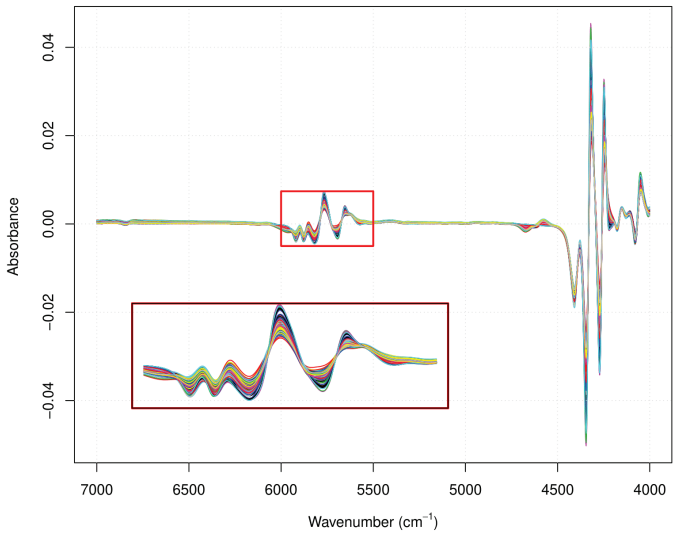

Figure 4.3 ~ *D_{NIR}: first Savitzky-Golay derivatives of the NIR spectra of 208 samples. Bottom subplot: magnification of the red box.*

4.4.3 Model settings: number of latent component selection

Selecting the appropriate number M of latent components is crucial when building a regression model. It balances between model complexity and prediction accuracy (degrees of freedom), preventing the risk of overfitting. This issue is especially important when using PLS and its extensions in chemometrics. In practice, one may use this variant of a proposal in [15], based on cross-validation with multiple random split. First, observations are split randomly several times into calibration and validation sets. Second, candidate models are constructed with different numbers of latent components. Third, each prediction is evaluated on the validation set with MSE. The latter are averaged for each model. Finally, the smallest model with the lowest averaged MSE reveals an adequate number of latent components. With this method, two parameters are necessary: the splitting ratio and the number of times observations are divided.
We do not use this procedure in Section 4.5. We evaluate Dual-sPLS performance by comparing it to other regression methods, and exploring model orders. We vary the number of latent components from 1 to 10 and assess each case.

4.4.4 Calibration and validation

The evaluation of prediction models traditionally divides the dataset into two representative sets called calibration and validation. Three main methods are used. In the first one, observations are randomly selected. The second only considers the distribution of values in the response $\mathbf{y}$ [81, Stratified sampling]. A third class is known as Kennard and Stone (KS) method [54]. It optimizes relative distances between observations according to variables of $\mathbf{X}$. In chemometrics, one may expect the existence of a yet unknown dependence between analytical measurements and properties. Taking both $\mathbf{X}$ and $\mathbf{y}$ values for a proper calibration and validation split would be desirable. The attempts of [94] to consider $\mathbf{X}$ and $\mathbf{y}$ in a single distance with appropriate weights is not straightforward. It is difficult to adequately weight variables that do not belong to the same space. We have recently proposed a CalValXy for that purpose. It consists in dividing the dataset into subgroups according to the repartition of $\mathbf{y}$ and applying the Kennard and Stone to each subgroup. It is summarized in Algorithm 5, and extensively described in [2].

Algorithm 5: Calibration and validation CalValXy

Input: $\mathbf{X}, \mathbf{X}_{type}$(index of which set belongs each observation of $\mathbf{X}$),
Listecal (number of calibration points to pick from each subset)
G = mean($\mathbf{X}$) (centroid)
$C_1 = \max_n \|\mathbf{x}_G - \mathbf{x}_n\|$, $n \in \{1, \ldots, N\}$ (first calibration point)
s = subset where C_1 is located
while Listecal is not empty **do**
  $s \leftarrow s + 1$
  Find the minmax point C in subset s
  Remove C from $\mathbf{X}$ and **Listecal**
  Store C in a vector of calibration index **cal**
end while

4.5 Comparative evaluation and discussion

We benchmark each proposed Dual-SPLS regression flavor (respectively pseudo-lasso, least squares and ridge) against its classical counterpart, and comparable sparse SPLSs, when applicable. We follow a common procedure to state the main results. First, we split observations into calibration (80 %) and validation (20 %). We replaced the traditional Kennard and Stone method [54] — using explanatory variables $\mathbf{X}$ only — with CalValXy (cf. Section 4.4.4 and [2]). The latter incorporates the response variable $\mathbf{y}$ in the splitting and proves slightly better than KS in terms of prediction. Comparative performance is assessed in both accuracy and quality of interpretation. For the first one, common objective metrics are root mean squared error (RMSE), mean absolute error (MAE), or determination coefficient (R^2) (see end of Section 4.1). As metrics yield similar outcomes, we only compare, in the topmost figures, RMSE values for either calibration (left) or validation (right) CalValXy splits, as we increase the number M of latent components from one to ten. For the second one, we assess both variable selection and localization by vertically stacking regression coefficients for each compared algorithm in the bottom figure. Results are extensively discussed on simulated and real data for Dual-SPLS$_l$, and

in less details for the least squares and ridge flavors. Complementary outcomes are provided in the supplementary materials.

4.5.1 Dual-sPLS pseudo-lasso evaluation (D_{SIM}, D_{NIR})

Dual-sPLS$_{\text{l}}$ is compared to standard PLS, three alternative sparse PLS (sPLS$_{\text{LeCao}}$ [61], sPLS$_{\text{Chun}}$ [25], sPLS$_{\text{Durif}}$ [37]) and lasso [95]. Their respective parameters are selected by cross-validation (Section 4.4.1). Both sPLS$_{\text{LeCao}}$ and Dual-sPLS$_{\text{l}}$ explicitely specify a sparsity parameter: the (approximate) proportion of variables ς to be discarded (ℓ_0^c/P). We set it here to 99 %.
We first evaluate Dual-sPLS$_{\text{l}}$ on simulated data D_{SIM} (Section 4.4.1) in Figure 4.4. Top-left and right plots entail that accuracy (RMSE) globally improves as the number of latent variables M increases for all five PLS-related methods — in both calibration and validation. The lasso performance, independent on the number of components, is represented by the sixth dotted curve. From six to ten latent variables, all curves tend to plateau, with close RMSE values. Dual-SPLS$_{\text{l}}$, sPLS$_{\text{Chun}}$ and PLS provide the best results (lowest curves). Thus, adding more components seems uncessary. We choose six latent variables to compare coefficient localization. On Figure 4.4-bottom, we stack seven panels: original spectra (1) and the coefficients for: PLS (2), Dual-sPLS$_{\text{l}}$ (3), sPLS$_{\text{LeCao}}$ (4), sPLS$_{\text{Chun}}$ (5), sPLS$_{\text{Durif}}$ (6), lasso (7). PLS coefficients (panel 2) match the shape of the simulated data (panel 1). However, it fails to localize the most important variables, unlike sparse PLS. The ℓ_0 criterion (Section 4.1) quantifies the sparsity induced by each method. Dual-sPLS$_{\text{l}}$, sPLS$_{\text{LeCao}}$ and lasso perform best, selecting as expected a small number of variables, with an ℓ_0 value around 40 to 60. It however is not sufficient to hint at improvements in interpretability. Looking only at variables affecting the response (shaded red background in panel 1), most compared methods exhibit significant coefficients in many (useless) areas (transparent background). Only Dual-sPLS$_{\text{l}}$, sPLS$_{\text{LeCao}}$ present concentrated coefficients that can help chemical interpretation. On this rudimentary yet explainable model, we hint that Dual-sPLS$_{\text{l}}$ provides a predictive quality comparable to its challengers, and is the best in providing at the same time accurate localization on simulated data, with a verifiable (yet simplified) prediction model.
We are now able to evaluate the performance of Dual-sPLS$_{\text{l}}$ on real near-infrared data D_{NIR} (4.4.2) for density prediction. Similarly to D_{SIM}, RMSE curves in Figure 4.5 for calibration (top-left) and validation (top-right) globally decrease with an increasing number of components. Errors plateau after six components, indicating that additional latent structure orders might be weakly helpful. The performance gap for sPLS$_{\text{Durif}}$ could occur as it was mainly designed for classification. Again, we assess model interpretation in Figure 4.5 (bottom) for six latent vectors. By nature, location of the most influential features of spectra for a specific property is yet to be unveiled. One may expect that most of the meaningful variables are located in the active parts of the signal, e.g. spectral bands with relatively higher intensities, with some others possibly in quieter wavenumber ranges. On the top panel, NIR spectra are mainly active[(3)] from 4000 cm^{-1} to 4800 cm^{-1} and 5500 cm^{-1} to 6000 cm^{-1}. Meaningful PLS coefficients are visible on a much wider support, provoking ambiguity on the identification of spectral bands related to density. All sPLS actually have smaller support, sPLS$_{\text{LeCao}}$ and Dual-sPLS$_{\text{l}}$ being the sparsest

[(3)] We do not endeavour a chemical explanation here. It ought to be substanciated in forthcoming paper [38]

with ℓ_0 respectively equal to 88 and 82. The first singularity of Dual-sPLS$_\text{l}$ is the contiguous and smoothness of its coefficients. By contrast, sPLS$_\text{Chun}$ and sPLS$_\text{Durif}$ coefficients location appear to be more scattered across the wavenumber axis, in non-contiguous small chunks and even isolated spikes. The second is the absence of response in the 5500 cm^{-1} to 6000 cm^{-1} bands[(3)] in Dual-sPLS$_\text{l}$. We are not able to chemically explain the discrepancy of absence/presence results in this band. However, Dual-sPLS$_\text{l}$ does not need it to remain almost as accurate as its competitors.

4.5.2 Dual-sPLS pseudo-least squares evaluation ($\overline{D}_\text{SIM}$)

The Dual-sPLS$_\text{LS}$ requires data to be represented by a non-singular matrix $\mathbf{X}$, as explained in Section 4.3.2. Since real data D_NIR is singular, we use simulated data $\overline{D}_\text{SIM}$ presented in Section 4.4.1. As the number of variables in $\overline{D}_\text{SIM}$ is already small, we only shrink 60 % of its variables to evaluate the Dual-sPLS$_\text{LS}$ against classical least squares. The latter is denoted by dashes, as the number of latent components is meaningless in this case.

For calibration (Figure 4.6 top-left) the RMSE for Dual-sPLS$_\text{LS}$ decreases mildly as the number of components increases. It approaches the least squares performance. For validation (Figure 4.6 top-right) Dual-sPLS$_\text{LS}$ performs similarly or better than least squares all over model orders. This contrast in performance might be explained by a tendency to overfit for least squares. A better prediction performance is expected with our model. Similarly to the Dual-sPLS$_\text{l}$, we also choose to evaluate it with six components in the bottom of Figure 4.6. Again, redish regions indicate active variables for the unknown linear model. We observe an overall similarity in the dynamics of both regression coefficients: strong amplitude in the first five and last ten variables corresponding to active regions. The main difference resides in the intermediate part, irrelevant to the response. Least squares as expected shrinks inactive variables towards zero but not as much as Dual-sPLS$_\text{LS}$ does. This is exemplified in the zoomed panels, where Dual-sPLS$_\text{LS}$ exhibit much less non-zero coefficients.

4.5.3 Dual-sPLS pseudo-ridge evaluation (D_SIM, D_NIR)

Dual-sPLS$_\text{r}$ is compared to classical ridge regression (Section 4.2.2) either applied to simulated data D_SIM or real data D_NIR. Ridge hyper parameter t (equation (4.16)) is fixed using cross-validation. We set λ_2 for Dual-sPLS$_\text{r}$ (equation (4.25)) to $\frac{1}{t}$ for easier comparison. All other parameters are kept as for Dual-sPLS$_\text{l}$ (Section 4.5.1). Looking at top-left and -right in Figure 4.7 Dual-sPLS$_\text{r}$ reaches a plateau for D_SIM after five latent components. Moreover, its RMSE values are slightly lower than ridge's for both calibration and validation. We can safely select six latent components as before. Reference coefficients for ridge are misleading because the largest ones do not reside in influencing areas. They therefore can not be used for data interpretation. By selecting only fifty variables, located in red regions governing the model, Dual-sPLS$_\text{r}$ better succeeds in both prediction and localization. Similar conclusions can be drawn for real data D_NIR on RMSE values. Dual-sPLS$_\text{r}$ even better predicts the response $\mathbf{y}$ with only four components. Regression coefficients (Figure 4.8 bottom) yield comments akin to above. While ridge

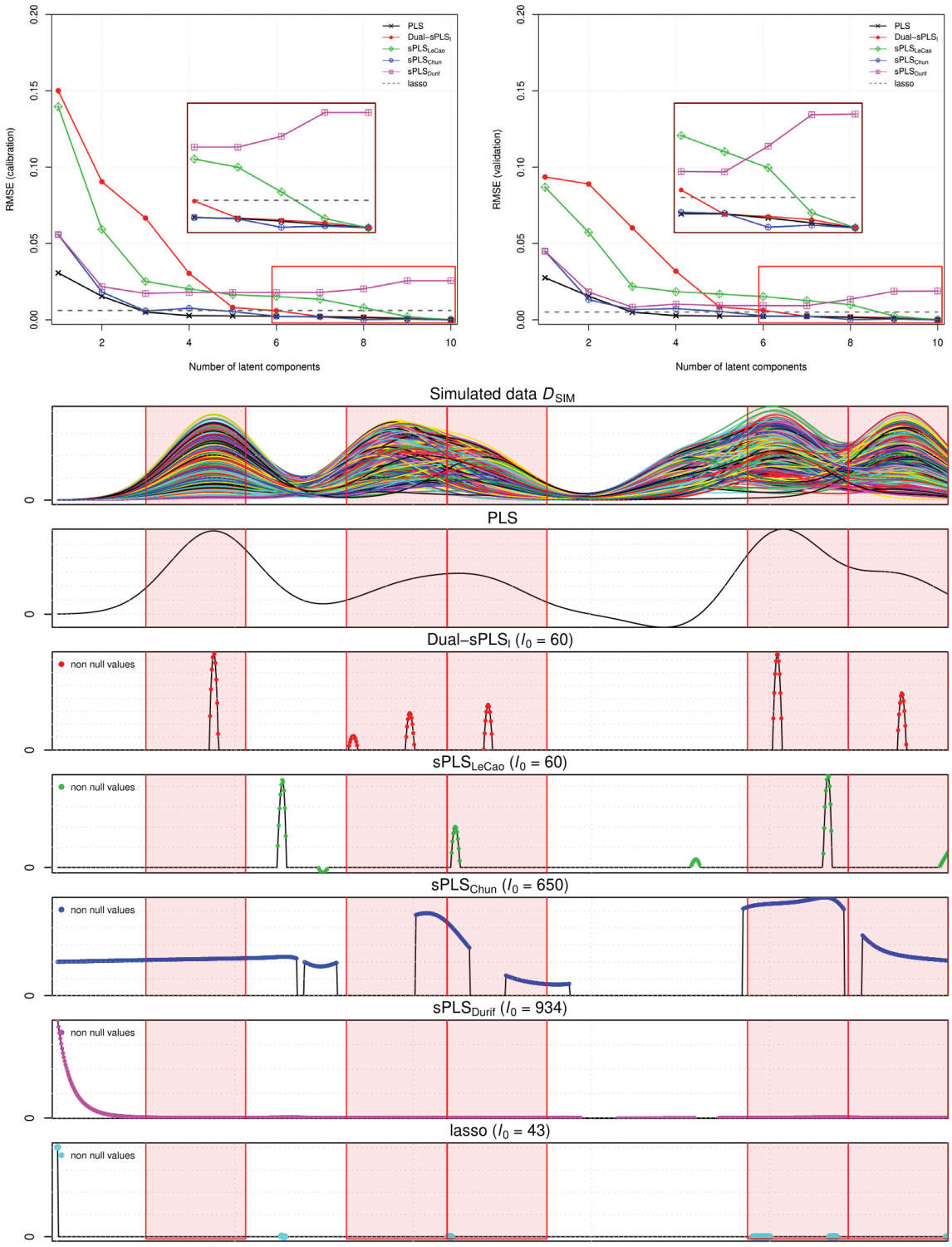

Figure 4.4 ~ *Dual-sPLS$_l$ evaluation on simulated data D_{SIM}. (Top) RMSE values for calibration (left) and validation (right) with respect to the number of latent components. (Bottom) From top to bottom: simulated data D_{SIM}, regression coefficients of PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ for six components, and lasso.*

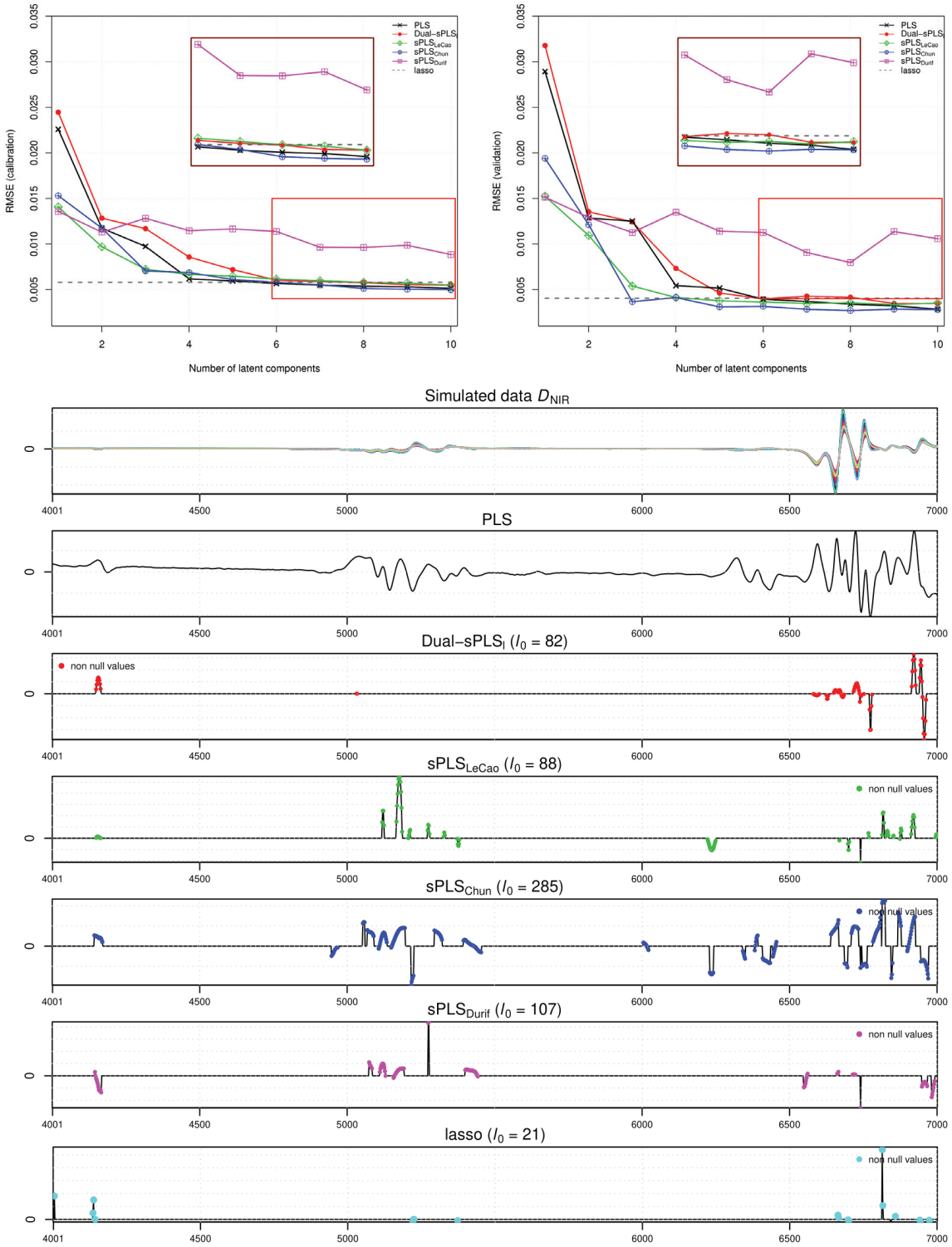

Figure 4.5 ~ *Dual-sPLS$_l$ evaluation on real data D_{NIR}. (Top) RMSE values for calibration (left) and validation (right) with respect to the number of latent components. (Bottom) From top to bottom: original data D_{NIR}, regression coefficients of PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ for six components, and lasso.*

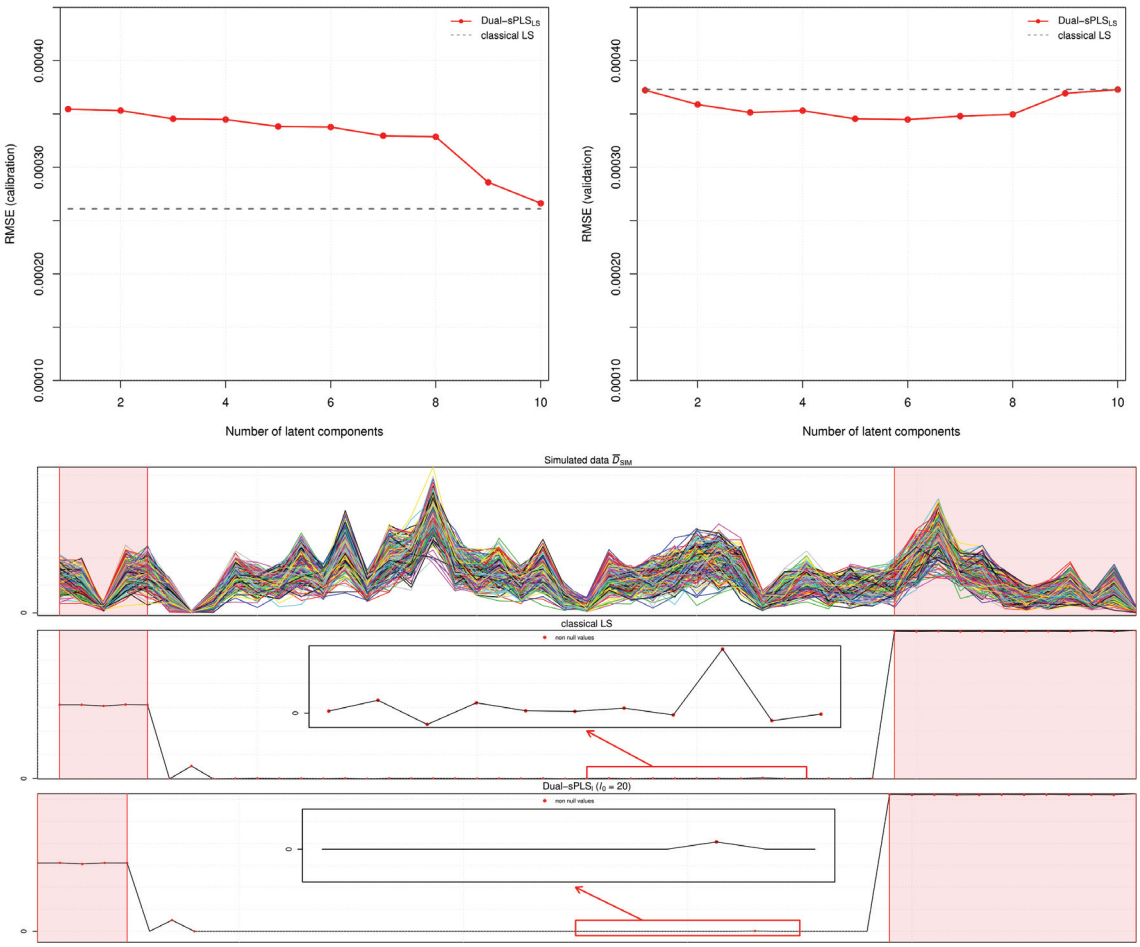

Figure 4.6 ~ *Dual-sPLS$_{LS}$ evaluation on simulated data $\overline{D}_{\text{SIM}}$. (Top) RMSE values for calibration (left) and validation (right) with respect to the number of latent components. (Bottom) From top to bottom: simulated data $\overline{D}_{\text{SIM}}$, regression coefficients of least squares and Dual-sPLS$_{LS}$ for five components.*

apparently emphasizes unimportant features, Dual-sPLS$_r$ seems more reliable in identifiying of relevant variables to predict density using chemical data.

4.6 Conclusion and perspectives

We propose a family of dual sparse Partial Least Squares algorithms that broadens the compass of standard PLS. Along with competitive prediction accuracy with respect to PLS as used in chemometrics, we expect additional benefits in dimension reduction or model interpretability. This is achieved by supplementing the traditional optimization problem with well-chosen dual norms.
We chiefly validate this approach by borrowing three classical regression penalties: lasso, least-squares, ridge. Each proposed Dual-sPLS draws close to the reference in calibration/validation performance with a reduced number of latent components. This is assessed in a benchmark on both realistic simulated models and real near infrared spectroscopy data, against a standard baseline and sparse contenders. Coefficients are sieved with a user-defined sparsity target. They are well-located in influential data ranges, suggesting a means for better interpretability of the trained prediction reduced model. Pseudo-lasso and ridge Dual-sPLS avatars exhibit close colocation of selected features in both datasets despite different penalties. This suggests a robust identification of meaningful information in signals.
The Dual-sPLS framework is thus a good candidate for a host of applications. We provide it as an open-source package in R [5]. It can be prolonged to other field-favorite penalties, for instance elastic net. We plan to evaluate the alluded "pseudo-group lasso" option, to refine feature selection on important contiguous areas, or to combine datasets providing complementary information on the predicted response. To improve prediction robustness or reduce the number of necessary latent components (toward three or four instead of six), we explore additional diversity enhancement preprocessing, such as higher-order derivatives and discrete wavelet transforms. Last, as PLS deserves sounder statistical foundations, we endeavor a study of asymptotic convergence bounds.

4.7 Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

4.8 Acknowledgements

This work was performed within the framework of the LABEX MILYON (ANR-10-LABX- 0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX- 0007) operated by the French National Research Agency (ANR). We acknowledge the financial support of the Research Council of the Saint Joseph University of Beirut. Ghislain Durif contributed in the code validation procedure in R. IFPEN provided the real NIR data set used in the applications. We thank Noémie Caillol, Luca Castelli and Irène Gannaz for useful comments.

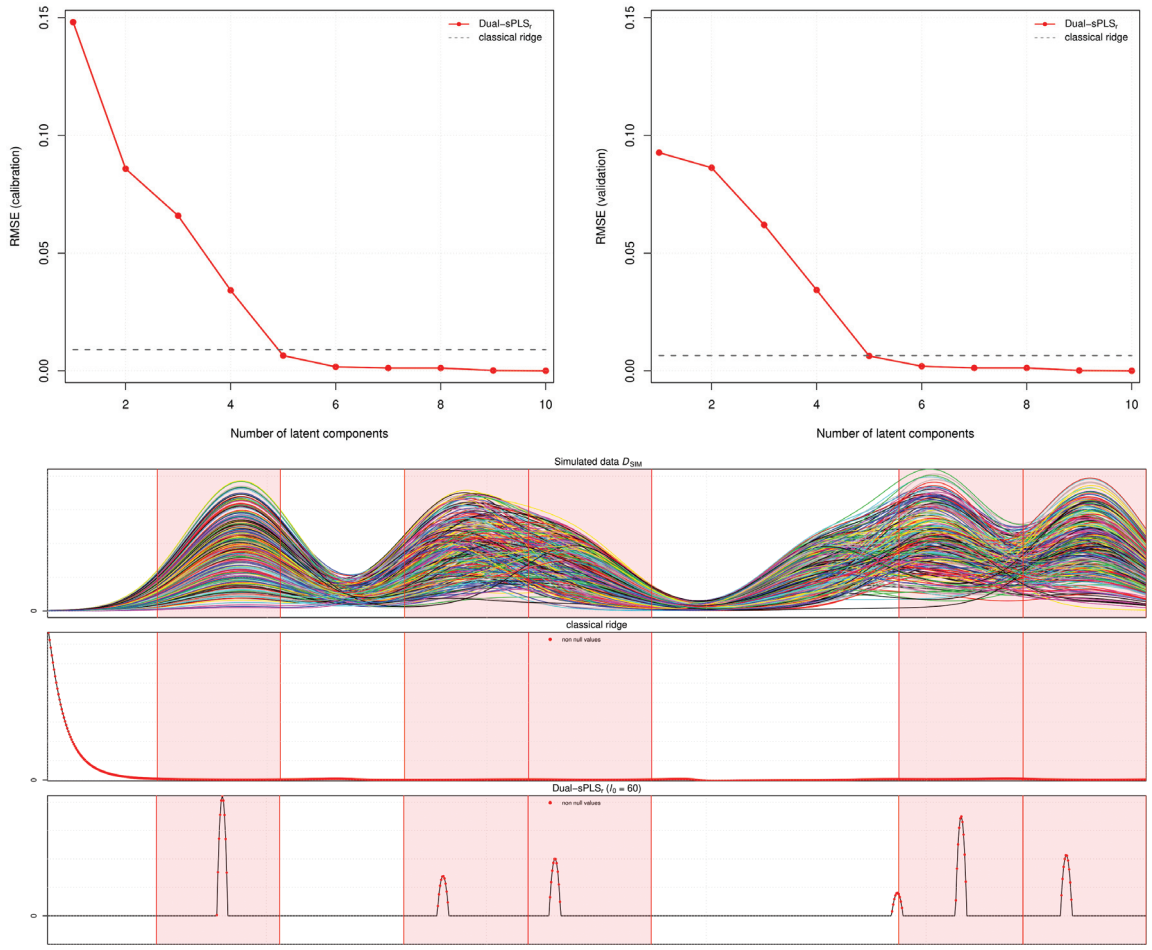

Figure 4.7 ~ *Dual-sPLS$_r$ evaluation on simulated data D_{SIM}. (Top) RMSE values for calibration (left) and validation (right) with respect to the number of latent components. (Bottom) From top to bottom: original data D_{SIM}, regression coefficients of ridge and Dual-sPLS$_r$ for five components.*

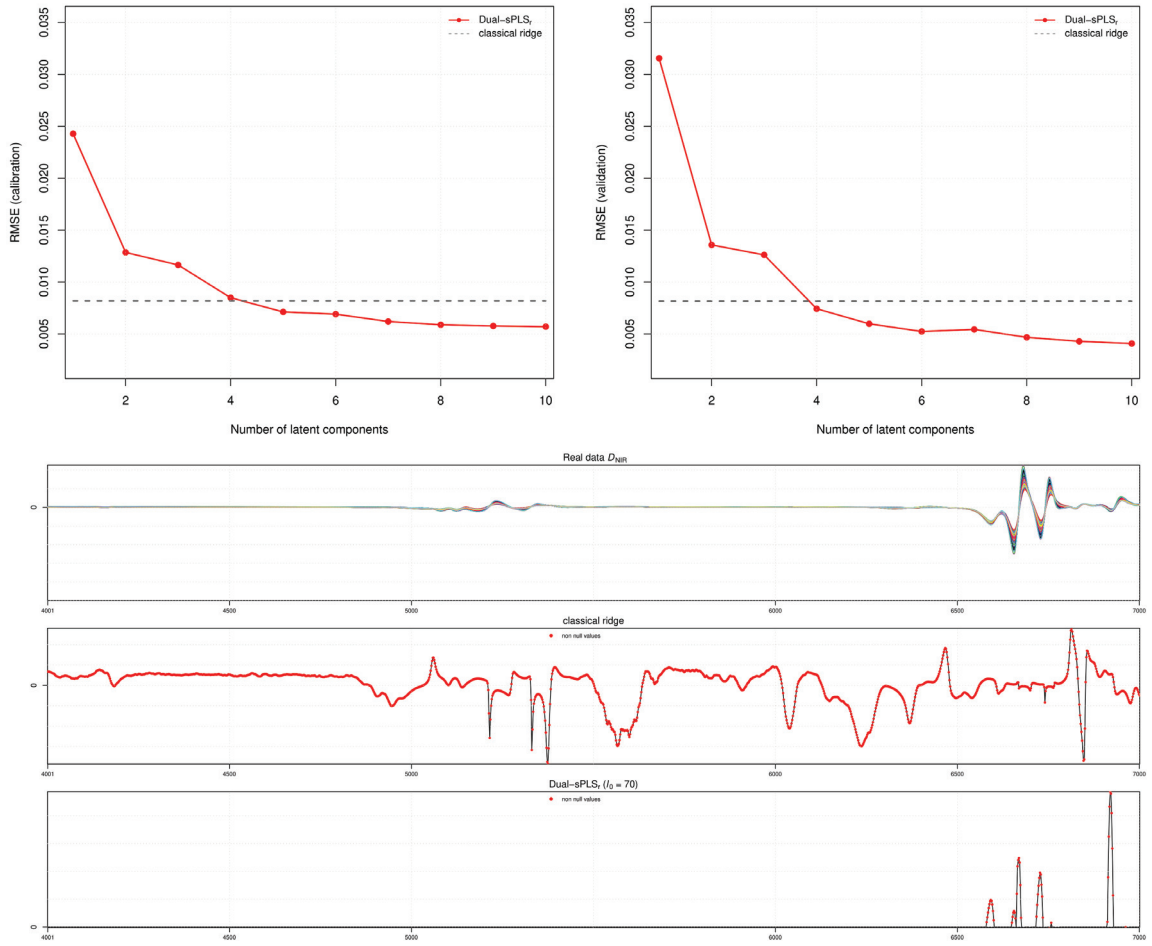

Figure 4.8 ~ *Dual-sPLS$_r$ evaluation on real data D_{NIR}. (Top) RMSE values for calibration (left) and validation (right) with respect to the number of latent components. (Bottom) From top to bottom: original data D_{NIR}, regression coefficients of ridge and Dual-sPLS$_r$ for five components.*

Our mentor, colleague or friend François Wahl passed away unexpectedly during the writing of this paper. He was the driving force of our team.

Appendix

Appendix 4.A Detailed resolution of Dual-sPLSs

4.A.1 Dual-sPLS pseudo-group lasso

We recall Equation (4.23): the Dual-sPLS$_{\text{gl}}$ norm case applied to optimization Problem (4.20). Note that here

- g represents a group of $P(g)$ index extracted from $\{1,\dots,P\}$;
- G represents the number of groups;
- $\mathbf{w}_g$ represents the values of index g in the loading vector $\mathbf{w}$.

We denote $\mathbf{z}_g$ the variables of $\mathbf{z}$ belonging to group g. We impose $\mathbf{z}_g$ and $\mathbf{w}_g$ to be in the same orthant. Let $\boldsymbol{\delta}_g$ be their vector of signs. By differentiating equation (4.23) we obtain

$$\frac{\partial \Omega(\mathbf{w})}{\partial w_g} = \frac{\alpha_g w_g}{\|\mathbf{w}_g\|_2} + \alpha_g \lambda_g \delta_g \,. \tag{4.39}$$

Using Lagrange multipliers as in Section 4.3.1, we compare (4.26) to (4.39) and obtain for $g \in \{1,\dots,G\}$:

$$\frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2} = \frac{\mathbf{z}_g}{\alpha_g \mu} - \lambda_g \delta_g \,, \tag{4.40}$$

which is simplified by

$$\frac{\mathbf{w}_g}{\|\mathbf{w}_g\|_2} = \frac{1}{\mu \alpha_g} \mathbf{z}_{\nu_g} \,, \tag{4.41}$$

where

$$\mathbf{z}_{\nu_g} = \boldsymbol{\delta}_g (|\mathbf{z}_g| - \nu_g)_+ \quad \text{for } g \in \{1,\dots,G\} \,. \tag{4.42}$$

Here $\nu_g = \mu \alpha_g \lambda_g$ and controls the amount of variables that we would like to shrink to zero. By applying ℓ_2-norm to (4.41), we conclude that for $g \in \{1,\dots,G\}$,

$$\mu = \sum_{g=1}^{G} \|\mathbf{z}_{\nu_g}\|_2 \qquad \text{and} \qquad \alpha_g = \frac{\|\mathbf{z}_{\nu_g}\|_2}{\mu} \,. \tag{4.43}$$

The term $\|\mathbf{w}_g\|_2$ is more involved. Thus, we simply use grid search. For each group g, ten possible values are chosen to be tested. The selection is done by detecting the maximum value

of $\|\mathbf{w}_g\|_2$ for each group g, denoted $\|\mathbf{w}_g\|_2^{max}$. The latter is computed by zeroing $\|\mathbf{w}_{g'}\|_2$ for all groups $g' \neq g$ and is expressed as:

$$\|\mathbf{w}_g\|_2^{max} = \frac{\mu}{\Omega_g(\mathbf{z}_{\nu_g})}\,. \tag{4.44}$$

Then, ten values of each group g are selected inside the interval $[0, \|\mathbf{w}_g\|_2^{max}]$. The grid search tests all the possible combinations and retains the one that allows the smallest error. We summarize the methodology with Algorithm 6.

Algorithm 6: Dual-sPLS$_{\text{gl}}$ algorithm

Input: $\mathbf{X}^1, \ldots, \mathbf{X}^G, \mathbf{y}$, M (number of components desired), ς (shrinking ratio), $\alpha_1, \ldots, \alpha_g$.
for $m = 1, \ldots, M$ **do**
  $\mathbf{X}_m = (\mathbf{X}^1, \ldots, \mathbf{X}^G)$ (combining data)
  $\mathbf{z}_m = \mathbf{X}_m^T \mathbf{y}$ (weight vector)
  Find ν adaptively according to ς for each group seperatly
  $\mathbf{z}_{\nu_g} = \boldsymbol{\delta}_g(|\mathbf{z}_g| - \nu_g)_+$ for $g \in \{1, \ldots, G\}$ (applying the threshold)
  $\mu = \sum_{g=1}^{G} \|\mathbf{z}_{\nu_g}\|_2$
  $\alpha_g = \dfrac{\|\mathbf{z}_{\nu_g}\|_2}{\mu}$ and $\lambda_g = \dfrac{\nu_g}{\alpha_g \mu}$ for $g \in \{1, \ldots, G\}$
  $\|\mathbf{w}_g\|_2^{max} = \dfrac{\mu}{\Omega_g(\mathbf{z}_{\nu_g})}$ for $g \in \{1, \ldots, G\}$
  selection of the values of $\|\mathbf{w}_g\|_2$ for each group
  $\mathbf{w}_g = \dfrac{\|\mathbf{w}_g\|_2}{\mu \alpha_g} \mathbf{z}_{\nu_g}$ for $g \in \{1, \ldots, G\}$ (loadings)
  $\mathbf{w}_g = \left(\mathbf{w}_g\right)_{g=1}^{G}$
  $\mathbf{t}_m = \mathbf{X}_m \mathbf{w}_m$ (component)
  $\mathbf{X}_{m+1} = \mathbf{X}_m - \mathcal{P}_{\mathbf{t}_m} \mathbf{X}_m$ (deflation)
end for
Compute $\hat{\boldsymbol{\beta}}$.

4.A.2 Dual-sPLS pseudo-least squares

We recall Equation (4.24): the Dual-sPLS$_{\text{LS}}$ pseudo case applied to optimization Problem (4.20). We impose $\mathbf{N}_1\mathbf{z}$ and $\mathbf{N}_1\mathbf{w}$ to be in the same orthant. Let $\boldsymbol{\delta}_2$ be their vector of signs. By differentiating (4.24) we obtain

$$\nabla\Omega(\mathbf{w}) = \lambda \mathbf{N}_1^T \boldsymbol{\delta}_2 + \frac{\mathbf{X}^T\mathbf{X}\mathbf{w}}{\|\mathbf{X}\mathbf{w}\|_2}\,. \tag{4.45}$$

Using Lagrange multipliers as in Section 4.3.1, we compare (4.26) to (4.45) and obtain

$$\frac{\mathbf{w}}{\|\mathbf{X}\mathbf{w}\|_2} = (\mathbf{X}^T\mathbf{X})^{-1}\frac{\mathbf{z}}{\boldsymbol{\mu}} - \lambda(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{N}_1^T\boldsymbol{\delta}_2\,, \tag{4.46}$$

imposing the invertibility of $\mathbf{X}^T\mathbf{X}$. We choose $\mathbf{N}_1$ such as

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{N}_1^T\boldsymbol{\delta}_2 = \text{sign}\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}\right). \tag{4.47}$$

The resolution steps are be similar to the ones from Dual-sPLS$_\text{l}$ but instead of applying the threshold on $\mathbf{z}$, we apply it on $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}$ which is exactly the classical Least Squares regression coefficients $\hat{\boldsymbol{\beta}}^{LS}$. So, the simplified solution is

$$\frac{\mathbf{w}}{\|\mathbf{X}\mathbf{w}\|_2} = \frac{1}{\mu}\text{sign}(\hat{\boldsymbol{\beta}}_{LS_j})(|\hat{\boldsymbol{\beta}}_{LS_j}| - \nu)_+\,, \tag{4.48}$$

where ν is chosen adaptively.
For a simpler algorithm, $\|\mathbf{X}\mathbf{w}\|_2$ is not computed as it is not mandatory in this case. Additionally, $\mathbf{w}$ only depends on ν and $\hat{\beta}_{LS}$, which means $\mathbf{N}_1$ does not intervene in the computation of the optimal solution. Thus, proving that $\mathbf{N}_1$ exists is enough. (4.47) implies the following

$$\mathbf{N}_1^T\boldsymbol{\delta}_2 = (\mathbf{X}^T\mathbf{X})\text{sign}\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}\right). \tag{4.49}$$

Let $\mathbf{w}$ be an eignvector of $\mathbf{N}_1$, and $\mathbf{N}_1'$ be such as

$$\mathbf{N}_1' = \mathbf{N}_1 - \mathbf{w}^T\mathbf{w} \quad \text{and } \mathbf{N}_1'\mathbf{w} = 0\,. \tag{4.50}$$

Therefore, using (4.49) we have

$$\mathbf{N}_1'\boldsymbol{\delta}_2 = (\mathbf{X}^T\mathbf{X})\text{sign}\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}\right) - \mathbf{w}\mathbf{w}^T\boldsymbol{\delta}_2 \quad \text{with } \mathbf{N}_1'\mathbf{w} = 0\,. \tag{4.51}$$

With $\mathbf{N}_1$ a square matrix of P variables, (4.51) is a system of P^2 unknowns, P equations and P contraints. It can be verified by an infinite number of solutions.
The following algorithm reformulates the previous steps:

Algorithm 7: DUAL-SPLS$_\text{LS}$ ALGORITHM

Input: $\mathbf{X}, \mathbf{y}, M$ (number of components desired), ς (shrinking ratio)
$\mathbf{X}_1 = \mathbf{X}$
for $m = 1, \ldots, M$ **do**
  $\mathbf{z}_m = \mathbf{X}_m^T\mathbf{y}$ (weight vector)
  $\hat{\beta}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{z}$
  Find ν adaptively according to ς and $\hat{\beta}_{LS}$
  $\mathbf{z}_\nu = (\text{sign}(\hat{\boldsymbol{\beta}}_{LS})(|\hat{\boldsymbol{\beta}}_{LS}| - \nu)_+)$ (applying the threshold)
  $\mathbf{w}_m = \dfrac{\mathbf{z}_\nu}{\mu}$ (loadings)
  $\mathbf{w}_m = \dfrac{\mathbf{w}_m}{\|\mathbf{w}\|_2}$ (normalizing loadings)
  $\mathbf{t}_m = \mathbf{X}_m\mathbf{w}_m$ (component)
  $\mathbf{X}_{m+1} = \mathbf{X}_m - \mathcal{P}_{\mathbf{t}_m}\mathbf{X}_m$ (deflation)
end for
Compute $\hat{\boldsymbol{\beta}}$.

4.A.3 Dual-sPLS pseudo-ridge

We recall Equation (4.25): the Dual-sPLS$_\mathrm{r}$ pseudo case applied to optimization Problem (4.20). We impose $\mathbf{z}$ and $\mathbf{w}$ to be in the same orthant. Let $\boldsymbol{\delta}$ be their vector of signs. By differentiating (4.25), we obtain

$$\nabla\Omega(\mathbf{w}) = \lambda_1\delta + \lambda_2\frac{\mathbf{X}^T\mathbf{X}\mathbf{w}}{\|\mathbf{X}\mathbf{w}\|_2} + \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\,. \tag{4.52}$$

Using Lagrange multipliers as in Section 4.3.1, we compare (4.26) to (4.52) and obtain

$$\frac{\mathbf{w}}{\|\mathbf{w}\|_2} = \left(\nu_2\mathbf{X}^T\mathbf{X} + I_P\right)^{-1}(\mathbf{z} - \nu_1\boldsymbol{\delta})\,, \tag{4.53}$$

where $\nu_1 = \lambda_1\mu$ and $\nu_2 = \lambda_2\dfrac{\|\mathbf{w}\|_2}{\|\mathbf{X}\mathbf{w}\|_2}$.

In line with Dual-sPLS$_\mathrm{l}$, we note $\mathbf{z}_{\mathbf{X},\nu_2} = \left(\nu_2\mathbf{X}^T\mathbf{X} + I_P\right)^{-1}\mathbf{z}$ and $\boldsymbol{\delta}_\mathbf{X}$ its vector of signs. We exhibit a solution imposing that $\mathbf{w}$ and $\mathbf{z}_{\mathbf{X},\nu_2}$ are in the same orthant, which leads to the following reformulation of (4.53):

$$\frac{\mathbf{w}}{\|\mathbf{w}\|_2} = \frac{1}{\mu}\delta_\mathbf{X}(|\mathbf{z}_{\mathbf{X},\nu_2}| - \nu_1)_+\,. \tag{4.54}$$

The threshold ν_1 is chosen with the adaptive procedure described in Section 4.3.2 and Figure 4.1. However, in this case, we compare ν_1 to $|\mathbf{z}_{\mathbf{X},\nu_2}|$. Since the latter is colinear to $\mathbf{z}$, the shrinkage is adequate. Denoting $\mathbf{z}_\nu = \delta_\mathbf{X}(|\mathbf{z}_{\mathbf{X},\nu_2}| - \nu_1)_+$, simple computations lead to

$$\mu = \|\mathbf{z}_\nu\|_2\,, \tag{4.55}$$

and

$$\mathbf{w} = \frac{\mu}{\nu_1\|\mathbf{z}_\nu\|_1 + \nu_2\|\mathbf{X}\mathbf{z}_\nu\|_2^2 + \mu^2}\,. \tag{4.56}$$

It is summarized in Algorithm 8:

Algorithm 8: DUAL-SPLS$_{\mathrm{R}}$ ALGORITHM

Input: $\mathbf{X}, \mathbf{y}, M$ (number of components desired), ς (shrinking ratio), ν_2
$\mathbf{X}_1 = \mathbf{X}$
for $m = 1, \ldots, M$ **do**
  $\mathbf{z}_m = \mathbf{X}_m^T \mathbf{y}$ (weight vector)
  $\mathbf{z}_{\mathbf{X},\nu_2} = \left(\nu_2 \mathbf{X}^T \mathbf{X} + I_P\right)^{-1} \mathbf{z}$
  Find ν adaptively according to ς and $|\mathbf{z}_{\mathbf{X},\nu_2}|$
  $\boldsymbol{\delta}_{\mathbf{X}}$ vector of signs of $\mathbf{z}_{\mathbf{X},\nu_2}$
  $\mathbf{z}_\nu = \delta_{\mathbf{X}}(|\mathbf{z}_{\mathbf{X},\nu_2}| - \nu_1)_+$ (applying the threshold)
  $\mu = \|\mathbf{z}_\nu\|_2$ and $\lambda = \dfrac{\nu}{\mu}$
  $\mathbf{w}_m = \dfrac{\mu}{\nu_1 \|\mathbf{z}_\nu\|_1 + \nu_2 \|\mathbf{X}\mathbf{z}_\nu\|_2^2 + \mu^2}$ (loadings)
  $\mathbf{t}_m = \mathbf{X}_m \mathbf{w}_m$ (component)
  $\mathbf{X}_{m+1} = \mathbf{X}_m - \mathcal{P}_{\mathbf{t}_m} \mathbf{X}_m$ (deflation)
end for
Compute $\hat{\boldsymbol{\beta}}$.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Appendix 4.B Complementary plots

As mentioned in Section 4.5, metrics MAE and R^2 were also computed. They support our findings based on RMSE, as they yield similar results (see Figures 4.B1, 4.B2,4.B3,4.B4 and 4.B5).
Figures 4.B6 and 4.B7 represent a clearer perpective on regression coefficients for Dual-sPLS$_{\mathrm{l}}$ applied D_{SIM} and D_{NIR} from Section 4.5.1.
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

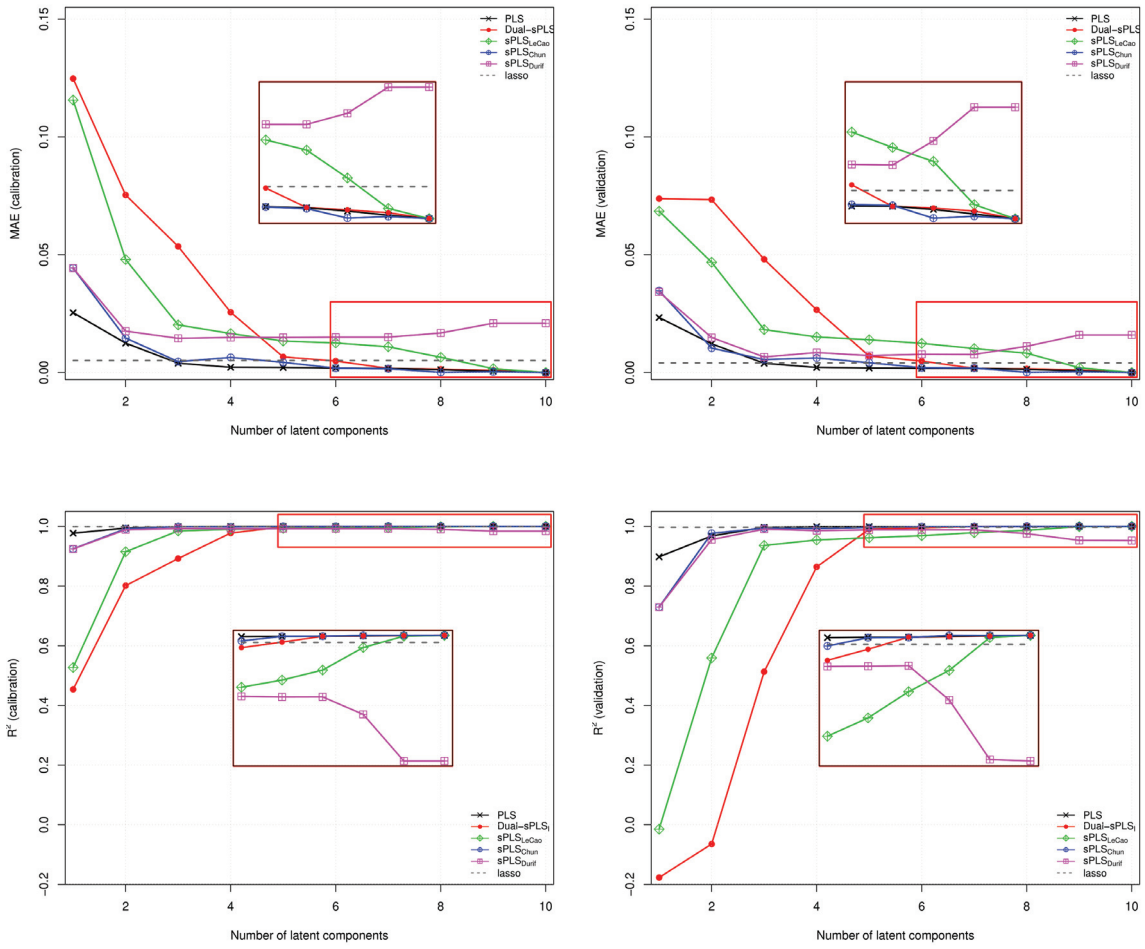

Figure 4.B1 ~ *Dual-sPLS$_l$ evaluation on simulated data D_{SIM}. MAE (top) and R^2 (bottom) values for calibration (left) and validation (right) with respect to the number of latent components derived from PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ and lasso regressions.*

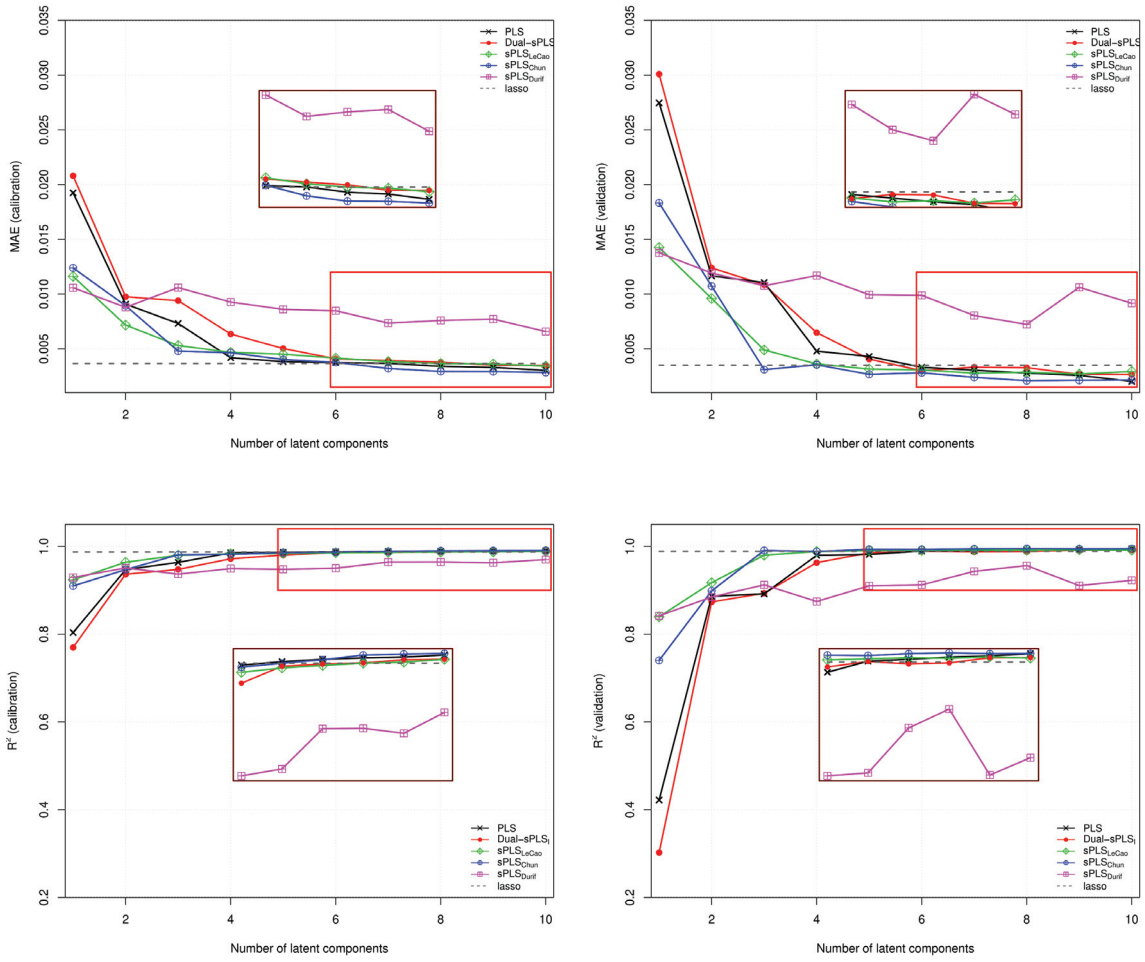

Figure 4.B2 ~ *Dual-sPLS$_l$ evaluation on real data D_{NIR}. MAE (top) and R^2 (bottom) values for calibration (left) and validation (right) with respect to the number of latent components derived from PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ and lasso regressions.*

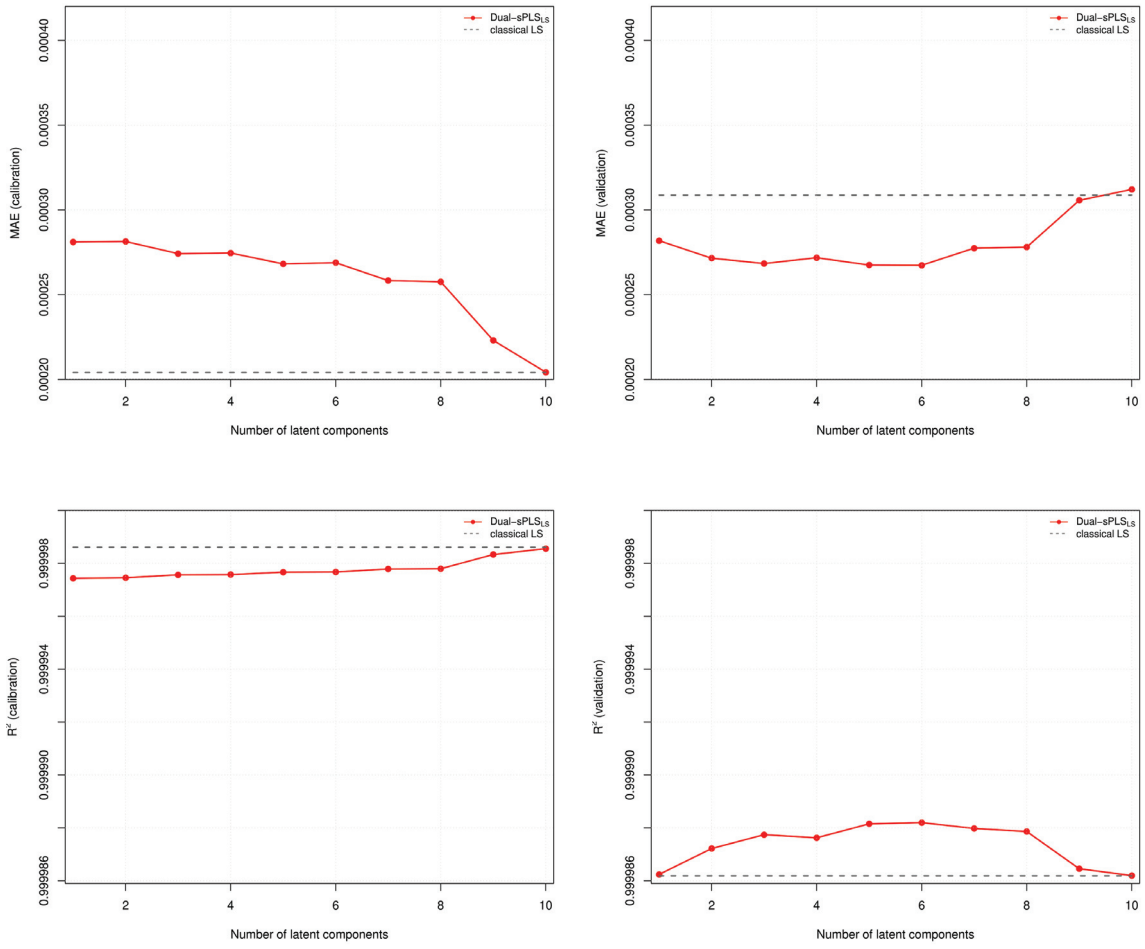

Figure 4.B3 ~ *Dual-sPLS$_{LS}$ evaluation on simulated data $\overline{D}_{\mathrm{SIM}}$. MAE (top) and R^2 (bottom) values for calibration (left) and validation (right) with respect to the number of latent components derived from Dual-sPLS$_{LS}$ and least squares regressions.*

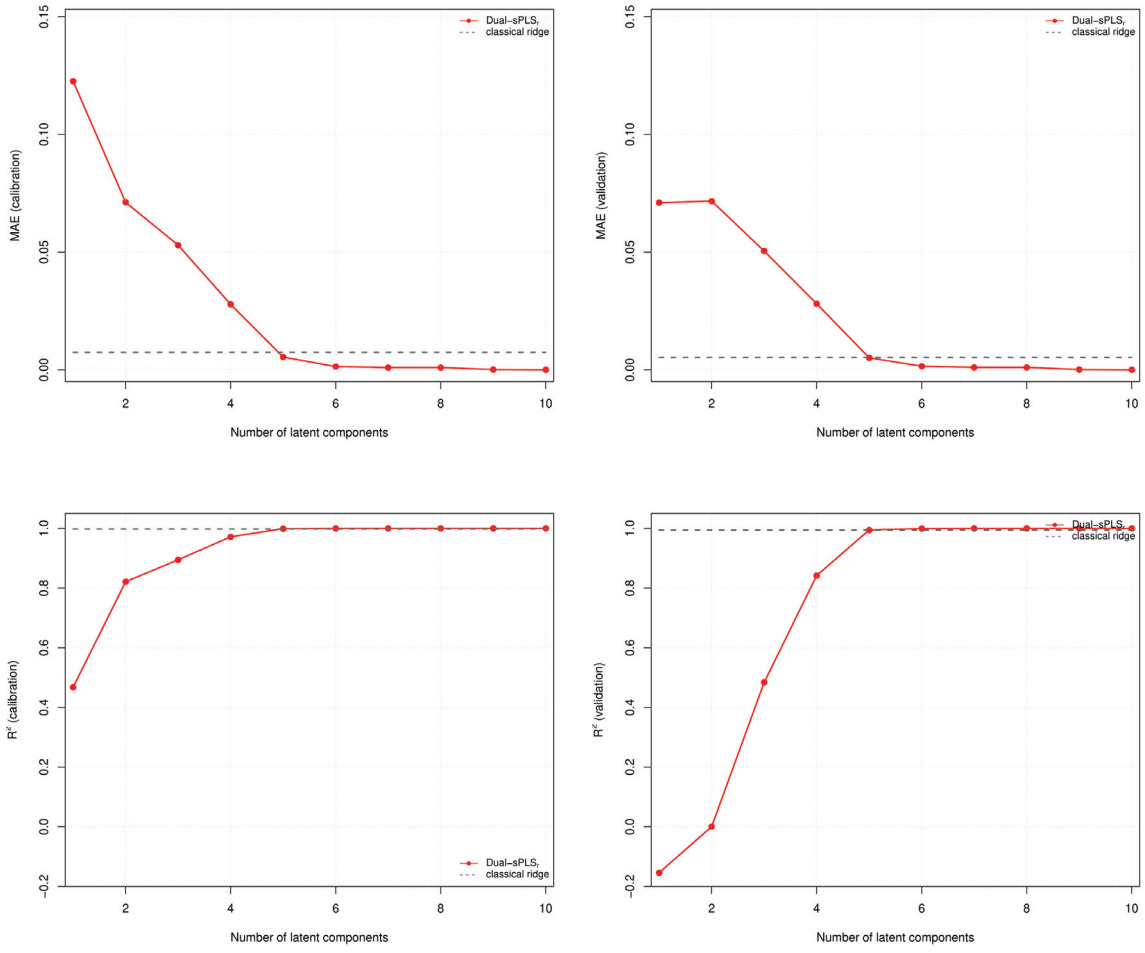

Figure 4.B4 ~ *Dual-sPLS$_r$ evaluation on simulated data D_{SIM}. MAE (top) and R^2 (bottom) values for calibration (left) and validation (right) with respect to the number of latent components derived from Dual-sPLS$_r$ and ridge regressions.*

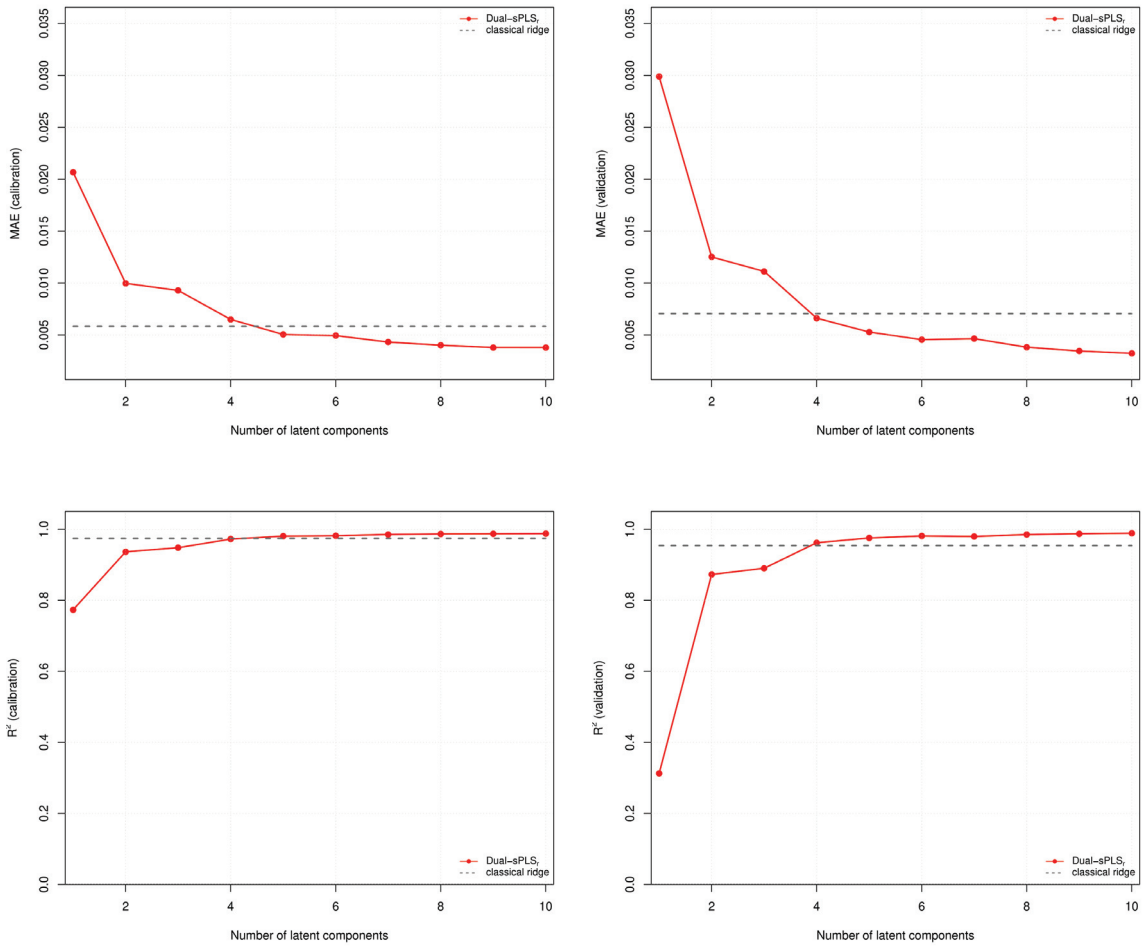

Figure 4.B5 ~ *Dual-sPLS$_r$ evaluation on real data D_{NIR}. MAE (top) and R^2 (bottom) values for calibration (left) and validation (right) with respect to the number of latent components derived from Dual-sPLS$_r$ and ridge regressions.*

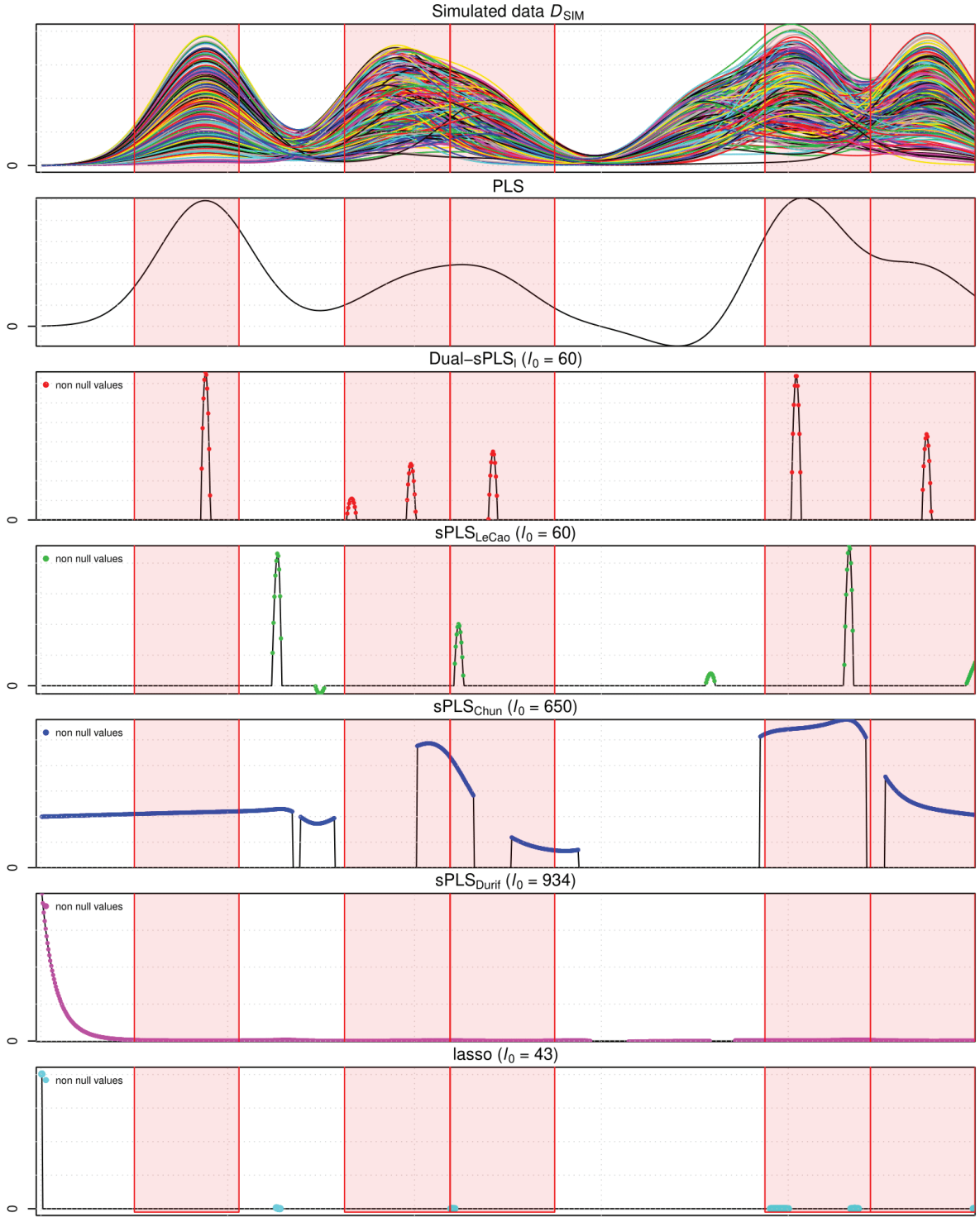

Figure 4.B6 ~ *Dual-sPLS$_l$ evaluation on simulated data D_{SIM}. From top to bottom: simulated data D_{SIM}, regression coefficients of PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ for six components, and lasso.*

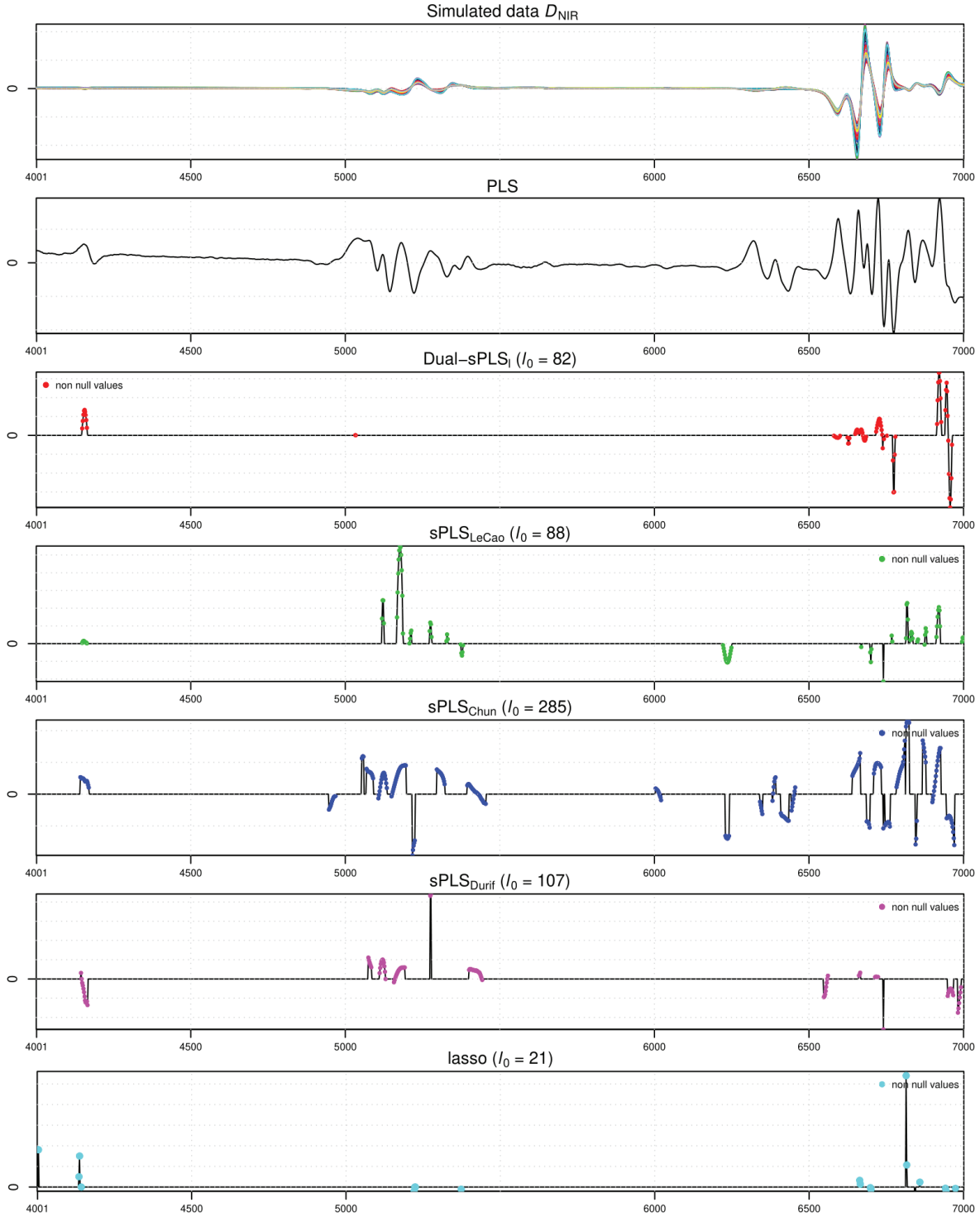

Figure 4.B7 *~ Dual-sPLS$_l$ evaluation on real data D_{NIR}. From top to bottom: simulated data D_{SIM}, regression coefficients of PLS, Dual-sPLS$_l$, sPLS$_{LeCao}$, sPLS$_{Chun}$, sPLS$_{Durif}$ for six components, and lasso.*

5

Package dual.spls

This chapter is dedicated to the introduction of the dual.spls package, which implements the Dual sparse partial least squares (Dual-sPLS) method in R. The statistical background behind the method is briefly explained, including Partial Least Squares and lasso algorithm, as well as the theory underlying Dual-sPLS. It then goes on to provide a comprehensive description of the package and how to use it, with illustrating chunks of code. The package includes the four primary Dual-sPLS penalties, a calibration and validation splitting routine, and a weighted data simulation algorithm. As Dual-sPLS can be used to combine explanatory variables related to the same response, the chapter benchmarks the method against state-of-the-art techniques using suitable simulated data. This chapter is a preprint and is aimed to be submitted to Journal of Statistical Software for June 2023.

Contents

5.1 Introduction

Data has always played a crucial role in scientific research due to the insights it can provide. However, the rapid advancement of technology has brought new challenges to the field of data analysis. High-dimensional data, in particular, has emerged as a significant obstacle especially found in regression contexts. In the latter, $N \in \mathbb{R}$ observations are represented by $Q \in \mathbb{R}$ response variables (stored in $\mathbf{Y} \in \mathbb{R}^{N\times Q}$) and $P \in \mathbb{R}$ explanatory variables (stored in $\mathbf{X} \in \mathbb{R}^{N\times P}$) Dimensionality issues are common when trying to relate both sets of features. First, it prevents the use of naive statistical prediction models like the classical Ordinary Least Squares (OLS) inapplicable do to the singularity of the large matrix $\mathbf{X}$. Second, as data are high dimensional, more noise appears which hinders insightful interpretation of data. Dimension reduction techniques are commonly used when facing these challenges and include projection and penalization methods. On the one hand, projection strategies particularly address multi-collinearity problems between variables. They project data onto a smaller space summarizing original data. They also reduce storage requirements and algorithm running time. With a smaller number of variables and reduced noise in the data, the model is thus improved and data visualization becomes feasible. Principal Component Analysis [51] is a popular projection technique that builds new uncorrelated and orthogonal components that successively maximize the variance between the projected new axes. Partial Least Squares (PLS) [103] regression differs from PCA by generating components using both predictors $\mathbf{X}$ and target variables $\mathbf{Y}$. It is an iterative method that deals with highly correlated data and results in accurate forecasts. PLS is implemented in multiple are straightforward and simple to handle algorithms NIPALS[103], SIMPLS [32], etc.. However, statistical interpretation is often deceiving as its associated regression coefficients fail to accurately localize most important variables. On the other hand, penalization-based methods most often address dimensionality issues with shrinkage. The latter provides insights about the relevance of the variables and their localization when dealing with functional data allowing better interpretability compared to PLS. The lasso procedure ([95]) is popularly adopted. It performs ℓ_1 regularization which induces sparsity. Nevertheless, lasso has some recognizable limitations [112, 47] as it is known to be sensitive to the data and has the tendency to select moderately representative variables when using strongly correlated variables

The sparse PLS approach [61] combines both dimension reduction techniques. It adds to the PLS framework a selection step inspired by the lasso. Several sPLS variants exist in the literature [25, 37], but despite their sparse results, they do not always yield pertinent feature localization for instance in some functional data cases. Dual sparse PLS [3] was recently suggested for univariate responses $\mathbf{y} = \mathbf{Y}$, with $\mathbf{y} \in \mathbb{R}^N$. It generalizes the standard PLS1 [57] algorithm by extending it with adequate norm-based regularization inspired by state-of-the-art methods: (group) lasso, least squares, ridge. It balances accuracy in predictions and satisfactory interpretation. In the present work, we introduce the associated **dual.spls** package implemented in R.. It is self-contained including the four primary Dual-sPLS penalties, a new calibration and validation splitting routine called CalValXy [2] and a weighted data simulation algorithm.

The paper is structured as follows: first we detail some notations, then the statistical background (PLS and penalization) and the theory underlying the Dual-SPLS is briefly introduced in Section 5.2. Next, Section 5.3 presents a full description of the package with illustrating chunks of codes. Afterward, Section 5.4 benchmarks the pseudo group lasso penalty against its

counterpart and classical PLS using suitable simulated data. Finally, we conclude by providing insight in Section 5.5 and further information in the appendix.

Notation and definitions

Matrices, vectors and scalars are respectively denoted by boldface uppercase letters, boldface lowercase and light lowercase letters, e.g. $\mathbf{X}$, $\mathbf{y}$ and λ. The transpose of matrix $\mathbf{X}$ is $\mathbf{X}^T$. The identity matrix of size P is represented by I_P. The ℓ_1-norm and the ℓ_2-norm of vector a $\mathbf{w}$ of length P are respectively:

$$\|\mathbf{w}\|_1 = \sum_{p=1}^{P} |w_p| \qquad \text{and} \qquad \|\mathbf{w}\|_2 = \sqrt{\sum_{p=1}^{P} |w_p|^2}\,. \tag{5.1}$$

We denote by $\ell_0(\mathbf{w})$ the sparsity index or count measure of the non-zero coordinates of $\mathbf{w}$ and $\ell_0^c(\mathbf{w})$ its complement i.e. $\ell_0^c(\mathbf{w}) = P - \ell_0(\mathbf{w})$. To choose the number of latent variables we rely on the mean squared error (MSE) expressed as

$$\text{MSE} = \frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2\,, \tag{5.2}$$

for a response vector $\mathbf{y}$ of N observations and a given corresponding estimate vector $\hat{\mathbf{y}}$. For performance evaluation, we choose the root mean squares error (RMSE), the mean absolute error (MAE) and the determination coefficient (R^2), respectively defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y_n - \hat{y}_n)^2} = \frac{1}{\sqrt{N}} \|\mathbf{y} - \hat{\mathbf{y}}\|_2\,, \tag{5.3}$$

$$\text{MAE} = \frac{1}{N} \sum_{n=1}^{N} |y_n - \hat{y}_n| = \frac{1}{N} \|\mathbf{y} - \hat{\mathbf{y}}\|_1\,, \tag{5.4}$$

$$\mathrm{R}^2 = \frac{\sum_{n=1}^{N} (y_n - \bar{y})^2}{\sum_{n=1}^{N} (y_n - \hat{y}_n)^2} \qquad \text{where} \qquad \bar{y} = \frac{\sum_{n=1}^{N} y_n}{N}\,. \tag{5.5}$$

The vector of signs of $\mathbf{w}$ entries is noted $\text{sign}(\mathbf{w})$, and $(\mathbf{w})_+$ is[(1)] the vector composed of scalars $\mathbf{w}_p$ if $\mathbf{w}_p \geq 0$ and 0 if $\mathbf{w}_p < 0$ (assuming $(w_p)_{p=1,\dots P}$ are the entries of the vector $\mathbf{w}$).
In the following, matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ represents the explanatory data where rows and columns respectively denote the N observations and P variables. Vector $\mathbf{y} \in \mathbb{R}^N$ denotes the response variable. Without loss of generality, we assume that $\mathbf{X}$ and $\mathbf{y}$ are mean-centered.

[(1)] It corresponds to the Rectified Linear Unit (ReLU), a popular activation function for neural networks.

5.2 Dual-sPLS in a nutshell

5.2.1 Statistical background

Regression models are effective when an observable numerical feature $\mathbf{y} \in \mathbb{R}^N$ is potentially related to a group of variables $\mathbf{X} \in \mathbb{R}^{N \times P}$. A linear dependence is expressed as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}\,, \tag{5.6}$$

where the noise $\boldsymbol{\epsilon}$ is expected to be independent of $\mathbf{X}$, with zero mean. When $P < N$ and $\mathbf{X}$ is of full rank, the classical ordinary least squares estimator is

$$\hat{\boldsymbol{\beta}}^{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}\,. \tag{5.7}$$

Otherwise, when $P \gg N$, the matrix $\mathbf{X}^T\mathbf{X}$ becomes singular and one must resort to alternative estimation techniques.

Partial Least squares (PLS)~ PLS is aims to estimate $\boldsymbol{\beta}$ from Equation (5.6) while avoiding singularity poblem of matrix $\mathbf{X}^T\mathbf{X}$. Its main idea is to compress predictor matrix $\mathbf{X}$ into a score matrix $\mathbf{T}$ encoding $M < \min(N, P)$ components while taking into account the covariance of $\mathbf{X}$ and $\mathbf{y}$ and nicely handling the correlation between the variables in $\mathbf{X}$. Therefore, PLS builds a more compact latent space spanned by a set of new components $\mathbf{t}_m$ for $m \in \{1, \ldots, M\}$ on which response $\mathbf{y}$ is projected. Each one is a linear combinations of original variables using weight vectors $\mathbf{w}_m$ for $m \in \{1, \ldots, M\}$ as $\mathbf{t}_m = \mathbf{X}\mathbf{w}_m$. Several algorithms have been proposed. NIPALS [nonlinear iterative partial least squares, 103] and SIMPLS [32] are most popular. When applied to a one-dimensional reponse, as in our case, both are shown to be equivalent [57]. They solve the PLS following (known as PLS1) optimization problem:

$$\max_{\mathbf{w}} (\mathbf{y}^T\mathbf{X}\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1\,. \tag{5.8}$$

For the first loading vector $\mathbf{w}_1$, solution of Equation (5.8) is

$$\mathbf{w}_1 = \mathbf{X}^T\mathbf{y}\,. \tag{5.9}$$

NIPALS iteratively computes weight vectors by deflation while SIMPLS is more straightforward by avoiding this step. Let $\mathcal{P}_{\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}}$ denotes the orthogonal projection onto the space spanned by components $\mathbf{t}_1, \ldots, \mathbf{t}_{m-1}$. The algorithm considers the part of $\mathbf{X}$ that is orthogonal to $\mathbf{t}_k$, $k < m$. For the m^{th} component, $\mathbf{X}$ is replaced by $\mathbf{X}_m$ such that:

$$\mathbf{X}_m = \mathbf{X} - \mathcal{P}_{\mathbf{t}_1,\ldots,\mathbf{t}_{m-1}}\mathbf{X} = \mathbf{X}_{m-1} - \mathcal{P}_{\mathbf{t}_{m-1}}\mathbf{X}_{m-1}\,. \tag{5.10}$$

After M iterations, $\mathbf{X}$ is summarized by $\mathbf{T} \in \mathbb{R}^{N \times M}$. Based on Proposition 1 from [57], the regression coefficients for M components are computed as:

$$\hat{\boldsymbol{\beta}}_M^{PLS} = \mathbf{W}(\mathbf{T}^T\mathbf{X}\mathbf{W})^{-1}\mathbf{T}^T\mathbf{y}\,. \tag{5.11}$$

The vector of regression fitted values $\hat{\mathbf{y}}$ for M components is the projection of response vector $\mathbf{y}$ onto the space spanned by columns of $\mathbf{T}$.

Least absolute shrinkage and selection operator (lasso) ~ The least square regression optimization problem for estimating $\boldsymbol{\beta}$ in (5.6) is stated as:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^P}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 . \tag{5.12}$$

Variable selection methods supplement (5.6) with a penalty function $pen(\boldsymbol{\beta})$. This is transcripted by solving:

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^P}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + pen(\boldsymbol{\beta}) . \tag{5.13}$$

Lasso [95] is popularly used owing it to its ℓ_1 regularization : $pen(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1$. The latter induces shrinkage of irrelevant variable coefficients to exactly zero, thus retaining a smaller number of features. The penalized Problem (5.13) is initially formulated in the lasso case as constraint Problem as the following:

$$\underset{\boldsymbol{\beta}}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{subject to } \|\boldsymbol{\beta}\|_1 \leq t . \tag{5.14}$$

Note that there is a one-to-one correspondence between parameters λ and t.

Lasso allows to explain possible relationship between $\mathbf{X}$ and $\mathbf{Y}$ with a neater model due to sparsity in its results. One challenge is the choice of appropriate threshold t. In fact, it controls the amount of sparsity; that is the number $\ell_0(\boldsymbol{\beta})$ of coefficients set to zero. A closed form solution of optimization problem (5.14) exists in the orthonormal design case, i.e. $\mathbf{X}^T\mathbf{X} = I_P$. It is known as the *soft thresholding* operator and verifies:

$$\hat{\beta}_p^{\mathrm{l}} = \mathrm{sign}(\hat{\beta}_p^{\mathrm{LS}})(|\hat{\beta}_p^{\mathrm{LS}}| - \lambda)_+ \quad \forall p \in \{1,\dots,P\} . \tag{5.15}$$

Coefficients magnitudes are compared to threshold λ and insignificant variables coefficients are set to zero.

Ridge regression ~ Ridge [49] regression is another popular penalization technique that uses an ℓ_2 penalty: $pen(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_2$. Its closed-form solution is stated as:

$$\hat{\boldsymbol{\beta}}^{\mathrm{r}} = (\mathbf{X}^T\mathbf{X} + \lambda I_P)^{-1}\mathbf{X}^T\mathbf{y} . \tag{5.16}$$

Unlike lasso, it shrinks irrelevant variable coefficients close to zero retaining the totality of features.

Group lasso ~ One may suppose that some variables in $\mathbf{X}$ present related patterns, for example close intensities or similar shape. Then, grouping them into G different subsets may give insights about each group effect on the response. Similarly, in some cases, $\mathbf{y}$ can be explained by several sets of independent variables. All are regrouped in the same matrix $\mathbf{X}$ and each is affected to a group $g \in \{1,\dots,G\}$, thus more information is gathered to predict he response. These cases can be managed with group lasso [110] method where instead of selecting individual variables

like in the classical lasso, groups of derived input variables are retained. The corresponding optimization problem is

$$\arg\min_{\boldsymbol{\beta}\in\mathbb{R}^P} \frac{1}{2}\|\mathbf{y} - \sum_{g=1}^{G} \mathbf{X}_g\boldsymbol{\beta}_G\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{P_g}\|\boldsymbol{\beta}_g\|_2 \,. \tag{5.17}$$

where $\mathbf{X}_g$ is the submatrix of $\mathbf{X}$ of variables in group g, $\boldsymbol{\beta}_g$ is the corresponding vector of coefficients for group g and P_g is the length of $\boldsymbol{\beta}_g$ for each $g \in \{1, \ldots, G\}$.

Similar to lasso and ridge, the magnitude of the tuning parameter controls the amount of sparsity. More details can be found in [110] and [85].

Sparse Partial Least Squares (sPLS) ~ Sparse Partial Least Squares (sPLS) combines both PLS and lasso by adding an ℓ_1 penalty to the PLS framework, i.e. Equation (5.8).

For $\lambda_s > 0$ and with an orthogonality constraint on components, the sPLS optimization problem is for the first one:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^p}\{-\mathbf{y}^T\mathbf{X}\mathbf{w} + \lambda_s\|\mathbf{w}\|_1\}, \qquad \text{for } \mathbf{w}^T\mathbf{w} = 1\,, \tag{5.18}$$

Several avatars of sPLS were introduced along the years. In 2008, Problem (5.18) was solved [61] using elements from sparse PCA [84]. We denote it as $\text{sPLS}_{\text{LeCao}}$. In 2010, it was reformulated by resorting to a surrugate for an approximated solution [25], $\text{sPLS}_{\text{Chun}}$. In 2018, proximal optimization [8] was used for another reformulation of Equation (5.18) [37], denoted $\text{sPLS}_{\text{Durif}}$.

5.2.2 Dual Sparse Partial Least Squares (Dual-sPLS)

Extending and generalizing sPLSs previous formulations, Dual-sPLS main objective is to achieve balance between accurate prediction compared to state-of-art methods and statistically interpretable localization of features. Additionally, it handles variable grouping: the possibility to gather explanatory variables into more meaningful subsets, and the combination of heterogeneous data related to the same response[(2)].

Dual-sPLS defines a family of approaches that differ by the choice of with the notion of dual norm defined in the following:

Definition 5.2.1 *Let $\Omega(\cdot)$ be a norm on $\mathbb{R}^P$. For any $\mathbf{z} \in \mathbb{R}^P$, the associated dual norm, denoted $\Omega^*(\cdot)$, is defined as*

$$\Omega^*(\mathbf{z}) = \max_{\mathbf{w}}(\mathbf{z}^T\mathbf{w}) \quad \text{s.t.} \quad \Omega(\mathbf{w}) = 1\,. \tag{5.19}$$

When $\Omega(.) = \|.\|_2$ and $\mathbf{z} = \mathbf{X}^T\mathbf{y}$, Equations (5.8) and (5.19) are equivalent, indeed optimizing the PLS function in Equation (5.8) amounts to finding the vector $\mathbf{w}$ that goes with the dual of the norm ℓ_2 applied to $\mathbf{z}$. Thus, Dual-sPLS proposes to evaluate different penalties depending on the use case.

[(2)] Applications of this extention are only performed in detail in this paper.

For any chosen norm penalty $\Omega(.)$, the first component will be:

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^p}{\arg\min}\{-\mathbf{z}^T\mathbf{w}\}, \quad \text{s.t. } \Omega(\mathbf{w}) = 1\,. \tag{5.20}$$

The following components are computed iteratively after applying a deflation step like in Equation (5.10) from the NIPALS algorithm.
Four norm penalties are considered with their corresponding function implemented in the **dual.spls** package.

1. **Dual-sPLS$_\mathbf{l}$** *(pseudo-lasso norm,* `d.spls.lasso()`*).* Similar to the sPLS Problem (5.18), an intuitive norm combines ℓ_1 to ℓ_2:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 + \|\mathbf{w}\|_2\,. \tag{5.21}$$

2. **Dual-sPLS$_\mathbf{gl}$** *(pseudo-group lasso norm,* `d.spls.GL()`*).* Inspired by group lasso [86], it combines groups of measurements. It applies pseudo-lasso to each group individually while constraining the total set. For G groups, $\mathbf{w}_g$ represents the variables of the loading vector $\mathbf{w}$ that belongs to group g. The corresponding norm is formulated as:

$$\Omega(\mathbf{w}) = \sum_{g=1}^{G} \alpha_g\|\mathbf{w}_g\|_2 + \lambda_g\|\mathbf{w}_g\|_1\,, \tag{5.22}$$

   where $\alpha_g \geq 0, \forall g \in \{1,\ldots,G\}$ and $\sum_{g\in 1,\ldots,G}\alpha_g = 1$.

3. **Dual-sPLS$_\mathbf{LS}$** *(pseudo-least squares norm,* `d.spls.LS()`*).* It introduces $\mathbf{N}_1$, a matrix of p columns, and applies when $\mathbf{X}$ is not singular:

$$\Omega(\mathbf{w}) = \lambda\|\mathbf{N_1}\mathbf{w}\|_1 + \|\mathbf{X}\mathbf{w}\|_2\,. \tag{5.23}$$

   The classical least squares solution is recovered for $\lambda = 0$.

4. **Dual-sPLS$_\mathbf{r}$** *(pseudo-ridge norm,* `d.spls.ridge()`*).* It deals with cases where $\mathbf{X}$ is singular and resorts to a ridge-like penalization:

$$\Omega(\mathbf{w}) = \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{X}\mathbf{w}\|_2 + \|\mathbf{w}\|_2\,. \tag{5.24}$$

The resolution procedure for each are found in [3]. For the four penalty norms, $\mathbf{w}$ is colinear to $g_\nu(\mathbf{z})$, where the latter is the soft thresholding operator:

$$g_\nu(\mathbf{z}) = \text{sign}(\mathbf{z})(|\mathbf{z}| - \nu)_+\,. \tag{5.25}$$

Thus parameter ν controls the shrinking ratio and Dual-sPLS provides an adaptive way of selecting the appropriate parameter. In a few words, instead of choosing ν, users specify the expected null confficients and ν is computed accordingly. The approach is also detailed in [3].

5.3 Package dual.spls description

5.3.1 Package overview

The **dual.spls** package is self-contained providing various functions related to the Dual-sPLS regression method. The package includes a range of functions that enable the user to simulate, fit, evaluate, and visualize data. We list each function below:

- d.spls.lasso, d.spls.GL, d.spls.LS, d.spls.ridge and d.spls.pls represent the most useful features of the package by implementing each version of Dual-sPLS introduced in Section 5.2.2 and the classical PLS1 algorithm. These regression functions allow the user to select the most appropriate model for their data and to reduce the dimensionality of the data in a way that best suits their needs.

- d.spls.cv is a cross-validation variant function proposed by [15]. It is particularly useful for selecting the appropriate number of components to use in the regression analysis.

- d.spls.print displays the value of the shrinkage parameter and the number ℓ_0^c of variables selected by the model for any specified component.

- d.spls.predict is a function typically found in a regression package for predicting new responses based on the fitted model.

- d.spls.plot is used to visualize the coefficients of the model. It allows evaluating the latter interpretability by stacking the coefficients plot below the original data mean. It also quantify the model sparsity by specifying the number of zeros estimated.

- d.spls.metric helps in evaluating the accuracy of the predictions made by the model. It computes the RMSE, MAE, and R-squared values introduced in Section 5.1. These metrics are commonly used to assess the predictive performance of a model and can help the user to identify areas where the model may need improvement.

- d.spls.NIR provides a real data set that can be analyzed using the Dual-sPLS algorithm and provides an opportunity for users to experiment with the package and gain practical experience or also use it for other kind of studies.

- d.spls.simulate generates simulated data using mixture of Gaussians. It allows the user to generate data sets with sparse linearity dependence to test the performance of the Dual-sPLS algorithm under different conditions.

- d.spls.calval is used to split the data into a calibration and a validation set using CalValXy algorithm [2].

Main arguments and values for fitting a Dual-sPLS regression are represented respectively in Tables 5.31 and 5.32.

Arguments	Desription
X	predictor matrix or data frame
y	response vector
ncp	number of components
ppnu	proportion of variables to shrink to zero
verbose	wether to display or not the iteration number

Table 5.31 ~ *Main* **dual.spls** *arguments.*

Values	Desription
Xmean	predictors $\mathbf{X}$ mean
scores	scores matrix $\mathbf{T}$
loadings	loadings matrix $\mathbf{W}$
Bhat	regression coefficients $\hat{\boldsymbol{\beta}}$
intercept	intercepts vector β_0
fitted.values	fitted values matrix $\hat{\mathbf{y}}$
residuals	residuals matrix $\hat{\boldsymbol{\epsilon}}$
listelambda	tuning parameter vector of λ
zerovar	number of variables shrinked to zero ℓ_0

Table 5.32 ~ *Main* **dual.spls** *arguments.*

5.3.2 Simulated and Real NIR data

Real data ~ The provided real data set is near-infrared (NIR) spectra of hydrocarbon samples. It is provided by *IFPEN*[(3)] and was partially published in [38]. NIR spectroscopic data are often used in Chemometrics quantifying the absorption of infrared radiation which depends on the chemical bonds of organic matter. Limits of corresponding wavelengths range are from 800 nm to 2500 nm. Response property $\mathbf{y}$ is the density obtained with standardized methods. Corresponding covariate matrix $\mathbf{X}$ consist of 208 samples (rows) and 1557 variables (columns). A simple discrete derivative for each variable, denoted D_{NIR} is considered as a pre-processing using a Savitzky Golay smoothing [83]. Such data are generally represented as functional data, i.e. the absorbance derivative depending on the wavelenght, see Figure 5.31. Loading the dataset from the package is described below:

```
R> # Data loading
R> data(d.spls.NIR)
R> summary(d.spls.NIR)

             Length       Class        Mode
NIR          1557      data.frame      list
density      208         -none-       numeric
```

[(3)] a French research institute studying hydrocarbon-based and renewable energies

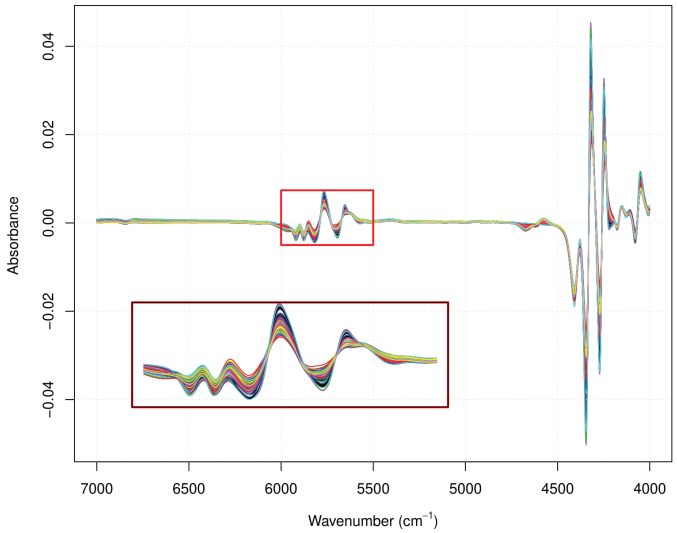

Figure 5.31 *~ D_{NIR}: first derivative of the NIR spectra of 208 samples.*

```
R> XNIR <- d.spls.NIR$NIR
R> XNIR <- as.matrix(XNIR)
R> yNIR <- d.spls.NIR$density
R> nNIR <- dim(XNIR)[1]  #number of observations
R> pNIR <- dim(XNIR)[2]  #number of variables
```

Note that we normalize the response variable between 0 and 1 as this can make model comparisons easier and help to ensure that the scale of the response variable does not unduly influence the model estimation. This process will be applied to every data set used in this paper, but since the coding procedure is identical, it will not be repeatedly illustrated in each segment. We provide thus the following code for the normalization of $\mathbf{y}$:

```
R> yNIR <- (yNIR-min(yNIR))/(max(yNIR)-min(yNIR)) #normalizing response
R>                                                #between 0 and 1
```

Simulated data ~ **dual.spls** provides a function d.spls.simulate that generates predictor matrix $\mathbf{X}$ and a linearly dependent response $\mathbf{y}$. $\mathbf{X}$ is a mixture K Gaussians with presetted scale σ and randomly picked amplitudes A_{ik} and location μ_k, for $i \in \{1, \ldots, N\}$ and $k \in \{1, \ldots, K\}$. Each row of $\mathbf{X}$ is generated as follows:

$$\mathbf{x}_i = \sum_{k=1}^{K} A_{ik} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma^2}\right). \tag{5.26}$$

They are uniformly sampled into P variables such that $\mathbf{X} \in \mathbb{R}^{N \times P}$. For $\mathbf{y}$ computation, first, as we consider linear regression context, we simulate the response as a linear combination of matrix $\mathbf{X}$ variables with a preset uncertainty $\boldsymbol{\epsilon}$. The linear coefficients depend on weights fixed by parameter `int.coef`. The latter can specified for ranges of variables. Second, to evaluate interpretabilty, we link it to sparsity by generating sparse models where only $S \ll P$ positive weights are imposed. Additionally, d.spls.simulate also allows simulating several predictors matrices explaining the same response $\mathbf{y}$.

Since Dual-sPLS$_{\text{LS}}$ requires invertible matrices, non-singular data $\overline{D}_{\text{SIM}}$ is simulated and represented in Figure 5.32. It includes an explanatory matrix $\mathbf{X}$ where $P < N$ and a response linearly and sparsely related to $\mathbf{X}$. The red highlighted bands from Figure 5.32 indicate positively weighted variables i.e. influential variables locations. The following chunks of code show the construction of $\overline{D}_{\text{SIM}}$.

```
R> # Parameters
R> set.seed(17)
R> nDS1 <- 200  # number of observations
R> pDS1 <- 50   # number of variables
R> K <- 100  # number of Gaussians
R> sigma <- 0.01   #Gaussians scale
R># Data simulation
R> DS1 <- d.spls.simulate(nDS1,pDS1,nondes=K,sigmaondes=sigma,sigmay=0.5,
R>        int.coef=c(100,0,0,0,200))
R> XDS1 <- DS1$X
R> yDS1 <- DS1$y
%R> yDS1 <- (yDS1-min(yDS1))/(max(yDS1)-min(yDS1)) #normalizing response
%R>                                                #between 0 and 1
```

Since Dual-sPLS$_{\text{GL}}$ allows dealing with heterogeneous data that explains the same property, another data set was simulated denoted D^2_{SIM}. It is composed of two explanatory matrices $\mathbf{X}_1$ and $\mathbf{X}_2$. Their combination is represented in Figure 5.33 where the dotted vertical line reflects the limit between the two sets. D^2_{SIM} also inlcudes a response $\mathbf{y}$ linearly linked to both matrices simultaneously. In fact, the construction of $\mathbf{y}$ depends on variables in colored bands in Figure 5.33. The red ones represent influential variables from the first data set $\mathbf{X}_1$ and the blue ones from the second data set $\mathbf{X}_2$. Following chunks of codes illustrate the simulation of D^2_{SIM}. D^2_{SIM} is also simulated and represented in Figure 5.33.

```
R> # Parameters
R> set.seed(1)
R> nDS2 <- 300       # number of observations
R> p1<- 5000         # number of variables of first explanatory matrix X1
R> p2 <- 2000        # number of variables of second explanatory matrix X2
R> pDS2 <- p1+p2     # number of variables of combination of X1 and X2
R> K <- c(10,4) # number of Gaussians for each X1 and X2
R> sigma <- c(0.03,0.2)    # standard deviation of X1 and X1 Gaussians
```

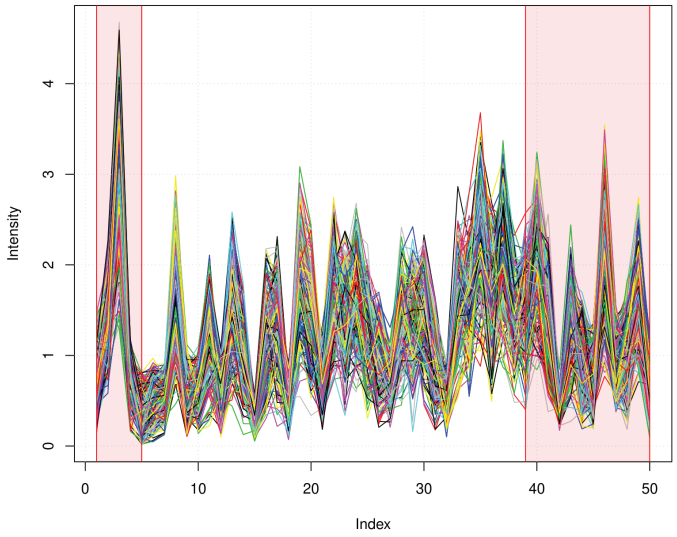

Figure 5.32 ~ *Simulated data* $\overline{D}_{\text{SIM}}$.

```
R> int.coef1 <- c(0,500,0,200,200, # depending ranges of variables in X1
R>                0,50,500,0)      # (highlighted in Figure 3 in red)
R> int.coef2 <- c(0,100,0,0,100) # depending ranges of variables in X2
R>                               # (highlighted in Figure 3 in blue)
R> # Data simulation

R> DS2 <- d.spls.simulate(nDS2,p=c(p1,p2),nondes=K,sigmaondes=sigma,
 R>         sigmay=0.5,int.coef=c(int.coef1,int.coef2))
R> XDS2 <- DS2$X
R> yDS2 <- DS2$y
%R> yDS2 <- (yDS2-min(yDS2))/(max(yDS2)-min(yDS2)) #normalizing response
%R>                                                #between 0 and 1
```

The package and data sets will be loaded and generated as described previously throughout the remainder of this section. To showcase the efficiency of Dual-sPLS, Dual-sPLS$_{\text{l}}$ and Dual-sPLS$_{\text{r}}$ will be applied on D_{NIR}, Dual-sPLS$_{\text{LS}}$ on $\overline{D}_{\text{SIM}}$ and Dual-sPLS$_{\text{gl}}$ on D^2_{SIM}.

5.3.3 Calibration and validation splitting

To enhance model's prediction performance, each data is first divided into calibration and validation sets. The regression model is built using calibration data set, and its performance is assessed using validation data set. The package **dual.spls** proposes a splitting technique called CalValXy [2] implemented in function d.spls.calval. The latter selects observations to build the calibration set to best represent the overall data in terms of its variability and range of values

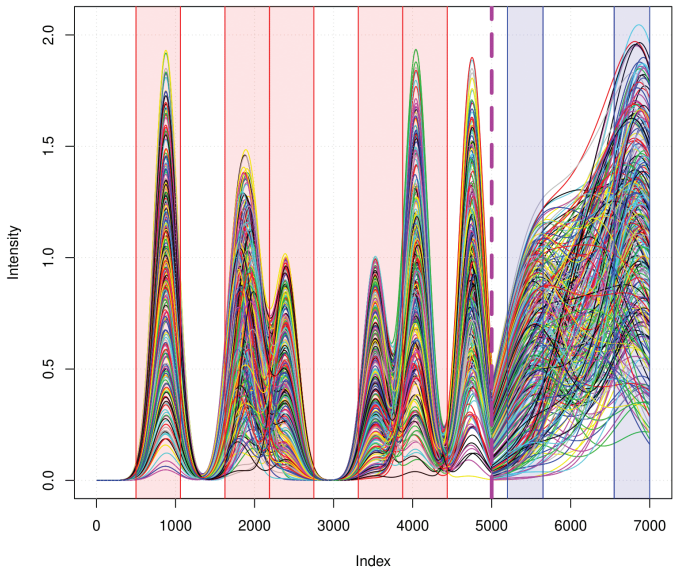

Figure 5.33 ~ *Simulated data* D^2_{SIM}.

while taking into account information provided by both $\mathbf{X}$ and $\mathbf{y}$. It does so by partitioning $\mathbf{y}$ into `ncells` subsamples (function input parameter) and applying the [54] algorithm to each subgroup. When the number of variables is high, data dimension can be reduced using PCA; this is established by selecting the appropriate `method` input parameter. Since CalValXy is based on computation of distances between observations, data is first centered. As the splitting procedure is similar to each data of Section 5.3.2, the following will only provide the application on $\overline{D}_{\text{SIM}}$. It is also illustrated in Figure 5.34. The code below shows how the procedure can be done using **dual.spls** functions.

```
R> # Centering Data
R> XDS1m <- apply(XDS1,2,mean)                # mean of each column of XDS1
R> XDS1c <- XDS1-rep(1,nDS1) %*% t(XDS1m)    # centering of XDS1 by column
R> # Splitting
R> cvDS1 <- d.spls.calval(XDS1c,pcal=80,y=yDS1, # index of calibration and
R>          ncells=10,method="euclidien")       # validation split of  DS1
R> indcalDS1 <- cvDS1$indcal      # calibration index of DS1
R> indvalDS1 <- cvDS1$indval      # validation index of DS1
R> ncalDS1 <- length(indcalDS1)   # number of calibration observations of DS1
R> nvalDS1 <- length(indvalDS1)   # number of validation observations of DS1
R> XcalDS1 <- XDS1[indcalDS1,]        # calibration covariates
R> XvalDS1 <- XDS1[indvalDS1,]        # validation covariates
R> ycalDS1 <- yDS1[indcalDS1]         # calibration response
```

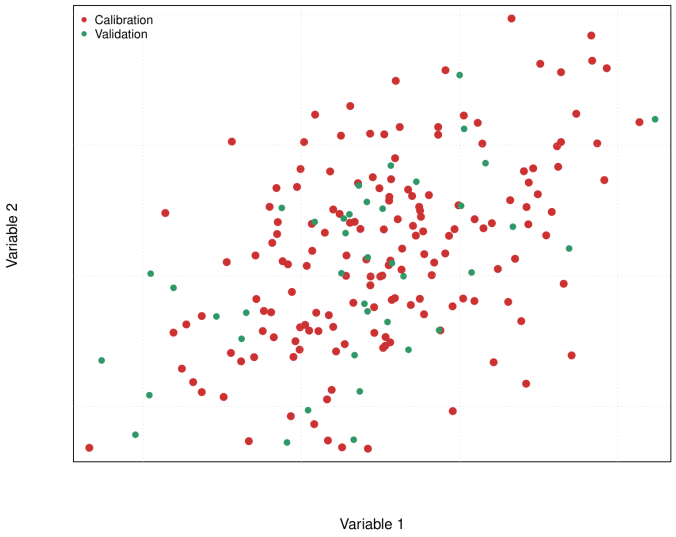

Figure 5.34 ~ *Calibration and validation splitting of* $\overline{D}_{\mathrm{SIM}}$

```
R> yvalDS1 <- yDS1[indvalDS1]          # validation response
```

5.3.4 PLS and Dual-sPLS fitting

The difference between a PLS and Dual-sPLS fitting is the input parameter `ppnu` that controls the sparsity of the model. Before fitting any regression model, input data matrix $\mathbf{X}$ is centered, as shown below, in order to enhances the numerical stability.

```
R> #Centering Data after splitting
R> # calibration
R> XcalcDS1 <- scale(XcalDS1 - rep(1, ncalDS1) %*% t(apply(XcalDS1, 2, mean))
R> ycalcDS1<- ycalDS1-mean(ycalDS1)
R> # validation
R> XvalcDS1 <- XvalDS1 - rep(1, ncalDS1) %*% t(apply(XcalDS1, 2, mean))
R> yvalcDS1<- yvalDS1-mean(ycalDS1)
```

The typical way of fitting a PLS and Dual-sPLS model is detailed in the following. We will apply the PLS1, Dual-sPLS$_{\mathrm{l}}$ and Dual-sPLS$_{\mathrm{r}}$ algorithms on real data D_{NIR}. Then, Dual-sPLS$_{\mathrm{gl}}$ and Dual-sPLS$_{\mathrm{LS}}$ will use respectively simulated data D^2_{SIM} and $\overline{D}_{\mathrm{SIM}}$. For each application, ten components are use and the proportion of sparsity is set to 0.990 for all the Dual-sPLSs except to Dual-sPLS$_{\mathrm{LS}}$ where we choose to shrink to zero only 60 % of the variables.

```
R> # PLS fitting on DNIR
R> mod.dspls.pls <- d.spls.pls(XcalcNIR,ycalcNIR,ncp=10,verbose = FALSE)
```

```
R>
R> # Dual-sPLS (lasso) fitting on DNIR
R> mod.dspls.l <- d.spls.lasso(XcalcNIR,ycalcNIR,ncp=10,
R>                             ppnu=0.99,verbose = TRUE)

Dual PLS ic= 1 lambda= 1.538579 mu= 0.1079445 nu= 0.1660811 nbzeros= 1541
Dual PLS ic= 2 lambda= 1.29071 mu= 0.00761122 nu= 0.009823877 nbzeros= 1526
Dual PLS ic= 3 lambda= 0.8785089 mu= 0.00146387 nu= 0.001286023 nbzeros= 1510
Dual PLS ic= 4 lambda= 0.4928369 mu= 0.003739323 nu= 0.001842877 nbzeros= 1501
Dual PLS ic= 5 lambda= 2.869796 mu= 0.0002254818 nu= 0.0006470867 nbzeros= 1489
Dual PLS ic= 6 lambda= 2.672436 mu= 0.000107955 nu= 0.000288503 nbzeros= 1475
Dual PLS ic= 7 lambda= 2.237073 mu= 4.334281e-05 nu= 9.696102e-05 nbzeros= 1461
Dual PLS ic= 8 lambda= 0.764029 mu= 0.0001691464 nu= 0.0001292327 nbzeros= 1451
Dual PLS ic= 9 lambda= 0.7945265 mu= 0.0001321681 nu= 0.0001050111 nbzeros= 1442
Dual PLS ic= 10 lambda= 0.696282 mu= 0.0001289014 nu= 8.975171e-05 nbzeros= 1434

R> # Dual-sPLS (group lasso) fitting on DS2
R> mod.dspls.GL <- d.spls.GL(XcalcDS2, ycalcDS2, ncp=10,ppnu=c(0.99,0.99),
R>                           indG = c(rep(1,p1),rep(2,p2)), verbose = FALSE)
R>
R> # Dual-sPLS (LS) fitting on DS1
R> mod.dspls.ls <- d.spls.LS(XcalcDS2,ycalcDS2,ncp=10,ppnu=0.6,verbose = FALSE)
R>
R># Dual-sPLS (ridge) fitting on DNIR
R> mod.dspls.r <- d.spls.ridge(XcalcNIR,ycalcNIR,ncp=10,ppnu=0.99,
                              nu2=2,verbose =FALSE)
```

As noticed in the output of the code above, when `verbose=TRUE`, every line displays the iteration number, `ic`, along with the parameter values of `lambda`, `mu`, and `nu`, as well as the number of zeros ℓ_0 in the regression coefficients.

5.3.5 Results visualization

The code above fits five models with 1 to 10 components. We can get an overview of each via d.spls.print and d.spls.plot. They both require which num- ber of components to use when displaying the results. They respectively print the values of the hyper-parameters along with the number of variables selected and plot the regression coefficients versus the mean of the original data. In particular, this helps in the visualization of the variable selection performed by Dual-sPLS models. We present an example for model Dual-sPLS$_{\text{LS}}$ on $\overline{D}_{\text{SIM}}$ for six components in the following and in Figure 5.35.

```
R> d.spls.print(mod.dspls.ls,6)

Dual Sparse Partial Least Squares Regression for the  LS  norm
```

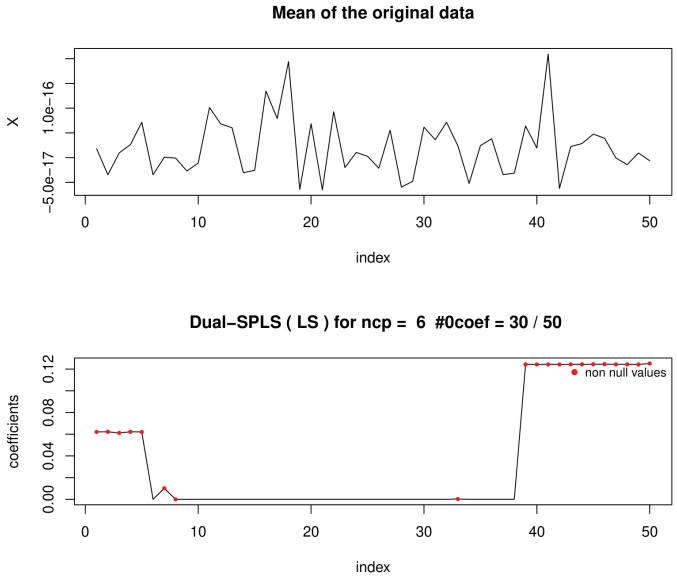

Figure 5.35 ~ *Dual-sPLS$_{LS}$ Regression coefficients for six components versus mean of centered original data* $\overline{D}_{\text{SIM}}$

```
-------------------------------------------

Parameters: lambda =  8.47570848970008e-13 , ncomp =  6
Dual-SPLS selected:  20  variables among  50  variables

R > d.spls.plot(mod.dspls.ls,6)
```

5.3.6 Prediction of validation set

Fitted models are frequently employed to forecast upcoming observations. **dual.spls** provides a d.spls.predict function for prediction and requires the fitted model to be used, the new batch of data, and the list of components chosen to be employed. As the data was initially split into calibration and validation, the latter is considered as new data for which corresponding response must be predicted. To assess this matter, **dual.spls** also includes d.spls.metric to compute RMSE, MAE and R^2 criteria. We also use $\overline{D}_{\text{SIM}}$ in this example.

```
R> # predictions on validation
R> yvalhat.ls=d.spls.predict(mod.dspls.ls,XvalcDS1,1:10)
R> d.spls.metric(yvalhat.ls,yvalcDS1)
$RMSE
 [1] 0.0003723408 0.0003588578 0.0003514178 0.0003531040 0.0003454870
 [6] 0.0003448199 0.0003480288 0.0003496940 0.0003694236 0.0003729228
```

```
$MAE
 [1] 0.0002818395 0.0002715141 0.0002683661 0.0002717894 0.0002674741
 [6] 0.0002673378 0.0002774522 0.0002780488 0.0003056253 0.0003121005

$Rsquared
 [1] 0.9999862 0.9999872 0.9999877 0.9999876 0.9999882
 [6]0.9999882 0.9999880 0.9999879 0.9999865 0.9999862
```

5.3.7 Choosing the number of latent components

The optimum number M of latent components needs to be determined while building a regression model. It avoids the risk of overfitting by striking a compromise between forecast accuracy and model complexity. **dual.spls** proposes a variant of popular cross validation procedure [15, multiple random split]. Function d.spls.cv implements it and requests the specification of maximum candidate number `ncomp` (input parameter) of Dual-SPLS components, the Dual-sPLS norm flavor with its associated parameters, the number (`nrepcv`) of repetitive iterations to be performed and calibration ratio (`pctcv`). Associated algorithm uses the following procedure:

1. Multiple random partitions of the observations are generated into `nrepcv` calibration validation set with (`pctcv`) ratio.

2. Different numbers (`ncomp`) of latent components are used while building potential models with the calibration sets.

3. On each validation set, MSE is computed and average.

4. The smallest number of `ncomp` that has the lowest averaged MSE discloses enough latent components.

An example of choosing the appropriate number of latent component from 1 to 10 for a Dual-sPLS$_{\mathrm{LS}}$ application on $\overline{D}_{\mathrm{SIM}}$ is detailed next:

```
R> d.spls.cv(XcalDS1,ycalDS1,ncomp=10,dspls="LS",ppnu=0.6,nrepcv=20,pctcv=75)

                        .
                        .
                        .
Dual PLS LS, ic= 6 nu= 0.1093369 nbzeros= 30
Dual PLS LS, ic= 7 nu= 8.947531e-05 nbzeros= 20
Dual PLS LS, ic= 8 nu= 5.201551e-05 nbzeros= 13
Dual PLS LS, ic= 9 nu= 2.241844e-05 nbzeros= 6
Dual PLS LS, ic= 10 nu= 1.051683e-05 nbzeros= 4
[1] 7
```

5.4 Results of Dual-sPLS pseudo group lasso applied on simulated data

This Section provides evaluation of Dual-sPLS$_{gl}$ regression. The latter is assessed in terms of accuracy of prediction and interpretation quality. First, data D^2_{SIM} is considered and divided into calibration and validation sets with a 80-20 ratio via CalValXy. Second it is centered by removing calibration variables average. Third, Dual-sPLS$_{gl}$ regression is applied and compared to its counterpart (group lasso) and classical PLS. Hyper-parameter of group lasso is selected by cross validation. For Dual-sPLS$_{gl}$, we choose to impose 99 % of the variables to be zero in each group separately.
Predictions performance is assessed by computing common objective measures: root mean squared error (RMSE), mean absolute error (MAE), or determination coefficient (R^2). As metrics produce comparable results, we simply show the RMSE values for calibration and validation. Since D^2_{SIM} is generated in a way that the location of influential variables linked to the response is known, interpretation quality is examined by vertically stacking the regression coefficients for each of the three examined algorithms. Following chunks of code detail the evaluation procedure with supporting interesting plots. They complement the D^2_{SIM} data simulation from Section 5.3.2.

```
R> # Centering Data
R> XDS2m <- apply(XDS2,2,mean)              # mean of each column of XDS2
R> XDS2c <- XDS2-rep(1,nDS2) %*% t(XDS2m)   # centering of XDS2 by column
########################
R> # Splitting
R> cvDS2 <- d.spls.calval(XDS2c,pcal=80,y=yDS2, # index of calibration and
            ncells=10,method="euclidien")       # validation split of  DS2
R> indcalDS2 <- cvDS2$indcal       # calibration index of DS2
R> indvalDS2 <- cvDS2$indval       # validation index of DS2
R> ncalDS2 <- length(indcalDS2)    # number of calibration observations of DS2
R> nvalDS2 <- length(indvalDS2)    # number of validation observations of DS2
R> XcalDS2 <- XDS2[indcalDS2,]        # calibration covariates
R> XvalDS2 <- XDS2[indvalDS2,]        # validation covariates
R> ycalDS2 <- yDS2[indcalDS2]         # calibration response
R> yvalDS2 <- yDS2[indvalDS2]         # validation response
########################
R> #Centering Data after splitting
R> # calibration
R> XcalcDS2 <- XcalDS2 - rep(1, ncalDS2) %*% t(apply(XcalDS2, 2, mean))
R> ycalcDS2<- ycalDS2-mean(ycalDS2)
R> # validation
R> XvalcDS2 <- XvalDS2 - rep(1, ncalDS2) %*% t(apply(XcalDS2, 2, mean))
R> yvalcDS2<- yvalDS2-mean(ycalDS2)
########################
R> # Application of Dual-sPLS group lasso
```

```
R> v.group=c(rep(1,p1),rep(2,p2))
R> mod.dspls.GL <- d.spls.GL(XcalcDS2, ycalcDS2, ncp=10,ppnu=c(0.99,0.99),
                             indG = v.group, verbose = F)
R> #calibration
R> ycalhat.dspls.GL <- mod.dspls.GL$fitted.values  # fitted values
R> metric.cal.dspls.GL <- d.spls.metric(ycalhat.dspls.GL,
R>                                      ycalc) # calibration metrics
R> #validation
R> yvalhat.dspls.GL <- d.spls.predict(mod.dspls.GL,XvalcDS2,
R>                                    liste_cp=1:ncp) # predictions on validation
R> metric.val.dspls.GL <- d.spls.metric(yvalhat.dspls.GL,
R>                                      yvalcDS2) # validation metrics
R> #coefficients
R> betahat.dspls.GL <- mod.dspls.GL$Bhat
########################
R> # Application of d.spls.pls (PLS)
R> mod.dspls.pls <- d.spls.pls(XcalcDS2,ycalcDS2,ncp=10,verbose = FALSE)
#calibration
R> ycalhat.pls <- mod.dspls.pls$fitted.values  # fitted values
R> metric.cal.pls <- d.spls.metric(ycalhat.pls,ycalc) # calibration metrics
R> #validation
R> yvalhat.pls <- d.spls.predict(mod.dspls.pls,XvalcDS2,
                                 liste_cp=1:ncp) # predictions on validation
R> metric.val.pls <- d.spls.metric(yvalhat.pls,yvalc) # validation metrics
R> #coefficients
R> betahat.dspls.pls=mod.dspls.pls$Bhat
########################
R> # Application of SGL (group lasso)
R> library(SGL)
R> set.seed(1)
R> data <-  list(x = XcalcDS2, y = ycalcDS2)
R> cvFit <-  cvSGL(data, index=v.group, type = "linear",alpha=0)
R> mod.GL<- SGL(data, index=v.group, type = "linear",alpha=0,verbose=T,
R>              lambdas = cvFit$lambdas)
R> lambda.SGL <- cvFit$lambdas[which.min(cvFit$lldiff)]
R> lambda.index <- which(mod.GL$lambdas==lambda.SGL)
#calibration
R> ycalhat.GL <- predict(mod.GL,as.matrix(XcalcDS2),lambda.index)  # fitted values
R> metric.cal.GL <- d.spls.metric(ycalhat.GL,ycalcDS2) # calibration metrics
#validation
R> ycalhat.GL <- predict(mod.GL,as.matrix(XvalcDS2),
                         lambda.index) # predictions on validation
R> metric.val.GL <-  d.spls.metric(yvalhat.GL,yvalcDS2) # validation metrics
#coefficients
```

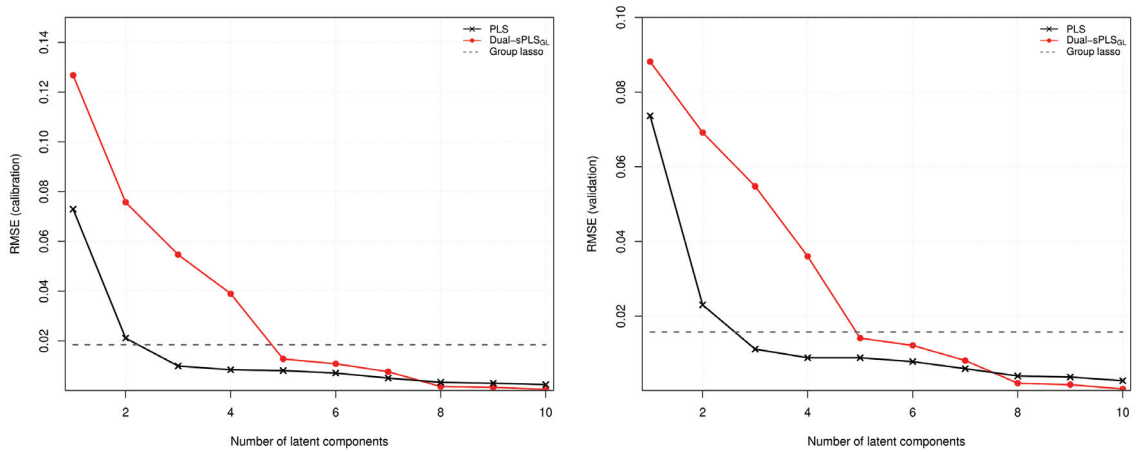

Figure 5.41 ~ *Dual-sPLS$_{GL}$ forecast evaluation on simulated data D^2_{SIM}, benchmarked against classical group lasso and PLS. RMSE values for calibration (left) and validation (right) with respect to the number of latent components.*

```
R> betahat.GL <- mod.GL$beta
```

Figure 5.41 illustrates RMSE outcomes for calibration (left) and validation (right). Both curves profiles look similar. As the number of components rises, Dual-sPLS$_{\text{GL}}$ and PLS error values significantly decrease and come close to overlap. The dotted black curve reflects the Group lasso prediction performance as it is independent of the number of components. Starting five components, RMSE Dual-sPLS$_{\text{GL}}$ and PLS curves outpass Group lasso line and decreases slope while remaining positive. A plateau is formed from component eight onwards where Dual-sPLS$_{\text{GL}}$ outperforms PLS. With this satisfactory result, localization is thus assessed at this point in Figure 5.42. In the later, four panels are stacked: original data (1) and regression coefficients from PLS (2) and Dual-sPLS$_{\text{GL}}$ (3) for eight components and Group lasso (4). Dotted purple vertical line distribute the data into the two original groups: G_1 (left) and G_2 (right). Highlighted areas (in red and blue) represent relevant variable real locations (see details in Section 5.3.2). A good interpretation is one that can provide the latter with precision. PLS coefficients (panel 1) mimics the shape of original data with relatively high amplitude curves in highlited areas from each group. However, it fails to localize the most important variables, unlike Dual-sPLS$_{\text{GL}}$. In panel (2), the latter selects a small number of variables $\ell_0(\mathbf{w})$ in each group and form sharp peaks matching emphasized ranges in red and blue. Classical group lasso provides coefficients in panel (3) that are less continuous and smooth. Additionally, they have high amplitude and emphasize unimportant features, indicating that they give too much importance to variables that do not contribute significantly to the outcome $\mathbf{y}$. In contrast, Dual-sPLS$_{\text{GL}}$ appears to be more reliable in identifying relevant variables in a regression context.

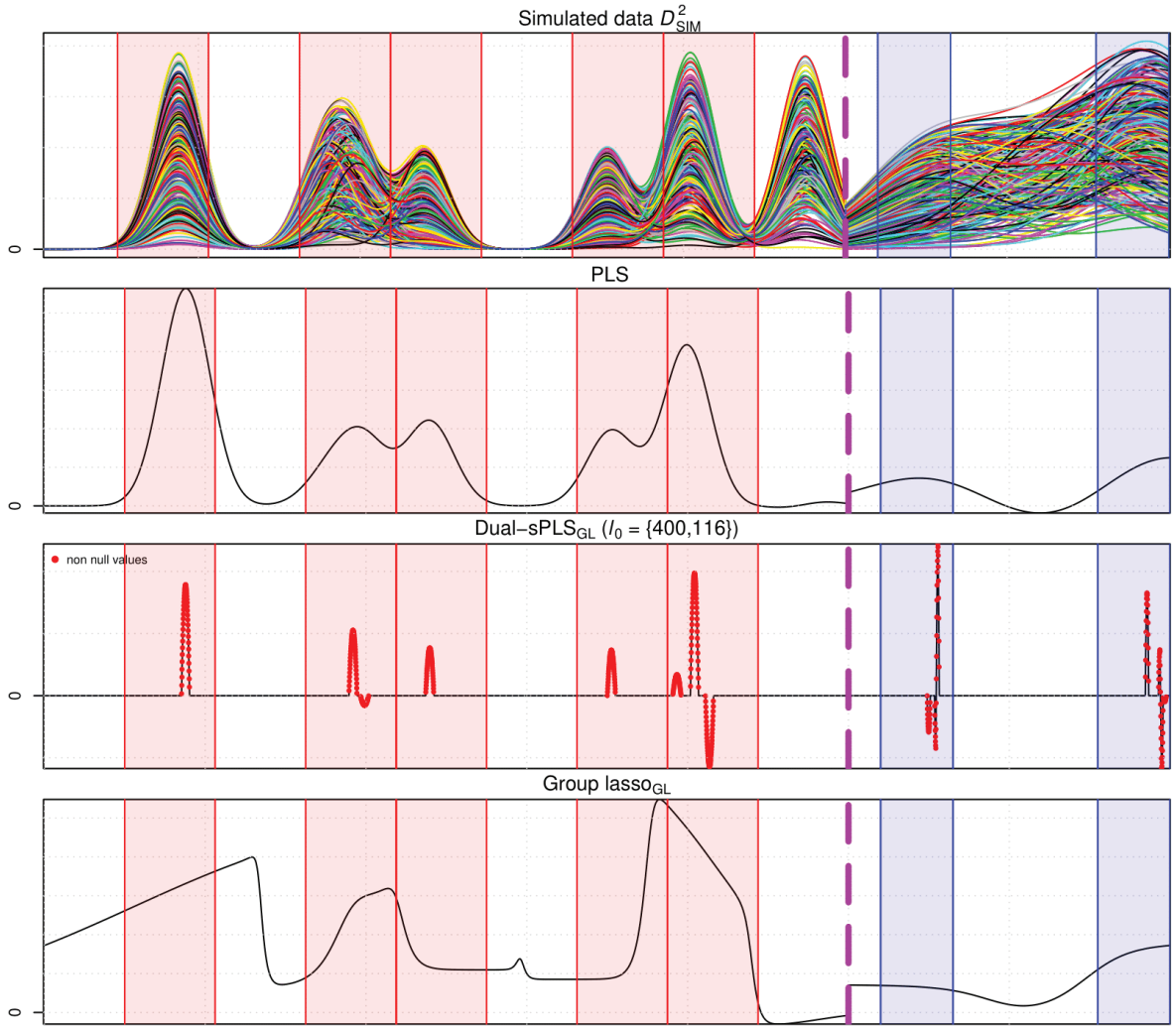

Figure 5.42 ~ *Dual-sPLS$_{GL}$ interpretation performance evaluation on simulated data D^2_{SIM}, benchmarked against classical group lasso and PLS. From top to bottom: D^2_{SIM} data compared to regression coefficients of PLS and Dual-sPLS$_{GL}$ for eight components and classical Group lasso.*

5.5 Conclusion

This paper describes the **dual.spls** R package. Dimensionality issues are becoming increasingly common, thus the purpose of this package is to make Dual-sPLS modeling available to the scientific community and so to enable its application to real-world issues. Package **dual.spls** also additional functions to make it complete. It contains both real and simulated data, splitting method and evaluation tools.
This paper also illustrate an detailed example of Dual-sPLS for the pseudo-group lasso. The latter covered the situation where two sets of independent variables are related to the same response. As illustrated by our example, Dual-sPLS$_{gl}$ successfully locates most relevant features by shrinking to zero unimportant variable coefficients. With sparsity acquired, prediction is also enhanced compared to PLS and group lasso. Indeed, Dual-sPLS$_{gl}$ outperforms both classical methods by striking a balance between accurate predictions and good interpretation while benefiting from information coming from both sets of explanatory variables.
So far, **dual.spls** includes just four types of penalties. However, other regularization procedure can be developed in the Dual-sPLS context. Accordingly, the package will be in continuous updating.

6 Conclusions and perspectives

Conclusions

Chemical analysis remains an important subject that is widely studied across various fields of science and industry. The study of chemical products and their properties is essential for understanding their behavior and interactions in various environments. It is includes in particular the characterization of petroleum cuts, an ongoing research subject, which is the main application of this thesis. Knowing the unique set of properties of these complex mixtures is essential for quality control, optimizing refining process to produce high-quality products efficiently, and safety and environmental concerns. While standardized physicochemical properties (like density, viscosity, flash point, pour point, etc.) allows classifying and comparing different types of mixtures, they require specialized equipment and skilled personnel, as well as extensive time and resources. Thus, rapid analysis (like infrared spectroscopy, X-ray fluorescence, and near-infrared spectroscopy) is often used instead. These methods offer a cost-effective way to rapidly assess the properties of petroleum products. However, they have relatively lower accuracy and the results may not be as comprehensive as those obtained from more standardized methods. Chemometric methods are thus used and provide a powerful and efficient way to process large datasets and extract meaningful information. For oil characterization multivariate calibration is often applied as it allows to use rapid analysis to predict physicochemical properties by conceiving regression models linking them together. Hence, this thesis addressed the need previously outlined. The focus of this work is to conceive a regression method that create models that balance between accurate forecasts of properties and good interpretability of rapid analysis spectra.

IFPEN supplied real data to support this study. Each incorporates a standardized physicochemical properties of a variety of petroleum cuts as well as one or more related rapid analysis spectrum. The latter are characterized by their functional aspect. They refer to a collection of measurements taken over a time-varying or continuous domain, rather than at fixed points in time or space. They are often inherently complex and high-dimensional. As a result, functional data sets can be challenging to analyze and visualize, requiring specialized tools and techniques for data processing and interpretation. We focus our study on one main real data denoted D_{NIR} which includes information on the density, represented by response vector $\mathbf{y}$ of $N = 208$ petroleum cuts and their pre-processed near-infrared spectra discretized in $P = 1557$ variables, regrouped in a matrix $\mathbf{X} \in \mathbb{R}^{N \times P}$ to build a predictive model. The near-infrared spectra provide insight into the molecule's atom bonds, which vary by length and strength, and are measured by how much near-infrared radiation a matter absorbs. Additionally, we composed a data generator that reproduce comparable sets to D_{NIR} and similar data. Simulated data plays a crucial role in data science projects because it allows data scientists to conduct experiments and test hypotheses without having to make decisions based on the limited amount of available real-world data. Thus we generated three data D_{SIM}, $\overline{D}_{\text{SIM}}$ and D^2_{SIM} compatible with the thesis objectives and adapted to different scenarios. Functional spectra represented by explanatory matrix $\mathbf{X}$ are simulated using mixtures of Gaussians and response $\mathbf{y}$ is linearly related to $S - P$ variables from $\mathbf{X}$. This allows us to evaluate prediction accuracy when using linear models and assess interpretability when creating sparse regressions.

Before embarking in the analysis and modeling of data, we set an evaluation procedure to assess

thesis objectives. On the one hand, the first main interest is to achieve accuracy. A typical method used is calibration and validation splitting. The first set is used to create the model and the second to evaluate forecast to prevent overfitting. We proposed the CalValXy algorithm that chooses the calibration observation by taking into account information from both $\mathbf{X}$ and $\mathbf{y}$. With the conception of the algorithm and the results obtained, we proved that the associated calibration covers the experimental space as uniformly as possible and provide accurate regression predictions compared to reference methods. This was concluded by computing the Euclidean distances and Φ_2 distances to evaluate the similarity between the calibration set and the whole dataset and the root mean squared error (RMSE), mean absolute error (MAE) and determination coefficient (R^2) when performing regression. On the other hand, we aimed at also providing insight on the signal according to the prediction of properties i.e. including further evaluation metrics, such as interpretability. We chose to address this challenge by detecting information with sparsity indicators. Sparsity in a regression model refers to the presence of a relatively small number of non-zero coefficients in the model. Sparse regression models are easier to interpret because they only include the most important variables in the model. This makes it easier to locate the most relevant factors that affect the outcome being predicted. They also involve fewer calculations, reducing the computational time required to build the regression model. This can be particularly beneficial when dealing with large datasets like ours. We propose to evaluate sparsity by computing the count measure ℓ_0 of regression coefficients and additionally plot them against original data spectra to compare localization.

With data that has a high number of dimensions, it can be useful to use dimension reduction techniques. These methods involve transforming the data from a high-dimensional space to a lower-dimensional one, while retaining the most important information from the original data. Two categories of approaches are commonly used: projection and penalized methods. Projection methods involve summarizing the original data matrix X in a lower-dimensional space, often using techniques such as PLS, which is commonly used in chemometrics. Penalized methods, on the other hand, involve penalizing the regression coefficient, often using techniques such as lasso, which can produce sparse results. To combine the benefits of both projection and penalized methods, a new generalized approach called dual-SPLS was developed. This method uses a mix of these regression methods and applies shrinkage adaptively based on the dual norm of a chosen penalty norm. We proposed four types of norms inspired by state-of-the art techniques: lasso, group lasso, least squares and ridge. They demonstrates a near-equivalent calibration and validation performance to the reference model, while using fewer latent components in prediction. A comparative benchmark test was conducted on both simulated models and real near infrared spectroscopy data. The resulting coefficients were found give precise location of influential data ranges. This provides greater interpretability of the trained prediction model.

This work was implemented as an R package called dual.spls. The purpose of this package is to make Dual-sPLS modelling available to the scientific community and so enable its application to real-world issues. Package dual.spls also propose additional functions to make it self-contained. It contains the real D_{NIR} data and the data generator algorithm. It also provide the CalValXy splitting method and evaluation tools.

Perspectives

This thesis proposed a comprehensive pipeline for the analysis of physico-chemical data, both from a practical and methodological perspective. From a practical point of view, we have made efforts to share data and code while explaining them. From a methodological point of view, we have developed two main blocks: first, a PLS related method, which is really the heart of the thesis, and secondly, a method of calibration and validation adapted to our motivation. Going back to the introduction, the steps on which we spent less time are on one hand the pre-processing of the data and on the other hand a more theoretical analysis.

We have utilized several pre-treatment procedures, including Savitzky Golay filtering and deriving spectra, to achieve these goals on real data. However, we are also keen to explore other treatments, such as wavelet treatment, which we believe can play an instrumental role in removing baselines, filtering, deriving and reducing noise. Wavelet treatment is particularly intriguing to us because it has the potential to reduce the dimensionality of the data by concentrating the information and increasing the gap between data points. In this way, we can identify the most suitable base that lies behind the data, allowing us to improve the overall performance of our predictive models. Additionally, we find that wavelet filtering potentially interesting to our problematic as they often use shrinkage in their applications. We hope that by leveraging these treatments, we can enhance the accuracy and effectiveness of Dual-sPLS or other PLS regression methods.
Figure 6.01 represents our first tests. On the upper left figure, we show the first NMR spectra of the data $D_{\text{NIR}}^{\text{NMR}}$, to which we applied a wavelet transformation of the type sym4 at a decomposition level of 1 that we show on the lower right side of the Figure. It reflects a diversity enhancement where information is concentrated in around the first 6000 variables. Beyond this range, this transformation shrinks the rest of the variables to zero leading to the sub-sampled spectra on the upper right plot. This hints that wavelet filtering reduces dimension and noise. With an inverse transformation of the data corresponding to the upper right plot, we reconstructed a spectra similar to the original on the lower left of Figure 6.01. This treatment was repeated to the 93 spectra in five levels of decomposition. We applied Dual-sPLS to each of the four sets of data.
Preliminary results seem to indicate that from an analytical standpoint, the transformation appears to effectively concentrate the information up to a certain level while respecting the positivity constraints of the data. In terms of prediction performance, have found that a more global decomposition of the data improves prediction accuracy and allows for a faster convergence to a stable number of components. However, there are still questions that need to be addressed, particularly regarding the location of the coefficients. Therefore, the results are worth to be developed to complete the analysis and draw accurate conclusions from the data.

Another intriguing subject for us is the statistical foundation of the PLS method, as research primarily focuses on PLS algorithms and avatars. As mentioned in this manuscript, PLS is a popular alternative to traditional regression techniques as it is effective in handling collinear and high-dimensional data sets. However, although PLS is commonly used and performs well in practical applications, the absence of statistical inference results for PLS presents a difficulty

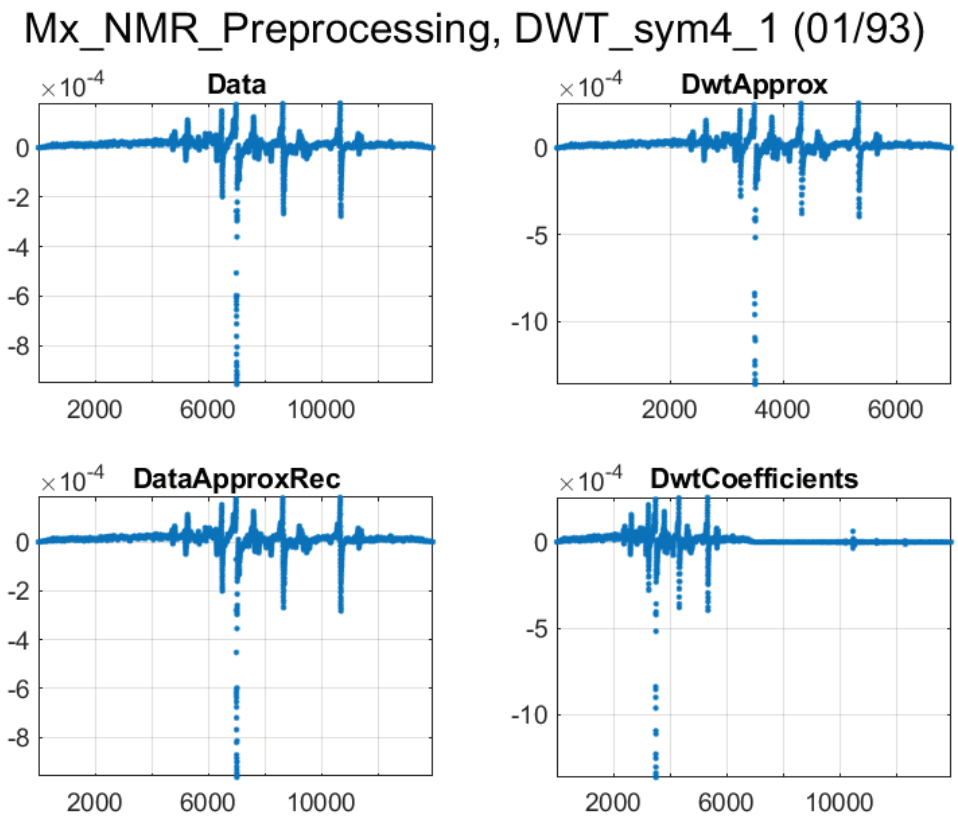

Figure 6.01 ~ *NMR wavelet transformation of the first spectra. Original spectra (top left), transformed spectra (bottom right), sub-sampled transformed spectra (top right) and inverse transformation spectra (bottom left).*

in determining the importance of the results and the level of uncertainty associated with them. Cook and Forzani [28] investigated the statistical properties of PLS in 2017, specifically the asymptotic behavior of prediction using the first PLS component. Their findings supported the idea that PLS regression is a useful approach for predicting in large, high-dimensional regression problems. They provided a solid support for PLS regression that is complementary to its practical use in chemometrics. They later developed their studies by moving on to multi-dimensional componenets [29]. However, they rely on statistical assumptions that may not align with what is typically observed in chemometric data. Therefore, it seems interesting to introduce slightly different and broader assumptions. In our future research, we aim to assess the level of uncertainty in the first direction and corresponding component of PLS, which indicates the degree of deviation from the true target. Our plan involves testing the null hypothesis and constructing confidence intervals. Through our research, we hope to contribute to the understanding of the statistical properties of PLS and provide insights into its practical application.

List of Figures

List of Tables

Bibliography

[1] H. Abdi. Partial least squares regression and projection on latent structure regression (pls regression). *WIREs Comp. Stat. (WARNING DUPLICATE with j-wiley-interdiscip-rev-comput-stat)*, 2(1):97–106, 2010.

[2] L. Alsouki, L. Duval, R. El Haddad, C. Marteau, and F. Wahl. CalValXy: well-balanced and stratified calibration/validation splitting using both predictors X and response y. *PREPRINT*, 2023. Submitted to Technometrics, June 2023.

[3] L. Alsouki, L. Duval, R. El Haddad, C. Marteau, and F. Wahl. Dual-sPLS: a family of dual sparse partial least squares regressions for feature selection and prediction with tunable sparsity; evaluation on simulated and near-infrared (NIR) data. *Chemometr. Intell. Lab. Syst.*, 2023.

[4] L. Alsouki, F. Wahl, and G. Durif. Dual sparse partial least squares: the R package dual.spls. *PREPRINT*, 2023. Submitted to Journal of Statistical Software, June 2023.

[5] L. Alsouki, F. Wahl, and G. Durif. dual.spls: Dual sparse partial least squares regression. CRAN, April 2023. R package version 0.1.4.

[6] J. C. Alves and R. Poppi. *Near-Infrared Spectroscopy in Analysis of Crudes and Transportation Fuels*, pp. 1–16. John Wiley & Sons, Ltd, 06 2015.

[7] S. B. Gadžurić, S. O. Podunavac Kuzmanović, M. B. Vraneš, M. Petrin, T. Bugarski, and S. Z. Kovačević. Multivariate chemometrics with regression and classification analyses in heroin profiling based on the chromatographic data. *Iranian Journal of Pharmaceutical Research : IJPR*, 15(4):e125240, 2016.

[8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, 2012.

[9] V. Baeten and P. Dardenne. Spectroscopy: Developments in instrumentation and analysis. *IGrasas y Aceites*, 53:45–63, Mar. 2002.

[10] R. J. Barnes, M. S. Dhanoa, and S. J. Lister. Standard normal variate transformation and de-trending of near-infrared diffuse reflectance spectra. *Appl. Spectrosc.*, 43(5):772–777, May 1989.

[11] B. Basu, G. Kapur, A. Sarpal, and R. Meusinger. A neural network approach to the prediction of cetane number of diesel fuels using nuclear magnetic resonance (nmr) spectroscopy. *Energy Fuels*, 17, Oct. 2003.

[12] D. Bertsimas, L. K., N. Azami, and F. Doucet. Novel mixed integer optimization sparse regression approach in chemometrics. *Anal. Chim. Acta*, 1137:115–124, 2020.

[13] M. Blanco, S. Maspoch, I. Villarroya, X. Peralta, J. González, and J. Torres. Determination of physical-chemical parameters for bitumens using near infrared spectroscopy. *Anal. Chim. Acta*, 434:133–141, April 2001.

[14] L. Bottmer, C. Croux, and I. Wilms. Sparse regression for large data sets with outliers. *European J. Oper. Res.*, 297(2):782–794, 2022.

[15] A.-L. Boulesteix and K. Strimmer. Predicting transcription factor activities from combined analysis of microarray and chip data: a partial least squares approach. *Theoretical biology and medical modelling*, 2:23, June 2005.

[16] A.-L. Boulesteix and K. Strimmer. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Brief. Bioinform.*, 8(1):32–44, 05 2007.

[17] F. Bovey, P. Mirau, and H. Gutowsky. *Nuclear Magnetic Resonance Spectroscopy*. Elsevier Science, 1988.

[18] BP. Statistical review of world energy. 2021.

[19] R. Bro. Multivariate calibration: What is in chemometrics for the analytical chemist? *Anal. Chim. Acta*, 500:185–194, Dec. 2003.

[20] J. Buendia Garcia, M. Lacoue-Negre, J. Gornay, S. Mas Garcia, R. Bendoula, and J. Roger. Diesel cetane number estimation from NIR spectra of hydrocracking total effluent. *Fuel*, 324:124647, sep 2022.

[21] D. Burns and E. Ciurczak. *Handbook of Near-Infrared Analysis*. Practical Spectroscopy. CRC Press, 2007.

[22] L. Carbognani, L. Diaz, T. Oldenburg, and P. Pereira-Almao. Determination of molecular masses for petroleum distillates by simulated distillation. *CT & F - Ciencia, Tecnología y Futuro*, 4:43–55, May 2012.

[23] J. M. Chalmers and P. R. Griffiths, editors. *Handbook of Vibrational Spectroscopy*. Wiley, 2002.

[24] A. Cherni, E. Chouzenoux, L. Duval, and J.-C. Pesquet. SPOQ ℓ_p-over-ℓ_q regularization for sparse signal recovery applied to mass spectrometry. *IEEE Trans. Signal Process.*, 68:6070–6084, 2020.

[25] H. Chun and S. Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(1):3–25, 2010.

[26] H. Chung. Applications of near-infrared spectroscopy in refineries and important issues to address. *Applied Spectroscopy Reviews*, 42:251–285, May 2007.

[27] W. Cochran, W. Cochran, and A. Bouclier. *Sampling Techniques*. Wiley Series in Probability and Statistics. Wiley, 1977.

[28] R. Cook and L. Forzani. Big data and partial least-squares prediction. *Can. J. Stat.*, 46, 04 2017.

[29] R. D. Cook and L. Forzani. Partial least squares prediction in high-dimensional regression. *Ann. Statist.*, 47(2):884 – 908, 2019.

[30] C. B. Y. Cordella. *PCA: The Basic Building Block of Chemometrics*. IntechOpen, Nov. 2012.

[31] R. D. Cramer. Partial least squares (pls): Its strengths and limitations. *Perspectives in Drug Discovery and Design*, 1:269–278, 1993.

[32] S. de Jong. SIMPLS: An alternative approach to partial least squares regression. *Chemometr. Intell. Lab. Syst.*, 18(3):251–263, Mar. 1993.

[33] L. K. DeNoyer and J. G. Dodd. *Smoothing and Derivatives in Spectroscopy*. John Wiley & Sons, Ltd, 2006.

[34] M. M. Deza and E. Deza. *Encyclopedia of distances*. Springer, 2009.

[35] M. Dhanoa, S. Lister, R. Sanderson, and R. Barnes. The link between multiplicative scatter correction (msc) and standard normal variate (snv) transformations of nir spectra. *J. Near Infrared Spectrosc.*, 2(1):43–47, 1994.

[36] X. Di and B. B. Biswal. Principal component analysis reveals multiple consistent responses to naturalistic stimuli in children and adults. *bioRxiv*, 2021.

[37] G. Durif, L. Modolo, J. Michaelsson, J. E. Mold, S. Lambert-Lacroix, and F. Picard. High dimensional classification with combined adaptive sparse PLS and logistic regression. *Bioinformatics*, 34(3):485–493, Feb. 2018.

[38] L. Duval, L. Alsouki, F. Wahl, J. Laxalde, and N. Caillol. MLnir IFPEN near-infrared spectroscopy dataset for property prediction: 208 NIR hydrocarbon spectra and density response. *PREPRINT*, 2023. Submitted to Data in Brief, June 2023.

[39] N. Gallagher. Savitzky-golay smoothing and differentiation filter, Jan. 2020.

[40] Z. German-Sallo. Nonlinear filtering in data signal denoising. In *9th RoEduNet IEEE International Conference*, pp. 85–88, 2010.

[41] J. L. Godoy, J. R. Vega, and J. L. Marchetti. Relationships between PCA and PLS-regression. *Chemometr. Intell. Lab. Syst.*, 130:182–191, jan 2014.

[42] A. Goldberger, W. Shenhart, and S. Wilks. *Econometric Theory*. WILEY Series in probability and statistics: applied probability and Statistics section Series. J. Wiley, 1964.

[43] D. L. Goodhue, W. Lewis, and R. Thompson. Does pls have advantages for small sample size or non-normal data? *MIS Quarterly*, 36(3):981–1001, 2012.

[44] S. Gorgunoglu, E. Ozkaynak, and I. Orak. Using pca algorithm in voice recognition. *Energy Education Science and Technology Part A: Energy Science and Research*, 30:759–764, Dec. 2012.

[45] C. J. Greenwood, G. J. Youssef, P. Letcher, J. A. Macdonald, L. J. Hagg, A. Sanson, J. Mcintosh, D. M. Hutchinson, J. W. Toumbourou, M. Fuller-Tyszkiewicz, and C. A. Olsson. A comparison of penalised regression methods for informing the selection of predictive markers. *PLoS One*, 15(11):1–14, Nov. 2020.

[46] S. Haber. A modified monte-carlo quadrature. *Mathematics of Computation*, 20:361–368, 1966.

[47] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.

[48] K. Hidajat and S. Chong. Quality characterisation of crude oils by partial least square calibration of nir spectral profiles. *J. Near Infrared Spectrosc.*, 8(1):53–59, Jan. 2000.

[49] A. E. Hoerl and R. W. Kennard. Ridge regression: Applications to nonorthogonal problems. *Technometrics*, 12(1):69–82, Feb. 1970.

[50] U. G. Indahl. The geometry of PLS1 explained properly: 10 key notes on mathematical properties of and some alternative algorithmic approaches to PLS1 modelling. *J. Chemometrics*, 28(3):168–180, 2014.

[51] I. Jolliffe. *Principal Component Analysis*, pp. 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[52] V. R. Joseph and A. Vakayil. Optimal ratio for data splitting. *Statistical Analysis and Data Mining*, 4(15):531–538, 2022.

[53] V. R. Joseph and A. Vakayil. SPlit: An optimal method for data splitting. *Technometrics*, 64(2):166–176, 2022.

[54] R. W. Kennard and L. A. Stone. Computer aided design of experiments. *Technometrics*, 11(1):137–148, Feb. 1969.

[55] M. Khanmohammadi, A. Bagheri Garmarudi, and M. Guardia. Characterization of petroleum based products by infrared spectroscopy and chemometrics. *Trends Anal. Chem.*, 35:135–149, 05 2012.

[56] L. Kish. *Survey Sampling*. John Wiley & Sons, Inc, 1965.

[57] N. Krämer, A.-L. Boulesteix, and G. Tutz. Penalized partial least squares with applications to B-spline transformations and functional data. *Chemometr. Intell. Lab. Syst.*, 94(1):60–69, nov 2008.

[58] A. Krishnan, L. J. Williams, A. R. McIntosh, and H. Abdi. Partial least squares (PLS) methods for neuroimaging: A tutorial and review. *Neuromimage*, 56(2):455–475, May 2011.

[59] J. Laxalde. *Analyse des produits lourds du pétrole par spectroscopie infrarouge*. PhD thesis, Université de Lille 1, 2012.

[60] J. Laxalde, C. Ruckebusch, O. Devos, N. Caillol, F. Wahl, and L. Duponchel. Characterisation of heavy oils using near-infrared spectroscopy: Optimisation of preprocessing methods and variable selection. *Anal. Chim. Acta*, 705(1-2):227–234, oct 2011.

[61] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, and P. Besse. A sparse PLS for variable selection when integrating omics data. *Stat. Appl. Genet. Mol. Biol.*, 7(1):35, 2008.

[62] P. Levy and S. Lemeshow. *Sampling of Populations: Methods and Applications*. Wiley Series in Survey Methodology. Wiley, 2013.

[63] T. y. Liu, L. Trinchera, A. Tenenhaus, D. Wei, and A. O. Hero. Globally sparse pls regression. In H. Abdi, W. W. Chin, V. Esposito Vinzi, G. Russolillo, and L. Trinchera, editors, *New Perspectives in Partial Least Squares and Related Methods*, pp. 117–127, New York, NY, 2013. Springer New York.

[64] S. L. Lohr. *Sampling: Design and Analysis*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2021.

[65] J. Lynch, editor. *Physico-Chemical Analysis of Industrial Catalysts. A Practical Guide to Characterisation*. Édition Technip, Sep. 2003.

[66] S. Ma and Y. Dai. Principal component analysis based methods in bioinformatics studies. *Brief. Bioinform.*, 12:714–22, 11 2011.

[67] F. Mabood, S. Gilani, M. Al-Broumi, S. Alameri, M. Nabhani, F. Jabeen, J. Hussain, A. Al-Harrasi, R. Boqué, S. Farooq, A. Hamaed, Z. Naureen, A. Khan, and Z. Hussain. Detection and estimation of super premium 95 gasoline adulteration with premium 91 gasoline using new nir spectroscopy combined with multivariate methods. *Fuel*, 197:388–396, June 2017.

[68] G. Mateos-Aparicio Morales. Partial least squares (PLS) methods: Origins, evolution, and application to social sciences. *Commun. Stat. Theory Methods*, 40(13):2305–2317, apr 2011.

[69] T. Mehmood and B. Ahmed. The diversity in the applications of partial least squares: an overview. *J. Chemometrics*, 30(1):4–17, Jan. 2016.

[70] P. Mishra, B. Brouwer, and L. Meesters. Improved understanding and prediction of pear fruit firmness with variation partitioning and sequential multi-block modelling. *Chemometr. Intell. Lab. Syst.*, 222, 2022.

[71] M. K. Moro, F. D. dos Santos, G. S. Folli, W. Romao, and P. R. Filgueiras. A review of chemometrics models to predict crude oil properties from nuclear magnetic resonance and infrared spectroscopy. *Fuel*, 303, 2021.

[72] M. K. Moro, F. D. dos Santos, G. S. Folli, W. Romão, and P. R. Filgueiras. A review of chemometrics models to predict crude oil properties from nuclear magnetic resonance and infrared spectroscopy. *Fuel*, 303:121283, 2021.

[73] M. G. Nespeca, R. R. Hatanaka, D. L. Flumignan, and J. E. d. Oliveira. Rapid and simultaneous prediction of eight diesel quality parameters through atr-ftir analysis. *J. Anal. Methods Chem.*, 2018, 2018.

[74] K. Nielsen, J. Dittmer, A. Malmendal, and N. Nielsen. Quantitative analysis of constituents in heavy fuel oil by 1h nuclear magnetic resonance (nmr) spectroscopy and multivariate data analysis. *Energy Fuels*, 22, Nov. 2008.

[75] X. Ning, I. W. Selesnick, and L. Duval. Chromatogram baseline estimation and denoising using sparsity (BEADS). *Chemometr. Intell. Lab. Syst.*, 139:156–167, Dec. 2014.

[76] H. Palo, S. Sahoo, and A. Subudhi. *Dimensionality Reduction Techniques: Principles, Benefits, and Limitations*, pp. 77–107. John Wiley & Sons, Ltd, Feb. 2021.

[77] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[78] L. F. Ramírez-Verduzco, J. E. Rodríguez-Rodríguez, and A. del Rayo Jaramillo-Jacob. Predicting cetane number, kinematic viscosity, density and higher heating value of biodiesel from its fatty acid methyl ester composition. *Fuel*, 91(1):102–111, Jan. 2012.

[79] o. Rinnan, F. Berg, and S. Engelsen. Review of the most common pre-processing techniques for near-infrared spectra. *Trends Anal. Chem.*, 28:1201–1222, Nov. 2009.

[80] R. P. Rodgers and A. M. McKenna. Petroleum analysis. *Anal. Chem.*, 83(12):4665–4687, 2011.

[81] C.-E. Särndal, B. Swensson, and J. Wretman. *Model Assisted Survey Sampling*. Springer, 2003.

[82] M. I. S. Sastry, A. Chopra, A. S. Sarpal, S. K. Jain, S. P. Srivastava, and A. K. Bhatnagar. Determination of physicochemical properties and carbon-type analysis of base oils using mid-ir spectroscopy and partial least-squares regression analysis. *Energy Fuels*, 12(2):304–311, 1998.

[83] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.*, 36(8):1627—1639, July 1964.

[84] H. Shen and J. Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Anal.*, 99(6):1015–1034, Jul. 2008.

[85] N. Simon and R. Tibshirani. Standardization and the group lasso penalty. *Statistica Sinica*, 22(3):983–1001, 2012.

[86] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *J. Comp. Graph. Stat.)*, 22(2):231–245, Apr. 2013.

[87] A. Sivasakthi and T. Nagalakshmi. Characterization of heavy crude oil through physical and chemical properties. *International Journal for Science and Advance Research in Technology (IJSART)*, 4:1379–1382, Mar. 2018.

[88] R. D. Snee. Validation of regression models: Methods and examples. *Technometrics*, 19(4):415–428, nov 1977.

[89] J. G. Speight. *Handbook of petroleum product analysis*. John Wiley & Sons, 2015.

[90] J. Speight. *The Chemistry and Technology of Petroleum, Fifth Edition*. Chemical Industries. Taylor & Francis, 2014.

[91] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 36(2):111–133, Jan. 1974.

[92] G. J. Székely and M. L. Rizzo. Partial distance correlation with methods for dissimilarities. *Ann. Statist.*, 42(6):2382–2412, 2014.

[93] M. Tenenhaus. *La régression PLS. Théorie et pratique*. Éditions Technip, 1998.

[94] H. Tian, L. Zhang, M. Li, Y. Wang, D. Sheng, J. Liu, and C. Wang. Weighted SPXY method for calibration set selection for composition analysis based on near-infrared spectroscopy. *Infrared Phys. Technol.*, 95:88–92, Dec. 2018.

[95] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996.

[96] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, Feb. 2005.

[97] E. van Wyngaard, E. Blancquaert, H. Nieuwoudt, and J. L. Aleixandre-Tudo. Infrared spectroscopy and chemometric applications for the qualitative and quantitative investigation of grapevine organs. *Frontiers in Plant Science*, 12, 2021.

[98] S. Verdier, J. A. P. Coutinho, A. M. S. Silva, O. F. Alkilde, and J. A. Hansen. A critical approach to viscosity index. *Fuel*, 88(11):2199–2206, Nov. 2009.

[99] A. P. Vieira, N. A. Portela, A. C. Neto, V. Lacerda, W. Romão, E. V. R. Castro, and P. R. Filgueiras. Determination of physicochemical properties of petroleum using 1h nmr spectroscopy combined with multivariate calibration. *Fuel*, 253:320–326, 2019.

[100] D. Villalanti, J. Raia, and J. Maynard. *High-temperature Simulated Distillation Applications in Petroleum Characterization*, pp. 6726–6741. John Wiley & Sons, Ltd, Sept. 2006.

[101] H. H. Willard, L. L. Merritt, and J. A. Dean. *Méthodes physiques de l'analyse chimique*. Dunod, 1965.

[102] H. Wold. Nonlinear iterative partial least squares (nipals) modelling: Some current developments. In P. R. Krishanaih, editor, *Multivariate Analysis–III*, pp. 383–407. Academic Press, 1973.

[103] H. Wold. Path models with latent variables: The NIPALS approach. In *Quantitative Sociology. International Perspectives on Mathematical and Statistical Modeling*, pp. 307–357. Elsevier, 1975.

[104] S. Wold. Chemometrics; what do we mean with it, and what do we want from it? *Chemometr. Intell. Lab. Syst.*, 30:109–115, 1995.

[105] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometr. Intell. Lab. Syst.*, 2(1):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[106] S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometr. Intell. Lab. Syst.*, 58(2):109–130, 2001.

[107] J. Wright and Y. Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[108] C. Yang, G. Zhang, M. Serhan, G. Koivu, Z. Yang, B. Hollebone, P. Lambert, and C. E. Brown. Characterization of naphthenic acids in crude oils and refined petroleum products. *Fuel*, 255, 2019.

[109] H. W. Yarranton. Prediction of crude oil saturate content from a simdist assay. *Energy Fuels*, 36(16):8809–8817, 2022.

[110] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):49–67, Feb. 2006.

[111] M. Yunus, A. Saefuddin, and A. Soleh. Characteristics of group lasso in handling high correlated data. *Appl. Math. Sci.*, 11:953–961, Jan. 2017.

[112] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(2):301–320, Apr. 2005.