TECHNISCHE
UNIVERSITÄT
WIEN

DISSERTATION

Contributions to robust and sparse estimation for regression, association, and dimension reduction

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften unter der Leitung von

Peter Filzmoser
E105 – Institute of Statistics and Mathematical Methods in Economics, TU Wien

eingereicht an der Technischen Universität Wien
Fakultät für Mathematik und Geoinformation

von

Pia Pfeiffer
Matrikelnummer: 01225133

Diese Dissertation haben begutachtet:

1. **Univ.Prof. Dr.techn. Peter Filzmoser**
   Institute of Statistics and Mathematical Methods in Economics, TU Wien
2. **Assoc.Prof. Ines Wilms, PhD**
   School of Business and Economics, Maastricht University
3. **Prof. Luca Greco, PhD**
   University Giustino Fortunato, Beneveto

Wien, am 29. April 2024

Kurzfassung

In dieser Arbeit werden die Herausforderungen bei der Analyse von Datensätzen aus empirischen Experimenten behandelt, wobei ein besonderer Schwerpunkt auf der Erkennung von Ausreißern und der Analyse hochdimensionaler Daten liegt. Durch robuste Ansätze und den Einsatz moderner Optimierungsalgorithmen bietet die Arbeit praktische Lösungen zur Verbesserung der Zuverlässigkeit und Effizienz von Datenanalysetechniken in komplexen realen Szenarien und bietet Erkenntnisse und praktische Werkzeuge für die Anwendung in verschiedenen Bereichen. Beiträge zur robusten und penalisierten Regression, Assoziation und Dimensionsreduktion werden anhand von Datensätzen aus der Tribologie veranschaulicht, einem multidisziplinären Gebiet, das Reibung, Verschleiß und Schmierung untersucht. Diese Daten stammen aus Experimenten mit Motorölen in unterschiedlichen Zuständen und umfassen Spektral-, Funktions- und Bilddaten mit jeweils nur begrenzter Anzahl von Beobachtungen. Robuste Methoden, die für niedrigdimensionale Daten entwickelt wurden, reichen nicht aus, um experimentelle Datensätze im hochdimensionalen Fall zu verarbeiten. Daher werden in dieser Arbeit geeignete Preprocessing- und Samplingstrategien für robuste Regression und Klassifikation unter diesen Bedingungen vorgestellt. Darüber hinaus wird eine Kombination von robusten statistischen Methoden mit gradientenbasierten Optimierungstechniken untersucht, um die Beziehung zwischen zwei multivariaten Datensätzen durch robuste und regularisierte CCA (kanonische Korrelationsanalyse) zu quantifizieren. Weiters wird eine Methode zur Dimensionsreduktion mittels robuster und regularisierter PCA (Hauptkomponentenanalyse) vorgestellt.

Abstract

In this thesis, challenges inherent to analyzing datasets from empirical experiments are addressed, with a particular focus on outlier detection and the analysis of high-dimensional data. By proposing robust approaches and leveraging modern optimization algorithms, the thesis offers practical solutions for enhancing the reliability and efficiency of data analysis techniques in complex real-world scenarios, offering valuable insights and practical tools for researchers and practitioners in various fields. Contributions to robust and sparse regression, association, and dimension reduction are illustrated on datasets from tribology, a multi-disciplinary field studying friction, wear, and lubrication. These data result from practical experiments with engine oils in different conditions and from several degradation pathways and include spectral, functional, and image data with only a limited number of observations. Robust methods tailored for low-dimensional data do not suffice for handling experimental datasets in high-dimensional settings. Therefore, this thesis presents suitable preprocessing and sampling strategies for robust regression and classification in this challenging setting. In addition, a combination of robust statistical methods with gradient-based optimization techniques is proposed for quantifying the relation between two multivariate datasets using robust and sparse CCA (canonical correlation analysis) and dimension reduction via robust and sparse PCA (principal component analysis).

Acknowledgement

First and foremost, I want to thank my supervisor Peter, for giving me the chance to pursue my doctorate at TU Wien. Thank you for your support, motivation, patience, and advice. You were always there to listen and provide knowledgeable input but also kind words when needed. I am very grateful that you encourage an environment that makes coming to work something to look forward to. It cannot be taken for granted to always have the support of one's colleagues, and I would like to thank the whole CSTAT group, but especially my predoc colleagues Marcus, for being the best "roommate", and also Patricia, Barbara, Roman, Lukas, and Jeremy, for our lunchtime traditions, all the coffee conversations, and, most importantly, for having become friends in the past years. I want to thank my colleagues from AC2T Research GmbH for providing insights into tribology and for the good collaboration, especially to Bettina, Georg, Josef, and Nicole. This research was funded by the Austrian COMET-Program (project InTribology2, No. 906860) via the Austrian Research Promotion Agency (FFG) and the federal states of Niederösterreich and Vorarlberg. I would also like to express my gratitude to my referees, Ines Wilms and Luca Greco, for their time and effort in reviewing this thesis.

Working towards a PhD degree without the support of the people in my personal life would have been rather lonely, and I would like to say thank you to all my friends and family, but especially: To Sabi, Bianca, Alex, Michi, Shelly, Carola, Valentina, and Angelika, for being by my side for longer than I can remember and never being further than a phone call away. To Manu, for never running out of ideas for destinations, and being the best travel buddy. To all the members of the "Sudern" group for sharing the love for breakfast and Austrian Grant. To Thomas, for your love and endless support, and for believing in me, when I wouldn't myself. There's no one I'd rather share this life with. To Birgit, for providing distraction and support whenever needed and for the best Sunday brunches (ever!). To my (not so little) sister Iris, for sharing my love for books, knitting, movies, and music, and for never missing a stadium concert. To Omi, who did not get the chance to see me finish my studies, thank you for always being there for me and for teaching me the importance of being a compassionate and kind person. Last, but not least, I want to thank my parents for encouraging me to chase my dreams and giving me the possibility to prioritize my education.
I would not have come far without any of you, and I am happy to have you in my life.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 29. April 2024

Pia Pfeiffer

Contents

1 Introduction

Datasets derived from empirical experiments often present challenges for (robust) statistical methods. Outliers, i.e. observations deviating from the majority of the data, may be present, and high-dimensional datasets, referring to the case when more variables than observations are recorded, need to be analyzed with appropriately designed statistical methods.

The necessity to systematically examine big sets of potentially noisy data can be seen as a prominent aspect of modern data analysis (Hastie et al., 2015). Especially in industries where the data originate from practical experiments, either in a laboratory or from field or bench tests, and only limited samples are available, efficient prediction methods to understand underlying mechanisms are crucial. This especially applies to the development of machinery, which should be sustainable and reliable. In industrial machines, but also in passenger cars, for example, not only the mechanic parts contribute to these properties, but also the used lubricants, i.e. engine or hydraulic oils. In the highly interdisciplinary field of tribology, friction, wear, and lubrication are studied using various analytical methods. The aim of this thesis is the development of robust statistical methods and algorithms designed to work with challenging datasets produced from practical experiments. The data obtained from oils at different degradation stages have been provided by AC2T research GmbH in Wiener Neustadt and include spectra, functional, compositional, but also image data and can roughly be grouped as lubricant chemistry, tribofilm and surface characteristics, and tribological behavior. However, tribometrical experiments are expensive and time-consuming, and often, only a limited number of observations (compared to the number of measured variables) are available. Furthermore, the data are prone to measurement errors, which should not influence the resulting models.

Our contribution aims to simplify the analytical work and reduce cost- and time-intensive tribological experiments. Robust methods and algorithms are developed in a data-driven way, tailored to problems and questions from tribology. However, the methods are not limited to tribology, but can be applied to any field where there is a need to extract and combine relevant information from multivariate high-dimensional data with only limited numbers of observations.

1.1 Motivation

In collaboration with AC2T research GmbH, we aim to provide answers to questions related to the condition monitoring of engine oils. The lubricating properties of an engine oil depend on the oil condition, and quantifying this relation between lubricant chemistry and tribological performance is an active research topic in tribology (Felkel et al., 2010; Al-Ghouti et al., 2010; Hirri et al., 2017; Besser et al., 2013). It is also of interest to compare different degradation pathways of engine oils, i.e. oils altered in the laboratory versus oils used in

the field or in test rigs. For this purpose, multivariate datasets originating from different kinds of experiments have been provided by AC2T research GmbH. As experimental data is susceptible to inaccuracies, leading to outlying observations and/or cells, robust methods are needed for their analysis. While many robust methods have been proposed and studied for the low-dimensional setting (see, e.g., Maronna et al., 2006), there are not as many robust methods designed for high-dimensional data, both in the sense of many variables as well as many observations. This especially applies to robust estimators that cannot be derived analytically but via an iterative procedure. Proposed algorithms that rely on finding optimal subsets of the data or the repeated estimation of regression models do not scale well to growing dimensions. Optimization techniques based on versions of gradient descent have been developed for large-scale problems in computer science, and we demonstrate how they can be effectively applied to different kinds of problems in robust statistics.

Section 1.2 provides an introduction to tribology and the available data, Section 1.3 summarizes some basics of robust statistics, and Section 1.4 gives an overview of gradient-based optimization. Chapters 2 and 3 present selected sparse and robust regression and classification techniques and their application to data from tribology for the prediction of the engine oil condition based on spectral and image data. In Chapter 4, an algorithm for the efficient computation of sparse and robust maximum association measures to relate two multivariate datasets is developed, and in Chapter 5 we present an approach to cellwise robust dimension reduction via PCA. Chapter 6 gives details to the implementation and guidance for the usage of the developed R package `RobSparseMVA` (R Core Team, 2023; Pfeiffer et al., 2024). Finally, the findings are summarized in Chapter 7, and an outlook on interesting future research topics is given.

1.2 Tribology

The name tribology originates from the Greek word $\tau\rho\iota\beta\omega$, meaning "to rub", and has been coined by Jost (1966), who described it as "the science and technology of interacting surfaces in relative motion and of practices related thereto." Nowadays, it is more commonly referred to as the science of friction, wear, and lubrication, although it is a much wider interdisciplinary field including (non-exhaustively) physics, chemistry, materials science, and mechanics, but also applied mathematics (Bhushan, 2013; Hutchings and Shipway, 2017).

Before describing the available dataset from tribology, we give a short summary of how the keywords "friction", "wear", and "lubrication" are defined in Hutchings and Shipway (2017):

- Friction refers to "the resistance encountered by one body in moving over another" (Hutchings and Shipway, 2017). For both rolling and sliding friction, the tangential force F moves the upper body over a counterface. The frictional force or coefficient of friction corresponds to the ratio between F and the normal force W.

- Lubricants can be a variety of materials and introduce a layer between the surfaces to prevent asperity contact and, subsequently, to reduce the frictional force between surfaces (Hutchings and Shipway, 2017). This is important as a lack of lubrication

could lead to high friction forces and therefore frictional energy losses not acceptable for engineering/industrial applications.

- Wear occurs when two surfaces slide against each other and refers to a complex process that depends on the materials of the surfaces and the lubricant. Its main effects can be categorized into stress (causing deformation), damage, thermal effects, and even chemical reactions or interactions of surfaces. In addition, reaction layers (e.g. oxides in the case of metals) can be formed.

The understanding of these complex phenomena and how they interact with each other is crucial for the design of efficient machinery, as has already been recognized by Jost in 1966, who mainly considered financial savings, for example, in maintenance and replacement costs of mechanical components and extended lifetime of plants. Nowadays, environmental sustainability is also a prominent topic in tribology (Holmberg and Erdemir, 2017). Especially in transport, where over 30% of the energy is used to overcome friction, tribological innovations could reduce energy loss by 18-40%, corresponding to around 8.7% of the global energy use, as discussed in a study by Holmberg and Erdemir (2019). And while the market share of electric cars grows, the majority of cars are still powered by internal combustion engines (IEA, 2023). Thus, the study of engine oils and their tribological performance is a crucial step in the development of modern, energy-efficient automotive engines (Besser et al., 2019).

By combining robust statistical methods with efficient optimization algorithms, this thesis contributes to a better understanding of how lubricant chemistry and tribological performance are related, specifically in the example of conventional lubricants that are commonly used as engine oils. Lubricant chemistry is represented by Fourier-Transform-Infrared (FTIR) spectra of engine oils, and tribological performance is reflected by images of wear scar areas that were taken under a microscope after friction and wear tests on a Schwing-Reib-Verschleiss SRV® tribometer. While the specific datasets used for modeling are described in more detail in the respective chapters, the experiments that the given data are derived from are summarized in the following.

1.2.1 Artificial oil alteration

Besser et al. (2019) motivate the production of engine oils at different degradation stages for testing components in the development of automotive engines. It has been investigated by several authors that the oil condition has an influence on its lubricating properties (Ponjavic et al., 2017; De Feo et al., 2015), and as the engine oil in a passenger car will only be fresh at the beginning of its lifecycle, it is sensible to also use lubricants with different degrees of degradation in the development phase, e.g. for bench tests or engine test rigs.

"Used" engine oils can be produced in the laboratory (Agocs et al., 2020) or obtained from field tests in passenger cars (Agocs et al., 2021). Artificial alteration in the laboratory has the advantage that degraded oils can be generated in a relatively short time under well-defined conditions from small (1L) to larger (200L) amounts (Besser et al., 2019).

A rather simple procedure for artificially altering engine oils is a modified version of the MAN test, described by Dörr et al. (2019b). This test refers to an open-beaker setup, where the beakers are placed in an oven at temperatures ranging between 120 and 180°C and are

Figure 1.1: Alteration example: Open-beaker setup. Source: AC^{2}T research GmbH.

sampled and analyzed according to regular sampling intervals. An example setup is shown in Figure 1.1.

Thermo-oxidative artificial alteration methods involve oxidative stress in addition to thermal stress. On a small laboratory scale, this can be carried out as suggested by Besser et al. (2012): The lubricant is contained in a round-bottomed glass flask and placed in a heated bath. Then dried air flow is applied to the oil via a tube. The schematic on the left-hand side of Figure 1.2 shows the setup. Again, the oil is regularly sampled and analyzed.

The amounts of altered oils that can be produced by such a small-scale alteration are still restricted to about 1L, which is why a large-scale alteration that is able to produce up to 200L of degraded engine oil has been developed by Besser et al. (2019). On this scale, a chemical reactor with heating, air management, and cooling units and a stirring mechanism is needed. The setup is described in detail in Besser et al. (2012). The schematic on the right of Figure 1.2 refers to this large-scale alteration.

1.2.2 Experimental data

The oil samples taken during different time points in either of the artificial alteration procedures are analyzed using different methods. "Conventional" oil analysis refers to the evaluation of oil attributes like the viscosity of the oil, the water content, as well as neutralization number or total base number (Besser et al., 2019). In addition, oxidative components and residual components of anti-oxidants and anti-wear additives can be evaluated from specific absorption bands in FTIR spectra of the respective oils.

FTIR spectroscopy measures the absorption of infrared radiation by a sample. It works by passing infrared light through a sample and detecting how much of the light is absorbed at different wavelengths, providing information about the sample's molecular structure and composition (Griffiths and de Haseth, 2007). An example of a typical FTIR spectrum of an engine oil is shown in Figure 1.3. The absorption bands around 3000 cm^{-1} and 1450 cm^{-1} correspond to C-H stretching and bending vibrations (Chimeno-Trinchet et al., 2020), which

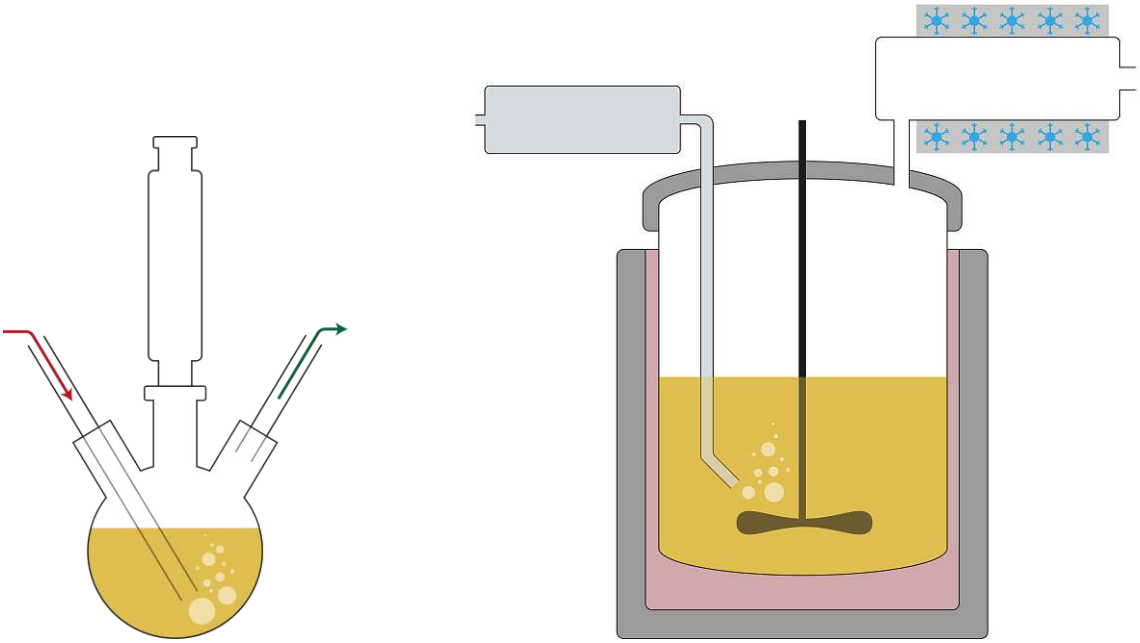


Figure 1.2: Left: Schematic for small-scale alteration. Right: Schematic for large-scale alteration. Source: AC^{2}T research GmbH.

are always high-absorption areas in engine oils and are therefore removed before analysis, either manually or by filtering procedures (Pfeiffer et al., 2022).

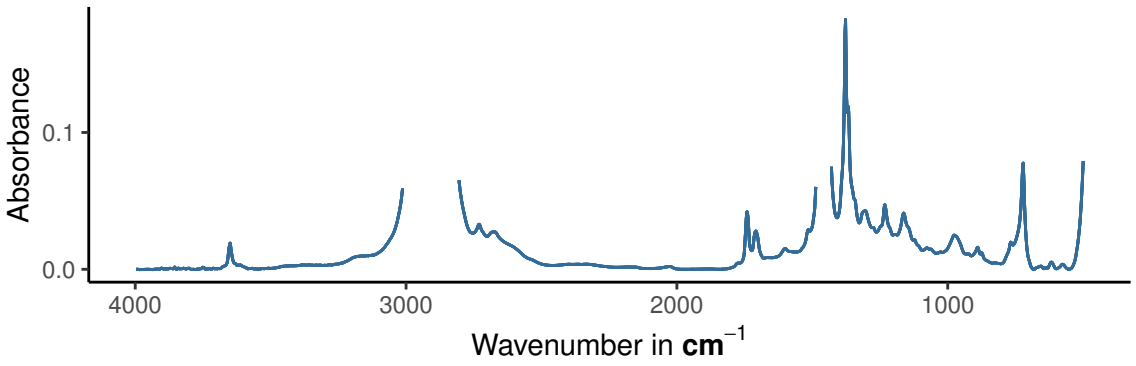


Figure 1.3: FTIR spectrum of an automotive engine oil.

For the evaluation of tribological performance, friction and wear experiments are performed on an SRV® 3 tribometer, yielding the coefficient of friction. For the datasets used in this thesis, the experiments were performed under sliding conditions using a ball-on-disc reciprocating contact with the lubricant in between the surfaces, as shown in the schematic on the left of Figure 1.4. The condition of the surfaces after the tribometer experiment is documented by photos taken under a microscope, in the following referred to as wear scar images. On the right side of 1.4, both the wear scar images of a ball and disc are shown

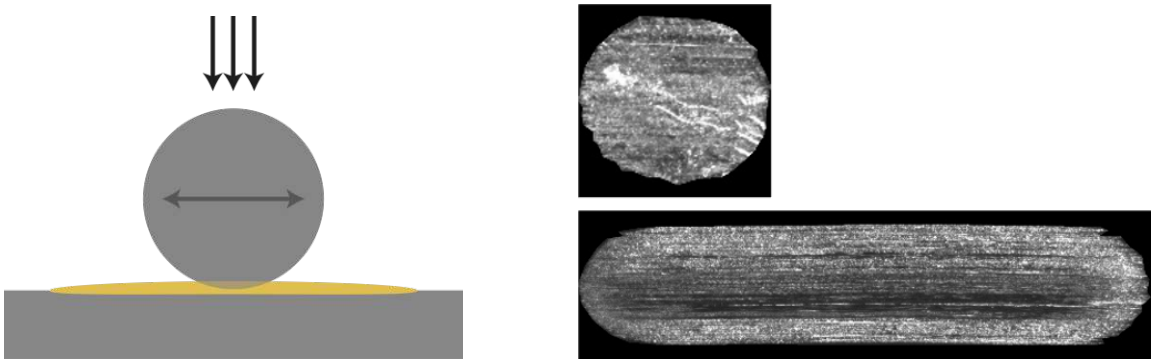


Figure 1.4: Left: Schematic for tribometer setup. Right: Wear scar areas. Source: AC^{2}T research GmbH.

after pre-processing. As we are mainly interested in the texture of the wear scar areas, the original RGB images were converted to greyscale based on brightness and the background of the wear scars was removed. Then, the images were cut to fit around the wear scars and the image dimensions unified across the samples. To account for different lighting conditions, normalization was performed using cv2.normalize from the OpenCV library (Bradski, 2000) with a reference image of zeros and the MINMAX type, scaling pixel values between 0 and 255.

As the data has been produced in a laboratory with possibly varying conditions and risk of contamination, the analysis will call for robust methods that can deal with outlying observations. In Section 1.3, we give a brief introduction to robust statistics, the different paradigms, and concepts needed for estimation.

1.3 Robust statistics

Classical statistics is based on assumptions such as the normality or linearity of the observations. However, even if the majority of a dataset satisfies the assumptions, there is often a small proportion of the data that behaves differently. These observations, not following the pattern of the bulk of the data, are commonly referred to as outliers (Maronna et al., 2006). Barnett et al. (1994) have defined these points as "a set of data to be an observation (or subset of observations) which appears to be inconsistent with the remainder of the dataset." The importance of robustness against outliers has already been recognized by Box (1953) and Tukey (1960), and Huber (1964) and Hampel (1974) contributed the theoretical foundations thereof. In Hampel et al. (1986), robust statistics is defined as follows: "In a broad informal sense, robust statistics is a body of knowledge, partly formalized into 'theories of robustness', relating to deviations from idealized assumptions in statistics."

The robust statistical approach aims to develop models that follow the majority of the data and downweight the influence of outlying observations. In addition, robust diagnostics provide a more reliable way to identify outliers (Maronna et al., 2006; Hampel et al., 1986).

1.3.1 Casewise versus cellwise outliers

We can distinguish between two paradigms in robust statistics, namely the casewise and cellwise approaches. In casewise robustness, whole observations are flagged as either outlying or regular data points. Formally, this corresponds to the Tukey-Huber contamination model which is given as

$$X = (1 - B)Y + BZ, \tag{1.1}$$

where $Y \sim F$ with F corresponding to the model distribution, and $Z \sim G$ with G corresponding to the outlier-generating distribution. $B \sim Bin(1, \varepsilon)$, for a small value ε, can be interpreted as a *contamination indicator* (Alqallaf et al., 2009).

In the multivariate setting, this model has been criticized for only allowing rowwise contamination, when in reality–especially in high-dimensional settings–it is likely that only a few columns in many rows are affected by outliers.

For this cellwise contamination framework, Alqallaf et al. (2009) have formalized the independent contamination model as

$$X = (\boldsymbol{I} - \boldsymbol{B})Y + \boldsymbol{B}Z, \tag{1.2}$$

where $\boldsymbol{B} = diag(B_1, B_2, \ldots, B_p)$ are independent $B_i \sim Bin(1, \varepsilon_i)$, for $i = 1, \ldots, p$.

As pointed out by Raymaekers and Rousseeuw (2023b), the cellwise contamination model is related to the idea of explaining the outlyingness of an observation by the contributions of each variable. This includes the SPADIMO algorithm developed by Debruyne et al. (2019) and the explanation of outliers using Shapley values, as proposed by Mayrhofer and Filzmoser (2023).

Figure 1.5 illustrates the difference between the contamination paradigms: Both datasets include the same number of outlying cells, but on the right side (cellwise contamination), many more rows are affected than on the left side (casewise contamination).

While many robust methods have been developed and studied for the casewise contamination model (see, e.g., Maronna et al., 2006), the cellwise scenario has been an active research topic in recent years (Raymaekers and Rousseeuw, 2023b). Raymaekers and Rousseeuw (2023b) discuss the challenges that come with the cellwise contamination model and summarize the existing literature on cellwise robust estimation of location, correlation, covariance and precision matrices, regression, principal components analysis (PCA), clustering, and time series analysis, and generalize theoretical concepts like breakdown points (Maronna et al., 2006) for the cellwise framework.

In the following sections, we give an overview of (classes of) robust estimators, as well as strategies for dealing with high-dimensional data, i.e. the scenario when more variables than observations are present.

1.3.2 Robust estimation

A big class of robust estimators includes the M- and S-estimators, a generalization of maximum likelihood estimators for location and scatter (Maronna et al., 2006).

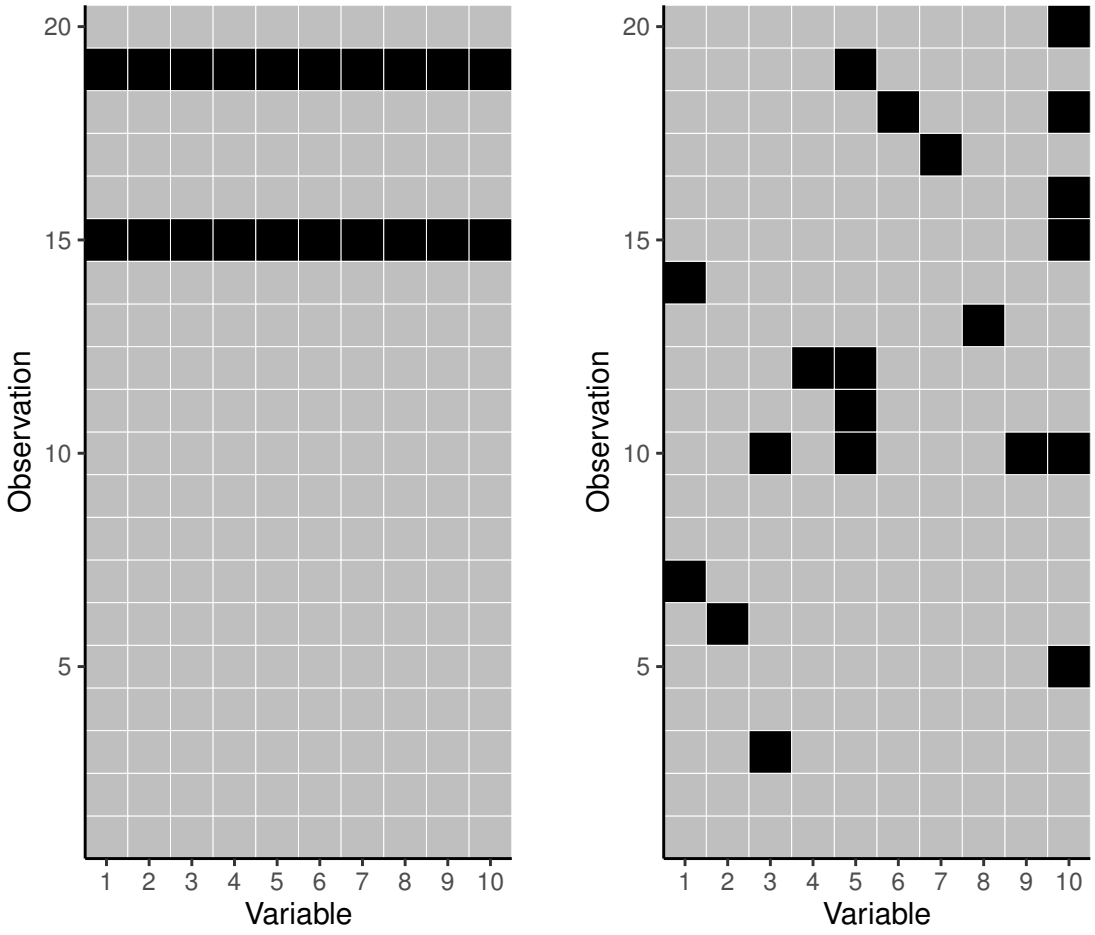


Figure 1.5: Casewise (left) versus cellwise (right) contamination scenario.

In the univariate case, M- and S-estimators of location and scatter work by re-weighting the observations based on a non-decreasing function ρ (Huber, 1964). For the multivariate case, the estimators work by applying the ρ-function to the Mahalanobis distances (Maronna, 1976). For linear regression, the re-weighting is based on the (scaled) residuals.

Trimmed estimators are another class of estimators, already hinting at the applied strategy: A proportion of the smallest and largest values is trimmed from the dataset before computing the mean, resulting in a trimmed mean. For robust scale, a similar concept has been introduced with the Q_n estimator (Rousseeuw and Croux, 1993), using the k-th order statistic of the $\binom{n}{2}$ distances, $Q_n = d\{|x_i - x_j|, i < j\}_{(k)}$. For robust covariance, the Minimum Covariance Determinant (MCD) - estimator (Rousseeuw, 1984, 1985) is a popular choice: It is based on finding the subset of observations resulting in the minimum determinant of the covariance matrix. The resulting location estimator is then the mean of the selected subset, and the covariance estimator is the sample covariance of the subset, multiplied by a consistency factor. The idea of trimming observations can also be applied to regression, resulting in the Least Trimmed Squares (LTS) estimator (Rousseeuw, 1984),

where observations are again down-weighted based on the corresponding residuals.

A more technical introduction to robust regression based on ρ–functions and trimmed residuals is given in Chapter 3.

Popular ρ–functions are shown in Figure 1.6. In Chapter 3, their application is demonstrated in regression, and in Chapter 5, different ρ–functions are used for robust matrix reconstruction.

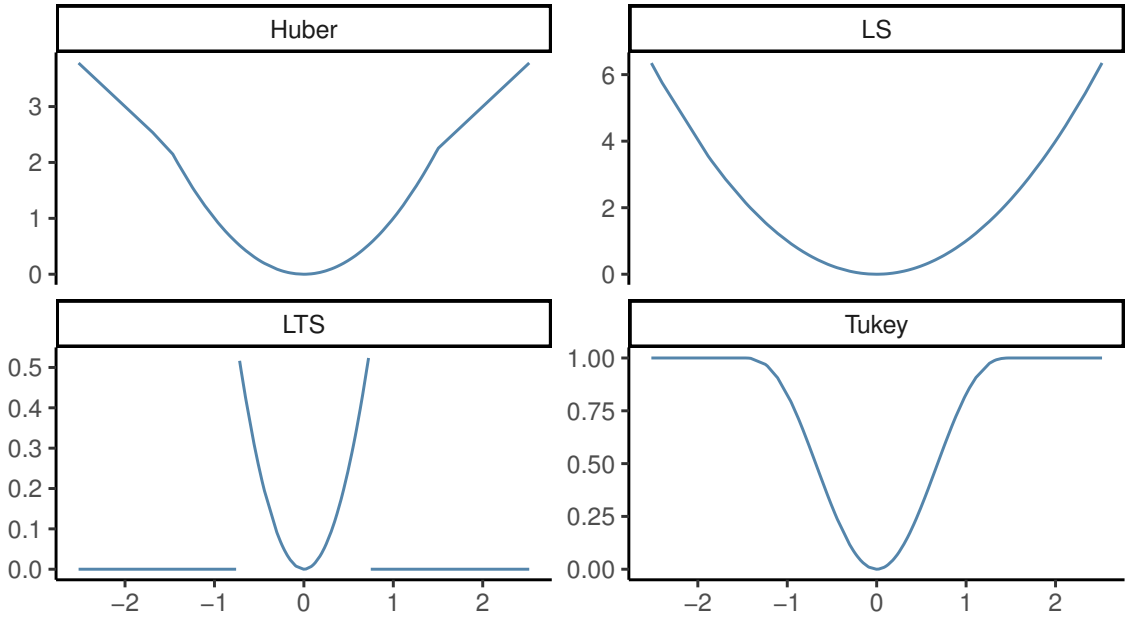


Figure 1.6: Different types of ρ-functions. The functions are shown for the parameters $b = 1.5$ for the Huber function, $c = 1.5$ for the Tukey function, and $h = 0.5$ for the LTS function for $\mathcal{N}(0, 1)$ distributed residuals.

The above methods are casewise robust, but generalizations for cellwise robustness have been studied: For covariance estimation, the MCD has been extended to cellwise MCD (Raymaekers and Rousseeuw, 2023b), for regression, the shooting algorithm, operating variable-wise, has been proposed (Bottmer et al., 2022). Another approach is to first detect cellwise outliers, and continue with estimation on the corrected dataset (Hubert et al., 2019). For cellwise robust correlation matrices, Öllerer et al. (2015) proposed to use pairwise rank-correlation measures. An alternative take on this idea is detailed by Raymaekers and Rousseeuw (2021), who proposed to use column-wise data transformations.

1.3.3 Dealing with high-dimensional data

High-dimensional datasets, in the sense that there are more variables than observations, lead to ill-conditioned estimators, even in the classical framework. When the application additionally requires robustness, this "curse of dimensionality" becomes even worse, as some observations (or cells) are discarded.

In order to extract useful information from such datasets, additional assumptions such as sparsity need to be made or dimension reduction needs to be performed. Hastie et al. (2015) gives an overview of sparse regression methods, such as the LASSO (Tibshirani, 1996) and generalizations, as well as sparse multivariate methods. The idea of the LASSO, i.e. adding a sparsity-inducing constraint via the L1-norm, has been applied to robust regression as well, yielding, for example, the sparse LTS estimator for regression (Alfons et al., 2013) or the sparse partial M-estimator (Hoffmann et al., 2015). Some multivariate methods can also be reformulated in a way that they can be solved via repeated estimation of a (robust) regression model (see, e.g., Waaijenborg et al., 2008; Wilms and Croux, 2015a; Maronna and Yohai, 2008). In Chapter 3, we give a more detailed overview of sparse and robust regression and classification methods that are suitable for high-dimensional datasets and demonstrate their application to a dataset from tribology.

The computation of reliable covariance estimators in the high-dimensional setting is another challenge that has received a lot of attention, as it is needed for many multivariate methods. In general, there are three ways to get a well-conditioned covariance matrix in large dimensions:

1. Regularization: The final estimate is a linear combination of the sample covariance and a target matrix (Ledoit and Wolf, 2004). This could be an identity matrix, but generalizations are possible. Boudt et al. (2020) extended this approach to the MCD, and Ollila et al. (2020) to M-estimators.

2. Thresholding: Under certain structural assumptions, thresholding small values of the covariance matrix yield consistent estimators (Bickel and Levina, 2008; Wainwright, 2019). In combination with robust M-estimators, Avella-Medina et al. (2018) showed similar results for a broader class of distributions.

3. Eigenvalue correction: For computational efficiency, an initial robust covariance estimate can also be derived from pairwise product-moment-correlations (Raymaekers and Rousseeuw, 2021). Then, the final covariance is defined as the "nearest" positive definite matrix in the Frobenius norm. Öllerer et al. (2015) proposed to apply the algorithm by Higham (2002). The OGK estimator (Maronna and Zamar, 2002) also relies on a pairwise identity, and an orthogonalization step ensures positive definiteness of the resulting covariance matrix.

Note that for some multivariate methods, a regularizing penalty on the coefficients, loadings, or directions has an implicit regularization effect on the plug-in covariance estimators. This phenomenon is similar to the effect of a Ridge-penalty in regression and will be discussed for Canonical Correlation Analysis (CCA) in Chapter 4.

1.4 Algorithms

For robust problem formulations, there often does not exist an analytical solution and numerical algorithms are required to get an approximation of the solution. Furthermore, the solution for a number of multivariate problems requires the eigenvalue decomposition

or inversion of the covariance matrix. In the high-dimensional setting, these computations are not possible, and other optimization as well as regularization strategies are required.

Optimization problems arising from multivariate statistical problems often also include constraints, such as the uncorrelatedness with lower-order loadings in Principal Component Analysis (PCA), or the restriction to normed directions.

We give a brief overview of algorithms designed for constrained optimization problems, then we explain how modern gradient-based optimization techniques can be exploited for efficient computation, both for a growing number of observations as well as variables.

1.4.1 Constrained optimization

Consider the optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \tag{1.3}$$

$$\boldsymbol{x} \in \mathcal{X}, \tag{1.4}$$

where $\mathcal{X} \subset \mathbb{R}^p$ and $f : \mathcal{X} \to \mathbb{R}$. The constraint (1.4) can be an equality or inequality, denoted with the help of a function $h : \mathbb{R}^p \to \mathbb{R}^m$. Line (1.4) then becomes $h(\boldsymbol{x}) = \mathbf{0}$ or $h(\boldsymbol{x}) \leq \mathbf{0}$, depending on the application (the equality and inequality are to be understood elementwise). With small tweaks, these two problems can be treated similarly (Bertsekas, 1996; Boyd and Vandenberghe, 2004; Boyd et al., 2011), and for ease of notation, we will continue the introduction only for the case $h(\boldsymbol{x}) = 0$.

The Lagrangian for problem (1.4) is

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \lambda' h(\boldsymbol{x}), \tag{1.5}$$

where λ corresponds to the Lagrange multiplier. We say that $\boldsymbol{x}^*$ is an optimal point of the *primal* problem, if there holds

$$\boldsymbol{x}^* = \operatorname{argmin}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \lambda^*), \tag{1.6}$$

where λ^* maximizes the *dual* function $\inf_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \lambda)$.

Several algorithms use this concept of duality for obtaining a solution to the optimization problem (1.4), which corresponds to a saddle point of the Lagrangian (1.5) and is therefore hard to find numerically. The dual ascent method was one of the first approaches (see, e.g., Boyd et al., 2011, for an overview of the available literature), it consists of alternating the updates of $\boldsymbol{x}$ and the ascent step of the dual variable λ. The update is shown in the left flowchart in Figure 1.7. As this algorithm is potentially unstable and only converges under strict assumptions, the augmented Lagrangian and the corresponding MM (method of multipliers) algorithm have been developed. The augmented Lagrangian for problem (1.4) is defined as

$$\mathcal{L}_c(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \lambda' h(\boldsymbol{x}) + \frac{c}{2}\|h(\boldsymbol{x})\|^2, \tag{1.7}$$

with the parameter c determining the strength of the regularization. Note that the augmented Lagrangian corresponds to the Lagrangian of problem (1.4) with an added penalty

of $\frac{c}{2}\|h(\boldsymbol{x})\|^2$ (Bertsekas, 1996; Boyd et al., 2011). The iterations look very similar to the simple dual ascent method, see the middle plot of Figure 1.7. A disadvantage of this method is that the augmented Lagrangian is not decomposable anymore, even if the original objective function was, i.e. $f(\boldsymbol{x}) = \sum_i f_i(x_i)$. The ADMM (alternating direction method of multipliers) remedies this by suggesting alternating updates in the different directions (Boyd et al., 2011). The steps are shown in the example of two variables in the right part of Figure 1.7. Unless the function f is convex, convergence to a global optimum can unfortunately not be guaranteed, however, the different variants of the algorithm have been successfully applied in many practical applications, as described by Boyd et al. (2011).

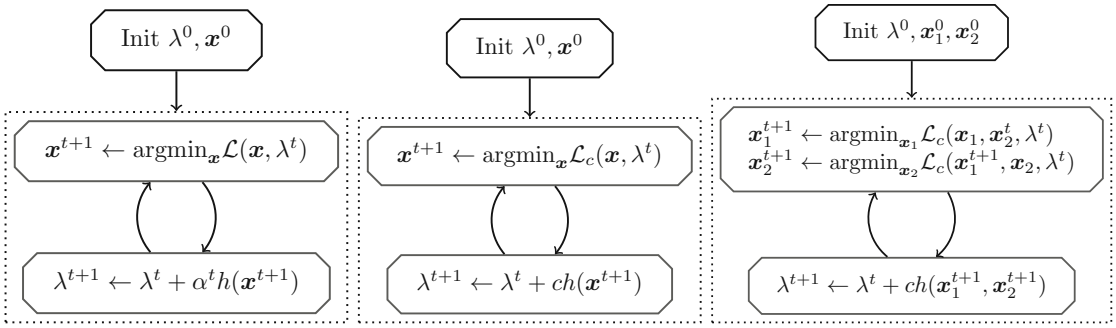


Figure 1.7: Overview of different algorithms exploiting duality. Left: dual ascent, middle: MM (method of multipliers), right: ADMM.

We could also say that the constrained problem (1.4) has been converted into a series of unconstrained problems, and now efficient numerical algorithms can be applied to find the optimum in the $\boldsymbol{x}$ minimization step. In various statistical applications, this corresponds to estimating a series of regression problems (Waaijenborg et al., 2008; Wilms and Croux, 2015a; Maronna et al., 2019), evaluating a number of projection directions (Croux and Ruiz-Gazen, 2005; Alfons et al., 2016a; Croux et al., 2013), or performing coordinate descent (Hastie et al., 2015). When the number of variables grows, these procedures are not sustainable computationally. What is more, the objective function may have multiple local minima, and techniques that make it possible to escape these without needing to evaluate a very large number of initial values. Optimization based on adaptive and stochastic gradient descent is able to combine both: It has been developed to process huge amounts of data for deep learning algorithms, and can also be applied for the efficient computation of robust statistical estimators. In the next section, we give a short introduction to these algorithms, and how they can be modified to work with different types of constraints.

1.4.2 Adaptive stochastic gradient descent

A natural method for finding a minimizing sequence for a differentiable objective function is the gradient method, which takes repeated steps in the opposite direction of the gradient of the function (Boyd and Vandenberghe, 2004). When the entire dataset is used for the computation of the gradient, the method is referred to as deterministic or batch learning, when one sample at a time is used, it is called online or stochastic gradient descent (Goodfellow et al., 2016). Minibatch gradient descent uses a subset of the available samples to approx-

imate the true gradient of a function and provides a compromise between the two. Let θ denote the parameters to be optimized. Then, a minibatch $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ is sampled from the training data, and the gradient estimate is computed as $\hat{\boldsymbol{g}} \leftarrow {}^1/_m \nabla_\theta \sum_i \mathcal{L}(x^{(i)})$. Then, the update step is executed as

$$\theta^{t+1} \leftarrow \theta^t - \epsilon^t \hat{\boldsymbol{g}}, \tag{1.8}$$

with ϵ^t corresponding to the learning rate. The updates are iterated until a convergence condition is reached (Goodfellow et al., 2016).

Boyd and Vandenberghe (2004) summarize theoretical properties of gradient descent algorithms and Goodfellow et al. (2016) provides an overview of popular extensions developed for the use case of big amounts of data and ill-conditioned problems. Two strategies are commonly applied to avoid the model getting caught in local minima: Using momentum (exponentially decaying moving averages of gradients, e.g., Nesterov momentum, Nesterov, 1983), or adaptive learning rates, where the ADAM algorithm by Kingma and Ba (2015) combines both.

Sometimes, it is also useful to apply a learning rate decay, meaning that an initially higher learning rate is decreased according to a specific rule. Many options exist, and popular choices are, for example, linear or exponential decay, adjusting the learning rate in each update step (1.8), see Goodfellow et al. (2016) for an overview.

1.4.3 Gradient descent on manifolds

In Section 1.4.1, the constraints are incorporated with the MM algorithm, but certain types of constraints can be included more naturally. In the case of PCA, for example, we are looking for subspaces described by matrices with orthonormal columns, which corresponds to the Stiefel manifold. When the constraints correspond to a smooth manifold, gradient algorithms can be modified to stay on the manifold during the update. While the procedure has already been proposed by Edelman et al. (1998), extensions to adaptive and stochastic gradient algorithms (Bonnabel, 2013; Bécigneul and Ganea, 2019) make it applicable to more general problems.

In Chapter 5, we describe an approach that uses Riemannian optimization to compute a sparse and cellwise robust PCA estimator. The algorithm is described in more detail there, but the visualization in Figure 1.8 gives an intuition of what happens: The gradient step is divided into first projecting the "naive" gradient onto the tangent space at the current parameter value θ, denoted by $\mathcal{T}_\theta\mathcal{M}$. This could also be interpreted as a first-order approximation of the manifold at θ. After the gradient step is executed in the tangent space, the new parameter value θ' is mapped to the surface of the manifold $\mathcal{M}$ by the exponential map or a retraction $\mathcal{R}$ for improved computational efficiency (Douik and Hassibi, 2019).

1.4.4 Sparsity inducing constraints

So far, we have covered the treatment of differentiable loss functions, but for sparsity-inducing regularization such as the $L1$ norm, this assumption does not hold. Several strategies for the application of gradient descent for constrained non-smooth functions have been developed. In the following, we give a short introduction to a selection.

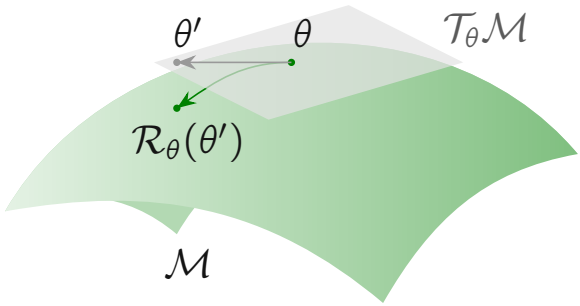


Figure 1.8: Visualization of a gradient step on a manifold.

One possible strategy is the application of proximal operators (Parikh et al., 2014). An example would be the well-known soft-thresholding operator for the $L1$ norm. Generally, this approach works by first applying the gradient step to the differentiable part of the objective function, followed by the projection. This approach is widely used for sparsity-inducing penalties in statistics, such as the LASSO and variants (Hastie et al., 2015). In combination with adaptive gradient techniques, however, its application can be tricky, as averaged moments of previous gradients would need to be updated in a suitable way.

Another option, which is also used in software implementing the backpropagation algorithm and different types of optimizers such as `torch` (Falbel and Luraschi, 2023), is resorting to the subgradient. We use this idea in the algorithm presented in Chapter 4 and give a formal definition there. A visualization of the subgradient of the $L1$ norm is plotted in the left part of Figure 1.9.

Alternatively, it is often also possible to approximate a non-differentiable function by a differentiable one. For the $L1$ norm, we use $|x| \approx x \tanh cx$ in Chapter 5 (also proposed by Öllerer et al., 2015), but other options are available. The functions for different values of c are plotted in the right part of Figure 1.9.

Note that in the case of the $L1$ norm, for both the approach using the subgradient and the approximation with a differentiable function, gradient descent does not yield truly sparse solutions, and a thresholding step is needed. We describe this procedure as part of the proposed algorithms in Chapters 4 and 5.

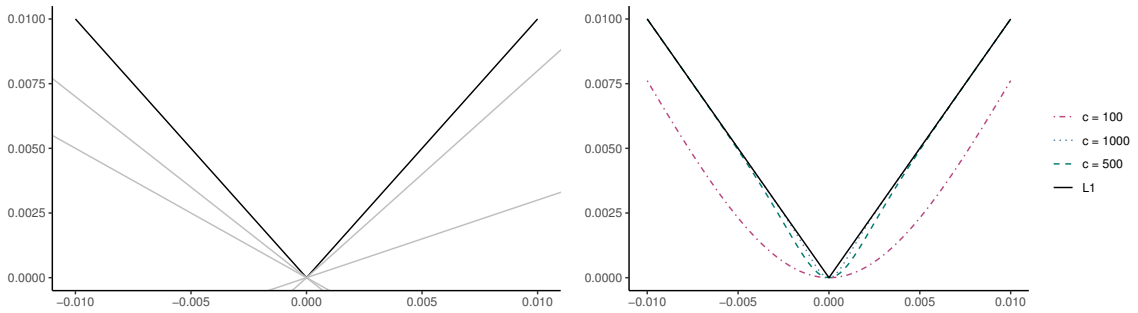


Figure 1.9: Left: Subgradient for L1 penalty. Right: Approximation of L1 penalty with $|x| \approx x \tanh cx$, for different values of c.

1.5 Overview of the contents

In Chapter 2, we present a procedure for the analysis of FTIR spectra from engine oils, we propose a suitable preprocessing method and model the relation between different degradation pathways using weighted LASSO regression models. This chapter has been published as an article in Chemometrics and Intelligent Laboratory Systems, Volume 228, 104617, Pia Pfeiffer, Bettina Ronai, Georg Vorlaufer, Nicole Dörr, Peter Filzmoser, Weighted LASSO variable selection for the analysis of FTIR spectra applied to the prediction of engine oil degradation, Copyright Elsevier (2022). P. Pfeiffer participated in several discussions with the coauthors to develop the idea and methodology. Furthermore, she performed the data analysis, implemented the R code and contributed to the overall writing and editing of the paper as well as the review following discussions and suggestions of the reviewers.

In Chapter 3, we give an overview of robust statistical methods for regression and classification and discuss their applicability in the high-dimensional setting. We demonstrate the application of those methods on the example of FTIR spectra and image data from tribology, illustrating the benefits of robustness. This chapter is a reprint of an article published in Analytica Chimica Acta, Volume 1279, 341762, Pia Pfeiffer, Peter Filzmoser, Robust statistical methods for high-dimensional data, with applications in tribology, Copyright Elsevier (2023). P. Pfeiffer participated in discussions to develop the ideas, performed data analysis and contributed to the writing and editing of the paper.

Chapter 4 considers robust and sparse maximum association estimators, a generalization of CCA. It studies how challenges in the computation of robust estimators can be overcome by applying efficient algorithms based on gradient descent. A simulation study comparing the proposed method to competitors indicates superior performance, and high-dimensional empirical examples are analyzed to underline the usefulness of this approach. This chapter is available as a preprint on arXiv: Pfeiffer, Pia, Andreas Alfons, and Peter Filzmoser. Efficient Computation of Sparse and Robust Maximum Association Estimators. arXiv preprint arXiv:2311.17563 (2023). P. Pfeiffer participated in several discussions with the coauthors to develop the methodology. She implemented the R code, conducted the simulation study, and performed the data analysis. She also contributed to the overall writing and editing of the paper.

Chapter 5 considers cellwise robust and sparse PCA based on low-rank matrix approximation. An algorithm based on Riemannian gradient descent for the resulting optimization problem is presented, and the superiority of this approach in comparison with existing methods, both in the cellwise and casewise setting, is shown in a simulation study. An application to two datasets from tribology illustrates the effectiveness of the proposed method.

In Chapter 6, we give an overview of the R package `RobSparseMVA` (Pfeiffer et al., 2024) that has been implemented for the methods in Chapters 4 and 5, demonstrating the usage of the most important functions, analysis of the results, and possibilities for customization and tuning of the models.

Finally, Chapter 7 summarizes the contributions and gives an outlook on future research topics.

2 Weighted LASSO variable selection for FTIR spectra

This chapter was published in Chemometrics and Intelligent Laboratory Systems, Volume 228, 104617, Pia Pfeiffer, Bettina Ronai, Georg Vorlaufer, Nicole Dörr, Peter Filzmoser, Weighted LASSO variable selection for the analysis of FTIR spectra applied to the prediction of engine oil degradation, Copyright Elsevier (2022).

2.1 Introduction

FTIR (Fourier-transform infrared) spectroscopy in combination with chemometric methods is used in various fields: Applications range from differentiation of document paper types in forensics and the analysis of ingredients in terms of quality or adulteration in food chemistry and pharmaceutics to oil condition monitoring in tribology. In tribology, lubrication plays a crucial role to reduce wear and control friction in machinery, e.g., engines and gears (Mang and Dresel, 2017). For maintenance of machinery, oil condition monitoring describes the health status of the lubricant but also the lubricated system by monitoring of one or more critical parameters to identify a significant change that is indicative of a developing fault, e.g., temperature, acids, water, and viscosity (Whitby, 2021). In R&D of lubricants, condition monitoring provides valuable information about the progress of the lubricant‘s degradation over time or, to express it differently, about its stability. Several authors investigate how spectroscopic methods can be applied for the efficient assessment of used engine oils, especially for the prediction of oil attributes such as Viscosity Index (VI), kinematic viscosity, Total Acid Number (TAN) or Total Base Number (TBN) (Felkel et al., 2010; Hirri et al., 2017; Macian et al., 2020; Al-Ghouti et al., 2010), and oil adulteration (Bassbasi et al., 2013). Wolak et al. (2021) model the mileage of engine oils used in cars as a response to the band area of selected FTIR absorption bands, and Sejkorová (2017) develops a PLS (Partial Least Squares) regression model to predict diesel contamination in engine oil. Moreover, Besser et al. (2013) compare engine oils altered in the laboratory and in a chassis dynamometer by their FTIR spectra using PCA (Principal Component Analysis). Recently, there have also been advances in the production of "used" engine oils in the laboratory (Agocs et al., 2020) and the analysis of degradation patterns in field tests with passenger cars (Agocs et al., 2021).

Artificial alteration methods offer a great benefit: they allow to generate degraded oils under laboratory-controlled conditions in small to large quantities (Besser et al., 2019) and in relatively short time. Exemplarily, the condition of an engine oil that is in use for a year or typically 15 000 km can be reproduced in the laboratory in a week. Furthermore, control of artificial alteration enables the production of "used" oil at a defined degree of degradation. However, the choice of artificial alteration parameters and duration is currently

done empirically based on experts' knowledge. Therefore, a quantitative association between alteration parameters, time and mileage in a real car, for example, is highly desirable.

We develop an analysis pipeline for FTIR spectroscopic data that allows to quantify the relationship between different series of FTIR spectra, and apply these methods to understand how field use and artificial alteration of engine oils are associated. In contrast to other approaches, we do not aim to predict defined attributes of the oils but relate the oil condition based on the runtime of artificial alteration methods and mileage of a real car, respectively. While FTIR absorption bands that are suitable for the analysis are typically manually selected (Macian et al., 2020; Bassbasi et al., 2013; Wolak et al., 2021, see, for example,), we present a pre-processing method that can filter non-informative variables objectively. Moreover, interpretation of the results is simplified, as the applied regression method yields sparse results, i.e., only a small number of variables is selected for the model.

IR spectroscopy in general is based on the excitation of vibrations and rotations in molecules by infrared radiation and primarily provides information about the functional groups of molecules present. Evaluation of characteristic absorption bands enables a qualitative and quantitative identification of engine oil components like base oil, additives, and their degradation products, as well as contaminations like fuel, water, or soot. FTIR spectroscopy presents a quick and powerful analytical technique that reveals valuable information about the oil composition and condition (Wolak et al., 2020).

FTIR spectra are high-dimensional data: the number of variables (wavenumbers) is usually much larger than the number of observations, as is the case with the given dataset. However, it can be assumed that only a certain number of variables contribute to explaining oil degradation, which motivates the use of variable selection methods to simplify the interpretation of the coefficients.

On a high level, the common procedure(see, for example, Felkel et al., 2010; Macian et al., 2020; Al-Ghouti et al., 2010; Bassbasi et al., 2013) applied when regressing on FTIR spectra consists of the following steps:

1. manual selection of intervals of wavenumbers that are known to be important
2. removal of non-informative variables: either manual or by application of filters
3. exploratory data analysis using PCA
4. application of regression methods, such as PLS regression

This procedure can be quite subjective and lacks reproducibility.

The proposed data analysis method consists of two steps. In the first step, two types of pre-processing are performed: an automatic procedure to remove non-informative variables based on the reconstruction error from PCA, and a baseline correction of the FTIR spectra to ensure comparability among different methods. Then, the LASSO (least absolute shrinkage and selection operator) regression estimator (Tibshirani, 1996) with inherent variable selection is applied, and measures of variable importance are retrieved using post-selection inference, introduced by Lee et al. (2016). By using a weighted version of the LASSO as described by Hastie et al. (2015), expert knowledge can be integrated with the mathematical model.

The remainder of the paper is organized as follows: The proposed analysis pipeline is demonstrated through the example of a dataset of series of FTIR spectra measured on used and artificially altered engine oils, a detailed description of the data is given in Section 2.2.1. Then, the pre-processing steps involving an automatic filtering method are explained in Section 2.2.2, followed by a detailed model description in Section 2.3: Variable selection and suitable statistical tools for inference are discussed in Sections 2.3.1 and 2.3.2 before different models are estimated. The results are analyzed and interpreted in terms of lubricant chemistry in Section 2.4.2 and the best models applied to relate different degradation methods in Section 2.4.3. Eventually, an outlook for the application of proposed analysis methods for tribological research is given in Section 2.5.

2.2 Dataset

The dataset used in this work contains the FTIR spectra of automotive engine oils in different conditions. The underlying engine oil is a commercially available SAE 5W-30 engine oil that meets the specifications of ACEA C3 and API SN. FTIR spectra, elemental analysis (detecting e.g., Zn, P, S, Ca) and other conventional analyses indicate the application of additives commonly used in automotive engine oils, like ZDDP (zinc dialkyldithiophosphates), antioxidants, detergents with a base reserve, and dispersants. Two batches of this engine oil were taken as a fresh oil and were subjected to three different treatments:

(a) an artificial small-scale alteration (duration of 288 h, 11 samples)

(b) an artificial large-scale alteration (duration of 143 h, 24 samples)

(c) a field test consisting of two oil change intervals (mileage of 19 800 km, 21 samples)

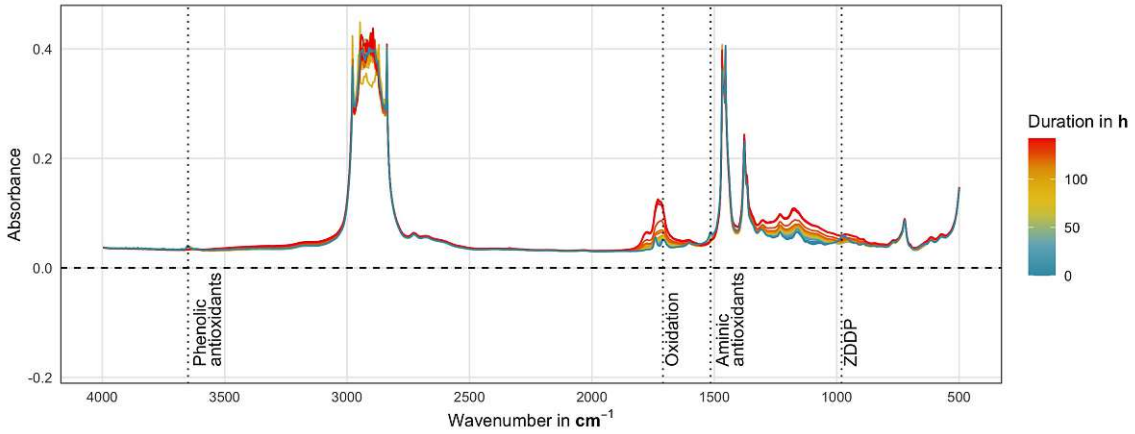
2.2.1 Data source

Generally, an artificial alteration is used to achieve an accelerated degradation of a lubricant in the laboratory, in order to obtain an oil condition that is close to reality. The two artificial alteration methods performed in this study were both thermo-oxidative degradations, i.e., involving thermal and oxidative stress.

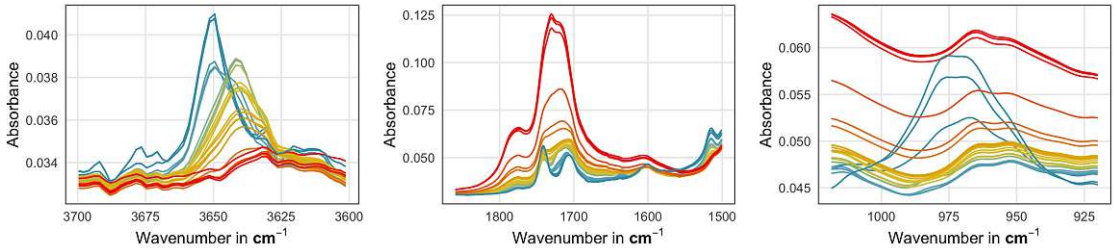
The small-scale alteration was carried out on a laboratory scale according to Besser et al. (2012). Here, 300 g of engine oil contained in a round bottom flask were placed in a heating bath at 180 °C, with a dried air flow of 10 L/h being applied to the oil. Sampling took place in regular intervals during the total duration of 288 h, yielding 11 artificially altered small-scale samples.

The principle of the large-scale alteration is based on the mentioned small-scale method. The device used to artificially alter 100 L of engine oil at a temperature of 180 °C and with a dried air flow of 2160 L/h is described in detail by Besser et al. (2019). The total alteration duration was 143 h, producing 24 samples in total. The alteration that provided the large-scale data used in this work is also described by Agocs et al. (2020).

In the field test, the engine oil was used in a conventional passenger car with a modern 4-cylinder turbocharged internal combustion engine of 1.4 liter displacement powered by petrol. For the duration of the field test, the car was mainly used for commuting and thus



(a) Original FTIR spectra: large-scale artificial alteration series



(b) (c) (d)

Figure 2.1: Plots of FTIR spectra of large-scale alteration series colored according to duration. For regions of interest, a zoomed view is provided: (a) Phenolic antioxidants, (b) Oxidation, ester, aminic antioxidants and (c) ZDDP (zinc dialkyldithiophosphates).

mostly operated on freeways. "Field 1" refers to the first run, where the oil was in use for 19 800 km, and "Field 2" describes the second run, where after an oil change the oil was used for 10 800 km. During the field test, which is presented by Agocs et al. (2020) and Dörr et al. (2019a), samples were taken regularly via the engine oil dipstick tube, providing a total of 21 used oil samples.

FTIR spectra were recorded of all 58 samples (including the 2 fresh oil batches). Each FTIR spectrum contains the absorbances at 1814 wavenumbers (variables) in the range of $3997 - 500$ cm^{-1}. Figure 2.1a shows the FTIR spectra of the large-scale alteration series before pre-processing methods are applied, Figures 2.1b - 2.1d provide a zoomed-in view of the absorption bands that reveal important information for the evaluation of the oil condition.

2.2.2 Preprocessing: filtering of non-informative variables and baseline correction

FTIR spectra, such as those shown in Figure 2.1a, represent typical IR spectra of automotive engine oils. The regions that can be seen around $3030 - 2770$ cm^{-1} and $1480 - 1430$ cm^{-1} are areas of high or total absorption. The absorption bands found here are typical for the C-H stretching and bending vibrations of hydrocarbons. Since hydrocarbons form the basis of engine oils, they are always present to a large extent. Hence, they do not only cause total absorption, but also do not provide any relevant information and are usually not considered in evaluation. In the context of statistics, these regions of an FTIR spectrum can therefore be considered as non-informative variables.

For automatic identification of these non-informative variables, i.e., variables that do not contribute to a model, we propose to use an approach based on reconstruction error. PCA is performed on the scaled and centered data, then the data is reconstructed from the number of principal components needed to explain 95% of the variance. Variables that are characterized by a higher reconstruction error are assumed to not contain relevant information; these wavenumbers are candidates for removal. An R implementation for the pre-processing method is available at `https://github.com/piapfeiffer/FTIR-filtering`.

Let $X = (x_i)_{i=1}^n$ denote the centered and scaled data matrix, collecting n observations for a p-dimensional vector of features $x_i = (x_{i1}, \ldots, x_{ip})$. The p columns of X correspond to the wavenumbers, the n rows contain the absorbances of the respective wavenumbers for every observation. PCA, described in more detail for example in Anderson (1958), represents data by linear combinations of specific components, resulting in the matrix Z of principal components. The linear transformation $Z = X\Gamma$ is constructed such that the variance of the columns of Z is maximized. Furthermore, Γ is an orthogonal matrix ($\Gamma^T = \Gamma^{-1}$) and its columns γ_j are unitary vectors ($\gamma_i^T\gamma_i = 1$ and $\gamma_i^T\gamma_j = 0$ if $i \neq j$). The original data matrix can be reconstructed from the score matrix Z and the loadings matrix Γ via the identity $X = Z\Gamma^T$. Let $z_j, j = 1, \cdots, p$ denote the column vectors, or components, of Z. We can describe the proportion of variance explained by the first k components as

$$\mathrm{Var}_{\mathrm{explained}}^{(1:k)} = \frac{\sum_{j=1}^{k} \mathrm{Var}(z_j)}{\sum_{i=1}^{p} \mathrm{Var}(z_i)} \tag{2.1}$$

We can now choose k such that $\mathrm{Var}_{\mathrm{explained}}^{(1:k)} \geq 95\%$ and approximate the data matrix using only the first k components of the scores $Z_{(1:k)}$ and loadings $\Gamma_{(1:k)}$ matrix for reconstruction:

$$\hat{X}_{(1:k)} = Z_{(1:k)}\Gamma_{(1:k)}^T \tag{2.2}$$

A plot of the centered and scaled data series is given in Figure 2.2. It can be observed that apart from the bands with total absorption, there is a logical progress over duration, this variation can be modelled by the first k principal components, whereas other effects are not included. When the data matrix is reconstructed using only a part of the components, the reconstruction error is higher at these absorption bands.

The reconstructed data is back-transformed to its original scaling and the mean reconstruction error (MRE) is computed for the j-th variable (wavenumber), $j = 1, \ldots, p$ by

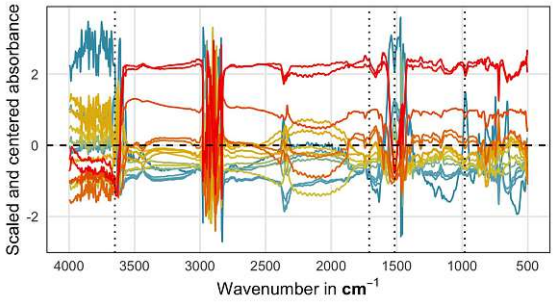


Figure 2.2: Data preparation for filtering procedure: FTIR spectra (large-scale alteration) are scaled and centered.

taking the average over the squared differences between the original and reconstructed data matrices for the observations:

$$\text{MRE}_j = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \hat{x}_{ij})^2 \tag{2.3}$$

In order to achieve a small number of intervals, we propose to smooth the mean reconstruction error using the Nadaraya-Watson kernel-weighted average, as described in Hastie et al. (2009):

$$S(\text{MRE}_j) = \frac{\sum_{l=1}^{p} K_\lambda(j,l)\text{MRE}_l}{\sum_{l=1}^{p} K_\lambda(j,l)} \tag{2.4}$$

with $K_\lambda(j,l) = D\left(|j-l|/\lambda\right)$ and $D(t) = 3/4(1-t)^2$ if $|t| < 1$ and 0 otherwise. K_λ defines the kernel function and λ controls the width of the local neighborhood. A plot of the smoothed MRE is given in the right plot of Figure 2.3. The threshold to distinguish between informative and non-informative variables is given by the 95% percentile of the mean reconstruction error sample distribution and is shown as a red line. For the subsequent analyses, the variables characterized by an MRE above the 95% percentile were deleted - these default values were found to work well for the presented analysis. However, several parameters can be tuned: the number of components used for reconstruction, the smoothing function, but also the threshold determining the cutoff value.

After this filtering procedure, a baseline correction is applied to the original FTIR spectra using the standard rubberband-method (Prizer and Sawatzki, 2008; Wartewig, 2006). Due to the nature of the experiments, there is a baseline shift mostly caused by the presence of soot for the oils used in the field. Normalizing the baseline ensures that spectra can be compared among different pathways of degradation.

To base the models on the changes during oil degradation and as an approach to make the process more generalizable, spectral subtraction was utilized. Difference spectra were obtained by subtracting the appropriate FTIR spectrum of the fresh oil from those of the degraded oils. A plot of the resulting dataset is given in Figure 2.3.

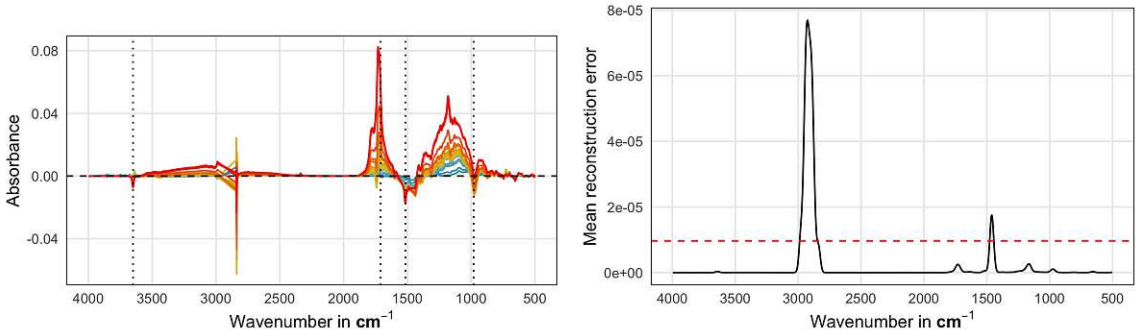


Figure 2.3: Dataset after pre-processing: The difference spectra for the large-scale alteration are shown in the left plot. In the right plot, the smoothed MRE (mean reconstruction error) including a threshold to filter non-informative variables is shown.

2.3 Model

Let us now consider the linear regression model. Given n samples $(x_i, y_i)_{i=1}^n$, where each $x_i = (x_{i1}, \ldots, x_{ip}) \in \mathbb{R}^p$ is a p-dimensional vector of features, the p columns again corresponding to the wavenumbers, and $y_i \in \mathbb{R}$ is the respective response, given in runtime (h) for artificial alteration and mileage (km) for use in the field. For simplicity, the number of variables is again denoted by p, now referring to the filtered set of predictors. The multiple linear regression model reads: $y = X\beta + \varepsilon$, where X is the $n \times (p+1)$-dimensional design matrix collecting a vector of ones (for the intercept) in the first column and the data $(x_i)_{i=1}^n$. As the regressors usually do not describe the response y perfectly, an additive error term ε is added. The least squares estimator is based on minimizing the residual sum of squares: $\hat{\beta}_{LS} = \text{argmin}_{\beta \in \mathbb{R}^p} ||y - X\beta||_2^2$, which is not well-defined in the case $p > n$. There are several approaches to address this: PLS, which is done by regressing on latent variables, as well as penalized approaches such as LASSO (Tibshirani, 1996) or ridge regression. The respective estimates are given by

$$\hat{\beta}_{LASSO} = \text{argmin}_{\beta \in \mathbb{R}^p} ||y - X\beta||_2^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.5}$$

$$\hat{\beta}_{ridge} = \text{argmin}_{\beta \in \mathbb{R}^p} ||y - X\beta||_2^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{2.6}$$

Elastic net Zou and Hastie (2005) combines the LASSO and ridge penalties to reduce the number of variables. PLS constructs latent variables w_k according to the criterion $w_k = \text{argmax}_w \text{cov}(y, Xw)$, which means the covariance between the response y and a linear combination of the input variables is maximized. Regression is then performed on the scores $t_k = Xw_k$, for $k = 1, \ldots, K$, where $K < p$ is an appropriate number of PLS components. The resulting regression coefficients can be back-transformed to be interpreted in terms of the original variables. However, one must be careful with interpretation and significance estimates for PLS results. Generally, interpretation is easier if the resulting model is sparse,

i.e., only few variables are used for the final model (see, for example, Varmuza and Filzmoser, 2009). Sparse PLS (SPLS) as described by Chun and Keleş (2010) combines the PLS approach with regularization techniques. Still, inference can only be done using bootstrap, and there is no straight-forward possibility to integrate domain-experts' knowledge with the models.

2.3.1 Variable selection with the (weighted) LASSO

In addition to being hard to interpret, a model using many variables in relation to observations is prone to overfitting. This means a good fit is achieved on training data, but the model is not able to generalize well and results in low prediction performance for unseen data. Hence using all available variables in a model is not advisable, even though methods like PLS can handle high-dimensional data. Exhaustive search of all possible subsets is not feasible, but there are several approaches to variable selection. In general, the aim is to find a (small) subset of predictors that is best for prediction. For example, this can be done by excluding variables with low potential for prediction, e.g., almost constant or outlying variables, or searching for variables with high predictive potential, e.g., variables with high variance. Other strategies are stepwise selection or models based on latent variables derived using PCA or PLS. A practical overview of available strategies is given in Varmuza and Filzmoser (2009), for example. Penalized regression estimators such as LASSO Tibshirani (1996) and elastic net Hastie et al. (2009) are also very popular methods for variable selection, as variable selection is inherent to the estimation process and the resulting models are sparse, i.e. only a small number out of all predictors is selected.

We address the objectives of variable selection and parameter estimation simultaneously by applying the weighted LASSO regression method (Hastie et al., 2015). The objective function reads:

$$\hat{\beta}_{wLASSO} = \text{argmin}_{\beta \in \mathbb{R}^p} ||y - X\beta||_2^2 + \lambda \sum_{j=1}^{p} \gamma_j |\beta_j| \tag{2.7}$$

where λ controls the complexity of the solution (number of non-zero coefficients) and γ_j allows to modify the penalty for single coefficients to account for expert knowledge. For standard LASSO, $\gamma_j = 1$ for all j.

Adaptive LASSO was introduced by Zou (2006) and can be considered a special variant of the weighted LASSO: Let $\hat{\beta}^{(0)}$ denote an initial estimate for $\hat{\beta}$, computed using ordinary LS in the standard case ($n > p$) or ridge regression in the case $p > n$. The penalty modification can then be defined as $\gamma_j = 1/|\hat{\beta}_j^{(0)}|^\alpha$ for an $\alpha > 0$.

In the case $\gamma_j \neq 0$, the penalty modification directly corresponds to a rescaling of the input data if the resulting coefficients are also scaled accordingly. To see this, we re-express the objective function (2.7) in terms of the KKT-conditions as described in Tibshirani

(2013). Then, β is a solution to (2.7) if the following equations are satisfied:

$$-\sum_{i=1}^{n} x_{ij} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right) + \lambda\gamma_j s_j = 0 \tag{2.8}$$

$$-\sum_{i=1}^{n} \frac{x_{ij}}{\gamma_j} \left(y_i - \sum_{j=1}^{p} \frac{x_{ij}}{\gamma_j} \underbrace{(\beta_j\gamma_j)}_{\tilde{\beta}_j} \right) + \lambda s_j = 0 \tag{2.9}$$

where the subgradient s_j is defined as the sign of β_j, if $\beta_j \neq 0$, or between -1 and 1 otherwise. $\tilde{\beta}$ is the solution to the rescaled problem and can be computed as for the LASSO without penalty modification. The original coefficients can be recovered by rescaling with the penalty vector γ: $\beta_j = \tilde{\beta}_j/\gamma_j$.

As certain regions of an FTIR spectrum of an engine oil are known to provide more specific and interpretable chemical information than others, it is reasonable to include this knowledge in the model building considerations. Some of the absorption bands that are usually evaluated (see Besser et al. (2019)) and reveal essential information about the engine oil condition are the ones of oxidation, antioxidants and ZDDP. Oxidation occurs at elevated temperatures in the presence of atmospheric oxygen and leads to the formation of undesirable degradation products. Absorption bands of oxidation products like ketones, aldehydes, carboxylic acids and esters can be seen in the region of $1860 - 1660$ cm^{-1} (see Figure 2.1c). Antioxidants are additives that protect the engine oil against oxidation. The absorption band around 3650 cm^{-1} (see Figure 2.1b) is typical for phenolic antioxidants, while the one around 1515 cm^{-1} (see Figure 2.1c) is formed by aminic antioxidants. ZDDP is a widely used anti-wear and extreme-pressure additive that protects metal surfaces and reduces wear in the engine. Its typical absorption band can be found in the region of $1050 - 900$ cm^{-1} (see Figure 2.1d).

Based on this knowledge, several configurations of the LASSO were estimated and compared. A penalty modifier with $\gamma_j = 0.1$, reducing the penalty for the respective coefficients by 90%, is applied according to the configurations listed below. The exact value of the penalty modifier was not found to have a big impact on the selected model. In addition, the penalty modification derived from an initial ridge estimate, i.e. adaptive LASSO, with $\alpha = 1$ was considered. For comparison, a PLS and SPLS regression for achieving a baseline result were estimated as well. The first model, configuration 0, does not include a penalty modifier. The other configurations are defined as follows:

1. all regions of interest
2. ZDDP, wavenumbers $990 - 950$ cm^{-1}
3. phenolic antioxidants, wavenumbers $3651 - 3649$ cm^{-1}
   aminic antioxidants, wavenumbers $1516 - 1514$ cm^{-1}
4. oxidation, wavenumbers $1860 - 1660$ cm^{-1}
5. combination oxidation + antioxidants

6. combination ZDDP + antioxidants

7. combination ZDDP + oxidation

Each of the configurations emphasizes different absorption bands of the FTIR spectra by applying the penalty modifier γ_j exclusively to the respective absorption bands. The aim of this setup is to investigate which of these regions can contribute most to achieve good prediction performance and therefore explain the degree of degradation.

2.3.2 Inference and reliability

For the ordinary least-squares estimator there exist statistical tests to evaluate the importance of single variables. The same task is much more difficult when variable selection is inherent to the estimation process on the same data, as this means the target, the coefficients that are estimated, are changing with the selected model (Berk et al., 2013). Furthermore, LASSO in general does not have oracle properties: According to Fan and Li (2001), an oracle procedure is defined as a method that is able to identify the right subset of variables, or the underlying true model, and in addition fulfills an optimal estimation rate. For LASSO, Meinshausen and Bühlmann (2006) have discussed the conflict of consistent variable selection and optimal prediction and Zou (2006) has shown that non-trivial conditions need to be fulfilled for consistent variable selection. Let β_j^M denote the coefficient of the j-th variable in model M and C_j^M the respective confidence interval for β_j^M. If variable selection is not consistent, we cannot compare β_j^M and $\beta_j^{M'}$ over different models M and M', as β_j^M is only defined if the j-th variable is selected for model M. Therefore, the probability $P(\beta_j^M \in C_j^M) \geq 1 - \alpha$ may not be defined (Lee et al., 2016). One alternative to this "traditional" construction of confidence intervals is data splitting and bootstrapping as discussed in Meinshausen et al. (2009); Dezeure et al. (2015), but is not feasible in the given context with only a handful of observations. We therefore use the concept of post-selection inference introduced by Berk et al. (2013), which allows for statistical inference for any type of variable selection, including the weighted LASSO. Based on this concept, Lee et al. (2016) construct intervals for the LASSO coefficients by conditioning on the selected model, i.e. $P(\beta_j^{\hat{M}} \in C_j^{\hat{M}} | \hat{M} = M) \geq 1 - \alpha$, and characterizing the selection event of the LASSO as a union of polyhydra. After model selection, their implementation in the R package "selectiveInference" Tibshirani et al. (2019) was used to derive p-values and confidence intervals for the selected variables and coefficients.

2.3.3 Estimation

The different model configurations are trained on the data series generated by the artificial large-scale alteration. This dataset is split into training (2/3) and test (1/3) data randomly, which are scaled and centered. The response is transformed to (approximate) normality using a power transform of 2/3. Then the optimal value for λ is selected using 5-fold cross-validation on the training set as implemented in the R package "glmnet" Friedman et al. (2010b). Then the predictive ability of the resulting models is compared using the mean

squared error of prediction (MSEP):

$$\text{MSEP} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.10}$$

The value of λ resulting in minimum MSEP is denoted as λ_{min}, the model resulting from this parameter is then used for the evaluation on the test set. For estimation of the PLS model, the R package "pls" Liland et al. (2021) is used, the optimal number of components is found using cross-validation as implemented in the package. Then, the predictive performance of this PLS model is evaluated on the test set. Similarly, for SPLS the R package "spls" Chung et al. (2019) is used, the optimal regularization strength and number of components is estimated using cross-validation as implemented in the package. The predictive performance of the model is again evaluated on the test set.

2.4 Results and discussion

The results for different model configurations are given in Table 2.1, the best models are highlighted: Configurations 1, 3 and 5 reach a low prediction error during estimation, as well as on the test set. For configuration 4, we are also able to achieve a good prediction during the estimation process, but the error on the test set is much larger, potentially due to overfitting, but it might also be an indicator for the presence of outliers in the test set as the performance during estimation seems to be consistent. For configurations 1 and 2, the error is not stable during estimation, potentially due to inconsistencies. The results of adaptive LASSO are comparable to weighted LASSO based on experts' knowledge. Depending on the configuration, LASSO clearly leads to an improvement compared to PLS. For the investigation of the association between different oil degradation models, the best LASSO configurations (1, 3 and 5) and the PLS model are selected. As configuration 1 and 5 result in the same model, i.e., the same wavenumbers are selected (see also Figure 2.4), adaptive LASSO (as the next best in stable training and good test performance) is also selected.

2.4.1 Computational results

An overview of the resulting coefficients for λ_{min} including a comparison to the PLS coefficients using two components is given in Figure 2.4. Empirical studies (see, for example, Zou and Hastie, 2005) show that in the case of high-dimensional data, as FTIR spectra, the LASSO estimator selects only one predictor in a group of correlated variables. Keeping this in mind, we interpret a selected wavenumber as a representative of a variable group: Wavenumbers associated with oxidation, for example, are in the range of $1860 - 1660$ cm^{-1} and neighboring variables are highly correlated. The selection of any wavenumber in this interval therefore points to a contribution of oxidation to explaining oil degradation. Among all models we can observe strong negative contributions of the absorption bands related with phenolic and aminic antioxidants, as well as positive contributions from the region associated with oxidation processes.

Model	MSEP (train)	MSEP (test)
PLS	0.052	0.055
SPLS	0.014	0.027
Config 0	0.200 ± 0.189	0.006
Config 1	**0.005 ± 0.002**	**0.011**
Config 2	0.230 ± 0.017	0.203
Config 3	**0.020 ± 0.005**	**0.008**
Config 4	0.016 ± 0.004	0.027
Config 5	**0.005 ± 0.002**	**0.012**
Config 6	0.003 ± 0.001	0.025
Config 7	0.017 ± 0.006	0.014
Adaptive LASSO	**0.009 ± 0.003**	**0.017**

Table 2.1: Comparison of performance for difference model configurations on the large-scale alteration series, split into training and test set. For (weighted) LASSO, the error for λ_{min} is given.

The advantage of a highly sparse solution is apparent in direct comparison to the PLS and SPLS coefficients: While single wavenumbers act as a representative for the underlying processes and can be easily interpreted for the different LASSO models, PLS regression results in a contribution of all available variables. Regions selected by SPLS coincide with wavenumbers with high contribution for the PLS model. While SPLS is able to reduce the number of variables, there are still more than 1000 variables left in the model. The LASSO configurations, in contrast, select only a small number of variables. Moreover, the different LASSO models agree on certain regions as being important for prediction, even if no emphasis (in form of a penalty modifier) is put on them.

2.4.2 Interpretation of coefficients

Classical LASSO (configuration 0) already selects some of the variables in regions of the FTIR spectrum that are important for the description of the engine oil condition, like that of the phenolic and aminic antioxidants. Moreover, said variables mostly coincide with the respective band maximum, where also conventional evaluation would be performed and where penalty modifiers were applied in other configurations. Penalty modifiers that are automatically determined by the adaptive LASSO also coincide with some of the regions that would be considered for conventional analysis. Similar to classical LASSO, some of these selected variables correspond to the respective band maximum.

Adding penalty modifiers based on lubricant chemistry knowledge enhances the model in terms of MSEP in most cases. Especially the configurations where the regions of antioxidants are penalized less, result in lower prediction errors and thus better models. Even though there is a shift of the phenolic antioxidants' absorption band towards slightly smaller wavenumbers in the course of degradation (see Figure 2.1b), the absorption at 3650 cm^{-1} seems to contribute to good models.

The signs of the coefficients represent how an absorption band in the FTIR spectrum is

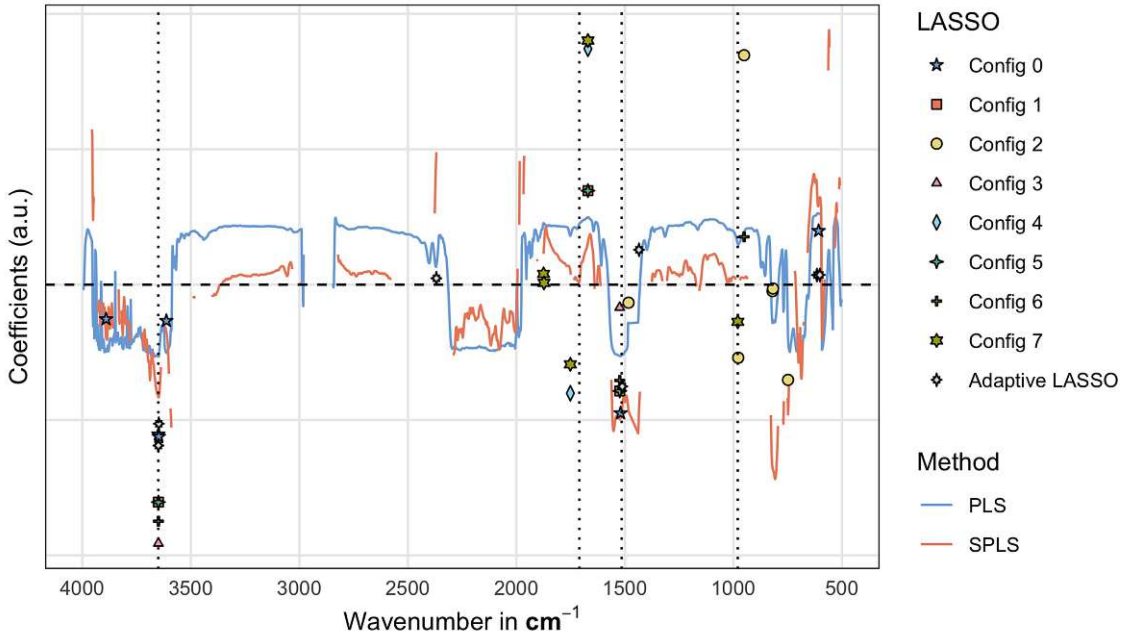


Figure 2.4: Plots of selected coefficients for different models.

associated with oil degradation, in detail whether an increase or decrease of the respective absorption band is related with an increase in oil degradation. During engine oil degradation, additives like antioxidants are consumed or decomposed and therefore their absorption bands decrease. This is reflected in the negative sign of the coefficients of the variables 3650 and 1514 cm^{-1}. In comparison, the coefficients of variables that represent oxidation, have a positive sign, meaning that an increase at the respective absorption bands corresponds to an increase of oil degradation. The coefficients of variable 1742 cm^{-1}, which represents the absorption band of esters, show a negative sign. This absorption band, which can be seen in Figure 2.1c, decreases at the beginning of the alteration because esters that are present in the fresh engine oil can be decomposed by degradation processes, before it increases again due to the formation of oxidation products. The selected variables around the ZDDP absorption band have coefficients of both negative and positive sign. This can result from the combination of rapid ZDDP degradation (Dörr et al., 2019b) and the formation of other compounds that takes place in that region of the FTIR spectrum (see Figure 2.1d). An interesting observation is that the variables chosen by LASSO in the region of oxidation are located on the fringe of absorption bands. The conventional evaluation of oxidation values is usually carried out at 1710 cm^{-1} DIN (Deutsches Institut für Normung) (2004) or 1720 cm^{-1} (see Besser et al., 2019), while LASSO selects the wavenumbers 1865, 1863 and 1661 cm^{-1}. They are located on the left fringe of the absorption band at 1780 cm^{-1}, and on the right fringe of the one at 1710 cm^{-1}, respectively.

Table 2.2 summarizes the results concerning significance of variables that were computed using the R package `selectiveInference` (Tibshirani et al., 2019): The significant coefficients correspond to the wavenumbers 3650, 1661, 1742 and 1514 cm^{-1} which are therefore

Model	variables	confidence interval
Config 1	3650	[-0.727, -0.300]
	1661	[0.166, 0.307]
	1514	[-0.459, -0.083]
Config 3	3650	[-1.266, -0.734]
	1514	[-0.262, 0.276]
Config 5	3650	[-0.727, -0.300]
	1661	[0.166, 0.307]
	1514	[-0.459, -0.083]
Adaptive LASSO	3650	[-0.956, 0.285]
	3648	[-1.279, -0.038]

Table 2.2: Confidence intervals for selected variables. The signs of the coefficients correspond to how the respective FTIR absorption bands are associated with oil degradation.

especially important for the prediction of the degree of oil degradation. Penalty configurations 1 and 5 result in the same model and significance estimates, while configuration 3 also selects a wavenumber in the absorption band characteristic for esters. The signs for these variables again correspond to the underlying chemical processes.

2.4.3 Relating different degradation pathways

The top three models estimated using the large-scale alteration series in Section 2.3.3 are used to quantify the relation between different degradation pathways. These results are then interpreted and qualitatively assessed. For reference, the result using the SPLS estimator is included as well.

Figures 2.5a - 2.5d visualize the relationship between field use and artificial large-scale alteration. Using the SPLS model and the best three LASSO configurations as given in Table 2.1, the field data is given as input and the duration in terms of the artificial large-scale alteration is predicted. Figures 2.6a - 2.6d visualize the relationship between artificial small-scale and large-scale alteration methods. It can be observed that the prediction for the field data based on the SPLS model (Figure 2.5a) is not as smooth as the curves based on different LASSO configurations (Figures 2.5b - 2.5d). While the SPLS model is able to map the order of degraded oils, it does not cover the same value range as the predictions based on the other models. For the prediction of small-scale alteration, on the other hand, the performance of SPLS (Figure 2.6a) covers a similar range as the LASSO configurations (Figures 2.6b- 2.6d), but the shape of the predictions differ. SPLS (without additional variable selection measures) is much more vulnerable to overfitting (see, for example, Varmuza and Filzmoser, 2009) which explains a worse performance of the model for a more general use case, while the prediction of a similar alteration procedure (artificial small-scale vs. large-scale) yields reasonable results.

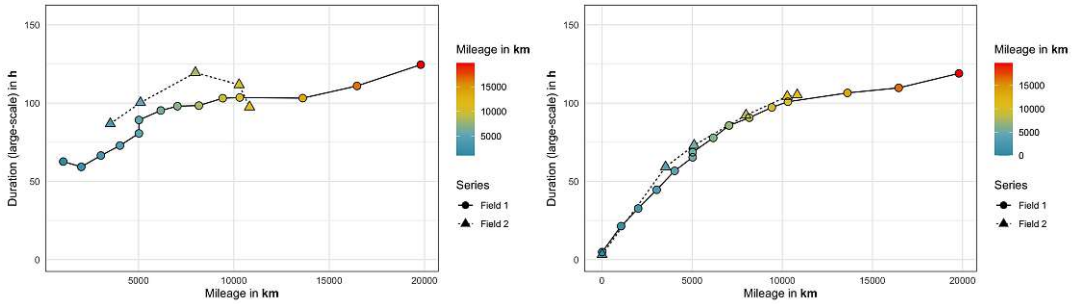
For field use, at the beginning of the process, all LASSO models under investigation agree

on an almost linear relationship up to about 5 000 km, then the curve starts to flatten. For all LASSO configurations (Figures 2.5b-2.5d), the curve stops increasing at around 10 000 km. The maximum prediction for configuration 1 is around 125 h, for configuration 3 around 100 h and for adaptive LASSO around 90 h. For a more detailed interpretation, we can use the significance results from the previous chapter: Adaptive LASSO with important coefficients at wavenumbers 3650 and 3648 cm^{-1} results in the lowest prediction range of the LASSO models, while model 1 with significant coefficients at wavenumbers 3650, 1661 and 1514 cm^{-1} results in the largest range; model 3 (wavenumbers 3650 and 1514 cm^{-1}) places in between. Configurations relying on wavenumbers associated with oil components that are consumed during use (configurations 1 and adaptive LASSO) yield a prediction that reaches its maximum at around 100 h, as said components are completely decomposed at this point and the absorption bands associated with them reach a mostly constant state. This can especially be observed for the phenolic and aminic antioxidants (see Figures 2.1b and 2.1c) as they are consumed during their use as a protection additive against oxidation. When looking at the absorption band of the anti-wear additive ZDDP in Figure 2.1d, one can see that it decreases in height while changing shape in the beginning, whereas for the remaining time the shape remains mostly unchanged with the overall height increasing in the respective area of the FTIR spectrum. As already mentioned in Section 2.4.2, the former originates from the ZDDP degradation and the latter possibly indicates the formation of other compounds.

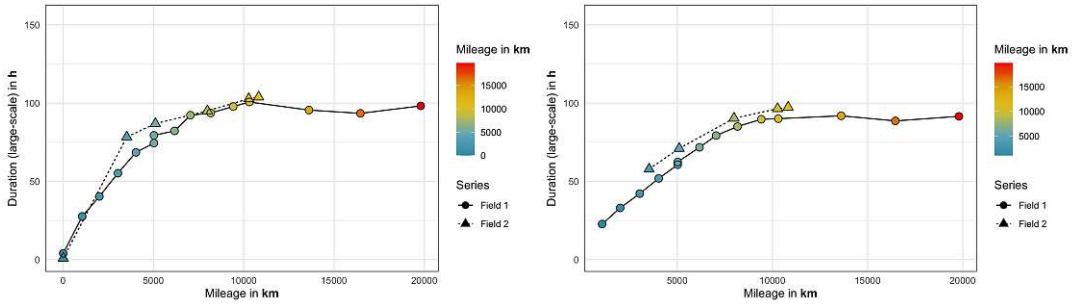
However, the configuration also based on oxidation predicts a higher duration, as oxidation processes keep going during the progression of degradation and the respective absorption bands keep changing. Oxidation processes include very comprehensive and complex mechanisms that vary in the nature and quantity of the chemical compounds formed, depending on the exact conditions. This influences the position, shape and height of the respective absorption bands. Despite this variability in terms of the underlying chemical processes, oxidation seems to make an important contribution to the prediction models, especially when it comes to oil conditions where additives are depleted to a great extent.

In addition to oxidation processes being different from those during an artificial alteration, other processes such as nitration play a role in the real-life use of engine oils. Nitration accounts for one of the biggest differences between oils degraded in a real engine and oils altered in a classical thermo-oxidative laboratory method (Ronai, 2021). Since the training of the models in this work is based on a data set generated by the mentioned alteration method, nitration does not appear in this context.

Regarding the association between small-scale alteration and large-scale alteration, the resulting curves are much smoother, and all models retain the order of the degradation samples. In general, the prediction between the artificial alteration methods yields more consistent results, most likely due to the similarity of the alteration processes as the two methods are very closely related being based on the same principles but being carried out on different scales. It should be noted, however, that the timescale of the two artificial alteration methods is not equivalent, i.e., the large-scale alteration allows a certain degraded condition to be attained more quickly than the small-scale one. A possible explanation for this phenomenon is the more intensive mixing that is implemented in the large-scale method (see Besser et al., 2019), that results in a more extensive contact of the lubricant with the air that passes through it.

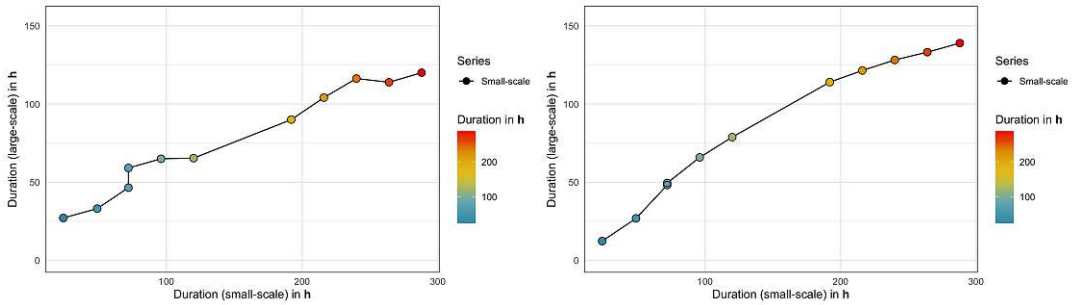


(a) Prediction using SPLS regression (b) LASSO regression, penalty configuration 1

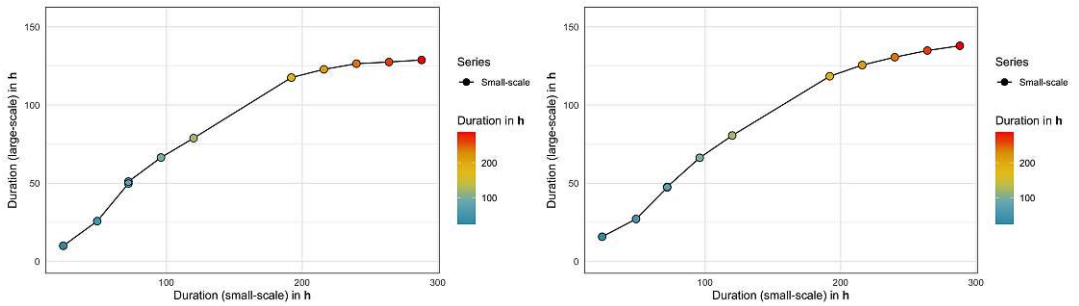


(c) LASSO regression, penalty configuration 3 (d) Adaptive LASSO regression

Figure 2.5: Models for relationship between field test and artificial large-scale alteration.



(a) Prediction using SPLS regression (b) LASSO regression, penalty configuration 1



(c) LASSO regression, penalty configuration 3 (d) Adaptive LASSO regression

Figure 2.6: Models for relationship between artificial small-scale and large-scale alteration.

2.5 Conclusion

As FTIR spectroscopy is a popular and widely applied method in analytical chemistry, a comprehensive procedure to simplify the pre-processing, data cleaning and application of chemometrical methods to FTIR spectra has been developed. An automatic filtering procedure for non-informative variables was introduced, model estimation and variable selection were performed simultaneously using the weighted LASSO, and confidence intervals for the selected wavenumbers were derived using post-selection inference. The analysis pipeline was demonstrated on a real-world dataset of FTIR spectroscopic data of artificially altered and used engine oils, our model achieving high predictive performance.

This analysis pipeline offers several advantages: Our reconstruction-error based method is an objective alternative to manual selection of absorption bands for further analyses. The combination of weighted LASSO and the post-selection inference methodology provides an effective tool for the analysis of high-dimensional spectroscopic data: Knowledge of domain experts can be integrated with the LASSO model, leading to higher predictive power than PLS and classical LASSO. As the resulting model is sparse, it is easy to interpret and simplifies subsequent analyses and interpretation. Furthermore, confidence intervals for the resulting coefficients can be obtained, providing more insight for the interpretation of selected wavenumbers.

By means of the presented procedure, a mutual correlation of degradation stages of engine oils from different sources could be quantified. For example, the mileage of a passenger car could be mapped to an alteration duration in the large-scale laboratory alteration device. With such correlations at hand, it is possible to develop laboratory-scale alteration methods tailored to simulate specific field applications where samples of used oils are typically scarce. The proposed analysis methods can be used to classify unseen samples from the field according to a scale calibrated on artificially altered engine oil samples. Moreover, alteration duration in the laboratory can be directly correlated with mileage in field, e.g., the alteration duration required to generate a "used" engine oil after 10 000 km. Such lab-to-field approach allows for bridging the gap between laboratory bench testing and real-world field applications. Within the domain of lubrication technology, a potential application is related to green lubricants, where long-term field tests are not yet available.

The most important aspect of the usage of lubricants is related to lubricating performance, i.e., friction and wear behavior. Currently, tribometrical experiments, such as a steel ball sliding against a steel disk with the oil of interest in between, are executed to capture lubricating performance. The knowledge of the oil condition and its relationship with lubricating performance is also a step to reduce or even make obsolete such time-consuming and costly experiments.

More general, the here proposed technique allows for building statistical models based on FTIR spectroscopic data: The presented pre-processing method filters non-informative variables automatically and especially for small sample sizes, the ability to integrate the knowledge of domain experts with the popular LASSO model proved to be a powerful technique to achieve high predictive performance.

3 Sparse robust regression and classification with FTIR spectra and image data

This chapter was published in Analytica Chimica Acta, Volume 1279, 341762, Pia Pfeiffer, Peter Filzmoser, Robust statistical methods for high-dimensional data, with applications in tribology, Copyright Elsevier (2023).

3.1 Introduction

The advance of digital technologies has transformed the way data are collected and analyzed. In tribology, these developments have motivated the use of data-driven methods for the design and validation of tribological systems. For oil condition monitoring, for example, several authors investigate the application of spectroscopic methods to monitor the lubricant's degradation process over time: FTIR (Fourier-transform infrared) spectra can be used to predict oil attributes (Al-Ghouti et al., 2010; Felkel et al., 2010; Rivera-Barrera et al., 2020). Other modeling objectives include the comparison of oil degradation in different laboratory alterations and field settings (Besser et al., 2013; Pfeiffer et al., 2022). Another aspect linked to oil condition is lubrication performance, i.e. friction and wear behavior. To investigate lubrication performance, SRV® (Schwing-Reib-Verschleiß) tribometer experiments (a steel ball sliding against a steel disk with the lubricant of interest in between) are carried out, resulting in a collection of several types of data for one oil, including functions of the coefficient of friction and optical data of wear scar areas.

However, in data produced from experiments, there may also be observations present that behave differently from the majority of data points. Those observations are called *outliers* in statistics and the data set is said to be *contaminated*. While for traditional methods one outlying observation can have a huge impact on the resulting model, robust methods aim to identify and downweight unusual data points. This way, observations that do not follow the majority of the data can be uncovered and further investigated.

In addition, high numbers of measured variables make the application of classical statistical methods difficult. A given data set is called *high-dimensional* if the number of variables p exceeds the number of observations n. In this setting, both the classical as well as robust regression and classification estimators are not well-defined and run into numerical problems. These can be handled by dimension reduction, using PCR (Principal Component Regression) or PLS (Partial Least Squares), for example. Other approaches for high-dimensional data are penalized regression or classification estimators such as Ridge (Hoerl and Kennard, 1970), LASSO (Tibshirani, 1996) or Elastic Net (Zou and Hastie, 2005) regression or penalized discriminant analysis (Witten and Tibshirani, 2011), as well

as sparse logistic regression, available in the R package `glmnet` (Friedman et al., 2010c). These approaches are suitable for high-dimensional data, however, they are not robust in the presence of outliers.

While there are many robust methods available for the low-dimensional case, the portfolio of robust methods for a high-dimensional setting is not that rich. In the following, we will mention some of these approaches, and also put emphasis on sparse methods, which are based on the underlying assumption that only a few variables of the high-dimensional data contribute to explaining the response. This work does not aim to give an exhaustive review of available methods but rather demonstrate the application of selected robust statistical methods for practitioners.

The remainder of the paper is organized as follows: In Section 3.2, an overview of selected sparse and robust methods as well as available implementations for regression and classification tasks is given. Section 3.3 illustrates the application of these statistical methods using two data sets from lubricant analysis and tribological experiments: FTIR spectra and image data of wear scar areas resulting from a tribometrical experiment, and Section 3.4 concludes with recommendations for the application of robust statistical methods in practice.

3.2 Robust statistical methods

First, selected robust regression and classification estimators are introduced for the low-dimensional setting. Then, approaches to extend robust methods to the high-dimensional case are discussed. For all mentioned methods, the availability of implementations in R software packages is indicated.

3.2.1 Robust linear regression

Consider n samples $(\boldsymbol{x}_i, y_i)$ with $i = 1, \ldots, n$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})$ contains information about the measurements on p variables. In a regression setting, the values y_i are collected in the vector $\boldsymbol{y}$, which is our response, and the information $(1, \boldsymbol{x}_i)$ is collected as rows of the predictor matrix $\boldsymbol{X}$. The linear regression model is given as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$ are the regression coefficients, with the intercept term β_0, and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)'$ are the error terms. Let $\hat{\boldsymbol{\beta}}$ denote an estimate for the unknown regression coefficients. Then the *residuals* $\boldsymbol{r} \in \mathbb{R}^n$ are given as $\boldsymbol{r}(\hat{\boldsymbol{\beta}}) = (r_1(\hat{\boldsymbol{\beta}}), \ldots, r_n(\hat{\boldsymbol{\beta}}))' = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}$.

The well-known Least Squares (LS) estimator is then defined as

$$\hat{\boldsymbol{\beta}}_{LS} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} r_i(\boldsymbol{\beta})^2. \tag{3.1}$$

The solution can be easily computed in explicit form: $\hat{\boldsymbol{\beta}}_{LS} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$. However, this only holds if the matrix $\boldsymbol{X}'\boldsymbol{X}$ is invertible, which would not be the case for high-dimensional settings ($n < p$).

As the LS estimator is based on the squared residuals, the influence of potential outliers is not bounded and therefore even one unusual observation can distort the estimation. One important step towards robustness is to introduce observation weights $\omega_i \in [0, 1]$, for

$i \in \{1, \ldots, n\}$. Outlying observations will receive a small weight. Outliers in regression are observations with large residuals, and they could either be outliers in the space of the x-variables (bad leverage points), or they could be in the normal x-range (vertical outliers). Observations with abnormal x-values but small residuals are often called good leverage points, because they could stabilize the regression fit. On the other hand, they could lead to underestimating the residual scale. For a robust estimator, the objective function (3.1) can be generalized as

$$\hat{\boldsymbol{\beta}}_M = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \rho \left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}} \right), \tag{3.2}$$

where ρ denotes an appropriate (bounded) function applied to the residuals and $\hat{\sigma}$ the residual scale estimate (Maronna et al., 2006). The resulting estimator is called M-estimator of regression and is computed by solving the system of estimating equations $\sum_{i=1}^{n} \omega_i(r_i(\boldsymbol{\beta}))\boldsymbol{x_i} = 0$ with a weight function $\omega(u) = \rho'(u)/u$ that determines the robustness of the estimator. This can be accomplished by using an iterative reweighted LS algorithm: For a given $\hat{\boldsymbol{\beta}}_t$ in iteration step t, the residuals and weights can be computed and the estimating equations solved for $\hat{\boldsymbol{\beta}}_{t+1}$. The starting value $\hat{\boldsymbol{\beta}}_0$ and the residual scale $\hat{\sigma}$ need to be estimated robustly. A popular choice is the *M-estimator of scale* or *S-estimator*, given as the solution $\hat{\sigma}$ of

$$\frac{1}{n} \sum_{i=1}^{n} \rho \left(\frac{r_i}{\hat{\sigma}} \right) = b \tag{3.3}$$

where ρ denotes an appropriate (bounded) function and b is a constant. Using the S-estimator (Equation (3.3)) as the initial estimator for the M-estimator leads to the *MM-estimator*, leading to a compromise between good efficiency and robustness (Maronna et al., 2006). It is implemented as `lmrob()` in the R package `robustbase` (Maechler et al., 2024).

An intuitive and computationally efficient alternative is given by the Least Trimmed Squares (LTS) estimator (Rousseeuw, 1984; Rousseeuw and Van Driessen, 2006). It is defined as

$$\hat{\boldsymbol{\beta}}_{LTS} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{h} r_{i:n}(\boldsymbol{\beta})^2 \tag{3.4}$$

with the order statistics of the squared residuals $r_{1:n}(\boldsymbol{\beta})^2 \leq \cdots \leq r_{n:n}(\boldsymbol{\beta})^2$. The efficiency and robustness of the estimator are determined by the parameter h, which is typically chosen as half or 3/4 of the number of observations. As for the above regression estimators, a limitation is that they can only be applied to settings with $n > p$, here, depending on the choice of h, even $n > 2p$. The LTS estimator is also available in the R package `robustbase` as `ltsReg()`.

3.2.2 Robust regression for high-dimensional data

For the case $p > n$, the PLS estimator is often chosen in chemometrics (Varmuza and Filzmoser, 2009). Several proposals exist to make this estimator robust against outliers:

They are based on robust covariance estimation (Gil and Romera, 1998) or replace LS regression by a robust estimator (Wakeling and Macfie, 1992; Cummins and Andrews, 1995; Hubert and Vanden Branden, 2003; Serneels et al., 2005; Xie et al., 2022). A discussion of robust PLS approaches and respective advantages and disadvantages can be found in Filzmoser et al. (2020b). In the following, we describe the Partial Robust M (PRM) estimator (Serneels et al., 2005). As for the M-estimator (Equation (3.2)), observation weights $\omega_i \in [0,1]$, for $i \in \{1,\dots,n\}$ are introduced to downweight outlying observations. The weights are collected in the diagonal of the diagonal matrix $\boldsymbol{\Omega} = \text{Diag}(\omega_1,\dots,\omega_n)$, and the weighted data information is obtained as $\tilde{\boldsymbol{X}} = \boldsymbol{\Omega}\boldsymbol{X}$ and $\tilde{\boldsymbol{y}} = \boldsymbol{\Omega}\boldsymbol{y}$. In PLS regression we construct latent components (or *scores*), which are linear combinations of the original variables with *weighting vectors*. The weighting vectors $\boldsymbol{a}_h$ for $h \in \{1,\dots,h_{max}\}$ are obtained by the maximization problem

$$\boldsymbol{a}_h = \underset{\boldsymbol{a}}{\text{argmax}}\, \text{cov}^2\left(\boldsymbol{y}, \boldsymbol{X}\boldsymbol{a}\right), \tag{3.5}$$

for $h \in \{1,\dots,h_{max}\}$ under the constraints that

$$\| \boldsymbol{a}_h \| = 1 \quad \text{and} \quad \boldsymbol{a}_h'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{a}_i = 0 \text{ for } 1 \le i < h. \tag{3.6}$$

Here, h_{max} is the maximum number of components we want to retrieve, and it is assumed that the response, as well as the predictor variables, are mean-centered. In PRM regression, the centering is done robustly, e.g. by the column-wise median. For estimating the covariance in Equation (3.5), the sample covariance matrix with the weighted observations has been proposed, and thus we maximize

$$\text{cov}^2\left(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{X}}\boldsymbol{a}\right) = \frac{1}{(n-1)^2}\boldsymbol{a}'\tilde{\boldsymbol{X}}'\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}'\tilde{\boldsymbol{X}}\boldsymbol{a}, \tag{3.7}$$

and the constraints (3.6) are also based on weighted predictors. The resulting weighting vectors are collected as columns in the matrix $\boldsymbol{A}$, and thus the matrix of scores is $\tilde{\boldsymbol{T}} = \tilde{\boldsymbol{X}}\boldsymbol{A}$, with rows $\tilde{\boldsymbol{t}}_i$, for $i = 1,\dots,n$. The crucial point is to obtain the weights. As the name already suggests, the PRM regression estimator makes use of the concept of the robust M-estimator, see Equation (3.2), by regressing the weighted response on the robustified scores,

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\text{argmin}} \sum_{i=1}^{n} \rho\left(\tilde{y}_i - \tilde{\boldsymbol{t}}_i'\boldsymbol{\gamma}\right), \tag{3.8}$$

where $\tilde{y}_i$ are the elements in $\tilde{\boldsymbol{y}}$. This yields robust residuals $\tilde{r}_i = \tilde{y}_i - \tilde{\boldsymbol{t}}_i'\hat{\boldsymbol{\gamma}}$, and by employing a robust scale estimator, such as the MAD, a robustly estimated residual scale $\hat{\sigma}$ can be obtained. The weights are defined by

$$\omega_i^2 = \omega_R\left(\frac{\tilde{r}_i}{\hat{\sigma}}\right)\omega_T\left(\frac{\|\tilde{\boldsymbol{t}}_i - \text{med}_j(\tilde{\boldsymbol{t}}_{.j})\|}{\text{med}_i\|\tilde{\boldsymbol{t}}_i - \text{med}_j(\tilde{\boldsymbol{t}}_{.j})\|}\right), \tag{3.9}$$

where $\tilde{\boldsymbol{t}}_{.j}$ is the jth column of $\tilde{\boldsymbol{T}}$, for $j = 1,\dots,h_{max}$. The weight function $\omega_R(u)$ takes care about downweighting large (scaled) residuals, whereas the weight function $\omega_T(u)$ downweights leverage points. The specific choice of appropriate weight functions, as well as

initial weights to start the iterative algorithm, are discussed in Serneels et al. (2005). More recently, the effects of different weight functions have been studied in Polat (2020), and better guidance has been offered to select the most appropriate one.

The PRM method has been extended to a sparse PRM regression procedure by Hoffmann et al. (2015) which, similar to LASSO regression, yields zeros in the regression coefficient vector, and thus, in fact, performs variable selection. In the R package `sprm` (Serneels and Hoffmann, 2015), both PRM and SPRM regression are available via the functions `prms()` and `sprms()`. In Python, the package `direpack` (Menvouta et al., 2023) provides robust dimensionality reduction techniques for high-dimensional data.

Combining the LTS estimator with L1 regularization yields the *sparse LTS* estimator Alfons et al. (2013), a robust version of the LASSO. It is given by

$$\hat{\boldsymbol{\beta}}_{sparseLTS} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{h} r_{i:n}(\boldsymbol{\beta})^2 + n \cdot \lambda P(\boldsymbol{\beta}), \tag{3.10}$$

where $P(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|$.

Least Angle Regression (LARS) was proposed by Efron et al. (2004) and is closely related to the LASSO (Tibshirani, 1996). LARS provides an ordered sequence in which the variables enter the regression model. While this sequence is the same as for the LASSO, it is derived in a computationally more efficient way from the correlation matrix of the data. Based on this property, Khan et al. (2007) propose a robustification of LARS by replacing mean, variance, and correlation with robust location, scatter, and correlation estimators.

Both methods are available in the R package `robustHD` (Alfons, 2016) as `sparseLTS()` and `rlars()`.

3.2.3 Robust classification

It is assumed that a training set of multivariate data observations is available, together with information about their group membership. The task is to train a classifier which reliably assigns test set observations to the groups. For linear discriminant analysis consider g multivariate normally distributed populations $\pi_i, i = 1, \ldots, g$ with means $\boldsymbol{\mu}_i$ and the same covariance $\boldsymbol{\Sigma}$. Let p_i denote the prior probabilities that an observation belongs to group i. Then the discriminant values for an observation $\boldsymbol{x}$ are given by

$$d_i(\boldsymbol{x}) = \boldsymbol{\mu}_i' \boldsymbol{\Sigma}^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}_i' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_i + \ln p_i, \tag{3.11}$$

for $i = 1, \ldots, g$ Johnson and Wichern (2007). An observation $\boldsymbol{x}$ is assigned to group k, if

$$d_k(\boldsymbol{x}) = \max_i d_i(\boldsymbol{x}). \tag{3.12}$$

The discriminant values (3.11) depend on the group means and the joint covariance matrix. In the classical case, the arithmetic means of the data groups and a pooled sample covariance can be used as estimators Johnson and Wichern (2007). In order to achieve a robust classifier in presence of outliers, these estimators can be substituted with robust location and scatter estimates. In Croux and Dehon (2001), for example, the S-estimator is proposed, while

in Hubert and Van Driessen (2004) the FastMCD estimator is used. Todorov and Pires (2007) provide a comparative study between different robust covariance estimators. An implementation is available as `Linda()` in the R package `rrcov` Todorov and Filzmoser (2009b).

Robust classification can also be performed by applying robust regression estimators for a logistic regression model, where the posterior class probabilities with the group variable G are modeled by linear functions,

$$\log \frac{P(G = k|\boldsymbol{x})}{P(G = g|\boldsymbol{x})} = \beta_{k0} + \boldsymbol{\beta}_k'\boldsymbol{x}, \text{ for } i = 1, \ldots, g-1, \tag{3.13}$$

with the constraint that the probabilities remain in the interval $[0, 1]$ and that they sum up to 1. The model parameters are commonly estimated using the maximum likelihood (ML) method. In the classical case, this corresponds to an iteratively reweighted LS algorithm. In the R package `robustbase` (Maechler et al., 2024), several different algorithms for a robust estimator are implemented in the function `glmrob()`.

3.2.4 Robust classification for high-dimensional data

As discussed in Section 3.2.2, several proposals for robust regression in high dimensions have been developed. For classification purposes, however, fewer methods are available. One approach to robust discriminant analysis is by directly plugging in a regularized version of a robust covariance estimator to compute the discriminant values in (3.11). This can be done for example by applying the Minimum Regularized Covariance Determinant (MRCD) estimator from Boudt et al. (2020) to the group-wise robustly centered observations. Another approach is based on applying robust regression estimators in logistic or multinomial regression. Kurnaz et al. (2017) combine a trimmed estimator with the Elastic Net penalty to achieve a robust estimator suitable for high-dimensional data. An implementation is available in the R package `enetLTS` (Kurnaz et al., 2022). Another strategy to perform robust classification for high-dimensional data is to first reduce the dimensionality before applying a robust classification method. If the resulting classifier should be adjusted to a response variable, this can be done by constructing latent variables based on PCR or PLS, or selecting variables based on a robust and sparse regression method like sparseLTS. An example for this two-step approach will be given in Section 3.3.1.

The robust methods discussed in the above sections downweight potentially outlying rows $\boldsymbol{x_i}$ of a given data set $\boldsymbol{X}$. Especially in the high-dimensional case, however, it might be desirable to consider the concept of *cellwise* robustness: In contrast to rowwise robustness, outlying cells x_{ij}, not rows $\boldsymbol{x_i}$, are flagged. Rather recent proposals for cellwise robust estimators have been made by Machkour et al. (2020) and Bottmer et al. (2022), though unfortunately, their algorithms are not available in R packages yet.

3.3 Examples

3.3.1 Sparse robust regression and classification with FTIR spectra

Some of the methods above are illustrated on a data set consisting of FTIR spectra of ten automotive engine oils. The underlying engine oils are commercially available SAE 5W-

30 and SAE 0W-20 engine oils. FTIR spectra and other conventional analyses indicate the application of additives commonly used in automotive engine oils, like ZDDP (zinc dialkyldithiophosphates), antioxidants, detergents with a base reserve, and dispersants.The fresh oils were subjected to an artificial small-scale alteration as described by Dörr et al. (2019b), once with a temperature of 180°C, and once with 160°C, denoted by *Group A* and *Group B*, respectively. For both groups, samples were taken regularly during the total duration of 96 hours, yielding a data set of in total 50 samples per group.

For all of these samples, FTIR spectra were recorded, each consisting of the absorbance at 1814 wavenumbers. The resulting data set contains $p = 1814$ explanatory variables and $n = 100$ observations and includes two types of response: a grouping variable denoting the membership to Group A or B, respectively, and a numeric response referring to the alteration duration in hours. Hence, the statistical tasks at hand are classification according to group membership and regression on the alteration duration for each group separately. In this application, the interpretability of those models is also of interest: A sparse model with only few non-zero coefficients corresponding to specific wavenumbers can help to understand the underlying chemical processes distinguishing the groups or contributing to oil degradation.

As there are only 50 samples per group (the same ten oils in each temperature group, with varying levels of alteration duration in the groups), we have a high-dimensional setting with low sample sizes. In order to make the tasks even more challenging, we added 6 samples from a *large-scale* artificial alteration series according to Besser et al. (2019) to Group A (same temperature of 180°C). We will refer to these data as "contaminated" samples. This will call for robust methods, and their performance will be compared to non-robust counterparts.

The wavenumbers between 3030-2770 cm^{-1} and 1480-1430 cm^{-1} are areas of high or total absorption, i.e. are not reliable measurements. This is caused by vibrations of hydrocarbons that are always present in engine oils. As a result, these regions not only exhibit total absorption but also do not provide any useful information and are generally disregarded during evaluation. These sections are sometimes removed manually by domain experts but can also be identified as uninformative variable ranges by statistical methods, as proposed by Pfeiffer et al. (2022). After the filtering process applied in Pfeiffer et al. (2022), the spectra consist of 1668 out of 1814 wavenumbers.

Sparse regression

Due to the nature of FTIR data, neighboring variables are highly correlated and we can expect that only a few wavenumbers are sufficient for a reasonable prediction accuracy. We use the LASSO estimator (Tibshirani, 1996) to perform sparse regression with high-dimensional data, separately for the Group A and the Group B measurements.

Since Group A has been contaminated by 6 observations, we also fit a LASSO model to the uncontaminated Group A measurements for comparison. As a robust counterpart, the sparse LTS estimator, see Equation (3.10), is used separately for the contaminated Group A and for Group B. All methods are applied on randomly selected training sets: When fitting a model to the uncontaminated Group A and to Group B, about 2/3 of the samples were selected; when fitting the contaminated Group A, all 6 *large-scale* samples were added to the training set. The test sets consist of all remaining samples from both data sets.

Figure 3.1 shows the (selected) FTIR spectra of the training data, here for Group A,

together with vertical lines indicating the selected variables from the three approaches. All methods yield only very few variables. The variable selection by the robust method should not be influenced by the contamination. For the LASSO, however, a rather big difference in the clean and contaminated training data can be observed: not only the number but also the position of selected variables is different.

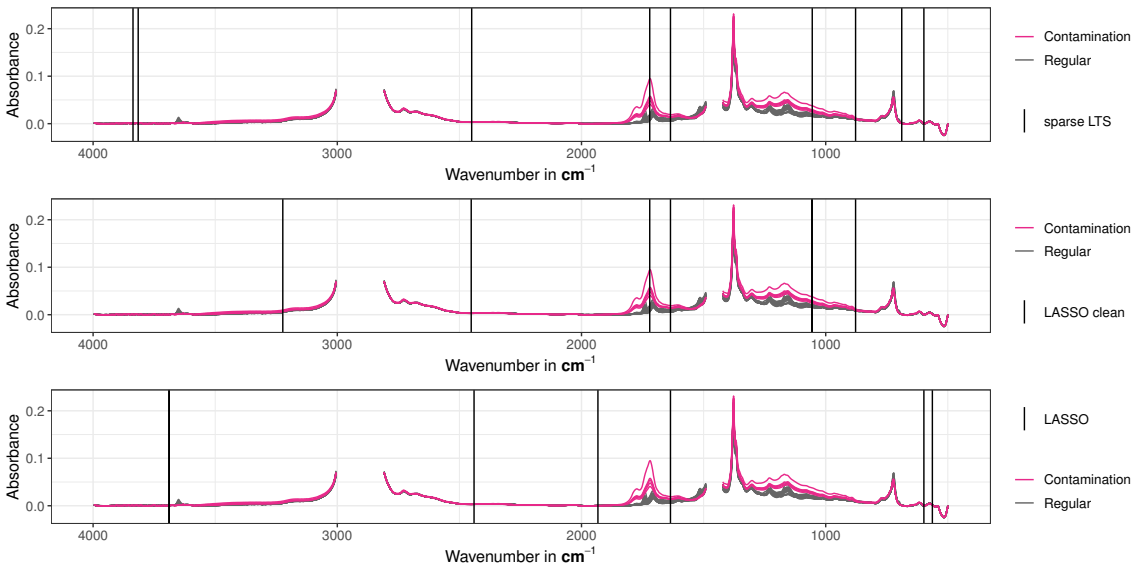
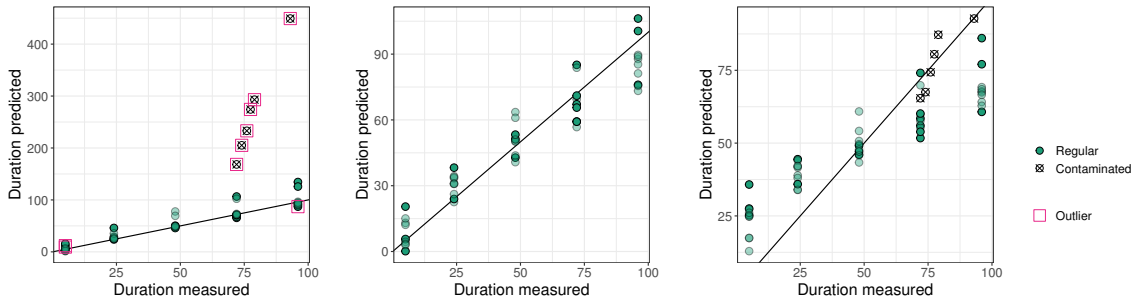


Figure 3.1: FTIR spectra of the training set of Group A, for robust and non-robust sparse regression on the contaminated data. The vertical lines indicate the wavenumbers selected by the models. For *sparse LTS*, the selected wavenumbers are 3836.62, 3815.40, 2449.73, 1720.60, 1635.72, 1055.12, 877.66, 688.62, and 597.96 cm^{-1}. For *LASSO* on clean data, the selected wavenumbers are 3223.22, 2451.66, 1720.60, 1635.72, 1057.05, 1055.12, and 877.66 cm^{-1}, and for *LASSO* on contaminated data, the selected wavenumbers are 3690.02, 3688.09, 2440.08, 1932.78, 1635.72, 597.96, and 563.24cm^{-1}.

Figure 3.2 shows the measured (horizontal axes) versus the predicted (vertical axes) response of Group A, for the three models, with the selected variables shown in Figure 3.1. The colors correspond to training (half transparent) or test dataset and the symbols show whether an observation is regular, contaminated, or identified as an outlier by the robust procedure. The solid line refers to the equality $y = \hat{y}$. The plot for the robust method (Figure 3.2a) reveals that the model does not follow the contaminated samples. This can also be verified by inspecting the observations flagged as outliers (encoded as squares): the contaminated samples in the training set, and also some additional observations, were fully downweighted when fitting the model. In addition to the contaminated samples, two additional observations are flagged as outlying. These two outliers consist of atypical x-information, but their prediction is still in a normal range (good leverage points). Since the procedure also yields a robustly estimated standard deviation, this outlying information can also be computed for the test set data: here, outliers are defined as observations with



(a) (b) (c)

Figure 3.2: Measured versus predicted response for training (half transparent) and test set, where the prediction is based on the selected variables from Figure 3.1. (a) Sparse LTS for contaminated data, (b) LASSO for clean data, (c) LASSO for contaminated data.The robust method (Figure 3.2a) fully downweights the contaminated samples, while the classical method (Figure 3.2c) is severely influenced by those samples.

standardized absolute residuals larger than 2.

Figures 3.2b and 3.2c reveal the effect of contamination on the non-robust LASSO estimator: When contamination is present, the LASSO fit changes significantly, as the model also tries to accommodate the contaminated samples. This implies that the selected variables are also influenced by these samples. In Figures 3.2b and 3.2c this difference between a fit on clean and contaminated data is illustrated.

The presented methods can identify outliers, and if there is the need to further investigate *why* an observation is outlying, i.e. which variable(s) contribute most to the outlyingness, some recently developed algorithms can be applied: In Mayrhofer and Filzmoser (2023), the outlyingness is decomposed using Shapley values, and in Debruyne et al. (2019), the outlyingness is regarded as a regression problem. In the latter approach, it is possible to use the weights, that are output of a robust linear regression fit, as input to the SPADIMO algorithm, which is implanted in the R package `crmReg` Filzmoser et al. (2020a). We use the function `spadimo` together with the weights from sparse LTS regression and show the resulting outlyingness scores for each observation that has been identified as outlying in Figure 3.3. The lines have been colored according to the outlyingness scores, and upon inspection of the figures the usefulness of robust methods becomes apparent: As several variables, that are selected for the resulting sparse model, also correspond to variables with high outlyingness scores, it is crucial to apply a statistical method that can deal with outliers.

Sparse classification

The second statistical task is to predict the group membership of the samples based on the wavenumbers. In our high-dimensional setting, a penalized estimator such as sparse logistic regression (see Friedman et al., 2010a) with a LASSO penalty on the negative log-likelihood is applied. This yields again variable selection among the wavenumbers. As a

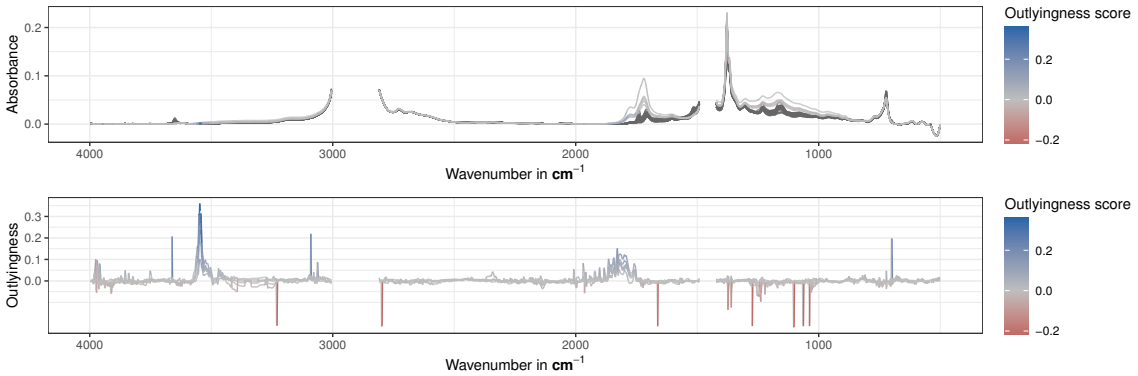


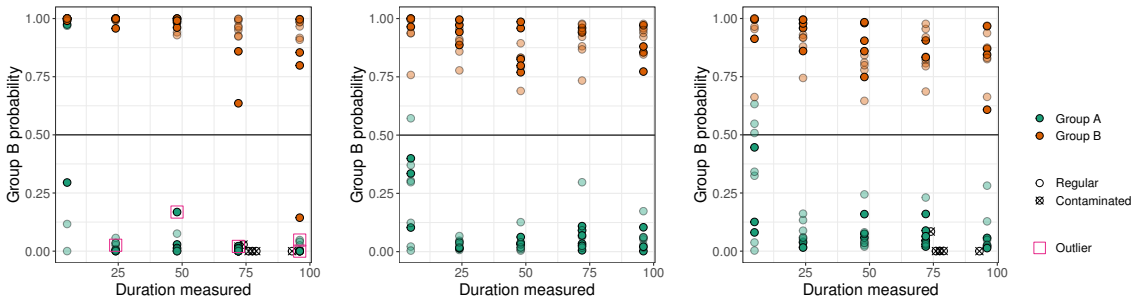
Figure 3.3: FTIR spectra of the training set of Group A, as well as the outlyingness scores for each variable resulting from the SPADIMO algorithm Debruyne et al. (2019). The lines have been colored according to the outlyingness direction and strength.

robust counterpart, the robust version of the Elastic Net estimator for logistic regression (enetLTS) is used (Kurnaz et al., 2017). We use again the same training data as in Section 3.3.1, and evaluate based on the test data; for the non-robust method, the estimator is also applied to the clean data set. The resulting misclassification errors are given in Table 3.1. For computing the misclassification rates in Table 3.1 we used the cutoff value 0.5 for the probabilities.

Table 3.1: Misclassification errors based on sparse (robust) logistic regression.

Misclassification error in %	**training set**	**test set**
enetLTS for contaminated data	7.46	9.68
Sparse logistic regression for clean data	1.53	0
Sparse logistic regression for contaminated data	4.35	0

In contrast to the results from the regression, outliers do not seem to have a negative influence on the classical estimators. When inspecting the corresponding plots of group probability over *duration* in Figure 3.4, however, it becomes apparent that the misclassification error is not evenly distributed over the different values of duration. In Figure 3.4, the cutoff value is displayed as a horizontal line, the colors refer to group membership and the symbols distinguish regular, contaminated and observations identified as outliers by the robust procedure (encoded as a square). The training data are again shown as half transparent points. While the clean data is classified almost perfectly (Figure 3.4b), the prediction for the sparse logistic regression model for contaminated data is worst for observations with duration zero (Figure 3.4c). The resulting plot for enetLTS is given in Figure 3.4a. While the robust method yields more confident predictions, it fails to detect the contaminated samples correctly. The misclassification error also seems biased and is worse at both minimum and maximum duration. This might be due to the enetLTS algorithm that evaluates all possible subsets of a given size of the explanatory variables. This process can become instable in the presence of many correlated predictors.



(a) (b) (c)

Figure 3.4: Response variable *duration*, used in the regression step, against the posterior probability for Group B, with the cutoff at 0.5. Misclassified observations from the training (half transparent) and test set are shown "on the wrong side" of this cutoff. (a) direct enetLTS for contaminated data, (b) direct sparse logistic regression for clean data, (c) direct sparse logistic regression for contaminated data.

In order to better adjust the classifier to the response *duration*, modeling the respective duration for Group A and Group B can be used as a variable screening step. Using the set of selected variables resulting from this first step, a classification model can now be fitted to discriminate the two groups. Again, the same training and test data as for step 1 are used. Here we did not employ a robust procedure for classification, as the first step as described in Section 3.3.1 already protected against a variable selection bias due to contamination. Moreover, a unified framework in this second step makes the effect of robust estimation in the first step easier identifiable.

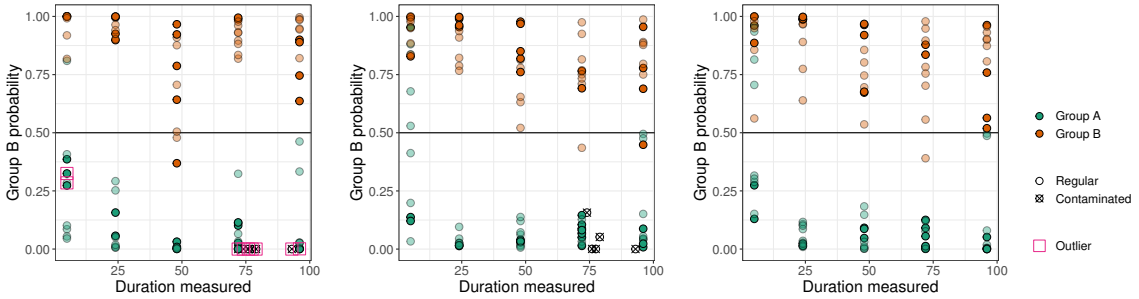
The misclassification errors resulting from the different approaches are presented in Table 3.2. While the robust procedure yields low errors for both training and test data, the errors for the classical procedure with the contaminated data are much higher.

Table 3.2: Misclassification errors based on sparse logistic regression with the pre-selected variables from Section 3.3.1.

Misclassification error in %	**training set**	**test set**
sparseLTS for contaminated data	2.86	3.03
LASSO for clean data	5.88	3.03
LASSO for contaminated data	7.46	6.25

Sparse logistic regression yields estimated posterior probabilities for each sample. Figure 3.5 shows the posterior probabilities of the samples to belong to Group B, again based on the models from the robust (Figure 3.5a) and the classical approach (Figure 3.5c for clean, Figure 3.5b for contaminated data). The horizontal axis in the plots is the response variable *duration*, which has been used in the screening step. Again, the colors correspond to group membership and the symbols refer to regular, contaminated, and identified outly-

ing observations. One can see that the non-robust models suffer from bias: for the smallest value of *duration*, only Group A observations are misclassified. In fact, this bias can already be seen in Figure 3.2c. For LASSO on the clean data, the model seems to be too much adjusted to the training data, as several test set observations are wrongly classified for small values of *duration*, see also Table 3.2. The robust procedure seems much more balanced for the two groups (Figure 3.5a). Here, the outliers as identified in the first step (see also Figure 3.2a) are indicated by pink squares. All these outliers, including the contaminated samples, are clearly assigned to the correct groups by the classifier, indicating an appropriate pre-selection of the variables.



(a) (b) (c)

Figure 3.5: Response variable *duration*, used in the regression step, against the posterior probability for Group B, with the cutoff at 0.5. Misclassified observations from the training and test set are shown "on the wrong side" of this cutoff. From left to right, the following methods are compared: (a) sparseLTS for contaminated data, (b) LASSO for contaminated data, and (c) LASSO for clean data.

3.3.2 Robust regression with image data

The performance of lubricants is measured in terms of friction and wear, and a reliable model associating the degradation stage of the engine oil with wear would therefore be useful for practitioners. Wear properties under laboratory conditions can be evaluated in a tribometrical experiment, where a steel ball and disc with the oil of interest in between are sliding against each other in a reciprocating contact on an SRV® tribometer (see Agocs et al., 2022, for a more detailed description of the experiment). For the present data set, wear scars were created from experiments with oil samples that were used on an engine test rig (according to ASTM D7484 (2021)) for up to 100 hours, with 38 samples taken at 0 minutes, 20 minutes, 20 hours, 50 hours and 100 hours. Then, images of the wear scars were recorded with an optical microscope. The statistical task is to predict the duration the engine oil was used based on the image data of the wear scar areas. For this analysis, only the wear-scar images of the balls were used. The original image data are recorded as RGB images in high resolution (2600×2000 pixels), and have to be pre-processed before training a regression model. In a first step, the images were converted to greyscale based on brightness. Then, the images were annotated to segment two classes: the wear scars in the foreground and the background, which was discarded. Next, the images were scaled to

size 128×128 pixels and pixel values were normalized to the same range using the *minimax* method, before Histogram of Gradients (HoG) features (see, for example, Prince, 2012) were extracted using the Python version of `opencv` (Bradski, 2000). HoG features can encode the texture of an image and therefore seem to be especially suitable for the presented case. The input image is first divided into cells, then the gradient magnitude and orientation for each pixel are computed, before they are normalized and collected in histograms. Depending on the cell and bin size for the histogram, a certain number of features is extracted from the input image. Note that there is a variety of textural image features available, with features based on Deep Learning being the most powerful ones Humeau-Heurtier (2019). However, training such models on as few as 38 images would only be feasible in combination with a suitable pre-trained model, while HoG features in combination with robust statistical models already lead to a good predictive performance.

Figure 3.6 shows example images of the ball as a result of wear experiments with oil at different degradation stages: from left to right, the duration the oil was used on the engine test rig is 0 minutes, 20 minutes, 10 hours, 50 hours, and 100 hours. It can be observed that with a longer duration on the engine test rig (and therefore worse oil condition) there are more and more artifacts in the images. The visible lines correspond to ridges along the direction the balls were moved in. Moreover, the shape of the wear scar gets more and more distorted with the degradation of the oil and the image. The 100 hours experiment also shows a dark spot, which might be due to soot or other small particles on the wear scar area.

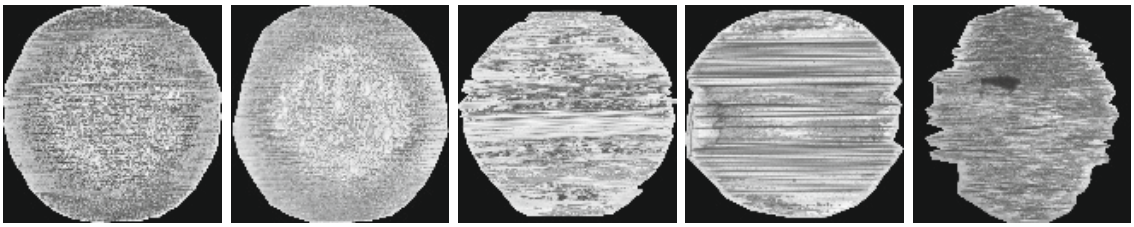


Figure 3.6: Example images of the wear experiments with varying oil condition. From left to right, the duration the oil was used is 0 minutes, 20 minutes, 10 hours, 50 hours, and 100 hours.

Since the distortion caused by the experiments is heavily varying (shape of the wear scar, type of striation, appearance of spots, etc.), the model needs to be robust against such effects. The method of choice here is linear regression, but there are some challenges for a robust approach:

- Our data set consists of only $n = 38$ images, and every image is encoded by $p = 7730$ variables. With a cell size of 8, bin size of 9, and block size of 2, the HoG feature extraction initially results in 8100 variables. After the removal of columns with only zeros due to the border, the final dimension is reached. This number is already much lower than unfolding the image pixels to 16384 variables, but still, the number of variables exceeds the number of observations by far. For this "flat" data set, sparse regression methods such as LASSO regression yield very poor models, because they can select at most n variables, which seems to be far too low in order to describe the

rather complex information of the images. Robust estimators using the Elastic Net penalty, like implemented in the `enetLTS` package, could be a compromise. However, due to the very high number of variables, the robust algorithm is not computationally feasible anymore.

- 26 out of the 38 images are taken at the beginning of the experiments (duration 0), and for the remaining durations (20 minutes, 10 hours, 50 hours, and 100 hours) we only have 3 images per duration time. Since this response variable y is extremely skewed, we will work with the transformed variable $y^{1/3}$. Still, robust regression methods either lead to very poor models, or the procedures even stop with an error. The reason is the imbalancedness of the response: The robust methods try to fit the data majority, which is for the group $y = 0$, and data with duration larger than zero are treated as outliers. A regression model only for the zero-group is of course useless.

In contrast to robust procedures, non-robust methods such as PLS regression work without any problem. Thus, the question is whether robustness can still be employed, and whether it leads to any advantage.

A first naive attempt is to exclude outliers in the x-space, i.e. we perform outlier detection only for the image data information. However, since we do not want to exclude images for the small groups with positive values of duration, outlier detection is only applied for the 26 images where the duration is zero. We use the method *pcout* as described in Filzmoser et al. (2008), implemented as function `pcout()` in the R package `mvoutlier`, which also works for very high-dimensional data. The algorithm identified 6 out of the 26 observations as multivariate outliers. PLS regression, as well as PRM regression, can then be applied to the cleaned data.

In order to evaluate and compare the different strategies, we randomly select around 2/3 of the observations (once for the complete and once for the cleaned data), and fit a model. We compare PLS regression with PRM, however, for PRM the internal weights are only used for the group with duration zero, and otherwise the weights are set to 1 in order to avoid downweighting of the observations for these small groups. The models are evaluated with the remaining test set observations by the RMSE as the measure of prediction quality. Here another issue occurs: As the discrete values of the response are very unevenly distributed, a pure random selection of observations could lead to training data where data groups are underrepresented or even absent. Therefore, we also compare the results for a stratified sampling approach: the training sets will consist of about 2/3 of the observations for the group with duration zero, and 2 out of the 3 randomly selected samples from every of the other groups. Each experiment is repeated 50 times.

The resulting RMSE values are presented as boxplots in Figure 3.7. There is not much difference between using all observations or the cleaned data if no stratification is used. PRM (modified) performs a bit better than PLS. Stratification clearly improves the results, and for PLS based on the uncleaned data, there are several outliers in the predictions. PRM on the uncleaned data gives very stable and good results, and the internal weighting seems to be better than first removing outliers.

More insights can be gained by the plots of the measured versus predicted (transformed) response, shown in Figure 3.8, for the different strategies and the two estimators. The predictions are separately shown for the training and test set observations, and for PLS and

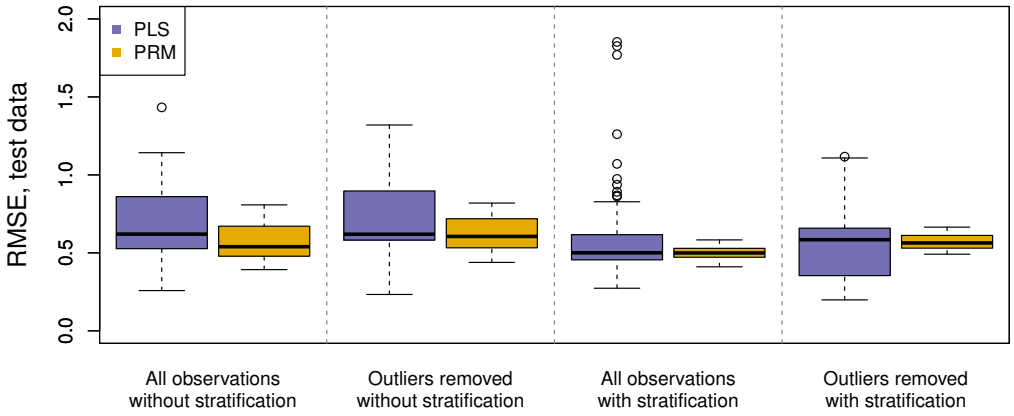


Figure 3.7: Prediction errors for classical (PLS) and robust (PRM) estimation, following different strategies for data cleaning and training/test sample selection.

PRM. In order to avoid overplotting, the values on the horizontal axis have been slightly changed in the plots. Overall, all models fit the training data quite well, but the test set prediction is rather poor, especially for higher values of duration. For all models 3 components were used, but the picture is the same when using e.g. 5 components, and it would become worse for a higher number.

For the approach with sample stratification we can see a reduction of the variability of the test set predictions for higher values of duration. This means that the main problem for these poor predictions is the imbalancedness of the data set. In the groups with duration zero there is a clear difference whether outliers are removed or not; in the latter case, outliers are visible in the test set predictions, for PLS as well as for PRM. However, as for PRM one also obtains a robust scale estimate of the residuals, these observations would be reliably identified as outliers. The same applies to the deviating predictions for higher values of duration.

To gain a better understanding of what the three latent components represent, the PRM and PLS loadings can be shown in the image domain (up to a normalization factor). In Figure 3.9, the loadings from an exemplary train-test split are shown as sections of the original images. The color scale ranges from red (negative) to blue (positive), and the intensity can be interpreted as the importance of the respective sections. While both PLS and PRM rely on similar features of the images, it can be observed that PRM is more confident, yielding more intense colorings. This is especially visible in the first, and most important, loading. Also, the first loading clearly corresponds to wear marks such as horizontal scratches on the ball's surface, while higher-order loadings also have contributions from the border of the wear scars. This emphasizes the usefulness of the presented methods to analyze the given data, also in terms of interpretation.

Overall, we can conclude that the high-dimensional image information yields good models for predicting the duration of the experiments, with the exception of the trials with duration 100 hours. A main difficulty here was the imbalancedness of the values of the response, but

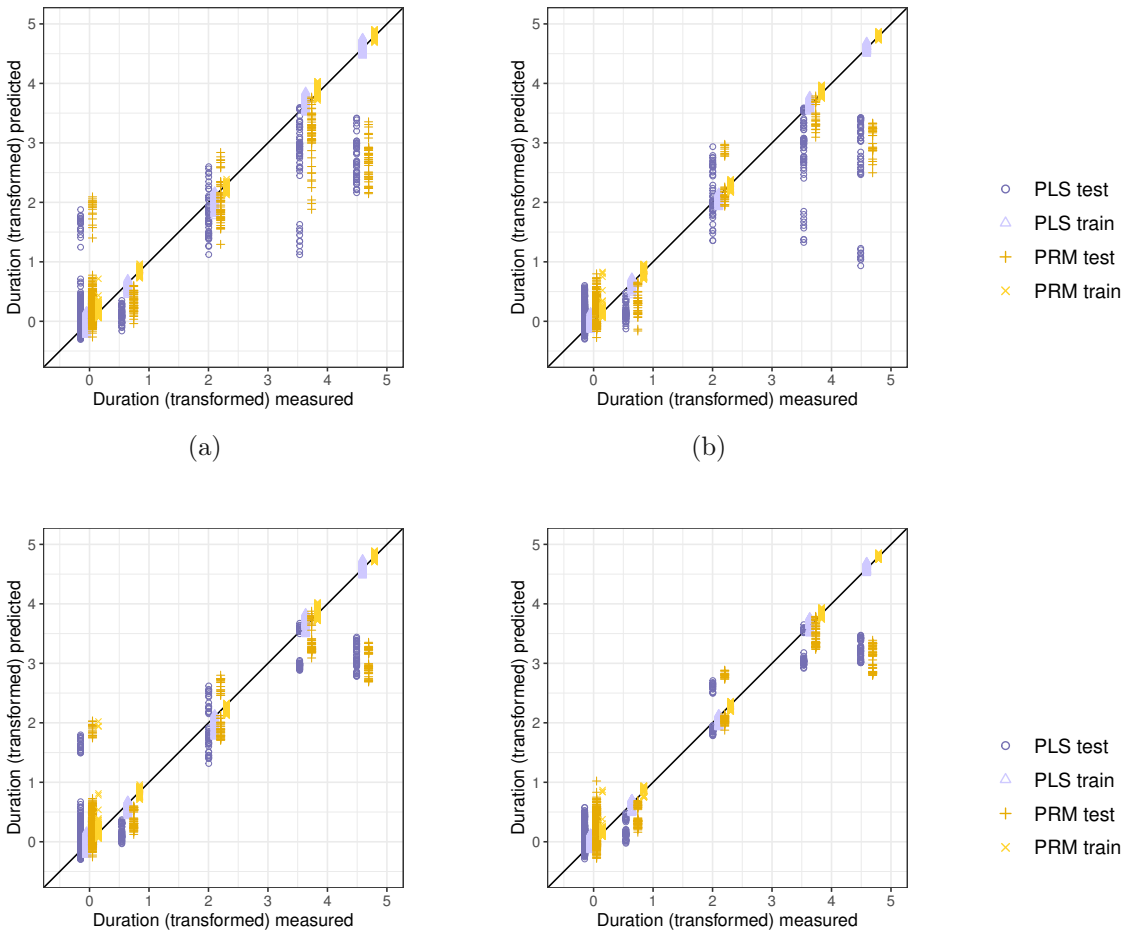
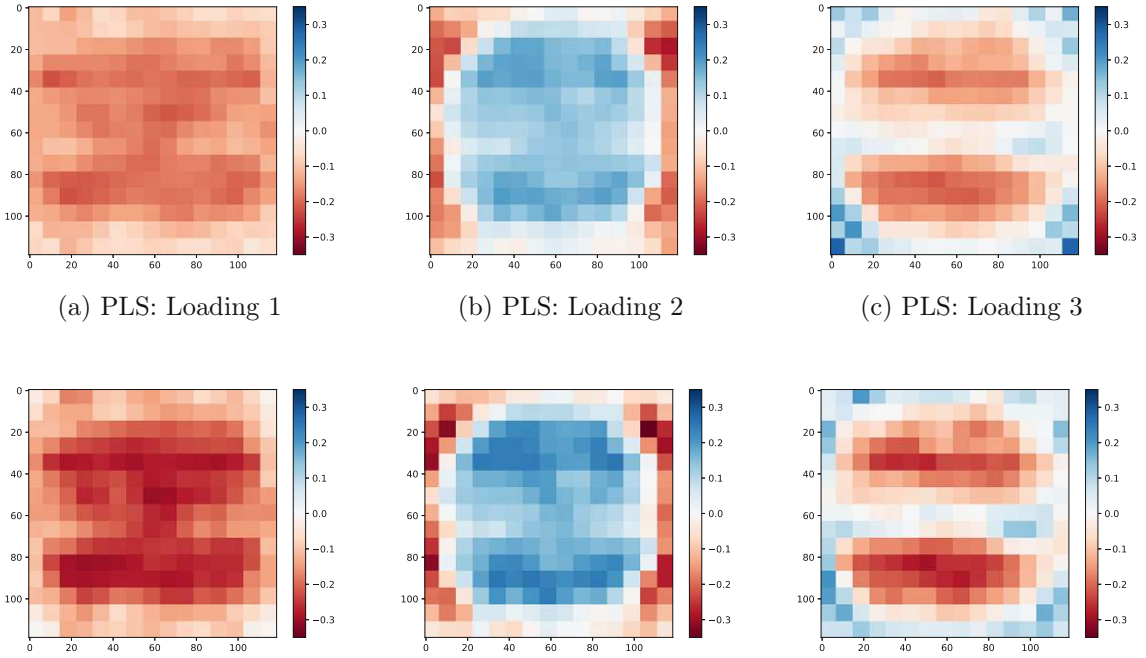


Figure 3.8: Training and test set predictions for all 50 PLS and PRM models, based on different strategies for data cleaning and selection. (a) All observations, without stratification, (b) Outliers removed, without stratification, (c) All observations, with stratification, (d) Outliers removed, with stratification.



(a) PLS: Loading 1 (b) PLS: Loading 2 (c) PLS: Loading 3

(d) PRM: Loading 1 (e) PRM: Loading 2 (f) PRM: Loading 3

Figure 3.9: The loadings traced back to the image domain for both PLS (top row) and PRM (bottom row) loadings. The loadings that are shown are those from the best-case scenario when the extracted features were robustly cleaned before applying stratified sampling and estimating the model. The color scale ranges from red (negative) to blue (positive) and the intensity corresponds to the respective section's contribution. It can be seen that while both methods seem to rely on similar areas in the original images, PRM is more precise and confident in its choice.

outliers also had a negative effect. Outlier cleaning before fitting the models makes almost no difference, for PLS as well as for PRM, but stratification clearly improves the results. The robust PRM method performs best in general; only for the stratified version on (robustly!) cleaned data, PLS can compete.

3.4 Conclusions

Imbalanced and flat data sets with fewer observations than variables pose challenges for statistical methods, especially for robust estimators, where outlying observations are down-weighted. In this paper it was demonstrated how robust methods can still be applied, and that they lead to an improved performance. To handle difficulties in model estimation, approaches that split the task in two or more steps have been shown to be successful. Especially for very imbalanced data sets, an appropriate sampling strategy was found to be crucial for the derivation of a good model as well. For a data set consisting of FTIR spectra of engine oils, a robust and sparse regression estimator was applied for the prediction of oil degradation, measured in the duration the oil was subjected to alteration. The resulting model was also demonstrated to be useful as a variable screening procedure: The selected variables, now adjusted to the different degradation stages, were used as input for a classification model. For very high-dimensional data like textural image features, sparse estimators like LASSO were found to yield very poor results, as they cannot select enough variables to represent the image information. PRM, a robust PLS method, could however be applied in combination with a stratified sampling strategy.

The given examples illustrate that, even when the direct application of robust methods is not possible, combined approaches with appropriate pre-processing and sampling methods yield improved results when compared to traditional methods. What is more, they can identify observations that do not follow the majority of the data and therefore offer additional insights.

4 Efficient computation of sparse and robust maximum association estimators

This chapter has been published as a preprint on arXiv: Pfeiffer, Pia, Andreas Alfons, and Peter Filzmoser. Efficient Computation of Sparse and Robust Maximum Association Estimators. arXiv preprint arXiv:2311.17563 (2023).

4.1 Introduction

With the availability of new measurement techniques, various different characteristics can be acquired from one and the same object. As an example from tribology, an engine oil can be investigated with respect to its chemical element composition, spectral information can be derived, or various properties concerning friction and wear of the oil can be measured, including image information of the degradation caused by the oil condition. Another example is biological data, specifically, the association between gene expressions and other variables, such as hepatic fatty acid concentrations related to a specific diet (see, e.g., Martin et al., 2007). The quantification of the relationships between different data sources can be very informative for a deeper understanding of already established mechanisms as well as for the generation of new hypotheses.

More formally, we are interested in the relationships between a p-dimensional real-valued random vector $\boldsymbol{x}$ and a q-dimensional real-valued random vector $\boldsymbol{y}$. We consider the problem of obtaining coefficient vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that the linear combinations $\boldsymbol{a}'\boldsymbol{x}$ and $\boldsymbol{b}'\boldsymbol{y}$ have maximum association, measured by an appropriate measure of association between univariate random variables. A widely applied method for this task is canonical correlation analysis (CCA) (see, e.g., Johnson and Wichern, 2007). The first canonical correlation coefficient ρ_1 and the first pair of canonical variables $(\boldsymbol{a}_1, \boldsymbol{b}_1)$ are defined via the maximization of the correlation coefficient between the two linear combinations (see, e.g., Johnson and Wichern, 2007), that is,

$$\rho_1 = \max_{\substack{\boldsymbol{a},\boldsymbol{b} \\ \|\boldsymbol{a}\|=\|\boldsymbol{b}\|=1}} \operatorname{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}), \tag{4.1}$$

$$(\boldsymbol{a}_1, \boldsymbol{b}_1) = \operatorname{argmax}_{\|\boldsymbol{a}\|=1,\|\boldsymbol{b}\|=1} \operatorname{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}). \tag{4.2}$$

The k-th canonical correlation coefficient ρ_k and the respective pair of canonical variables $(\boldsymbol{a}_k, \boldsymbol{b}_k)$ are obtained similarly as in (4.1), but under the additional constraint that they are uncorrelated with the previous $k-1$ directions, for $k \in \{2, \ldots, \min(p, q)\}$. Expression (4.1)

can be written in terms of the covariance:

$$\max_{\substack{\boldsymbol{a},\boldsymbol{b} \\ \|\boldsymbol{a}\|=\|\boldsymbol{b}\|=1}} \operatorname{Corr}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y}) = \max_{\substack{\boldsymbol{a},\boldsymbol{b} \\ \|\boldsymbol{a}\|=\|\boldsymbol{b}\|=1}} \frac{\operatorname{Cov}(\boldsymbol{a}'\boldsymbol{x}, \boldsymbol{b}'\boldsymbol{y})}{\sqrt{\operatorname{Var}(\boldsymbol{a}'\boldsymbol{x})}\sqrt{\operatorname{Var}(\boldsymbol{b}'\boldsymbol{y})}} = \max_{\substack{\boldsymbol{a},\boldsymbol{b} \\ \|\boldsymbol{a}\|=\|\boldsymbol{b}\|=1}} \frac{\boldsymbol{a}'\boldsymbol{\Sigma}_{xy}\boldsymbol{b}}{\sqrt{\boldsymbol{a}'\boldsymbol{\Sigma}_{xx}\boldsymbol{a}}\sqrt{\boldsymbol{b}'\boldsymbol{\Sigma}_{yy}\boldsymbol{b}}}, \tag{4.3}$$

where $\boldsymbol{\Sigma}_{xx} = \operatorname{Cov}(\boldsymbol{x})$, $\boldsymbol{\Sigma}_{yy} = \operatorname{Cov}(\boldsymbol{y})$ and $\boldsymbol{\Sigma}_{xy} = \operatorname{Cov}(\boldsymbol{x}, \boldsymbol{y})$. The analytical solution is given by the eigenvectors and eigenvalues of a combination of (inverse) covariance matrices: ρ_i^2 are eigenvalues of $\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$ with normed eigenvectors $\boldsymbol{a}_i$, and ρ_i^2 are also eigenvalues of $\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$ with normed eigenvectors $\boldsymbol{b}_i$, for $i = 1, \ldots, \min(p, q)$ (see, e.g., Johnson and Wichern, 2007).

Classically, the involved covariance matrices are estimated by the sample covariances, and this corresponds to maximizing the Pearson correlation coefficient as measure of association. However, these estimators are sensitive to outlying observations, and the solution is not well-defined in the high-dimensional setting, when more variables than observations are available.

There are several approaches in the literature to derive a robust solution. For the *plug-in approach*, the sample covariance is replaced by a robust estimator of the joint covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$. Croux and Dehon (2002) propose to use the minimum covariance determinant (MCD) estimator (Rousseeuw, 1984, 1985), and they derive influence functions for the canonical correlations and vectors based on this plug-in estimator, revealing their robustness properties. For a broader class of affine equivariant scatter and shape matrices, influence functions and limiting distributions of canonical correlations and vectors have been studied by Taskinen et al. (2006). Langworthy et al. (2020) present theoretical results about using the transformed Kendall correlation, which is more robust under violation of the normality assumption, for the estimation of a scatter matrix.

Another approach is to generalize (4.1) to a wider class of association estimators. Alfons et al. (2016a) define the optimization problem (4.1) in a robust way, and also consider rank-correlation measures such as the Spearman rank correlation. In that way, the search for linear relationships, as done with the Pearson correlation, is extended to looking for non-linear relationships. Results concerning Fisher consistency and the influence function underline the good theoretical properties of the corresponding robust maximum association measures, which represent the strongest association between linear combinations of two sets of random variables. The optimization is done using a grid algorithm (Alfons et al., 2016b) which, however, has its limitations concerning the dimensionality p and q of the two random variables.

The high-dimensional case, when more variables than observations are present, is another scenario where the sample covariance matrix performs poorly. This can be addressed by regularizing the covariance matrix as in the penalized matrix decomposition (PMD) method of Witten et al. (2009), where the relationship between the singular value decomposition (SVD) and the Frobenius norm is exploited and optimization is done via a soft-thresholded power method. Chen et al. (2013) develop a canonical pair model and a sparse power algorithm combined with iterative thresholding is applied to estimate the precision matrices. The

alternating regression approach (Waaijenborg et al., 2008; Wilms and Croux, 2015a) avoids the computation of covariance matrices and considers problem (4.1) from a predictive point of view. Wilms and Croux (2015a) derive sparse directions by applying sparsity-inducing regression estimators like the least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996) or its robustification, sparse least trimmed squares (sparseLTS), introduced by Alfons et al. (2013). Gu and Wang (2020) combine the alternating regression approach with an alternating direction method of multipliers (ADMM) algorithm for an $L1$ penalized setting, and Shu et al. (2020) describe a CCA method suitable for high-dimensional data based on methods identifying common and distinctive components such as joint and individual variation explained (JIVE) or simultaneous component analysis with rotation to common and distinctive components (DISCO-SCA).

To the best of our knowledge, the only robust *and* sparse method that does not require the repeated computation and inversion of high-dimensional covariance matrices is based on alternating regressions (Wilms and Croux, 2015b). For higher-order associations, however, there is no efficient implementation available. Several authors propose to use *deflated data matrices* for computing higher-order correlations (Alfons et al., 2016a; Wilms and Croux, 2015b). However, this approach requires solving several regression problems and can potentially destroy sparsity. Wilms and Croux (2015b) address this by applying a sparsity-inducing regression estimator.

The optimization problems (4.1)–(4.3) based on robust correlation or estimators of the covariance matrix lead to highly non-convex objective functions. To obtain a problem formulation that is easier to optimize, the robust estimation and the optimization are *decoupled*: In the first step, the covariance is estimated robustly. This estimator of the covariance matrix is then plugged into the subsequent problem formulation, yielding a biconvex problem. Sparsity can be introduced by adding appropriate constraints. Witten et al. (2009) suggest a similar formulation of the optimization problem for the non-robust case, and an iterative method is presented. Our method, however, also considers the denominator in (4.3), and offers flexibility in the choice of sparsity constraints as well as for the estimator of the covariance matrix. Since rank-based estimators of the covariance matrix will be considered as well, we will use the terminology "(robust) association measure" instead of "canonical correlation coefficient", and simply "linear combinations" instead of "canonical vectors" in the following.

The remainder of the paper is organized as follows: First, the reformulation of the problem is detailed, and an appropriate algorithm for its numerical solution is introduced. Then, the results of a simulation study are presented to illustrate the suitability of our approach for a high-dimensional setting with outliers and to compare its performance to existing approaches. We conclude with an outlook on other common statistical tasks that can be solved by applying the algorithm in a similar way.

4.2 Robust and sparse maximum association

4.2.1 Formulation as a constrained optimization problem

The optimization problems stated in Section 4.1 can also be formulated as constrained optimization problem (see, e.g., Anderson, 1958). This problem formulation has the ad-

vantage that the conditions for uncorrelatedness for directions of higher order and sparsity-inducing penalty terms can be stated directly and added as constraints. Starting from expression (4.3), $\boldsymbol{\Sigma}_{xx}, \boldsymbol{\Sigma}_{yy}$, and $\boldsymbol{\Sigma}_{xy}$ are substituted with suitable estimators for the covariance, denoted by $\boldsymbol{C}_{xx}, \boldsymbol{C}_{yy}$, and $\boldsymbol{C}_{xy}$. Then, the first order maximum association coefficient ρ_1 and the corresponding vectors $(\boldsymbol{a}_1, \boldsymbol{b}_1)$ can be obtained as a solution to the following optimization problem:

$$\min_{\boldsymbol{a}\in\mathbb{R}^p, \boldsymbol{b}\in\mathbb{R}^q} -F(\boldsymbol{a}, \boldsymbol{b}) \tag{4.4}$$

with $F : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R} : \; F(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}'\boldsymbol{C}_{xy}\boldsymbol{b}$ under the constraints

$$\boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a} = 1, \tag{4.5}$$
$$\boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b} = 1. \tag{4.6}$$

This problem formulation avoids the repeated evaluation of the correlation measure that is needed for a projection-pursuit approach as suggested by Alfons et al. (2016a). The covariance needs to be estimated only once and is then fixed for the optimization process. For higher-order coefficients ρ_k and vectors $(\boldsymbol{a}_k, \boldsymbol{b}_k)$, $k \in \{2, \ldots, \min(p, q)\}$, constraints for uncorrelatedness with the lower-order directions are needed:

$$\boldsymbol{a}_k'\boldsymbol{C}_{xx}\boldsymbol{a}_i = 0, \qquad i = 1, \ldots, k-1, \tag{4.7}$$
$$\boldsymbol{b}_k'\boldsymbol{C}_{yy}\boldsymbol{b}_i = 0, \qquad i = 1, \ldots, k-1. \tag{4.8}$$

Especially for the high-dimensional setting, where p and/or q are big, it can be desirable to set some coefficients to zero in the vectors for the linear combinations. Thus, penalty terms can be added as further constraints in the form of

$$P_{a_k}(\boldsymbol{a}_k) \leq c_{a_k}, \tag{4.9}$$
$$P_{b_k}(\boldsymbol{b}_k) \leq c_{b_k}, \tag{4.10}$$

where c_{a_k} and c_{b_k} denote positive constants. Here, the penalty terms (4.9)–(4.10) are taken as elastic net penalties

$$P_{a_k}(\boldsymbol{u}) = \alpha_{a_k}\|\boldsymbol{u}\|_1 + (1 - \alpha_{a_k})\|\boldsymbol{u}\|_2^2, \tag{4.11}$$
$$P_{b_k}(\boldsymbol{u}) = \alpha_{b_k}\|\boldsymbol{u}\|_1 + (1 - \alpha_{b_k})\|\boldsymbol{u}\|_2^2, \tag{4.12}$$

but other (convex) penalties are also applicable.

Witten et al. (2009) also suggest formulating CCA as an optimization problem and derive the canonical directions via an iterative power method. Our approach is more general in that (i) there are no additional assumptions imposed on the covariance, and (ii) the penalty function can be adapted for each order and can also differ for $\boldsymbol{a}$ and $\boldsymbol{b}$.

4.2.2 Robust estimation of the covariance matrix

The choice of a suitable estimator of the covariance matrix is crucial for obtaining robust estimators of the canonical vectors (Alfons et al., 2016a). The robustness of the estimator

of the covariance matrix and its stability in the high-dimensional case will influence the respective properties of the resulting coefficients and vectors (Taskinen et al., 2006). In this work, the focus is on the following estimators: For a base result, the sample covariance matrix is used to estimate $\boldsymbol{\Sigma}_{xx}, \boldsymbol{\Sigma}_{yy}$, and $\boldsymbol{\Sigma}_{xy}$, which corresponds to using the Pearson correlation coefficient as measure of association. To achieve robustness and to allow for the high-dimensional case, the minimum regularized covariance determinant (MRCD) (Boudt et al., 2020) and orthogonalized Gnanadesikan-Kettenring (OGK) (Maronna and Zamar, 2002) estimators are used to estimate the joint covariance matrix of $\boldsymbol{x}$ and $\boldsymbol{y}$, which is afterward decomposed into the matrices $\boldsymbol{C}_{xx}, \boldsymbol{C}_{yy}$, and $\boldsymbol{C}_{xy}$. The MRCD estimator is based on minimizing the determinant of a *regularized* covariance matrix over all possible subsets of a given size $h \leq n$, and it can also be seen as a robust version of the Ledoit-Wolf estimator (Ledoit and Wolf, 2004). The OGK estimator relies on applying the identity $\text{Cov}(x, y) = (\sigma(x + y)^2 - \sigma(x - y)^2)/4$, where σ is the standard deviation and x and y denotes a pair of random variables. This identity is applied for the pairwise combinations of the components in the joint vector of $\boldsymbol{x}$ and $\boldsymbol{y}$, by using a robust scale estimator. The final robust estimator of the covariance matrix is obtained after an orthogonalization step. Note that this result is not necessarily positive definite and eigenvalue correction to obtain a positive definite estimator of the covariance matrix is applied. For both the MRCD and the OGK estimator, the implementations in the package `rrcov` (Todorov and Filzmoser, 2009a) for the statistical computing environment R (R Core Team, 2023) are used. The OGK estimator is thereby applied with the default settings (using the initial covariance as proposed by Gnanadesikan and Kettenring (1972) and the τ scale (Yohai and Zamar, 1988) for univariate location and dispersion). For MRCD, the size of the h subset, controlled by the parameter α, is set to 75% of the number of observations.

As an alternative, pairwise correlation estimators based on Spearman's rank and Kendall's *tau* are also investigated. They can easily be computed in the high-dimensional case as well and have desirable robustness properties (Croux and Dehon, 2010; Alfons et al., 2016a). Denote $\boldsymbol{R}$ as the resulting correlation matrix of the joint vector of $\boldsymbol{x} = (x_1, \ldots, x_p)'$ and $\boldsymbol{y} = (y_1, \ldots, y_q)'$, and $\boldsymbol{D} = \text{diag}(\sigma(x_1), \ldots, \sigma(x_p), \sigma(y_1), \ldots, \boldsymbol{\sigma}(y_q))$, where σ corresponds to a (robust) scale estimate. Then the joint covariance is obtained as $\boldsymbol{DRD}$. For σ we used the median absolute deviation (MAD). Note that for asymptotic normality, it is necessary to apply the transformation $s_{ij} = \frac{6}{\pi} \arcsin\left(\frac{r_{ij}^S}{2}\right)$ to the raw Spearman's rank correlation coefficient r_{ij}^S, and the transformation $\tau_{ij} = \frac{2}{\pi} \arcsin(r_{ij}^K)$ to the raw Kendall's *tau* coefficient r_{ij}^K, where the indices i and j refer to a pair of univariate variables. As Langworthy et al. (2020) point out, a potential issue with those covariance matrices based on pairwise estimation is that they are not necessarily positive definite. Various methods have been proposed to adjust the estimated covariance matrix so that it is positive definite. Rousseeuw and Molenberghs (1993), for example, discuss transformations based on shrinkage and eigenvalues. Higham (2002) presents a method to find the nearest correlation matrix in the Frobenius norm. This algorithm is implemented as the function `nearPD()` in the R package `Matrix` (Bates et al., 2023). While the positive definiteness of the estimator of the covariance matrix is necessary for the existence of a solution, the presented algorithm does not rely on this property for the computation of the maximum association and corresponding

linear combinations. However, the results may not be reliable and in our implementation of the proposed algorithm in the R package `RobSparseMVA` (see Section 4.3 for more information), it is possible to include a check for positive definiteness and apply the `nearPD()` transformation in case the assumption is violated before starting the optimization algorithm.

4.2.3 Lagrangian formulation

The Lagrangian function related to the optimization problem (4.4)–(4.10) is given by

$$\mathcal{L}(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\lambda}) = -\boldsymbol{a}'\boldsymbol{C}_{xy}\boldsymbol{b} + \boldsymbol{\lambda}' \cdot H(\boldsymbol{a}, \boldsymbol{b}), \tag{4.13}$$

while the constraints are given by

$$H : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^{2k+2} : H(\boldsymbol{a}, \boldsymbol{b}) = \begin{cases} \boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a} - 1 \\ \boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b} - 1 \\ P_1(\boldsymbol{a}) - c_a \\ P_2(\boldsymbol{b}) - c_b \\ \boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a}_{1:(k-1)} \\ \boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b}_{1:(k-1)} \end{cases}. \tag{4.14}$$

For obtaining the first order association coefficients and vectors, the Lagrange multipliers are $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_4)'$, and by setting the derivative of $\mathcal{L}$ to $\boldsymbol{0}$, we obtain

$$-\boldsymbol{C}_{xy}\boldsymbol{b} + \lambda_1\boldsymbol{C}_{xx}\boldsymbol{a} + \lambda_3\frac{\partial}{\partial \boldsymbol{a}}P_{\boldsymbol{a}_1}(\boldsymbol{a}) = \boldsymbol{0}, \tag{4.15}$$

$$-\boldsymbol{C}_{xy}\boldsymbol{a} + \lambda_2\boldsymbol{C}_{yy}\boldsymbol{a} + \lambda_4\frac{\partial}{\partial \boldsymbol{b}}P_{\boldsymbol{b}_1}(\boldsymbol{b}) = \boldsymbol{0}. \tag{4.16}$$

When $P_{\boldsymbol{a}_k}$ and $P_{\boldsymbol{b}_k}$ are given as elastic net penalties, the derivatives can be written as

$$\frac{\partial}{\partial \boldsymbol{u}}P(\boldsymbol{u}) = \alpha\boldsymbol{M}_1\boldsymbol{u} + (1-\alpha)\boldsymbol{M}_2\boldsymbol{u} \tag{4.17}$$

with $\boldsymbol{M}_1 = \operatorname{diag}(1/|u_1|, \ldots, 1/|u_p|)$ for $\boldsymbol{u} \in \mathbb{R}^p$ and $\boldsymbol{M}_2 = 1/\|u\|_2\boldsymbol{I} = c\boldsymbol{I}$. The derivatives of the penalty function do not exist at entries $u_i = 0$. Then, the *subgradient* at this point is used instead. Let $\boldsymbol{M}_u := \alpha\boldsymbol{M}_1 + (1-\alpha)\boldsymbol{M}_2$ denote the resulting matrix. For a function $f : \mathbb{R}^p \to \mathbb{R}$, the *subgradient* at $\boldsymbol{x} \in \boldsymbol{dom} f$ is defined as the set of vectors $\boldsymbol{g} \in \mathbb{R}^p$, such that for all $\boldsymbol{z} \in \boldsymbol{dom} f$, there holds: $f(\boldsymbol{z}) \geq f(\boldsymbol{x}) + \boldsymbol{g}'(\boldsymbol{z} - \boldsymbol{x})$. In the present case, f corresponds to the absolute value function and $\boldsymbol{g} \in [-1, 1]^{p+q}$ for $\boldsymbol{x} = \boldsymbol{0}$ (see, e.g., Boyd and Vandenberghe, 2004).

Substitution in Equations (4.15) and (4.16) followed by applying an inverse transformation yields

$$\left[\boldsymbol{C}_{xx} + \frac{\lambda_3}{\lambda_1}\boldsymbol{M}_a\right]^{-1}\boldsymbol{C}_{xy}\left[\boldsymbol{C}_{yy} + \frac{\lambda_4}{\lambda_2}\boldsymbol{M}_b\right]^{-1}\boldsymbol{C}_{yx}\boldsymbol{a} = \lambda_1\lambda_2\boldsymbol{a}, \tag{4.18}$$

$$\left[\boldsymbol{C}_{yy} + \frac{\lambda_4}{\lambda_2}\boldsymbol{M}_b\right]^{-1}\boldsymbol{C}_{yx}\left[\boldsymbol{C}_{xx} + \frac{\lambda_3}{\lambda_1}\boldsymbol{M}_a\right]^{-1}\boldsymbol{C}_{xy}\boldsymbol{b} = \lambda_1\lambda_2\boldsymbol{b}. \tag{4.19}$$

It can be seen that in the case of an L_2 penalty on $\boldsymbol{a}$ or $\boldsymbol{b}$, respectively, this formulation corresponds to a regularization of the estimators of the covariance matrix $\boldsymbol{C}_{xx}$ and $\boldsymbol{C}_{yy}$.

An optimization problem is feasible if there exists at least one point that satisfies the constraints of the problem. In the following we show that with an appropriate choice of the sparsity parameters c_{a_k} and c_{b_k}, the stated optimization is feasible, implying that a solution exists. Let $\Omega \subset \mathbb{R}^p \times \mathbb{R}^q$ denote the set of points that satisfy the constraints (4.5)–(4.10). To show that $\Omega \neq \emptyset$, first note that $\boldsymbol{C}_{xx}$ and $\boldsymbol{C}_{yy}$ are positive definite matrices that induce a norm on $\mathbb{R}^p$ and $\mathbb{R}^q$, respectively. Constraints (4.5) and (4.7) are fulfilled by any basis of $\mathbb{R}^p$ that is orthonormal with respect to the norm induced by $\boldsymbol{C}_{xx}$. The same argument can be applied to constraints (4.6) and (4.8) with the norm induced by $\boldsymbol{C}_{yy}$. From the equivalence of norms, it follows that there exists a positive constant $c_{a_k} \in \mathbb{R}$ such that $1/c_{a_k} P_{a_k}(\boldsymbol{a}_k) \leq 1 = \|\boldsymbol{a}_k\|_{\boldsymbol{C}_{xx}}$, and a positive constant $c_{b_k} \in \mathbb{R}$ such that $1/c_{b_k} P_{b_k}(\boldsymbol{b}_k) \leq 1 = \|\boldsymbol{b}_k\|_{\boldsymbol{C}_{yy}}$.

It follows that the optimization problem (4.4)–(4.6) attains a global minimum over Ω: The function F in (4.4) is continuous and the feasible region $\Omega \subset \mathbb{R}^p \times \mathbb{R}^q$ is non-empty and compact. Then by Weierstrass' theorem, the function F attains a global minimum over Ω.

4.3 Algorithm

The conditions (4.5)–(4.6) in the constrained optimization problem (4.4)–(4.10) are not convex. However, they can be modified to be convex by replacing the equality with an inequality constraint:

$$\boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a} \leq 1 \tag{4.5a}$$
$$\boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b} \leq 1. \tag{4.6a}$$

The modified optimization problem is now biconvex (that is, convex in $\boldsymbol{a}$ if $\boldsymbol{b}$ is fixed and vice versa) and has, under the condition that the constants c_{a_k} and c_{b_k} are chosen such that $\boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a} \geq 1$ and $\boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b} \geq 1$ hold, the same solution as the original problem (see, e.g., Boyd and Vandenberghe, 2004), as cited by Witten et al. (2009).

Using the *augmented Lagrangian* or *method of multipliers (MM)* (see, e.g., Boyd and Vandenberghe, 2004), the problem can be rewritten as a minimization problem in an unconstrained form. The MM-algorithm has been studied extensively by Bertsekas (1996), and the ADMM variation has been brought back more recently due to its potential for distributed computing (Boyd et al., 2011). The main idea of the MM approach is to convert the constrained optimization problem to a series of unconstrained problems. The augmented Lagrangian function is given by

$$\mathcal{L}_c(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\lambda}) = -F(\boldsymbol{a}, \boldsymbol{b}) + \boldsymbol{\lambda}' \cdot H(\boldsymbol{a}, \boldsymbol{b}) + \frac{c}{2}\|H(\boldsymbol{a}, \boldsymbol{b})\|_2^2 \,, \tag{4.20}$$

Algorithm 1 Sparse and robust maximum association

1: Estimate covariance matrices $\boldsymbol{C}_{xx}, \boldsymbol{C}_{yy}, \boldsymbol{C}_{xy}$
2: Initialize $\boldsymbol{a}_k^0$ and $\boldsymbol{b}_k^0$
3: **for** $k = 1, 2, \ldots, \min(p, q)$ **do**
4: $\quad \boldsymbol{\lambda}^0 \leftarrow H(\boldsymbol{a}_k^0, \boldsymbol{b}_k^0, \boldsymbol{a}_{1:(k-1)}, \boldsymbol{b}_{1:(k-1)})$
5: $\quad$ **while** $\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^t\| > \delta$ **do**
6: $\quad\quad (\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1}) \leftarrow \operatorname{argmin} \mathcal{L}_c(\boldsymbol{a}_k^t, \boldsymbol{b}_k^t; \boldsymbol{\lambda}^t)$
7: $\quad\quad \boldsymbol{\lambda}^{t+1} \leftarrow \boldsymbol{\lambda}^t + cH(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})$
8: $\quad\quad$ **if** $0.25 \cdot |H(\boldsymbol{a}_k^t, \boldsymbol{b}_k^t)| < |H(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})|$ **then**
9: $\quad\quad\quad c \leftarrow 10 \cdot c$
10: $\quad\quad$ **end if**
11: $\quad\quad t \leftarrow t + 1$
12: $\quad$ **end while**
13: **end for**

where F denotes the primal objective. The constraints are given by

$$H : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^{2k+2} : H(\boldsymbol{a}, \boldsymbol{b}) = \begin{cases} \boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a} - 1 \\ \boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b} - 1 \\ \boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a}_{1:(k-1)} \\ \boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b}_{1:(k-1)} \\ P_1(\boldsymbol{a}) - c_a \\ P_2(\boldsymbol{b}) - c_b \end{cases} . \tag{4.21}$$

The corresponding Lagrange multipliers are denoted by $\boldsymbol{\lambda} \in \mathbb{R}^{2k+2}$, and the strength of the regularization term for the equality constraints is given by $c \in \mathbb{R}$. This problem can be solved iteratively: in an alternating fashion, first $\boldsymbol{a}$ and $\boldsymbol{b}$ are updated, then the dual variable $\boldsymbol{\lambda}$ is updated. The resulting algorithm for the sparse and robust maximum association procedure is provided in Algorithm 1.

As the solution to the minimization problem in line 6 of Algorithm 1 cannot be derived analytically in the general case, the minimization is done by adaptive gradient descent as introduced by Kingma and Ba (2015) and refined by Reddi et al. (2018). The minimization step is done using the AMSGrad optimizer, given in Algorithm 2 and implemented in the R package `torch` (Falbel and Luraschi, 2023). The maximum and division in lines 7 and 8, respectively, are executed element-wise, and in the thresholding step in lines 15 – 16, a_{k_j} and b_{k_j} refer to the components of $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$, respectively.

Other gradient-based optimizers could also be applied. Methods using an adaptive learning rate and momentum such as AMSGrad or Adam are preferred choices, as they are capable of escaping local optima and are less sensitive to the initial choice of the learning rate α_0. All constraints are subdifferentiable (i.e., for all points in the domain of $\mathcal{L}_c$, at least one subgradient exists), and the subgradient update as implemented in `torch` can be executed. In addition, a thresholding step is included in the algorithm (lines 15–16) to get true sparsity, which is not possible from the subgradient update alone. Thresholding is

Algorithm 2 AMSGrad algorithm (Reddi et al., 2018) for the minimization in line 6 of Algorithm 1, followed by a thresholding step.

1: Input $\boldsymbol{a}_k^0$, $\boldsymbol{b}_k^0$ and $\eta_{1i}, \eta_2, \alpha_i$
2: Initialize $\boldsymbol{m}_0 = \boldsymbol{0}$, $\boldsymbol{v}_0 = \boldsymbol{0}$, $\hat{\boldsymbol{v}}_0 = \boldsymbol{0}$
3: **while** $\|\nabla_{a,b}\mathcal{L}_c\| > \delta$ **do**
4: $\quad \boldsymbol{g}_i \leftarrow \nabla_{a,b}\mathcal{L}_c$
5: $\quad \boldsymbol{m}_i \leftarrow \eta_{1i}\boldsymbol{m}_{i-1} + (1-\eta_{1i})\boldsymbol{g}_i$
6: $\quad \boldsymbol{v}_i \leftarrow \eta_2\boldsymbol{v}_{i-1} + (1-\eta_2)\boldsymbol{g}_i^2$
7: $\quad \hat{\boldsymbol{v}}_i \leftarrow \max(\hat{\boldsymbol{v}}_{i-1}, \boldsymbol{v}_i)$
8: $\quad (\boldsymbol{a}_k^i, \boldsymbol{b}_k^i) \leftarrow (\boldsymbol{a}_k^{i-1}, \boldsymbol{b}_k^{i-1}) - \alpha_i \frac{\boldsymbol{m}_i}{\sqrt{\hat{\boldsymbol{v}}_i}}$
9: $\quad d_a^i \leftarrow \frac{\|\boldsymbol{a}_k^i - \boldsymbol{a}_k^{i-1}\|}{\|\boldsymbol{a}_k^{i-1}\|}$
10: $\quad d_b^i \leftarrow \frac{\|\boldsymbol{b}_k^i - \boldsymbol{b}_k^{i-1}\|}{\|\boldsymbol{b}_k^{i-1}\|}$
11: $\quad i \leftarrow i+1$
12: **end while**
13: $\bar{t}_a \leftarrow \text{avg}[d_a^m]_{m=i}^{i-M+1} + 2\text{sd}[d_a^m]_{m=i}^{i-M+1}$
14: $\bar{t}_b \leftarrow \text{avg}[d_b^m]_{m=i}^{i-M+1} + 2\text{sd}[d_b^m]_{m=i}^{i-M+1}$
15: $\boldsymbol{a}_k \leftarrow [a_{k_j} \text{ if } |a_{k_j}| > \bar{t}_a, 0 \text{ otherwise}]_{j=1}^{p}$
16: $\boldsymbol{b}_k \leftarrow [b_{k_j} \text{ if } |b_{k_j}| > \bar{t}_b, 0 \text{ otherwise}]_{j=1}^{q}$

done using the moving average of the last M step sizes; in practice we were successful with setting $M = 10$. Depending on the current value of H, the regularization parameter c is updated in lines 8–9 of Algorithm 1. The constants 0.25 and 10 in lines 8–9 were already proposed by Bertsekas (1996) and work well in our simulations.

Biconvex optimization problems are commonly treated in an alternating manner, for the problem (4.4)–(4.10) that would suggest updating $\boldsymbol{a}$ while fixing $\boldsymbol{b}$ and vice versa (this course of action would correspond to the ADMM algorithm). Even though the partial problems are convex, in general, there is no guarantee that the ADMM converges to the global (or even local) optimum in this case. Therefore, instead of alternating the updates of $\boldsymbol{a}$ and $\boldsymbol{b}$, we propose to perform the update at the same time with a gradient-descent-type algorithm. This way, the algorithm converges towards a solution that satisfies the necessary optimality condition of stationarity of the Lagrange function $\mathcal{L}_0$: via gradient-descent, we are able to identify a stationary point of $\mathcal{L}_c(\boldsymbol{a}_k^t, \boldsymbol{b}_k^t; \boldsymbol{\lambda}^t)$, denoted by $(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})$. This point fulfills $0 \in \nabla\mathcal{L}_c(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1}; \boldsymbol{\lambda}^t) = -\nabla F(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1}) + \boldsymbol{\lambda}^{t'}\nabla H(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1}) + cH(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})\nabla H(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})$.
With the update $\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + cH(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1})$, we then have $0 \in \nabla\mathcal{L}_0(\boldsymbol{a}_k^{t+1}, \boldsymbol{b}_k^{t+1}; \boldsymbol{\lambda}^{t+1})$.

In the optimization problem derived from classical CCA, only the canonical vectors (of all orders) satisfy the stationarity condition. Theoretically, the algorithm could end up in a stationary point corresponding to the maximum association and the respective linear combinations of a different order. However, our simulations indicate that this is not a problem in practice, as the applied variant of gradient descent is able to escape local optima.

4.3.1 Hyperparameter optimization

Another important aspect of the algorithm is the choice of the hyperparameters. The mixing parameters α_{a_k} and α_{b_k} of the elastic net penalties in (4.11) and (4.12) are often set in advance by the user, but the sparsity parameters c_{a_k} and c_{b_k} have to be determined in a data-driven manner. Grid search in combination with cross-validation quickly becomes infeasible if the search space becomes larger, especially in more than one dimension. An alternative is *Bayesian optimization* of the given hyperparameters. An introduction to Bayesian optimization for hyperparameter optimization can be found, for example, in Frazier (2018). The benefit of using Bayesian optimization instead of grid search is that the information from previous function evaluations can be used to determine the best next point to execute the function. This way, a much bigger search space can be covered, and, in addition, it is less likely to miss a good parameter configuration due to the size of the grid. For the basic algorithm, it is assumed that there is a budget of in total N function evaluations. We also need to define a score to be maximized during the hyperparameter optimization, and an appropriate acquisition function. A Gaussian prior is placed on the score function, then its value is observed at n_0 points. Until N iterations are reached, the following steps are repeated: (i) update the posterior probability distribution on the score function, (ii) determine the maximum of the acquisition function, and (iii) observe the score value at this parameter configuration.

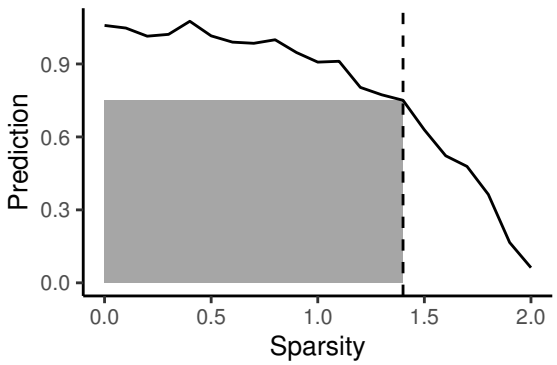


Figure 4.1: Visualization of the tradeoff product optimization (TPO) criterion (4.22) used as a score function in Bayesian hyperparameter optimization. The TPO score corresponds to finding the biggest area under the curve of prediction (robust association measure) over sparsity.

We used the implementation in the R package `ParBayesianOptimization` (Wilson, 2022) with the expected improvement as acquisition function and the tradeoff product optimization (TPO) as score function. It is similar to the TPO criterion used by Filzmoser et al. (2022) and models the tradeoff between sparsity in the estimated linear combinations, $\hat{\boldsymbol{a}}_k$ and $\hat{\boldsymbol{b}}_k$, and the estimated value $\hat{\rho}_k$ of the robust association measure. Figure 4.1 illustrates

this criterion. The original criterion

$$\text{score} = |\hat{\rho}_k| \cdot \left(2 - \frac{\#\{\hat{\boldsymbol{a}}_k \neq 0\}}{p} - \frac{\#\{\hat{\boldsymbol{b}}_k \neq 0\}}{q}\right)$$

where $\#\{\hat{\boldsymbol{a}}_k \neq 0\}$ returns the number of non-zero components in $\hat{\boldsymbol{a}}_k$, and similar for $\hat{\boldsymbol{b}}_k$, can be adapted for non-sparse regularization by including the elastic net parameters α_{a_k} and α_{b_k}:

$$\text{score} = |\hat{\rho}_k| \cdot \left(2 - \alpha_{a_k}\frac{\#\{\boldsymbol{a}_k \neq 0\}}{p} - \alpha_{b_k}\frac{\#\{\boldsymbol{b}_k \neq 0\}}{q}\right). \tag{4.22}$$

Both the sparsity and elastic net parameters can be chosen differently for each k and $\hat{\boldsymbol{a}}_k$ or $\hat{\boldsymbol{b}}_k$, respectively. In our simulations, presented in Section 4.4, the chosen elastic net parameters α_{a_k} and α_{b_k} are assumed to be the same for each k, while the Bayesian optimization procedure to determine the optimal sparsity parameters is run for each k. Furthermore, Section 4.4.2 of the simulation study is dedicated to the precision of the algorithm, i.e., the performance of the algorithm when the true covariance matrix and theoretically optimal sparsity parameters are provided.

4.3.2 Initialization

For the presented algorithm, suitable starting values $\boldsymbol{a}_k^0$ and $\boldsymbol{b}_k^0$ for the linear combinations $\boldsymbol{a}_k$ and $\boldsymbol{b}_k$, respectively, and for the associated Lagrange multiplier $\boldsymbol{\lambda}_k$ are needed. For the elements of $\boldsymbol{a}_1^0$ and $\boldsymbol{b}_1^0$, we use the average contribution of the respective row or column in $\boldsymbol{C}_{xy} = [c(x_i, y_j)]$ as a starting value,

$$a_{1_i}^0 = \frac{1}{q}\sum_{j=1}^{q} c(x_i, y_j) \quad \text{for} \quad i = 1, \ldots, p\ , \tag{4.23}$$

$$b_{1_j}^0 = \frac{1}{p}\sum_{i=1}^{p} c(x_i, y_j) \quad \text{for} \quad j = 1, \ldots, q\ , \tag{4.24}$$

and for the Langrange multipliers, the contraints are evaluated at the starting points,

$$\boldsymbol{\lambda}_k^0 = H(\boldsymbol{a}_k^0, \boldsymbol{b}_k^0). \tag{4.25}$$

If a non-robust estimator of the covariance matrix is used, the starting values may already be influenced by outlying observations. A more detailed investigation of what happens when the number of outlying observations is increased is given in Section 4.4.2.

For the computation of directions of higher order, the concept of "deflated" data matrices is often described in the literature (e.g. Branco et al., 2003). However, this step can be detrimental to the sparsity in the higher-order vectors. In our approach, constraints for uncorrelatedness to lower-order directions are added to the model. Higher-order directions need to satisfy Equations (4.7) and (4.8), respectively. Basically, this means that $\boldsymbol{a}_k$ is in the left null space of $\boldsymbol{C}_{xx}\boldsymbol{a}^{(i:k-1)}$ and $\boldsymbol{b}_k$ is in the left null space of $\boldsymbol{C}_{yy}\boldsymbol{b}^{(i:k-1)}$. These affine

constraints preserve the biconvexity and suggest the following variation for determining the starting values for higher-order linear combinations: The orthogonal complements $\boldsymbol{A}_k^{\perp} := \{\boldsymbol{a} : \boldsymbol{a}'\boldsymbol{C}_{xx}\boldsymbol{a}^{(i:k-1)} = 0\}$ and $\boldsymbol{B}_k^{\perp} := \{\boldsymbol{b} : \boldsymbol{b}'\boldsymbol{C}_{yy}\boldsymbol{b}^{(i:k-1)} = 0\}$ are computed, then the starting values $\boldsymbol{a}_k^0$ and $\boldsymbol{b}_k^0$ are chosen as the orthogonal projections of $\boldsymbol{a}_1^0$ and $\boldsymbol{b}_1^0$ on $\boldsymbol{A}_k^{\perp}$ and $\boldsymbol{B}_k^{\perp}$, respectively.

In our simulations, both the naive approach and this "orthogonal" initialization for the higher-order linear combinations are compared.

4.4 Simulation study

A simulation study was conducted to compare the performance of the proposed method using different (robust) estimators of the covariance matrix. The comparison is also done with other approaches already mentioned in Section 4.1, namely PMD by Witten et al. (2009), and SRAR by Wilms and Croux (2015a). For PMD, the R package `PMA` (Witten and Tibshirani, 2020) was used, for SRAR the code available from `https://sites.google.com/view/iwilms/software` was used. Our algorithm is implemented in the R package `RobSparseMVA` and available online `https://github.com/piapfeiffer/RobSparseMVA`.

A good sparse and robust method should be efficient when the number of variables grows, avoid misidentifying important variables, and should attain these properties in the presence of outliers in the data (see, e.g., Zou, 2006; Todorov and Filzmoser, 2013). In order to check those requirements, the following *performance measures* are used. For measuring accuracy, the angle $\theta_a = \arccos\left(\frac{\boldsymbol{a}'\hat{\boldsymbol{a}}}{\|\boldsymbol{a}\|\cdot\|\hat{\boldsymbol{a}}\|}\right)$ between the true and estimated canonical variables is computed. Note that only the results for one of the linear combinations are presented here as the results for the other are qualitatively similar. The true-positive rate (TPR), corresponding to the rate of correctly identified non-zero components, together with the true-negative rate (TNR), or the rate of correctly identified zero components, measure whether non-zero variables are identified correctly. For studying the scalability, the runtime for a growing number of variables is measured.

For the computation of above performance measures, the true linear combinations $\boldsymbol{a}$ and $\boldsymbol{b}$ have to be computed: They can be derived from the true covariance matrices $\boldsymbol{\Sigma}_{xx}$, $\boldsymbol{\Sigma}_{xy}$, and $\boldsymbol{\Sigma}_{yy}$ as eigenvectors of $\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}$ and $\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$, respectively, see Section 4.1.

4.4.1 Simulation design

Different simulation settings and contamination scenarios (similar to Wilms and Croux, 2015b) are considered. Clean data are generated from a multivariate normal distribution: $(\boldsymbol{x}, \boldsymbol{y})' \sim \mathcal{N}_{p+q}(\boldsymbol{0}, \boldsymbol{\Sigma})$. For the contaminated scenario, c_r% contamination is generated from a multivariate normal distribution with a mean shift: $(\boldsymbol{x}, \boldsymbol{y})' \sim \mathcal{N}_{p+q}(c_s \cdot \boldsymbol{1}, \boldsymbol{\Sigma})$, where c_s denotes the contamination strength. To simulate a heavy-tailed distribution, data are generated from a multivariate t-distribution: $(\boldsymbol{x}, \boldsymbol{y})' \sim t_3(\boldsymbol{0}, \boldsymbol{\Sigma})$. The joint covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}_{xy}' & \boldsymbol{\Sigma}_{yy} \end{bmatrix}$ is given according to the following simulation settings:

1. Low-dimensional, order 2: $p = q = 10, n = 100$ observations

$$\begin{aligned}
\boldsymbol{\Sigma}_{xx} &= \boldsymbol{I}_{10} \\
\boldsymbol{\Sigma}_{yy} &= \boldsymbol{I}_{10} \\
\boldsymbol{\Sigma}_{xy} &= \begin{bmatrix} 0.9 & 0 & \mathbf{0}_{1\times 8} \\ 0 & 0.7 & \mathbf{0}_{1\times 8} \\ \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 8} \end{bmatrix}
\end{aligned}$$

   The true associations in this setting are $\rho_1 = 0.9$ and $\rho_2 = 0.7$, and the true linear combinations are $\boldsymbol{a}_1 = \boldsymbol{b}_1 = (1, \mathbf{0}_{1\times 9})'$ and $\boldsymbol{a}_2 = \boldsymbol{b}_2 = (0, 1, \mathbf{0}_{1\times 8})'$.

2. High-dimensional, order 2: $p = q = 100, n = 50$ observations

$$\begin{aligned}
\boldsymbol{\Sigma}_{xx} &= \begin{bmatrix} \boldsymbol{S}^{\mathbf{1}}_{10\times 10} & \mathbf{0}_{10\times 10} & \mathbf{0}_{10\times 80} \\ \mathbf{0}_{10\times 10} & \boldsymbol{S}^{\mathbf{2}}_{10\times 10} & \mathbf{0}_{10\times 80} \\ \mathbf{0}_{80\times 10} & \mathbf{0}_{80\times 10} & \boldsymbol{I}_{80\times 80} \end{bmatrix} \\
\boldsymbol{\Sigma}_{yy} &= \boldsymbol{\Sigma}_{xx} \\
\boldsymbol{\Sigma}_{xy} &= \begin{bmatrix} \mathbf{0.9}_{10\times 10} & \mathbf{0}_{10\times 10} & \mathbf{0}_{10\times 80} \\ \mathbf{0}_{10\times 10} & \mathbf{0.5}_{10\times 10} & \mathbf{0}_{10\times 80} \\ \mathbf{0}_{80\times 10} & \mathbf{0}_{80\times 10} & \mathbf{0}_{80\times 80} \end{bmatrix}
\end{aligned}$$

   where $\boldsymbol{S}^{\mathbf{1}}_{ij} = 1$ if $i = j$ and $\boldsymbol{S}^{\mathbf{1}}_{ij} = 0.9$ for $i \neq j$ and $\boldsymbol{S}^{\mathbf{2}}_{ij} = 1$ if $i = j$ and $\boldsymbol{S}^{\mathbf{2}}_{ij} = 0.7$ for $i \neq j$. The true associations in this setting are $\rho_1 = 0.989$ and $\rho_2 = 0.685$, and the true linear combinations are $\boldsymbol{a}_1 = \boldsymbol{b}_1 = (\mathbf{0.105}_{1\times 10}, \mathbf{0}_{1\times 90})'$ and $\boldsymbol{a}_2 = \boldsymbol{b}_2 = (\mathbf{0}_{1\times 10}, \mathbf{0.117}_{1\times 10}, \mathbf{0}_{1\times 80})'$.

4.4.2 Simulation results

Precision of the algorithm

For evaluating the precision of the algorithm, we avoid estimating the covariance matrix but plug in the true covariance matrix $\boldsymbol{\Sigma}$ into the algorithm. We also compare the precision when the theoretically optimal sparsity parameters are provided (which can be derived from the true linear combinations $(\boldsymbol{a}, \boldsymbol{b})$ as $c_a = \|\boldsymbol{a}\|_1$ and $c_b = \|\boldsymbol{b}\|_1$) and when the sparsity parameters are estimated via Bayesian optimization. The results of these (deterministic) computations are presented in Table 4.1. For the first-order association measure, the algorithm always converges to the true solution. For higher-order association measures, it can be seen that the sparsity parameters need to be chosen with more care and that the orthogonal start is necessary for the high-dimensional setting. Furthermore, it can be concluded that starting with an orthogonal projection for the second-order directions is beneficial. Therefore, the results of all subsequent simulations are shown for this initialization.

Table 4.1: Performance measures given the true covariance. The true association measures for the low-dimensional and high-dimensional setting are $\rho_1 = 0.9$ and $\rho_2 = 0.7$, and $\rho_1 = 0.989$ and $\rho_2 = 0.685$, respectively.

Setting	Sparsity	Initialization	Order	Angle θ_a	TPR	TNR	Association
Low-dimensional	known	naive	1	0	1	1	0.9
		naive	2	0	1	0.89	0.7
		orthogonal	2	0	1	1	0.7
	estimated	naive	1	0	1	1	0.9
		naive	2	0	1	0.89	0.7
		orthogonal	2	0	1	1	0.7
High-dimensional	known	naive	1	0	1	1	0.989
		naive	2	1.57	0	0.89	0.989
		orthogonal	2	0.23	1	1	0.683
	estimated	naive	1	0	1	1	0.989
		naive	2	1.57	0	0.89	0.989
		orthogonal	2	0	1	1	0.685

Comparison to other methods

For the given scenarios and settings, the proposed method using different estimators of the covariance matrix is compared to SRAR by Wilms and Croux (2015b) and sparse CCA via PMD by Witten et al. (2009) in terms of estimation accuracy, measured by the angle θ_a, and sparsity control, measured by the TPR and TNR. For our algorithm, we used the orthogonal initialization to compute the second-order association. The naive initialization leads to worse results (not shown here).

The results over 100 repetitions for estimators of the covariance matrix used in our algorithm (left of the dashed line) with SRAR (Wilms and Croux, 2015b) and PMD (Witten et al., 2009) are summarized in Figure 4.2. The left column presents the results for the low-dimensional setting, the right column for the high-dimensional setting. The different plot symbols encode different contamination scenarios; black presents first-order results, and gray second-order results. Shown are the mean values over 100 repetitions for the metrics, together with error bars representing the standard error range. We present the results for uncontaminated data, for contamination of 5% of the observations with contamination strength $c_s = 2$, and for data generated from a multivariate t_3 distribution, see Section 4.4.1.

The results for the low-dimensional setting are shown on the left-hand side of Figure 4.2. The angle, TPR, and TNR are only presented for the estimated canonical variables $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$. For the non-contaminated data, the performance across all methods is similar. Estimating the second-order component leads to slightly worse results - with the exception of PMD, where the results are highly precise. The behavior under contamination and heavy tails is still comparable to the non-contaminated case; only SRAR and PMD have problems identifying the correct sparsity. For PMD, this is especially apparent in the figures depicting the TPR and TNR: Both for the first-order and second-order linear combinations, the TNR is 1, but the TPR is only around 0.5, indicating that the resulting linear combinations are too sparse.

For the high-dimensional setting (right-hand side of Figure 4.2), more differences can

be observed: When the data are uncontaminated, our algorithm based on the Pearson correlation is highly accurate. This can be in part attributed to the regularization effect on the estimators of the covariance matrix described in Equations (4.18)–(4.19). While the performance for the high-dimensional setting is overall worse, and all methods suffer from a decrease in accuracy for the second-order canonical vectors, it can be observed that for the robust OGK and MRCD estimators, the accuracy level for the second-order component is the same as it already is for the first-order component for SRAR (Figure 4.2: top-right). Our algorithm with robust estimators of the covariance matrix shows superior performance for the TPR. Especially interesting is the good result for the covariance based on Spearman's rank and Kendall's *tau* in the scenario using the t-distribution. This finding coincides with the work presented in Langworthy et al. (2020). The TNR is good for all variants of our algorithm and for PMD in the clean setting. SRAR performs worse, and similarly to the accuracy, the TNR for the second-order components estimated using our algorithm is on the same level as the TNR for SRAR in the first-order components. Difficulties with accuracy and estimating the correct sparsity are also reflected in the resulting association measure.

Increasing contamination

For both the low-dimensional and high-dimensional settings described in Section 4.4.1, the contamination proportion was increased from $cp = 0\%$ to $cp = 50\%$, and again, the same performance measures were evaluated. The results averaged over 50 repetitions comparing the different robust estimators are shown in Figure 4.3. On the left side, the performance metrics for the low-dimensional setting are given, on the right side, the results for the high-dimensional setting are shown. The different line types correspond to the performance metrics for our method using different estimators for the covariance matrix (Pearson, Spearman, Kendall, OGK, MRCD) and the sparse and robust alternating regressions (SRAR) technique on the other hand. The results for the penalized matrix decomposition (PMD) are omitted, as it is already affected by a small proportion of outlying observations. While the metrics for accuracy show in the low-dimensional setting an advantage for the alternating regressions approach, the other metrics are comparable across the methods. For the high-dimensional setting, the results are more distinguished: The proposed method outperforms the SRAR approach, while the robustness against an increased contamination proportion depends on the estimator of the covariance matrix. In these plots it is clearly visible that the h-subset parameter for MRCD was set to $\alpha = 0.75$, as the performance gets much worse over a contamination proportion of $cp = 25\%$. The OGK estimator also exhibits interesting behavior: The TPR metric decrease up to a contamination proportion of 40%, then it jumps to 1. The reason can be observed when analyzing the TNR metric: After a contamination proportion of 40% is reached, all components are identified to be non-zero. Another interesting observation is that while for all estimators the performance metrics decline, the predicted association measure still seems to be reliable. The contamination strength (mean shift) was also varied between $c_s = 2$ and $c_s = 10$, but was not found to have an effect on the performance of the different estimators. Therefore, these results are omitted here.

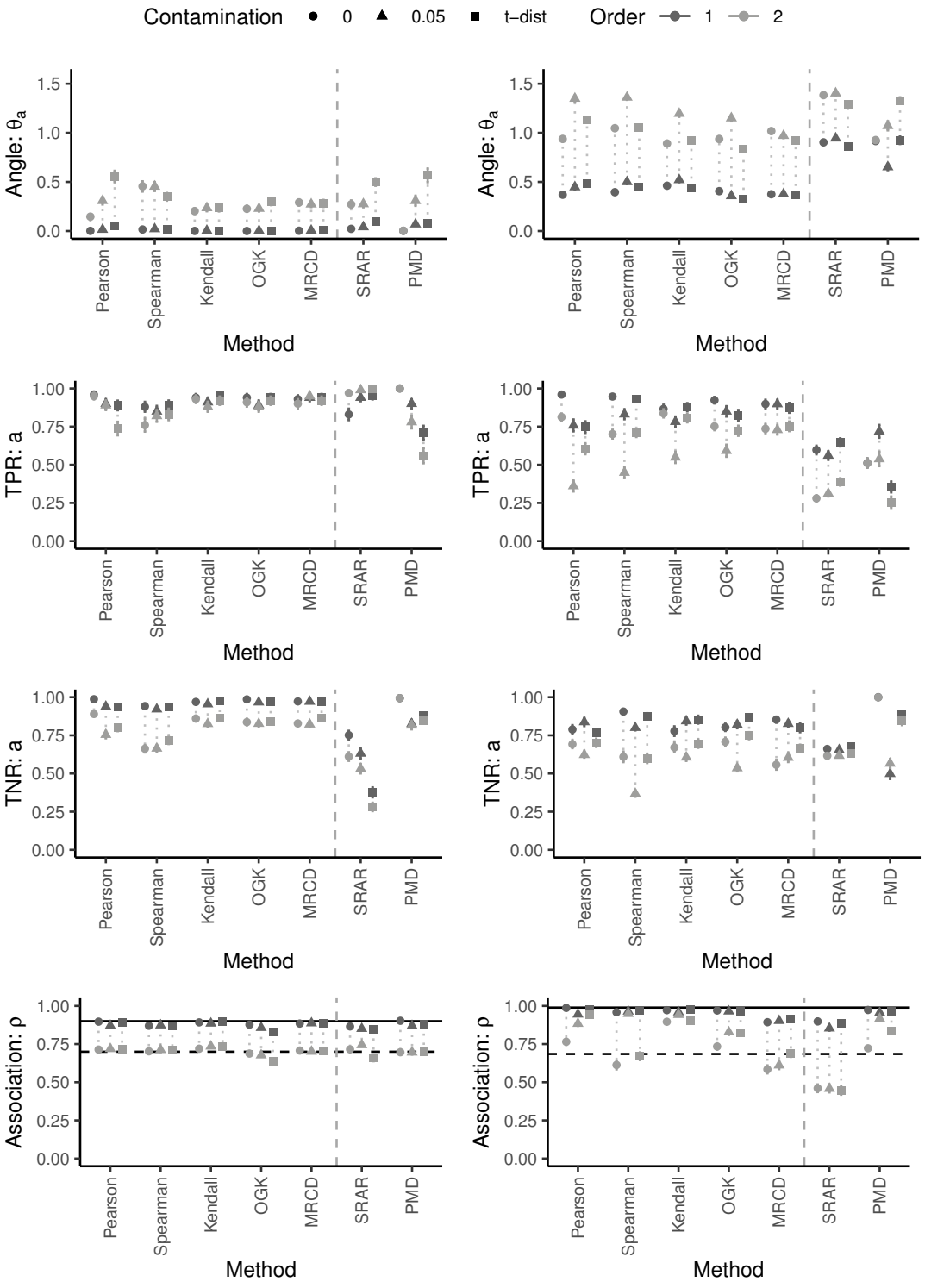


Figure 4.2: Comparison of performance measures for the different methods.

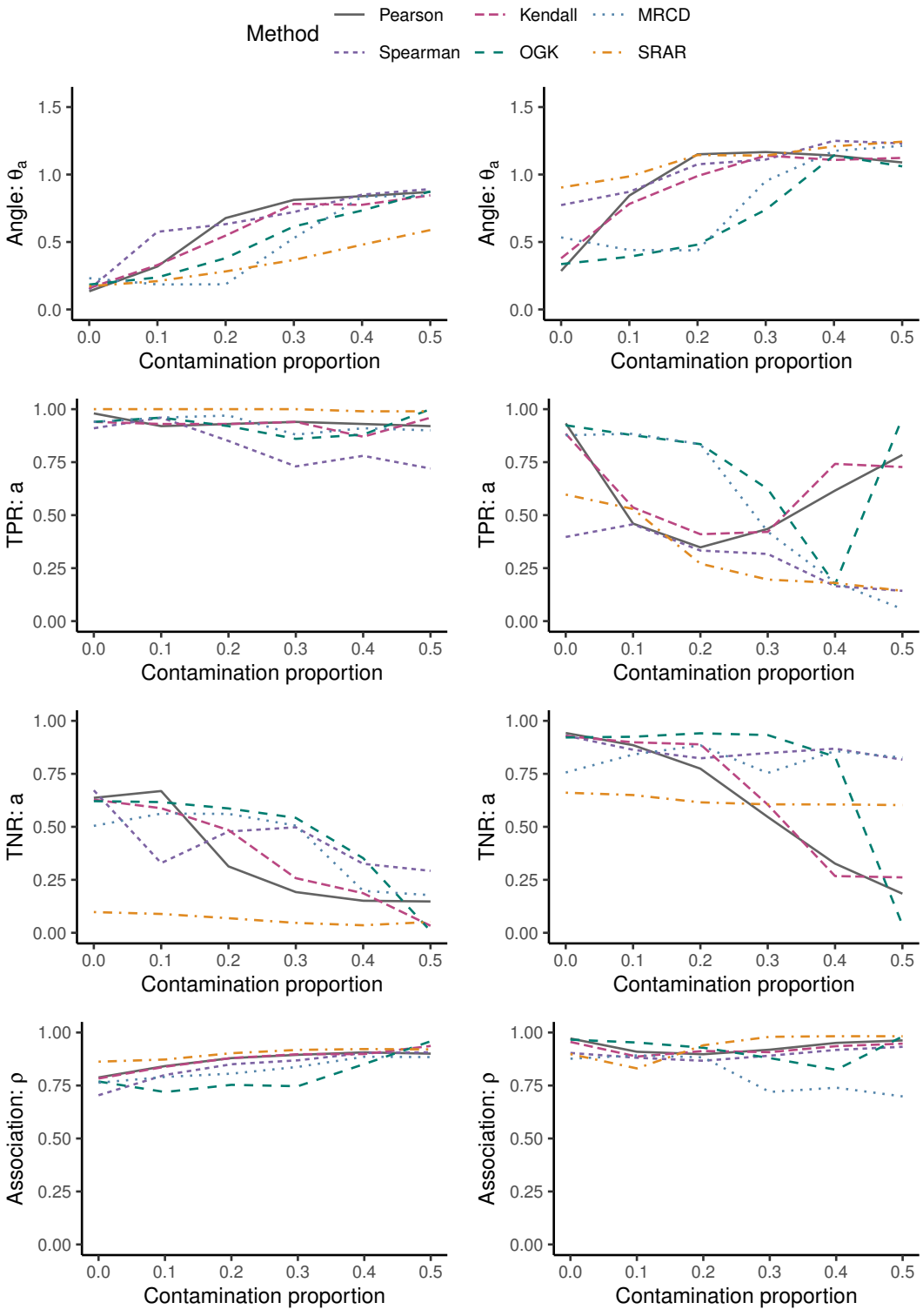


Figure 4.3: Increasing contamination ratio for different (robust) methods.

Runtime

For evaluating the runtime, the first-order canonical directions and association were computed using our algorithm and SRAR for increasing dimension $q = 50, \ldots, 10000$ while keeping the dimensionality of the other side, $p = 10$, and the number of samples, $n = 100$, fixed. The joint covariance matrix $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy} \\ \boldsymbol{\Sigma}'_{xy} & \boldsymbol{\Sigma}_{yy} \end{bmatrix}$ is given according to the settings:

$$
\begin{aligned}
\boldsymbol{\Sigma}_{xx} &= \left[\boldsymbol{S}^{\mathbf{1}}_{10\times 10}\right] \\
\boldsymbol{\Sigma}_{yy} &= \begin{bmatrix} \boldsymbol{S}^{\mathbf{1}}_{10\times 10} & \mathbf{0}_{10\times(q-10)} \\ \mathbf{0}_{(q-10)\times 10} & \boldsymbol{I}_{(q-10)\times(q-10)} \end{bmatrix} \\
\boldsymbol{\Sigma}_{xy} &= \begin{bmatrix} \mathbf{0.8}_{10\times 10} & \mathbf{0}_{10\times(q-10)} \\ \mathbf{0}_{(q-10)\times 10} & \mathbf{0}_{(q-10)\times(q-10)} \end{bmatrix}
\end{aligned}
$$

where $\boldsymbol{S}^{\mathbf{1}}_{ij} = 1$ if $i = j$ and $\boldsymbol{S}^{\mathbf{1}}_{ij} = 0.8$ for $i \neq j$.

To remove the effect of the hyperparameter search, fixed optimal sparsity parameters were used for all methods. The results are averaged over 10 replications and presented in Figure 4.4. The increasing dimension q is shown on the horizontal axis on a log scale, and the CPU runtime of the algorithm is shown on the vertical axis in minutes. It can be observed that the presented algorithm based on adaptive gradient descent has a comparable dependence of runtime to problem size to the only other robust and sparse alternative. However, it is obvious that the runtime heavily depends on the type of association estimator used, as the joint covariance needs to be estimated fully before starting the optimization. The covariance matrix only needs to be estimated once, and if it is already available, gradient descent scales better to high-dimensional data than regression-type algorithms. In combination with pairwise estimators of the covariance matrix, which are also easy to compute in high dimensions, the proposed algorithm could also serve as a more time-efficient alternative to alternating regressions.

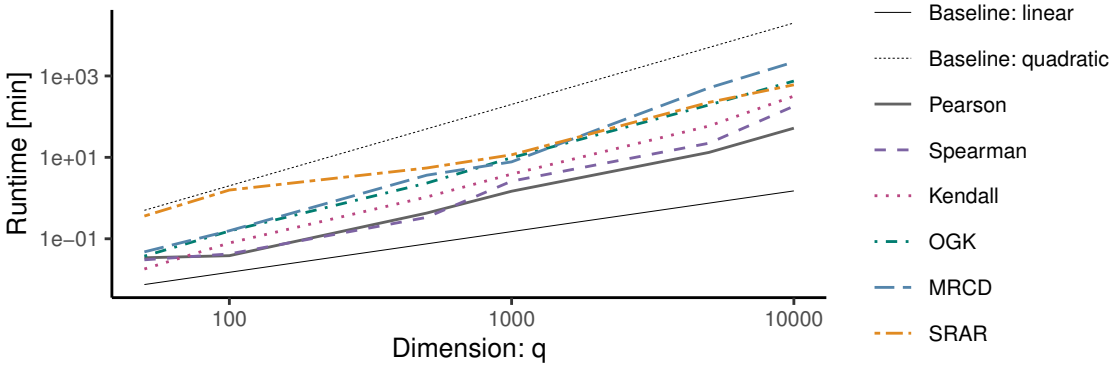


Figure 4.4: Log-log plot of CPU runtime in minutes versus dimensionality q of second variable.

In summary, the simulation results demonstrate the comparable or better performance of our proposed method, depending on the contamination scenario. Especially for the

high-dimensional scenario, our method in combination with an appropriate estimator for the covariance matrix performs well, also for higher-order directions. When runtime is of concern, the method based on Spearman or Kendall correlation should be considered, as these measures have better robustness properties than Pearson correlation, while still being efficient to compute. In general, the OGK estimator seems to be a sensible choice, leading to good performance metrics in both low-dimensional and high-dimensional settings and for an increased contamination ratio.

4.5 Examples

In this section, the proposed method will be applied to two data sets from different fields: biology and tribology. For both, the number of observations is significantly lower than the number of variables, and the flexibility of choosing the sparsity via the elastic-net penalty for each side is desirable.

In order to compare the performance of classical and robust estimation, we compare the out-of-sample performance of the robust and non-robust estimators by randomly splitting the data into a training and test set and computing the out-of-sample residual score

$$r = \frac{1}{n_{\text{test}}} \frac{1}{n_{\text{rep}}} \sum_{j=1}^{n_{\text{test}}} \sum_{i=1}^{n_{\text{rep}}} \|\boldsymbol{a}'_{\text{train}_i} \boldsymbol{x}_j - \boldsymbol{b}'_{\text{train}_i} \boldsymbol{y}_j\|^2, \tag{4.26}$$

where $\boldsymbol{a}_{\text{train}_i}$ and $\boldsymbol{b}_{\text{train}_i}$ are the first order linear combinations that are estimated on the i-th training set, n_{test} is the number of test set observations, and $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ are observations from the corresponding test sets. The random training and test splits are repeated n_{rep} times. Since outliers in the test sets could contaminate this residual score measure, we also use a trimmed version, by trimming the largest 10% of all the squared test residuals and dividing by the corresponding number of observations. We also report the (in-sample) association measure, averaged over n_{rep} random training and test splits.

4.5.1 Application to the `nutrimouse` dataset

The `nutrimouse` dataset is publicly available via the `CCA` package in R (González and Déjean, 2021) and has been discussed in the related literature, for example, by Wilms and Croux (2015b). It contains $n = 40$ observations of $p = 120$ gene expressions and $q = 21$ concentrations of fatty acids. Martin et al. (2007) provide a detailed description of the dataset, and investigate the influence of a certain diet on numerous gene expressions in mice. In this setting, the goal is to identify a set of genes that has a large association with a set of lipids (cf. Wilms and Croux, 2015b). The two datasets were robustly centered and scaled with median and MAD before continuing with the analysis. We compare the results of the sparse CCA method by Witten et al. (2009) with the proposed method by computing the residual score r, see Equation (4.26) in a leave-one-out cross-validation (CV) setting, i.e., the i-th training set consists of all observations except i, and the i-th test set only contains observation i.

The results are given in Table 4.2: The residual score for the robust method is much lower, the same holds for the trimmed version. The estimated association is 0.76 for PMD

and 0.89 for our method using the OGK estimator. Additionally, the contributions to the residual scores for both methods are presented as a scatterplot in Figure 4.5. Most of the points are above the 45° line, i.e. smaller for the robust method. Similar to Wilms and Croux (2015b) we can conclude that the out-of-sample performance of the robust method is better, and therefore present the estimated linear combinations of this method in Figure 4.6.

Table 4.2: Residual scores and association measure based on data splitting for the `nutrimouse` dataset.

Method	r	r (trimmed)	Association
sparse CCA	2.15	1.10	0.76
OGK	0.49	0.18	0.89

Out of the $p = 120$ gene expressions, 54 are selected by the algorithm, and out of $q = 21$ fatty acids, 16 are selected. Upon comparison with the selected variables presented by Wilms and Croux (2015b), we can identify the following sets of fatty acids that have been selected by both the SRAR method and ours: C.20.1n.9, C.20.2n.6 and C22.4n.6 with the highest (absolute) coefficients. Among the fatty acids related to the diets of the mice, namely C22:6n-3, C22:5n-3, C22:5n-6, C22:4n-3, and C20:5n-3 (see Martin et al., 2007), 5 are selected by our method. For the gene expressions, the coefficient values differ more than for the fatty acids. CYP3A11, which has been found to have a significant influence by Martin et al. (2007), is selected by both methods. The influence of diet on Lpin1 is also discussed by Martin et al. (2007) and has the highest absolute coefficient in our model.

4.5.2 Application in tribology

Fourier-transform infrared spectroscopy (FTIR) spectra to monitor a lubricant's degradation process and their relation to different indicators for oil condition have been studied by several authors (see, for example, Pfeiffer et al., 2022). However, the goal is to also understand the association between lubricant chemistry and lubrication performance. Pfeiffer and Filzmoser (2023) demonstrated how features from optical data of wear scar areas can be extracted and used in a robust partial least-squares (PLS) model to relate the wear scar to oil condition, measured by alteration duration.

We demonstrate that by using the method presented in this paper, the two high-dimensional datasets can be associated directly. The dataset consists of $n = 214$ observations, FTIR spectra with $p = 1668$ variables, and HoG (histogram of gradients) feature vectors, with $q = 1836$ variables, representing the wear scar images. As previous studies have shown, while sparsity is beneficial for the evaluation of FTIR spectra, it does not yield good results for HoG image features (Pfeiffer and Filzmoser, 2023). This example also illustrates the flexibility of our approach, being able to choose a sparsity-inducing penalty on one side, while applying L_2-regularization on the other. Before proceeding with the analysis (independent from which estimator of the covariance matrix was used), the HoG feature vectors were robustly centered and scaled using median and MAD, respectively.

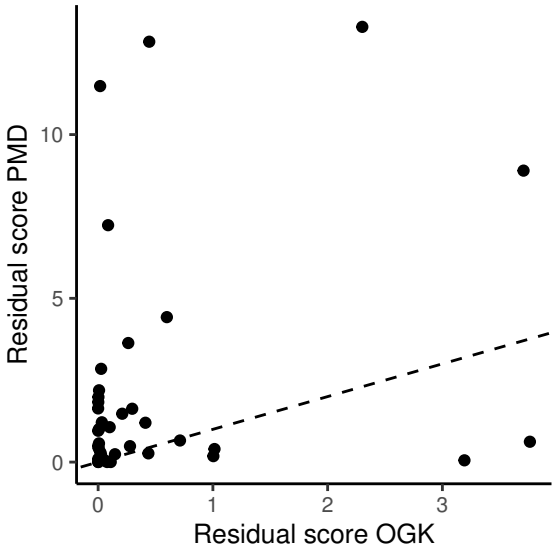


Figure 4.5: Scatterplot of leave-one-out cross-validation (CV) scores for PMD and the proposed method using the OGK estimator for the `nutrimouse` dataset. Almost all points are above the 45° line, indicating a better out-of-sample performance for the robust estimator.

We compute the residual score, see Equation (4.26), for a 90%–10% split, repeated 5 times. Here we run the algorithm using the Pearson correlation (sample covariance matrix), and as a robust counterpart we use the OGK estimator of the covariance matrix. The resulting residual scores are given in Table 4.3. It can be observed that the overall out-of-sample performance for the robust estimator is clearly better than for the classical one, while the estimated association is comparable.

Table 4.3: Residual scores and association measure based on data splitting for the tribology dataset.

Method	**r**	**r (trimmed)**	**Association**
Pearson	138.51	121.14	0.24
OGK	52.7	48.40	0.21

In Figure 4.7 (left), the wavenumbers of the FTIR spectra selected by the non-robust and the robust procedures are displayed. For the non-robust method, 79 wavenumbers are selected, and for the robust one 73, with an overlap of 52 non-zero elements in the two linear combinations. The selected wavenumbers between 1860-1660 cm^{-1} are known to be related to oxidation processes, while wavenumbers between 3651-3649 cm^{-1} correspond to phenolic antioxidants (Ronai, 2021). The selected wavenumbers are similar for the non-robust and robust estimators. For the HoG feature vectors, however, a difference in the size of the coefficients can be observed. The coefficient values for the sample covariance are

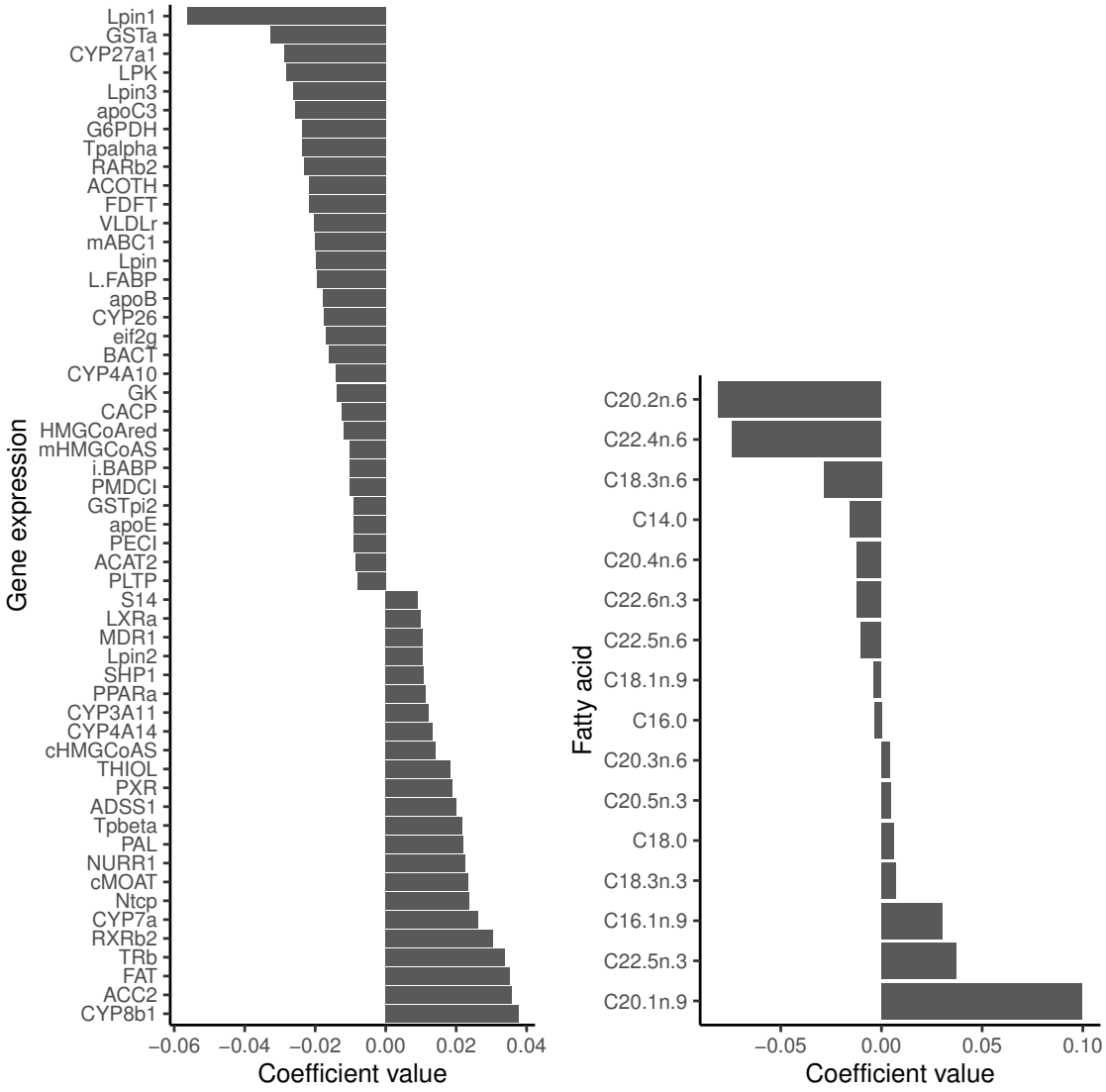
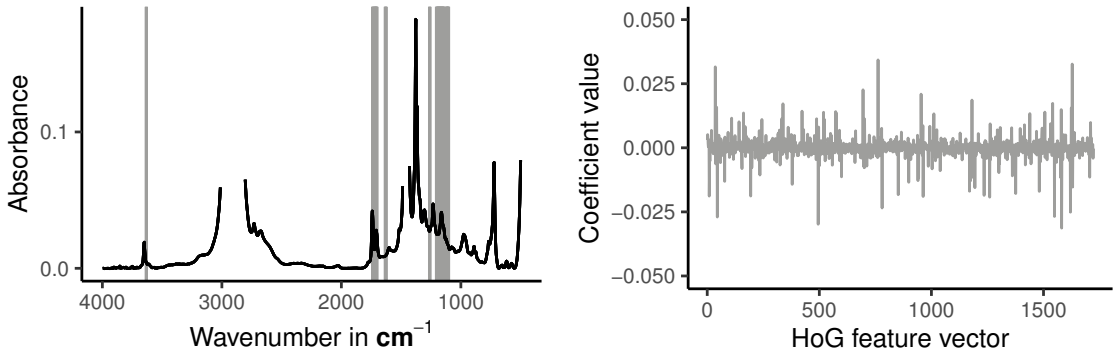


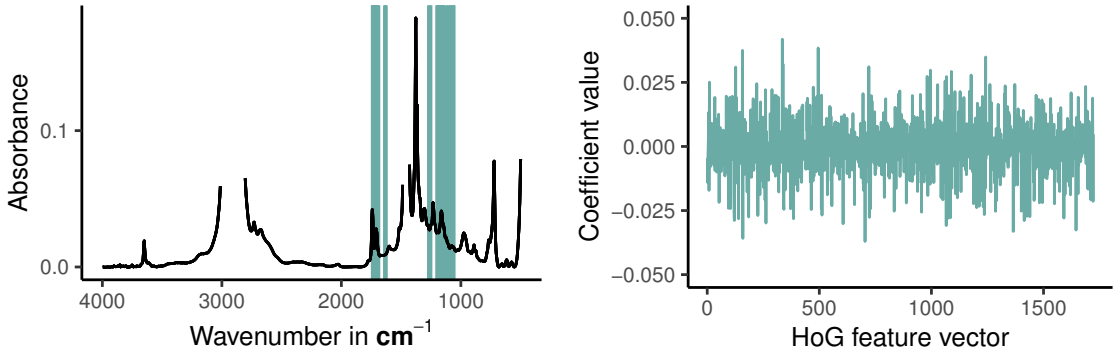
Figure 4.6: Estimated linear combinations using the OGK estimator of the covariance matrix for the `nutrimouse` dataset.

lower, which can be explained by a stronger regularization parameter being chosen during hyperparameter optimization. Note that the coefficients shown in Figure 4.7b and Figure 4.7d are normalized such that $\boldsymbol{b}'\boldsymbol{C}\boldsymbol{b} = 1$. As the HoG features have been extracted from images of wear scar areas, outliers can be expected to be present (cf. Pfeiffer and Filzmoser, 2023). While such outliers drive the non-robust method towards over-regularization of these features, the robust method based on the OGK estimator reduces the influence of outliers and is able to determine appropriate coefficient values. An example of wear scar images corresponding to outliers identified by the OGK estimator is given in Figure 4.8.



(a) Selected wavenumbers using the sample covariance.

(b) Coefficient values for HoG features using the sample covariance.



(c) Selected wavenumbers using the OGK estimator.

(d) Coefficient values for HoG features using the OGK estimator.

Figure 4.7: Selected wavenumbers (left) using the sample covariance (top, grey) and the OGK estimator (bottom, green). The plots on the right-hand side show the coefficient vector for the HoG features (classical on top, robust at the bottom). For these, no sparsity penalty was included.

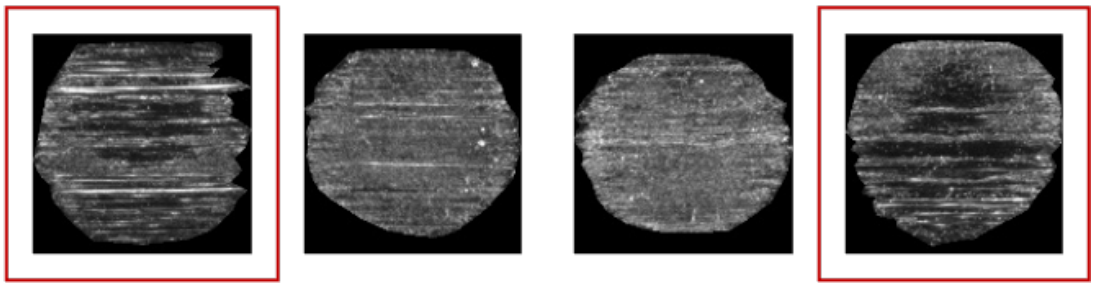


Figure 4.8: Wear scar areas on the ball for a duration of 552 hours of oil alteration. The framed images correspond to HoG features that were identified as outliers by the OGK estimator.

4.6 Summary and conclusions

We focused on the problem of maximizing the association between linear combinations of two sets of random variables (two multivariate data sets referring to the same observations). If the association measure is taken as the Pearson correlation, this corresponds to the framework of canonical correlation analysis. However, more general association measures can be considered, even allowing for an identification of monotone rather than just linear relationships. The stated problem can also be formulated as a constrained maximization problem, where the joint covariance of the two random variables is involved, and the coefficients for the linear combinations need to be identified. This is the problem considered in this paper, with two extensions: (i) we aim for robustness of the association measure against outliers in one or both data sets, and (ii) we want to have sparsity in one or both vectors of the linear combinations for applications in high dimensions. In this approach, robustness can easily be achieved by plugging in a robustly estimated covariance matrix. Several proposals for this purpose exist in the literature, also for high-dimensional data. We also investigated pairwise estimators of the covariance matrix, e.g., based on Spearman's rank correlation.

The main contribution of the paper is the development of an efficient algorithm to solve the optimization problem. The formulation in terms of a Lagrange problem makes the use of a gradient descent algorithm possible, where constraints for the linear combinations can easily be incorporated. The minimum requirements for the penalty functions are that they are convex and that (sub)gradients exist. Other than that, the presented algorithm allows flexibility in the choice of penalty functions, and—as seen in one of the examples—also enables to induce sparsity in only one component, while the other is L_2 regularized.

From a computational perspective, the main advantage of our approach is the scalability to a high number of variables. In comparison to solving a regression problem repeatedly (Wilms and Croux, 2015b), or to scanning a larger and larger search space in circles (Alfons et al., 2016b), an algorithm based on gradient descent is more efficient, as only the gradient information needs to be stored. Although the estimation of the plug-in covariance matrix can be computationally demanding, covariance estimation just needs to be done once. This is different from approaches, e.g., based on projection pursuit, where pairwise correlations or association measures need to be computed for all considered projection directions (Alfons et al., 2016b).

We provided numerical results for the precision and theoretical considerations concerning the existence of a solution of the optimization problem. For this, positive definiteness of the joint covariance matrix is a requirement, however, from the simulations we could see that even when this assumption is violated (high-dimensional setting with pairwise estimators), our algorithm is able to produce comparable or even better results than the alternative techniques. The results emphasize how important a good (and robust!) estimator of the covariance is. As the performance regarding robustness and computation time is dependent on the estimator of the covariance matrix, a sparse and robust estimator could lead to improvements. Avella-Medina et al. (2018) show that thresholding methods (see, e.g., Bickel and Levina, 2008) have desirable properties when a robust initial estimator is used. Extending the available implementation to exploit the sparsity structure of the covariance will be explored in our upcoming research.

Examples with high-dimensional data sets from biology and tribology underline the use-

fulness of our approach: It offers flexibility concerning penalty functions depending on the desired sparsity in each of the data sets, desirable robustness properties and maintains manageable computation times.

The combination of robust estimators and modern optimization techniques yields a powerful toolbox for solving several other common problems in statistics. Especially for robust procedures, where robust estimation (e.g., of a covariance matrix) and optimization can be decoupled, the proposed procedure is very promising. Examples of such extensions are robust principal component analysis and robust linear discriminant analysis, which are topics of our future research.

5 Cellwise robust and sparse principal component analysis

5.1 Introduction

The increasing prevalence of large data sets, especially high-dimensional data in the sense of many more variables than observations, motivates the use and development of dimension-reduction techniques. Principal Component Analysis (PCA), dating back to Pearson (1901) and Hotelling (1933), is one of the oldest and most widely applied dimension-reduction techniques. The idea of PCA is to find a low-dimensional representation of the data set in a way that preserves as much variance as possible (e.g. Jolliffe et al., 2003).

Based on a mean-centered (and possibly scaled) data matrix $\boldsymbol{X}$, with n observations in the rows, and p variables in the columns, the principal components (PCs) are defined by the linear combination $\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{V}$ under the constraint that the columns of the $p \times p$ matrix $\boldsymbol{V}$ are normed to length 1 and orthogonal to each other. Since the variances of the columns of $\boldsymbol{Z}$ have to be maximized, the solution for $\boldsymbol{V}$ can be obtained by the spectral composition $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{V}}\hat{\boldsymbol{A}}\hat{\boldsymbol{V}}'$, where $\hat{\boldsymbol{\Sigma}}$ is the estimated covariance matrix of $\boldsymbol{X}$, and $\hat{\boldsymbol{A}} = \text{Diag}(\hat{a}_1, \ldots, \hat{a}_p)$ is the diagonal matrix with the corresponding estimated eigenvalues, arranged in descending order (Jolliffe et al., 2003). The matrix $\hat{\boldsymbol{V}}$ is also known as loadings matrix, while $\hat{\boldsymbol{Z}} = \boldsymbol{X}\hat{\boldsymbol{V}}$ refers to the PCA score matrix. Traditionally, the sample covariance matrix $\boldsymbol{S} = \frac{1}{n-1}\boldsymbol{X}'\boldsymbol{X}$ is used for $\hat{\boldsymbol{\Sigma}}$, resulting in eigenvectors $\tilde{\boldsymbol{V}}$, eigenvalues $\tilde{\boldsymbol{A}}$, and classical principal components $\tilde{\boldsymbol{Z}} = \boldsymbol{X}\tilde{\boldsymbol{V}}$. The identical solution $\tilde{\boldsymbol{V}}$ can be obtained from a singular value decomposition (SVD) of $\boldsymbol{X}$ as $\boldsymbol{X} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{D}}\tilde{\boldsymbol{V}}'$ (e.g. Jolliffe et al., 2003).

As the main interest is usually in the first k PCs, where $k < p$, or even $k \ll p$ for high-dimensional data, it is not necessary to compute the whole $p \times p$ matrix $\hat{\boldsymbol{V}}$, but to only focus on the matrix $\hat{\boldsymbol{V}}_k \in \mathbb{R}^{p \times k}$ with the first k columns, to obtain the first k PCs $\hat{\boldsymbol{Z}}_k = \boldsymbol{X}\hat{\boldsymbol{V}}_k$. Especially for $p \gg n$, the approach based on a spectral decomposition of the estimated covariance matrix is numerically not attractive, and thus SVD is commonly employed in this case. This leads to a rank-k approximation $\tilde{\boldsymbol{X}}_k = \tilde{\boldsymbol{U}}_k\tilde{\boldsymbol{D}}_k\tilde{\boldsymbol{V}}_k'$ of $\boldsymbol{X}$, with $\tilde{\boldsymbol{U}}_k \in \mathbb{R}^{n \times k}$ and $\tilde{\boldsymbol{D}}_k \in \mathbb{R}^{n \times p}$ with elements $\tilde{d}_{ii} \geq 0$ for $i = 1, \ldots, k$ and $\tilde{d}_{ij} = 0$ otherwise. The rank-k SVD is the best rank-k approximation in the Frobenius norm (Eckart and Young, 1936),

$$\tilde{\boldsymbol{V}}_k = \text{argmin}_{\boldsymbol{V}_k} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{V}_k\boldsymbol{V}_k'\|_F^2 \tag{5.1}$$

for any $p \times k$ matrix $\boldsymbol{V}_k$ with $\text{rank}(\boldsymbol{V}_k) \leq k$ and $\boldsymbol{V}_k'\boldsymbol{V}_k = \boldsymbol{I}_k$. We can define the residual matrix

$$\boldsymbol{R} = \boldsymbol{X} - \boldsymbol{X}\boldsymbol{V}_k\boldsymbol{V}_k' \quad \text{with elements} \quad r_{ij}, \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, p, \tag{5.2}$$

and problem (5.1) is equivalent to minimizing the sum of all squared residuals by using a matrix $\boldsymbol{V}_k$ as defined above.

It is well known that a sum-of-squares criterion is sensitive to outlying entries (Maronna et al., 2019). Such outliers, however, refer to single outlying cells in the residual matrix $\boldsymbol{R}$, and not necessarily to outlying rows (observations). The latter is traditionally considered for robustly estimating a covariance matrix (e.g. Maronna, 1976; Rousseeuw, 1985; Rousseeuw and Leroy, 2005) in order to obtain robust PCs with the help of spectral decomposition (Maronna et al., 2019), but in the case $p > n$ those estimators cannot be computed.

The concept of cellwise outliers has been conceptually introduced in Alqallaf et al. (2009), but already Maronna and Yohai (2008) formulated a cellwise robust PCA version by downweighting outlying cells in the residual matrix rather than outlying rows. In fact, they proposed to replace the Frobenius norm in (5.1) with a more robust loss function. Cellwise robustness has particular advantages if $p \gg n$: Assuming that any data cell has the same chance to be contaminated, even a small amount of contamination can lead to many rowwise outliers, which at some point could even form the majority and cause breakdown of traditional rowwise robust methods (Maronna et al., 2019). A robust PCA method that cannot only deal with cellwise outliers but also with missing values is MacroPCA (Hubert et al., 2019). This method combines the DDC algorithm of Rousseeuw and Bossche (2018) to detect cellwise outliers with a version of ROBPCA (Hubert et al., 2005), a robust PCA method that can also deal with high-dimensional data.

None of the proposed cellwise robust PCA methods lead to sparse solutions. The contribution of this paper is sparsity: In the high-dimensional case it is desirable to obtain (many) zeros in the matrix $\boldsymbol{V}_k$, since this simplifies the interpretation of the PCs. In this paper we will introduce a cellwise robust and sparse PCA method. We first review robust and rowwise sparse PCA methods in Section 5.2 before introducing our method in Section 5.3. An algorithm for its computation based on manifold learning is presented in Section 5.4, and simulation results in Section 5.5 compare with alternative PCA methods. Applications in Section 5.6 demonstrate the usefulness of the method. The final Section 5.7 provides a summary and conclusions.

5.2 Related work

Robust PCA has been a very active research field, and thus many different approaches are available in the literature (see, e.g., She et al., 2016, for an overview). Since our focus is also on sparsity, we provide a short and thus possibly incomplete overview of robust and sparse PCA methods in the following, before discussing cellwise robust methods in this context.

One of the first papers on robust sparse PCA is Candès et al. (2011). However, sparsity here refers to a sparse residual matrix as the remainder of a robust rank-k approximation. In this paper, we are more interested in sparsity for the loadings matrix, as this simplifies the interpretation of the PCs. Although this work is highly visible, the method is intended to deal with additive outliers, and not with outliers in the orthogonal complement to the PCA subspace, see She et al. (2016); Hubert et al. (2005).

As PCA can also be seen as a projection-pursuit (PP) problem, with the task to search for a projection direction that maximizes the variance of the projected observations, Croux and Ruiz-Gazen (2005) introduced a robust procedure by considering a robust variance

(scale) estimator. Inspired by the SCoTLASS approach which adds a LASSO penalty on the direction vectors (Jolliffe et al., 2003), this PP approach was also reformulated to include an L_1 penalty, resulting in a robust and sparse PCA method (Croux et al., 2013).

The ROBPCA method by Hubert et al. (2005) combines a PP approach with the plug-in method by first projecting the data to a low-dimensional subspace and then applying a robust estimator of the covariance. This method was extended in Hubert et al. (2016), who proposed ROSPCA by integrating an L_1-penalty with the ROBPCA algorithm, which also leads to sparse solutions.

Zou (2006) suggest reformulating PCA as a regression problem. Then, sparse loadings can be derived using elastic net (Zou and Hastie, 2003) and LASSO (Tibshirani, 1996) regression. A robust extension that is based on robust plug-in estimators for the covariance was proposed by Greco and Farcomeni (2016). In the case $p > n$ they propose to use the unconstrained ROBPCA solution to obtain a plug-in estimator.

5.3 Cellwise robust sparse PCA for high-dimensional data

Following the ideas outlined in the previous section, the rank-k approximation criterion (5.1) can be combined with a criterion to obtain sparsity. When using an elastic net penalty, the modified problem formulation is

$$\hat{\boldsymbol{V}}_k = \text{argmin}_{\boldsymbol{V}_k'\boldsymbol{V}_k=\boldsymbol{I}_k} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{V}_k\boldsymbol{V}_k'\|_F^2 + \sum_{j=1}^{k} \lambda_j(\alpha\|\boldsymbol{v}_j\|_2^2 + (1-\alpha)\|\boldsymbol{v}_j\|_1), \tag{5.3}$$

where $\boldsymbol{v}_j$ refers to the j-th column of $\boldsymbol{V}_k$, $1 \leq j \leq k < p$, the strength of regularization for each component is controlled by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)'$, and the elastic net mixing parameter α controls the sparsity (Zou and Hastie, 2005).

As the least squares loss is highly susceptible to outlying observations, we propose to substitute it with a robust loss function, similar as suggested in Maronna and Yohai (2008),

$$\hat{\boldsymbol{V}}_k = \text{argmin}_{\boldsymbol{V}_k'\boldsymbol{V}_k=\boldsymbol{I}_k} \frac{1}{np} \sum_{j=1}^{p} \hat{\sigma}_j^2 \sum_{i=1}^{n} \rho\left(\frac{r_{ij}}{\hat{\sigma}_j}\right) + \sum_{j=1}^{k} \lambda_j(\alpha\|\boldsymbol{v}_j\|_2^2 + (1-\alpha)\|\boldsymbol{v}_j\|_1), \tag{5.4}$$

where r_{ij} are the residuals from (5.2), and $\hat{\sigma}_j$ is a column-wise estimator of residual scale. The function ρ corresponds to a robust loss function (Maronna et al., 2019), and popular choices are the Huber loss, defined as

$$\rho_H(r) = \begin{cases} r^2 & \text{for } |r| \leq b, \\ b|r| & \text{otherwise,} \end{cases} \tag{5.5}$$

the Tukey loss, defined as

$$\rho_T(r) = \begin{cases} \left(\frac{r}{c}\right)^2 \left(3 - 3\left(\frac{r}{c}\right)^2 + \left(\frac{r}{c}\right)^4\right) & \text{for } |r| \leq c, \\ 1 & \text{otherwise,} \end{cases} \tag{5.6}$$

or a trimmed version of the least squares loss, defined as

$$\rho_{LTS}(r) = \begin{cases} r^2 & \text{for } |r| \leq |r|_{(h)}, \\ 0 & \text{otherwise,} \end{cases} \tag{5.7}$$

where $|r|_{(i)}$ refers to the ordered values of the absolute residuals, i.e. $|r|_{(1)} \leq \ldots \leq |r|_{(n)}$ and $h \in \lfloor n/2, n \rfloor$. The trimming is applied column-wise.

The choice of the loss function will also determine the robustness properties of $\hat{\boldsymbol{V}}_k$, see also Maronna and Yohai (2008) for more detailed discussions. For either of these loss functions, the influence of data points with large scaled residuals is reduced, resulting in a more robust estimate (Maronna et al., 2019). Appropriate parameter choices for the constant b in (5.5) and (5.6) are proposed in the literature (Maronna et al., 2019). Throughout our experiments we will use the parameters $b = 1.35$, $c = 1.35$, and $h = 0.5$.

Note that a multiplication with $\hat{\sigma}_j^2$ in the objective function (5.4) is important, since this guarantees that for the special choice $\rho(r) = r^2$ one obtains the objective function (5.3) of the non-robust version. The estimation of the residual scale will be discussed in detail in the next section.

For outlier diagnostics, i.e. the distinction between regular observations, good and bad leverage points, and orthogonal outliers, a diagnostic plot can be constructed as proposed by Hubert et al. (2005). The score distances (SD) and orthogonal distances (SD) are computed based on the robust PCA result and plotted together with the cutoff values, also described in detail in Hubert et al. (2005).

5.4 Algorithm

Problem (5.4) is an optimization problem under constraints. The objective function

$$\mathcal{L}(\boldsymbol{V}) = \frac{1}{np} \sum_{j=1}^{p} \hat{\sigma}_j^2 \sum_{i=1}^{n} \rho\left(\frac{r_{ij}(\boldsymbol{V})}{\hat{\sigma}_j}\right) + \sum_{j=1}^{k} \lambda_j (\alpha \|\boldsymbol{v}_j\|_2^2 + (1-\alpha) \sum_{i=1}^{n} \|v_{ij}\|_1) \tag{5.8}$$

is minimized under the constraint that $\boldsymbol{V}'\boldsymbol{V} = \boldsymbol{I}_k$, which corresponds to the Stiefel Manifold, defined as the set of orthonormal matrices, i.e. $\mathrm{St}(p,k) = \{\boldsymbol{V} \in \mathbb{R}^{p \times k} : \boldsymbol{V}'\boldsymbol{V} = \boldsymbol{I}_k\}$.

We can therefore cast this problem in the framework of optimization on manifolds. Gradient methods such as the Newton or Conjugate Gradient algorithm on manifolds have already been studied in Edelman et al. (1998), and Bonnabel (2013) have extended SGD (Stochastic Gradient Descent) to the case when the objective function is defined on a Riemannian manifold and derived convergence properties for the algorithm. When the gradient is calculated on the whole dataset, it is referred to as batch gradient descent in the machine learning literature, see, e.g., Goodfellow et al. (2016). When big amounts of data have to be processed, (minibatch) SGD is preferable, this means that the gradient is computed on each sample, or a minibatch of samples. Mathematically, the true gradient of the objective function corresponds to the expectation of the gradient over the data-generating distribution. For batch gradient descent, the true gradient is approximated by the gradient over the whole training set. When there is a large number of observations, however, computing

this expectation over the whole dataset is not feasible, therefore it is computed over random subsamples, which are called minibatches (Goodfellow et al., 2016). Therefore, by application of a suitable variant of SGD we can ensure the scalability of the proposed algorithm to datasets containing a very large number of observations, which would not be possible with an approach based on alternating regressions, for example. In the following, we describe the algorithm for batch gradient descent for ease of notation.

5.4.1 Manifold optimization

Given a starting point $\boldsymbol{V}_0 \in \mathrm{St}(p, k)$, subsequent iterations lie on the same manifold. This is accomplished as follows: First, the gradient at step t, $\boldsymbol{H}_t := \nabla_{\boldsymbol{V}_t} \mathcal{L}(\boldsymbol{V}_t) = \left(\frac{\partial \mathcal{L}(\boldsymbol{V}_t)}{\partial \boldsymbol{v}_l} \right)_{l=1}^{k}$, is computed, then the gradient is projected on the tangent space of the manifold at the current parameter value, denoted by $\mathcal{T}_{\boldsymbol{V}_t}\mathrm{St}(p, k)$. Let $\boldsymbol{P}_{\boldsymbol{H}_t}$ denote the projection of the gradient, which can be computed as $\boldsymbol{P}_{\boldsymbol{H}_t} = -\gamma_t(\boldsymbol{I}_p - \boldsymbol{V}_t\boldsymbol{V}_t')\boldsymbol{H}_t$, where γ_t refers to the step size.

Finally, the gradient step is executed on the manifold. This can be done via an exponential map, $\boldsymbol{V}_{t+1} \leftarrow \exp_{\boldsymbol{V}_t}(-\gamma_t \boldsymbol{P}_{\boldsymbol{H}_t})$, or via a retraction, a first-order approximation of the exponential map, denoted by $\boldsymbol{V}_{t+1} \leftarrow R_{\boldsymbol{V}_t}(-\gamma_t \boldsymbol{P}_{\boldsymbol{H}_t})$. Numerically, the latter is preferable, and we can use $R_{\boldsymbol{V}_t}(\boldsymbol{P}) = \mathrm{qf}(\boldsymbol{P})$, where qf() extracts the orthogonal factor from the QR decomposition. This particular retraction was also studied in Proposition 3 in Bonnabel (2013) and essentially follows the gradient in the Euclidean space and then orthonormalizes the matrix at each step.

For the computation of the gradient, the continuous differentiability of the loss function ρ is a necessary condition. This assumption is fulfilled for the least squares loss without regularization, for the robust loss functions and different penalties, however, it does not hold. While the Huber loss (5.5) can be approximated by the differentiable Pseudo-Huber loss function (Hartley and Zisserman, 2003), $\rho_{PH}(r) = b^2(\sqrt{1 + (r/b)^2} - 1)$, the treatment of the LTS loss (5.7) requires more thought. It is easy to see though, that a gradient step in terms of the ρ_{LTS} function can be expressed as a re-weighting step: ρ_{LTS} is continuously differentiable on the set of points, for which $|r| \leq |r|_{(h)}$. Let H_t denote this h-subset computed based on the current iterate $\boldsymbol{V}_t$, then $\mathcal{L}(\boldsymbol{V}_t, H_t)$ corresponds to the value of the objective function depending on H_t. After each gradient step, H_{t+1} is updated using the current iterate $\boldsymbol{V}_{t+1}$ and we get the inequality

$$\mathcal{L}(\boldsymbol{V}_{t+1}, H_{t+1}) \leq \mathcal{L}(\boldsymbol{V}_{t+1}, H_t) \leq \mathcal{L}(\boldsymbol{V}_t, H_t). \tag{5.9}$$

Note that for the stochastic variant of the gradient step, the inequality only holds in expectation.

5.4.2 Initialization

As the loss function (5.8) is based on the approximation of the data matrix via rank-k SVD, it seems natural to consider the first k singular values as starting points. Despite the desirable properties of Riemannian SGD studied by Bonnabel (2013), however, we cannot hope to get convergence to a global minimum, as the loss function is not convex on $\mathrm{St}(p, k)$. Therefore, it is crucial to choose an initial estimate $\boldsymbol{V}_0$ that is robust in the presence of outliers.

We propose first applying a transformation g to the data matrix and then computing the SVD of $\boldsymbol{Y} = g(\boldsymbol{X})$, resulting in the first k right-singular vectors of $g(\boldsymbol{X})$ as the initial estimate $\boldsymbol{V}_0$. The procedure is inspired by the robust high-dimensional product-moment correlation, studied in Raymaekers and Rousseeuw (2021) where robustness properties of different data transformations are investigated. In the following, we consider one of the following options for the transformation, and later on, compare their results. Both options involve estimators of location t_j and scale c_j of the variables ($j = 1, \ldots, p$). Here we use the median for location and the Qn estimator (Rousseeuw and Croux, 1993) for scale.

1. Rank transformation: The elements of the transformed data matrix $\boldsymbol{Y}$ are given by $y_{ij} = g(x_{ij}) = {}^1\!/_n(\text{rank}_i(x_{ij}) - 0.5) \cdot c_j + t_j$.

2. Wrapping transformation: Denote $z_{ij} = \frac{x_{ij} - t_j}{c_j}$, for $i = 1, \ldots, n$ and $j = 1, \ldots, p$, as the column-wise robustly standardized data. The elements of the transformed data matrix $\boldsymbol{Y}$ are given by $y_{ij} = g(x_{ij}) = \psi_{b,c}(z_{ij}) \cdot c_j + t_j$. The ψ-function is given by

$$\psi_{b,c}(z) = \begin{cases} z & \text{if } 0 \leq |z| \leq b, \\ q_1 \tanh\left(q_2(c - |z|)\right)\text{sign}(z) & \text{if } b \leq |z| \leq c, \\ 0 & \text{otherwise,} \end{cases} \tag{5.10}$$

   where the values q_1 and q_2 can be derived for any combination of $0 < b < c$ (Raymaekers and Rousseeuw, 2021). We use the default values $b = 1.5$ and $c = 4$ as proposed in Raymaekers and Rousseeuw (2021) and implemented in the function `wrap()` of the R package `cellWise` (Raymaekers and Rousseeuw, 2023a).

5.4.3 Residual scale

Based on an initial estimate $\boldsymbol{V}_0$, we can compute the residuals $\boldsymbol{R}(\boldsymbol{V}_0) = \boldsymbol{X} - \boldsymbol{X}\boldsymbol{V}_0\boldsymbol{V}_0'$ with the elements $r_{ij}(\boldsymbol{V}_0)$. The objective function (5.8) needs an estimate of the residual scale, $\hat{\sigma}_j$, for the j-th column of this matrix ($j = 1, \ldots, p$). Minimizing the objective function then yields an updated estimate of $\boldsymbol{V}$, and thus also new residuals, from which the residual scale needs to be re-estimated. For estimating this residual scale, Maronna and Yohai (2008) suggest to use an M estimator of scale. As the proposed algorithm requires repeated computation of the residual scale, we propose to use the simple least median of squares estimator, given by $\hat{\sigma}_j = \text{median}_i|r_{ij}|$.

5.4.4 Sparsity inducing penalties

When the elastic net parameter in (5.8) is set to $\alpha = 0$, we get an L_1-penalty, a popular choice for inducing sparsity in the PCA loadings (Croux et al., 2013; Hubert et al., 2016). While the L_1-norm is not continuously differentiable, it can be approximated by a differentiable function, which converges towards the L_1-norm. In our implementation we use $|v_{jl}| = v_{jl}\text{sign}(v_{jl}) = \lim_{c\to\infty} v_{jl}\tanh(c \cdot v_{jl})$, as discussed by Öllerer et al. (2015). The more the constant c is increased, the better the absolute value function is approximated. In our implementation, we set $c = 1000$. Due to this approximation and the nature of gradient-based updates, this procedure does not lead to true sparsity in the loadings but only shrinks

the elements of the loadings matrix close to zero. In order to get truly sparse results, we propose to threshold the loadings in the following way: First, we track the relative change in each of the elements of the loadings matrix between the iterations $\boldsymbol{V}_t$ and $\boldsymbol{V}_{t+1}$. Then, the threshold value is computed as the average change of all elements during the last M iterations plus two standard deviations. If the absolute value of an entry of $\hat{\boldsymbol{V}}$ is lower than the threshold, it is set to 0. In the implementation, we use $M = 10$.

The complete algorithm is summarized in Algorithm 3.

Algorithm 3 Robust and Sparse PCA via Manifold Optimization

1: Compute transformations $\boldsymbol{Y} \leftarrow g(\boldsymbol{X})$
2: Initialize $\boldsymbol{V}_0$ as first k right-singular vectors of $\boldsymbol{Y}$
3: **while** $\|\mathcal{L}(\boldsymbol{V}_{t+1}) - \mathcal{L}(\boldsymbol{V}_t)\| > \delta$ **do**
4: $\quad \boldsymbol{H}_t \leftarrow \nabla_{\boldsymbol{V}} \mathcal{L}(\boldsymbol{V}_t)$ ▷ compute gradient H
5: $\quad \boldsymbol{P}_{\boldsymbol{H}_t} \leftarrow \text{proj}_{\mathcal{T}_{\boldsymbol{V}_t}\text{St}(p,k)}(\boldsymbol{H}_t)$ ▷ projection of gradient onto tangent space
6: $\quad \boldsymbol{V}_{t+1} \leftarrow R_{\boldsymbol{V}_t}(-\gamma_t \boldsymbol{P}_{\boldsymbol{H}_t})$ ▷ gradient step via retraction
7: $\quad$ **if** $\rho = \rho_{LTS}$ **then**
8: $\quad\quad$ update H_{t+1} based on $\boldsymbol{V}_{t+1}$ ▷ update h-subset
9: $\quad$ **end if**
10: $\quad d_{t+1} \leftarrow \|\boldsymbol{V}_{t+1} - \boldsymbol{V}_t\| / \|\boldsymbol{V}_t\|$ ▷ track relative change
11: **end while**
12: $\bar{t} \leftarrow \text{avg}[d_m]_{m=i}^{i-M+1} + 2\text{sd}[d_m]_{m=i}^{i-M+1}$ ▷ compute threshold
13: $\hat{\boldsymbol{V}} \leftarrow [v_{jl} \text{ if } |v_{jl}| > \bar{t}, 0 \text{ otherwise}]_{jl}$ ▷ thresholding
14: $\hat{\boldsymbol{Z}} \leftarrow \boldsymbol{X}\hat{\boldsymbol{V}}$ ▷ compute robust scores
15: $\hat{\boldsymbol{a}} \leftarrow \text{Qn}^2(\boldsymbol{X}\hat{\boldsymbol{V}})$ ▷ compute robust variances

Step 14 of Algorithm 3 returns the principal components, and Step 15 their variances, here estimated per component with the Qn-scale estimator (Rousseeuw and Croux, 1993).

5.4.5 Selection of sparsity parameter

While the elastic net mixing parameter α in (5.8) is set in advance by the user, the sparsity parameter λ is chosen depending on the data. Similarly to Croux et al. (2013), we choose the tradeoff-product criterion (TPO) that leads to a compromise between explained variance and sparsity in the loadings. In the following, we denote the columns of the estimate $\hat{\boldsymbol{V}}$ as $\hat{\boldsymbol{v}}_l$, for $l = 1, \ldots, k$. The original criterion

$$\text{score} = \sum_{l=1}^{k} \text{Qn}^2(\boldsymbol{X}\hat{\boldsymbol{v}}_l) \cdot \left(1 - \frac{\#\{\hat{\boldsymbol{v}}_l \neq 0\}}{p}\right)$$

where $\#\{\hat{\boldsymbol{v}}_l \neq 0\}$ returns the number of non-zero components in $\hat{\boldsymbol{v}}_l$, can be adapted for non-sparse regularization by including the elastic net parameters α:

$$\text{score} = \sum_{l=1}^{k} \text{Qn}^2(\boldsymbol{X}\hat{\boldsymbol{v}}_l) \cdot \left(1 - \alpha\frac{\#\{\hat{\boldsymbol{v}}_l \neq 0\}}{p}\right). \tag{5.11}$$

The maximum of this score function is now determined using Bayesian optimization: The advantage of this procedure is that contrary to cross-validation in combination with grid search, the information from previous function evaluations can be exploited. This way, a bigger search space can be covered. For the basic algorithm, it is assumed that there is a budget of in total N function evaluations. A Gaussian prior is placed on the score function, then its value is observed at n_0 points. Until N iterations are reached, the following steps are repeated: (i) update the posterior probability distribution on the score function, (ii) determine the maximum of the acquisition function, and (iii) observe the score value at this parameter configuration. We used the implementation in the R package `ParBayesianOptimization` (Wilson, 2022) with the expected improvement as the acquisition function and the tradeoff product optimization (TPO) as the score function.

The proposed algorithm is called SCRAMBLE (Sparse Cellwise Robust Algorithm for Manifold-based Learning and Estimation) and implemented in R (R Core Team, 2023), in the package `RobSparseMVA` (Pfeiffer et al., 2024), available from `https://github.com/piapfeiffer/RobSparseMVA`.

5.5 Simulations

A simulation study was conducted to evaluate the performance of the proposed method using different loss functions in comparison with other approaches, namely Robust and Sparse PCA (ROSPCA) (Hubert et al., 2016) for the rowwise contamination model, and MacroPCA Hubert et al. (2019) for the cellwise contamination model, although this method cannot induce sparsity.

5.5.1 Simulation settings

Similar to Croux et al. (2013); Hubert et al. (2016, 2019), several different contamination settings are considered. The casewise Tukey-Huber contamination model (1.1) and the independent contamination model for the cellwise framework (1.2).

A low- and a high-dimensional simulation setting is considered, and they are adapted from Croux et al. (2013) and Hubert et al. (2016). Clean data $\boldsymbol{X}$ are generated from a multivariate normal distribution $\mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$ with zero means and covariance $\boldsymbol{\Sigma} = \boldsymbol{C}^{1/2}\boldsymbol{R}\boldsymbol{C}^{1/2}$ given as follows:

1. Low-dimensional, order 2: $p = 10, n = 50$ observations

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{S}^1_{4\times4} & \mathbf{0}_{4\times4} & \mathbf{0}_{4\times2} \\ \mathbf{0}_{4\times4} & \boldsymbol{S}^2_{4\times4} & \mathbf{0}_{4\times2} \\ \mathbf{0}_{2\times4} & \mathbf{0}_{2\times4} & \boldsymbol{I}_{2\times2} \end{bmatrix},$$

where

$$\boldsymbol{S}^1_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0.9 & \text{otherwise,} \end{cases}$$

and

$$\boldsymbol{S}^2_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0.7 & \text{otherwise.} \end{cases}$$

The matrix $\boldsymbol{C}$ assigns the same scale to all variables of the same group, and it is given as $\boldsymbol{C} = \text{diag}(\mathbf{100}_4, \mathbf{25}_4, \mathbf{4}_2)$, where $\boldsymbol{a}_b$ is a vector with b replicates of the number a. The true loadings of the first component in this setting are $\boldsymbol{v}_1 = 1/2(\mathbf{1}_4, 0, \ldots, 0)$, and of the second component $\boldsymbol{v}_2 = 1/2(\mathbf{0}_4, \mathbf{1}_4, 0, \ldots, 0)$.

2. High-dimensional, order 2: $p = 500, n = 100$ observations

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{S}^1_{20\times 20} & \mathbf{0}_{20\times 20} & \mathbf{0}_{20\times 460} \\ \mathbf{0}_{20\times 20} & \boldsymbol{S}^2_{20\times 20} & \mathbf{0}_{20\times 460} \\ \mathbf{0}_{460\times 20} & \mathbf{0}_{460\times 20} & \boldsymbol{I}_{460\times 460} \end{bmatrix},$$

where $\boldsymbol{S}^1$ and $\boldsymbol{S}^2$ are defined as before. $\boldsymbol{C}$ is given as $\boldsymbol{C} = \text{diag}(\mathbf{100}_{20}, \mathbf{25}_{20}, \mathbf{4}_{460})$. Similarly, the true loadings for the first 2 components correspond to the vectors $\boldsymbol{v}_1 = 1/\sqrt{20}(\mathbf{1}_{20}, 0, \ldots, 0)$ and $\boldsymbol{v}_2 = 1/\sqrt{20}(\mathbf{0}_{20}, \mathbf{1}_{20}, 0, \ldots, 0)$.

Casewise contaminated data are generated by replacing $\varepsilon\%$ of rows with data generated from a p-variate normal distribution $\mathcal{N}(\boldsymbol{\mu}_{out}, \boldsymbol{I}_p)$,
with $\boldsymbol{\mu}_{out} = (2, 4, 2, 4, 0, -1, 1, 0, 1, -1, \ldots, 1, 0, 1, -1)$. For the cellwise contamination setting, data are generated using the `generateData(outlierType = "cellwiseStructured")` function in the R package `cellWise` (Raymaekers and Rousseeuw, 2023a), where $\varepsilon\%$ of cells are replaced by multiples of the last eigenvector of $\boldsymbol{\Sigma}$, restricted to the dimensions of the contaminated cells (Agostinelli et al., 2015; Rousseeuw and Bossche, 2018).

5.5.2 Performance measures

To measure the accuracy of the algorithm, we use the *principal angles* between subspaces, which are defined as follows: Let $\boldsymbol{V}$ and $\hat{\boldsymbol{V}}$ be orthonormal bases for the subspaces $\mathcal{V}$ and $\hat{\mathcal{V}}$, and assume that $\dim(\mathcal{V}) \leq \dim(\hat{\mathcal{V}})$. Then, the principal angle θ can be computed as $\theta = \sin^{-1}(\sigma_{\max}((\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}')\hat{\boldsymbol{V}}))$, where $\sigma_{\max}$ corresponds to the largest singular value of the projected matrix (Björck and Golub, 1973). We report the principal angle scaled to lie in $[0, 1]$, as implemented in the function `angle()` in the R package `rospca` (Reynkens, 2018) and described in Hubert et al. (2005).

The correct level of sparsity is evaluated by the true-negative rate (TNR) and true-positive rate (TPR), defined as

$$\text{TPR}(\boldsymbol{V}, \hat{\boldsymbol{V}}) = \frac{\#\{(j,k) : v_{jk} \neq 0 \text{ and } \hat{v}_{jk} \neq 0\}}{\#\{(j,k) : v_{jk} \neq 0\}} \tag{5.12}$$

$$\text{TNR}(\boldsymbol{V}, \hat{\boldsymbol{V}}) = \frac{\#\{(j,k) : v_{jk} = 0 \text{ and } \hat{v}_{jk} = 0\}}{\#\{(j,k) : v_{jk} = 0\}}, \tag{5.13}$$

where v_{jk} and $\hat{v}_{jk}$ are the corresponding elements of $\boldsymbol{V}$ and $\hat{\boldsymbol{V}}$, respectively. TNR and TPR correspond to the rate of correctly identified non-zero elements and zero elements, respectively, in the loadings.

The scaled principal angle corresponds to the accuracy of the subspace estimation and should be controlled at a low level even in the presence of outliers. The TPR and TNR depict how stable the correct estimation of the sparsity is; these performance measures should be close to 1.

5.5.3 Simulation results

The proposed method is compared to ROSPCA (Hubert et al., 2016) in the casewise contamination setting, using the implementation in the R package `rospca` (Reynkens, 2018), and MacroPCA as implemented in the R package `cellWise` (Raymaekers and Rousseeuw, 2023a) in the cellwise contamination setting. While the latter does not enforce sparsity in the loadings, it is the only other cellwise robust PCA method that software is readily available for, and a comparison in terms of accuracy of subspace estimation should be insightful.

Runtime

In addition to the algorithms ROSPCA and MacroPCA, we also include the PP algorithm from the `pcaPP` package (Filzmoser et al., 2022) as another sparse and robust PCA method for high-dimensional data in the comparison. However, this method is not included in the performance study, as the detailed comparisons to ROSPCA in Hubert et al. (2016) resulted in worse performance than the latter. Here we use Setting 1 described above, keeping the sparsity parameters for all methods fixed and varying p and n. The computation was repeated 100 times, and in Figure 5.1 the means are reported. In the left part of Figure 5.1, the number of variables is fixed at $p = 10$ and the number n of observations increased, on the right side, the number of observations is fixed at $n = 50$ and the number of variables p increased.

Figure 5.1 shows in most situations a higher runtime of the proposed method (SCRAMBLE). This is due to the other algorithms being implemented in C++. Still, with growing n and p, the advantage of the proposed method becomes apparent. If the number of observations n increases, the runtime even decreases, as the initial estimate is better, leading to faster convergence. In addition, very large n can also be handled by an appropriate SGD variant, as the data can be processed in batches of suitable sizes, a clear advantage in comparison to the repeated subset evaluations that need to be done for the ROSPCA algorithm, for example. When the number of variables p grows, the approach based on gradient descent also scales better, as several directions can be computed at once and no repeated cycling of candidate directions is necessary as for PP. While the runtime of MacroPCA as another cellwise robust PCA method is appealing, it is not able to include a regularization and produce sparsity in the loadings.

The computational complexity of the proposed algorithm depends on three steps: 1. the data transformation, 2. the SVD for the starting value, and 3. the gradient algorithm. For the rank transformation, this results in a complexity of $O(n\log(n)+np\min(n,p)+tnp+pk^2)$, and for the wrapping transformation in $O(np+np\min(n,p)+tnp+pk^2)$, where t refers to the number of iterations and k to the number of estimated components (refer to Raymaekers and Rousseeuw (2021) for the complexity of the wrapping transformation and Cunningham

and Ghahramani (2015) for a discussion of complexity of Riemannian gradient descent).

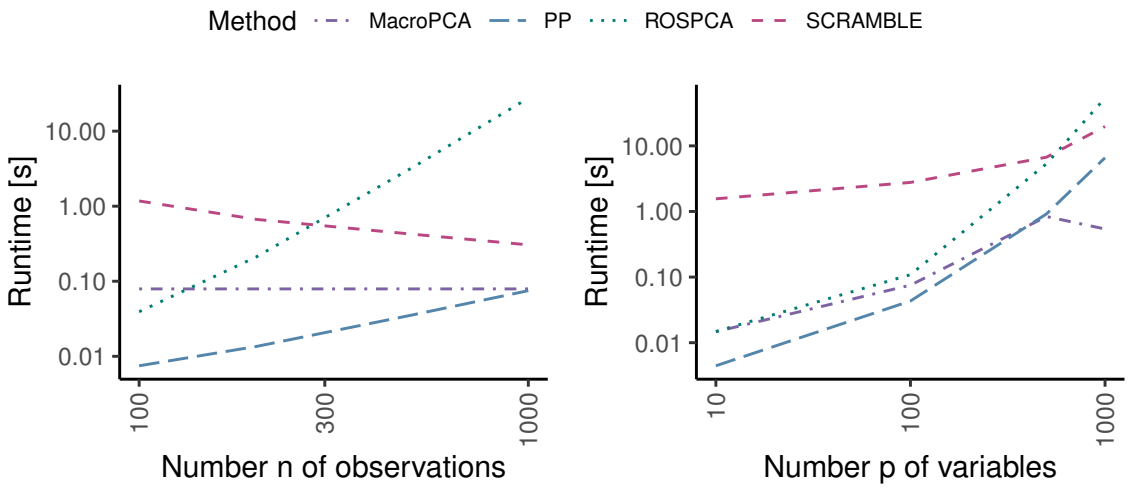


Figure 5.1: Comparison of runtime for the different methods. Left: The number of variables is fixed at $p = 10$, the number of observations n is increased from 100 to 1000. Right: The number of observations is fixed at $n = 50$, and the number of variables p is increased from 10 to 1000.

Performance

The robustness of the methods for an increasing proportion of casewise and cellwise outliers is studied. For the proposed method, performance measures of different combinations of the initial transformation and loss function are evaluated and compared to a suitable alternative method.

Figure 5.2 presents the results for casewise contamination. The performance results for the low-dimensional setting ($p = 10$ and $n = 50$) are shown in the plots on the left side, while those for the high-dimensional setting ($p = 500$ and $n = 100$) are on the right-hand side. In both cases, $k = 2$ is fixed, and ROSPCA is compared to different versions of SCRAMBLE, with different loss functions ρ (Huber, Tukey, LTS), see Section 5.3, and different transformations (rank, wrapping) for the initialization, see Section 5.4.2.

In the low-dimensional setting, the SCRAMBLE method clearly performs better than ROSPCA in all performance measures, especially in estimating the sparsity, described by the TNR. The increasing contamination has only little effect on the outcome. Also the different versions of SCRAMBLE show very similar results. For the high-dimensional setting we can see more effects. In the uncontaminated case, SCRAMBLE outperforms ROSPCA, particularly for the TNR. Interestingly, the TNR improves for ROSPCA with increasing contamination, and the same hold for SCRAMBLE based on the rank transformation. However, here an angle of about 0.75 for 20% contamination already suggests a solution very different from the target. The best results are achieved by SCRAMBLE initialized with the wrapping transformation, for the LTS and the Tukey loss function.

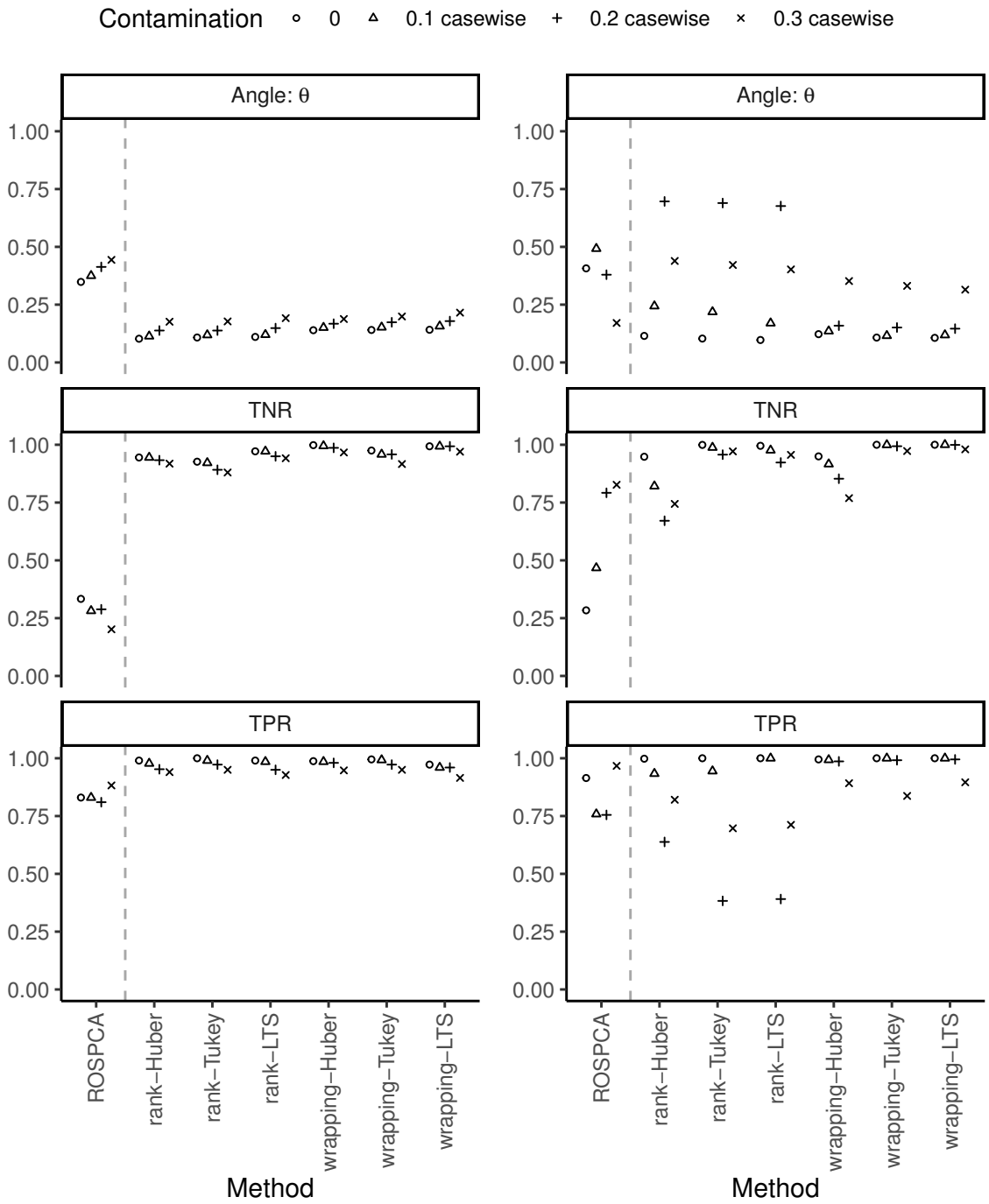


Figure 5.2: Comparison of performance measures for the methods ROSPCA and SCRAMBLE (six combinations of loss functions and transformations) for the casewise setting. Left: low-dimensional ($p = 10$, $n = 50$); right: high-dimensional ($p = 500$, $n = 100$).

The results for the cellwise contamination setting are shown in Figure 5.3, with the low-dimensional setting on the left and the high-dimensional setting on the right. Here we compare SCRAMBLE to MacroPCA as an alternative algorithm for cellwise robust PCA, although this method does not allow for sparsity. Consequently, MacroPCA fails to accurately predict the sparsity (resulting in a TPR of 1 and a TNR of 0) but yields very good results for the principal angle in the low-dimensional setting. Note that we selected the same proportions of contamination as for the casewise setting. However, in the cellwise setting, these amounts contaminate many more rows of the data matrix, or even all rows (Raymaekers and Rousseeuw, 2023b), leading to a faster decrease in performance among all metrics. For the high-dimensional setting, however, the proposed algorithm yields the best overall results. This is because SCRAMBLE processes the loss function cellwise, and thus both more rows or more columns yield more training set observations. As expected, there is a certain decrease in performance for increased contamination. In this setting, rank-based initialization is more robust than initialization with the wrapping transformation with default values (Raymaekers and Rousseeuw, 2021). The overall best results are in combination with the LTS or Tukey loss function.

5.6 Illustration on real data

The usefulness of the approach will be demonstrated on two datasets from tribology, the study of friction, wear, and lubrication. The presented data originate from chemical analyses and tribological experiments performed on automotive engine oils after they have been subjected to a varying duration of artificial alteration in the laboratory, see Dörr et al. (2019b) and Besser et al. (2019) for a description of the alteration methods. FTIR (Fourier-transform infrared) spectra consist of absorption values that are measured over about 2000 wavenumbers, with distinctive peaks associated with certain oil attributes. For this data structure, the sparsity assumption can be justified: It can be assumed that only a small set of wavenumbers is sufficient to explain most of the variability in the dataset (Pfeiffer et al., 2022). Another aspect is lubrication performance, which can be measured on an SRV tribometer experiment (a steel ball sliding against a steel disk with the lubricant of interest in between, see Dörr et al. (2019b) for a more detailed description of the experiment). One part of the resulting data consists of optical images (taken under the microscope) of the wear scar areas, which yield data matrices with $n \ll p$ when vectorized.

As both types of data are produced in the laboratory, we can expect outliers to be present in the datasets due to possibly high variability following the alteration process and experimental effects. In addition, the experiments are often costly and time-consuming, resulting in much fewer observations than variables and wide data matrices. Thus, dimension reduction is necessary before applying any further analysis. We demonstrate in the following, how the proposed method can be applied to perform this dimension reduction via robust PCA, and, in addition, yield sparse loadings, when appropriate.

5.6.1 FTIR spectra

The presented dataset consists of $n = 50$ FTIR spectra of 10 automotive engine oils. The fresh oils were subjected to a small-scale alteration in the laboratory as described

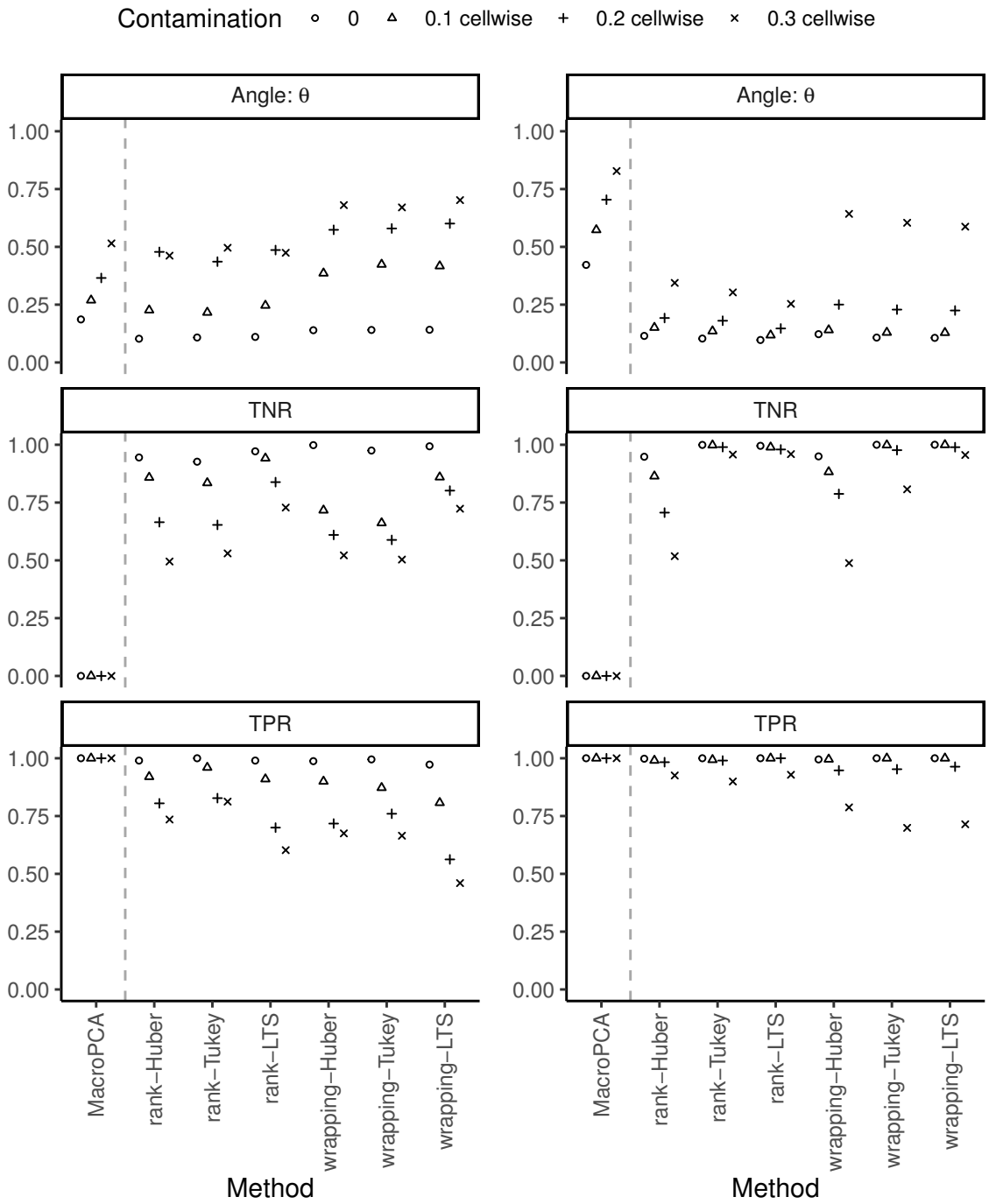


Figure 5.3: Comparison of performance measures for the methods ROSPCA and SCRAMBLE (six combinations of loss functions and transformations) for the cellwise setting. Left: low-dimensional ($p = 10$, $n = 50$); right: high-dimensional ($p = 500$, $n = 100$).

in Dörr et al. (2019b). During the alteration, samples were taken regularly and spectra were recorded, each containing the absorbance at $p = 1668$ wavenumbers. The task at hand is to understand which variables contribute most to the variability in the dataset, and often classical PCA is applied, see, for example, Besser et al. (2013). To make it more challenging, we contaminate the dataset with 6 observations originating from a large-scale alteration (Besser et al., 2019) to imitate a scenario when the origin of a sample may not be as clear as in the laboratory setting. Our aim is to identify observations that are different from the majority of the data, and also understand why they are outlying. As mentioned before, sparsity in the PCA loadings is desirable in this setting to enhance interpretability.

The resulting dataset consists of $n = 56$ observations and $p = 1668$ variables. As spectral data are already on the same scale, the data are not scaled, but only column-wise centered with the median before applying the methods. We compare the results from ROSPCA (Hubert et al., 2016) and the proposed SCRAMBLE method with the rank-based data transformation for the starting value and the Huber loss function. For both methods, the number of principal components is determined via the cumulative proportion of explained variance. For ROSPCA, this results in $k_{\text{ROSPCA}} = 10$, for SCRAMBLE in $k_{\text{SCRAMBLE}} = 7$ components. Then, hyperparameter optimization for the sparsity parameters is done for both algorithms. Figure 5.4 shows the original FTIR spectra in gray, together with the first (left plot) and second (right plot) loadings vectors of both methods. We find that SCRAMBLE leads to more sparsity, and therefore to results that are easier to interpret. Some of the selected variables are known to be associated with underlying chemical processes during oil alteration, like oxidation, conventionally evaluated at wavenumber 1720 cm^{-1}, or phenolic antioxidants at 3650 cm^{-1} (Besser et al., 2019).

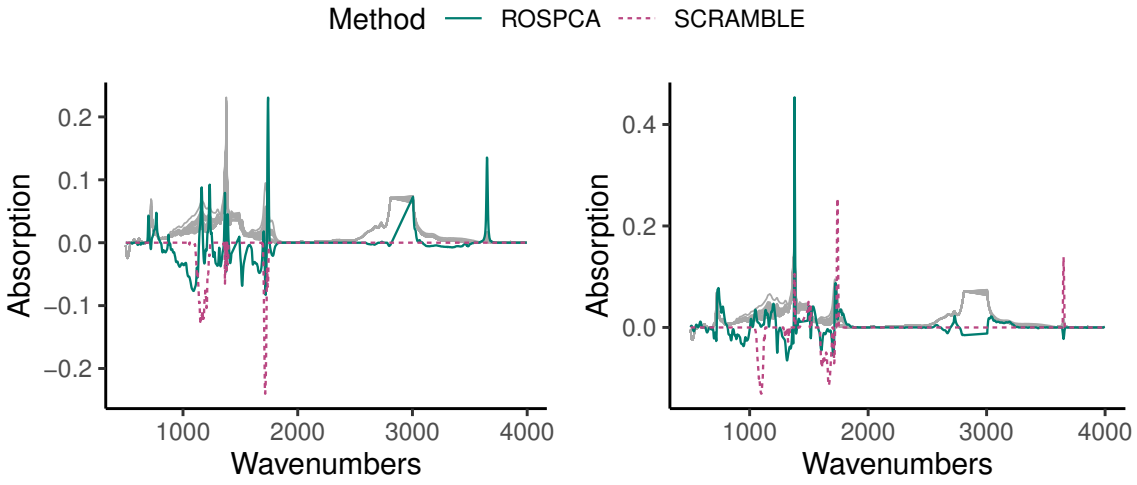


Figure 5.4: FTIR spectra of the original data, shown in gray, together with the first (left plot) and second (right plot) loadings vectors of each method.

Figure 5.5 shows PCA diagnostic plots based on the score distance SD and orthogonal distance OD, with the outlier cutoff values as dashed lines, see Hubert et al. (2005). A high value of the OD means that the observations are far away from the estimated PCA

subspace. The left plot presents the results for ROSPCA, and here almost all outliers, thus the observations from large-scale alteration, are identified with high OD values. However, also four regular observations yield high OD values. The right plot for the SCRAMBLE results corresponds to what we would expect.

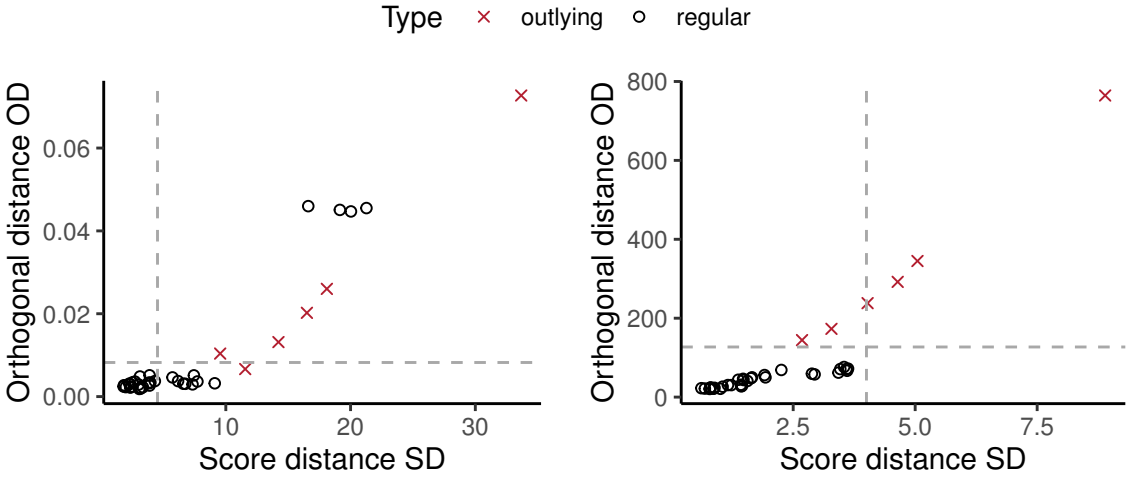


Figure 5.5: Score distance versus orthogonal distance for the FTIR spectra. Results for ROSPCA (left) and SCRAMBLE (right).

In order to investigate the differences between ROSPCA and SCRAMBLE in more detail, we show plots of the standardized residuals in Figure 5.6 for a selected wavenumber range and a subset of the observations. The left plot is for ROSPCA, the right plot is for SCRAMBLE, and each tile represents one element in the scaled residual matrix, with color according to the legend. The residuals were standardized robustly using the median and mad of the residual matrix. Note that the residuals scale is very small leading to very large scaled residuals for some cells. The first six rows correspond to the FTIR spectra of oils from a different alteration process, and SCRAMBLE clearly reconstructs these worst, meaning they are not as influential to the fit of the PCA subspace. In ROSPCA, on the other hand, only observation 6 is the only clearly visible observation out of the outlier subset, and four further observations also show larger residuals. A look at the original spectra with a zoomed-in view into this wavenumber range in Figure 5.7 explains this behavior: There, the outliers (first 6 rows) are shown by red dashed curves, and only one (observation 6) is clearly further away, while the other outliers partially overlap with regular observations. This overlap around wavenumber 1740 cm^{-1}, thus in a very restricted range, is the reason why four regular observations are falsely classified as outlying by the ROSPCA algorithm. The affected wavenumbers lie in the absorption band of oxidation products, ranging from 1860–1660 cm^{-1}, and are of interest in conventional analysis of FTIR spectra (Besser et al., 2019; Pfeiffer et al., 2022). A cellwise robust method can assist the practitioner in finding and understanding the differences between outlying and regular observations.

In summary, we can see the benefit of a cellwise robust method in contrast to only casewise robust estimation. As the differences only show at certain peaks in the spectra, a

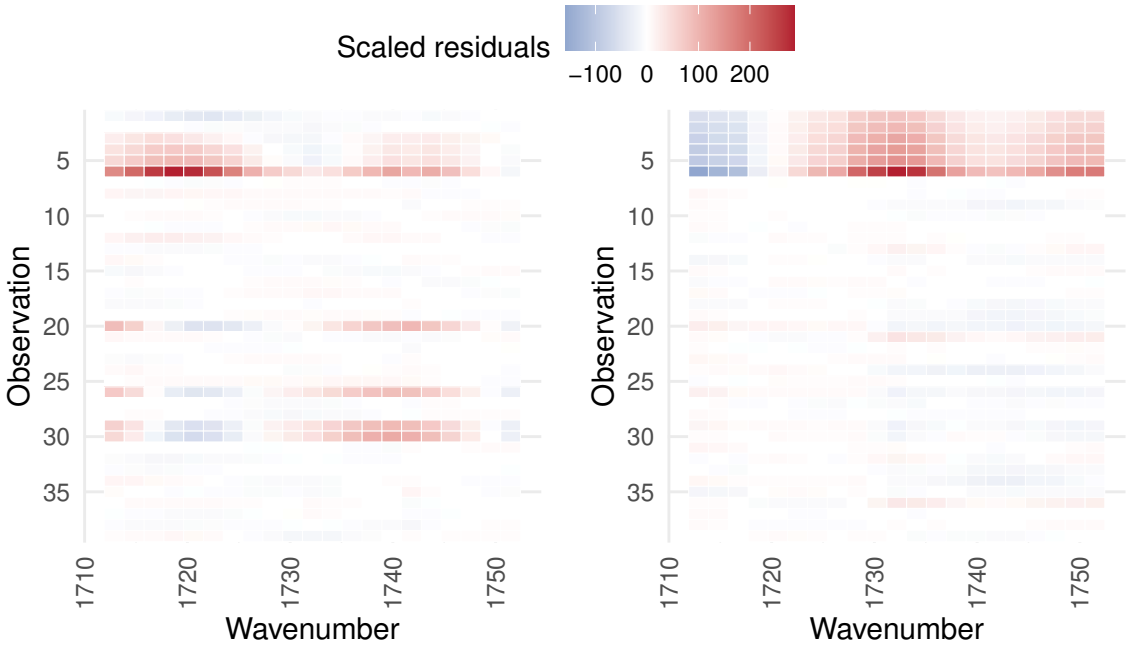


Figure 5.6: Residual plots for a selected range of wavenumbers and for a subset of the observations. Left: ROSPCA residuals; right: SCRAMBLE residuals.

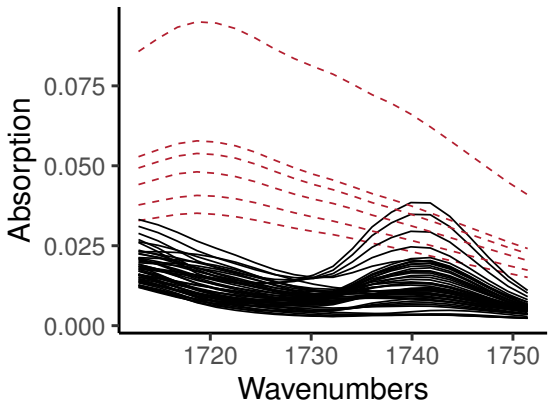


Figure 5.7: Zoomed-in view of the FTIR spectra from Figure 5.4 for the selected wavelength range shown in Figure 5.6. The red dashed lines are outliers with large-scale alteration.

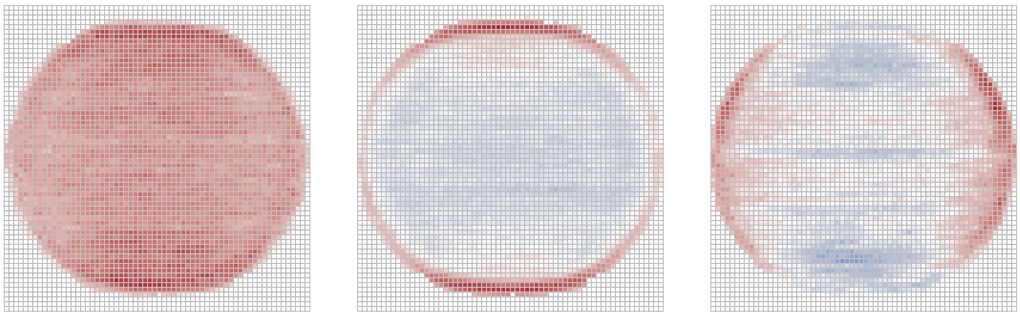


Figure 5.8: Classical loadings back-transformed to the image space.

cellwise robust method does not lose as much information as a casewise method, resulting in a better fit with fewer components. In addition, we can also identify the variables which contribute most to the outlyingness.

5.6.2 Tribology: wear scar images

Often, the task is not only to derive useful features but also to predict an outcome or a property. In this example, we demonstrate the flexibility of our approach in a PCR (Principal Component Regression) setting. As we are dealing with a wide data matrix, dimension reduction is necessary. In Pfeiffer and Filzmoser (2023), image features were derived from a similar image dataset, before robust regression methods were applied. For this demonstration, we derive robust features via SCRAMBLE directly from the vectorized images before applying a robust regression on the resulting principal components. We do not use a sparsity-inducing regularization, as we have found that this does not provide an advantage for images (Pfeiffer and Filzmoser, 2023). In the given setting, $n = 220$ gray-scale images of size 64×64, resulting in vectors of size $64^2 = 4096$, together with a response variable containing the alteration duration (in hours) of the lubricant used in the SRV experiment, are available. After the removal of all constant columns, $p = 3025$ columns are left. As the response variable, the alteration duration is given in hours, which is square-root transformed before estimating the model.

We compare classical PCA via SVD to the SCRAMBLE algorithm with rank-based pre-processing and the Huber loss. Therefore, the dataset is randomly split into a 70% training, a 20% validation, and a 10% test set. The principal components are estimated on the training set, and then the optimal number of components is evaluated for the validation set via the mean squared error of prediction (MSEP) using least-squares regression for the classic estimation and robust regression (the function `lmrob()` from the `robustbase` R package (Maechler et al., 2024)) for robust estimation. Finally, the MSEP is computed for the test set. The first three estimated loadings are shown in Figure 5.8 for classical PCA and in Figure 5.9 for robust PCA. We can observe that the first loadings look quite similar, while the order is different. Both methods distinguish between the border and the interior of the wear scar, as well as the overall contributions.

The 10% trimmed MSEP for the validation set based on different numbers of components

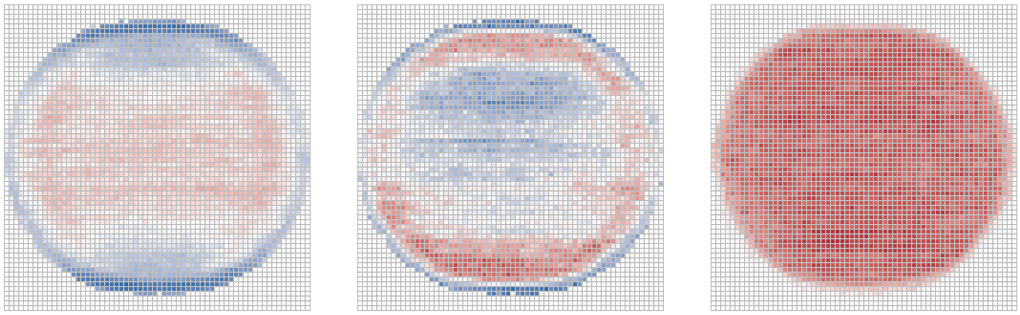


Figure 5.9: Robust loadings back-transformed to the image space.

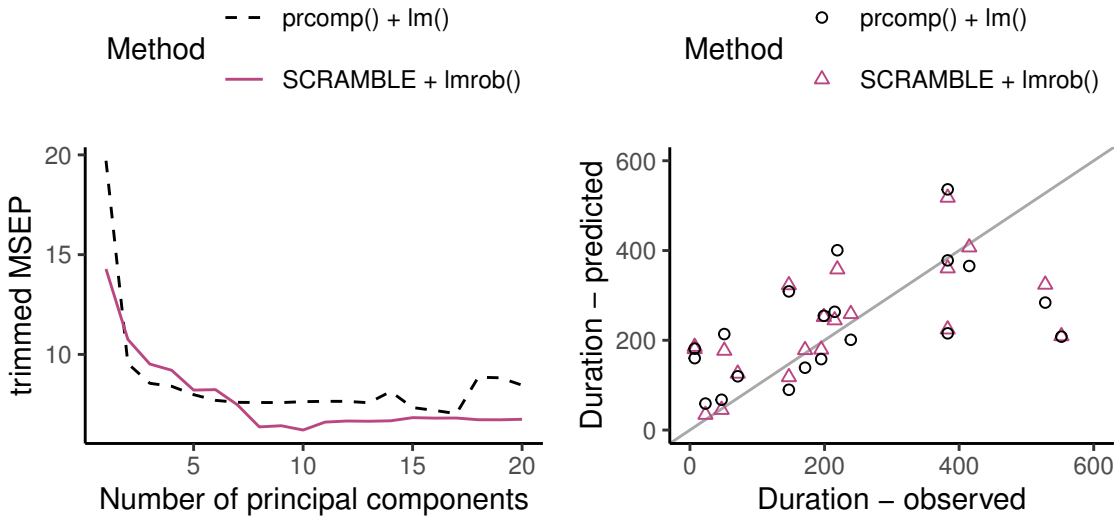


Figure 5.10: Left: Selection of best number of PCs based on 10% trimmed MSEP computed on the validation set. Right: Observed vs. predicted alteration duration for the classical and robust approach.

is shown on the left side in Figure 5.10 for both classical and robust PCR. For classical PCR, the error starts at a higher level, possibly indicating that the directions of the first few components are influenced by outliers, and thus they are not as effective for prediction.

For the optimal number of components, we select that number yielding the smallest trimmed MSEP, resulting in $k_{\text{classical}} = 17$ and $k_{\text{SCRAMBLE}} = 10$ components. Thus, the robust method leads to a smaller number of components, and also to a smaller prediction error.

In the right plot of Figure 5.10, the observed and predicted values of the alteration duration are shown for both methods for the test set observations. The predictions for higher values of duration for both methods are worse than for values in the beginning or middle of the duration range. In total, the model based on robust PCR performs better, with a trimmed MSEP of 13.76 for the test set observations, while classical PCR leads to an error of 17.88 (Figure 5.10).

Using this number of components, we can also reconstruct the data matrices and analyze the reconstruction errors. In Figure 5.11, the reconstruction errors per variable (pixel) are visualized for both the classical PCA (on the left) and SCRAMBLE (right). While for the classical method, no structure is left in these residuals, it is clearly visible that for the robust method, the border of the wear scars is not reconstructed well. This also makes sense because the borders of the balls in the wear scar images are not identical. In fact, the size of the balls in the image can slightly change due to the nature of the experiment, as the balls are placed manually under a microscope, but also due to preprocessing and cutting the images. Obviously, for the robust procedure, this change in size is not relevant for prediction, whereas the classical model takes this variability into account.

While the results for classical and robust PCR are not very different, the example still illustrates that the robust method is able to perform better for the majority of the data (in the middle of the duration range), while the classical predictions are influenced by more extreme values. Furthermore, we can use robust diagnostics to get further insight into why the prediction quality between certain values of the response differs.

5.7 Discussion and summary

In recent years, cellwise robust methods are becoming increasingly important. This is mainly due to the increased occurrence of high-dimensional data, as a result of modern measurement methods and devices. With high-dimensional data it becomes more likely that an observation contains outliers in single variables, and traditional rowwise (casewise) methods would no longer work if the majority of observations are contaminated. This is also an issue for principal component analysis (PCA), where rowwise robust methods could fail in the presence of many cellwise outliers.

One could think of several different approaches to obtain a cellwise robust PCA method. A first idea could be the identification of cellwise outliers and the replacement of those cells by values which would be expected according to some distributional assumptions (Rousseeuw and Bossche, 2018). With the cleaned data matrix one could proceed with classical PCA. Even in the casewise robust setting, the approach to detect and correct outlying observations prior to classical PCA would be a way to obtain a casewise robust PCA version.

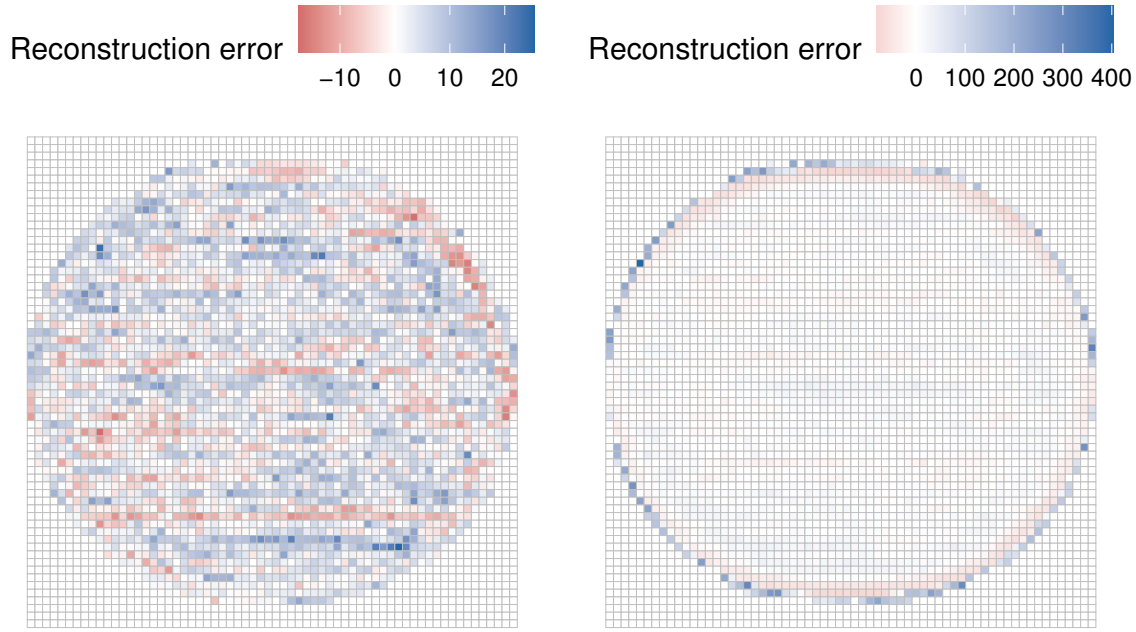


Figure 5.11: Reconstruction errors per variable, visualized in the image dimensions. Left: PCA via classical SVD, right: PCA via `SCRAMBLE`. The number of components corresponds to the number leading to the minimum trimmed MSEP for PCR.

However, outlier detection assumes an underlying model, usually a multivariate normal distribution, and outlier detection/correction identifies/corrects observations or cells according to this model assumption. In cellwise or casewise robust PCA, on the other hand, we are not limited to this distribution. Particularly for PCA based on low-rank approximation, the interest is rather in a robust data reconstruction, where the error loss function utilizes information from the single variables rather than from the joint multivariate distribution, see Equation (5.4).

Another approach to cellwise robust PCA is to use a plug-in estimator for the covariance to determine the principal components, for example, the cellwise robust Minimum Covariance Determinant (MCD) estimator (Raymaekers and Rousseeuw, 2023b). This is equivalent to a rowwise robust PCA version where a rowwise robust covariance estimator, such as the MCD (Rousseeuw, 1985) is plugged in. While such a procedure is straightforward to implement, it might not be so clear how to include sparsity.

Sparsity, or the natural requirement of interpretability of the principal components is especially important in a high-dimensional setting. For that reason, sparse (Jolliffe et al., 2003) and sparse robust (Croux et al., 2013) PCA versions were introduced which maximize the variance of the components subject to an L_1 penalty on the loadings vectors. The gain in sparsity or explainability leads to a loss in explained variance, and this compromise can be formalized by an appropriate objective function (Croux et al., 2013).

We have introduced a cellwise robust and sparse PCA method using low-rank matrix approximation. The objective function can be formulated in a very natural way (Maronna

and Yohai, 2008; Croux et al., 2013), and it combines a robust loss function for the reconstruction error of all cells of the data matrix with an elastic net penalty on the loadings. The specific choice of the loss function determines the robustness properties of the PCA solution (Maronna and Yohai, 2008). Both the robust loss function and the incorporation of an L_1 or elastic net penalty leads to computational challenges. We have developed an algorithm based on manifold learning to optimize the objective function. The L_1 penalty was incorporated by the use of a sparsity-inducing penalty, allowing for an approximation by a differentiable function. The choice of appropriate starting values is important, and we compared different approaches. Overall, the algorithm leads to an efficient computation, even for high dimensions p and many observations n. Simulations have demonstrated that the resulting method, called SCRAMBLE (Sparse Cellwise Robust Algorithm for Manifold-based Learning and Estimation), has superior properties when compared to alternative robust PCA approaches, both in the casewise and cellwise settings.

We applied the suggested method to two real data examples from tribology and compared the performance with existing estimators, illustrating the usefulness of a cellwise robust and sparse PCA method.

Possible extensions to groupwise PCA or robust data imputation are possible via a modification of the objective function (5.8). Furthermore, theoretical robustness properties like the influence function and breakdown point (Maronna et al., 2019) would be interesting topics for future research.

6 Implementation and practical use in R

The proposed methods from Chapters 4 and 5 have been implemented in the R package `RobSparseMVA` (Pfeiffer et al., 2024), which is currently available from GitHub in the repository `https://github.com/piapfeiffer/RobSparseMVA`. It can be installed using the `devtools` package (Wickham et al., 2022).

Both implementations rely on the backpropagation algorithm for computing the gradients of the objective functions, as it is implemented in the R package `torch` (Falbel and Luraschi, 2023). The code is structured into a *model* part, initializing the parameters and defining the objective function and evaluation. The *train* part calls the model and contains the training loops, with calls to the optimizing functions. For hyperparameter optimization, application-specific wrappers for Bayesian optimization `ParBayesianOptimization` (Wilson, 2022), including the proposed score functions, are implemented. Finally, the *compute* function wraps it together, providing the interface for the user and implementing options for (robust) data standardization, and algorithm-specific settings.

In this chapter, we demonstrate the functionality of the most important functions on examples and discuss possibilities for customization and fine-tuning.

6.1 Robust and sparse maximum association - `ccaMM()`

The easiest way to apply the algorithm proposed in Chapter 4 to a given dataset is via the function `ccaMM()`. The code for the call is given in Listing 6.1.

Listing 6.1: Arguments for `ccaMM()`

```
ccaMM <- function(data_x, data_y,
                  method = "Pearson",
                  ..., #keyword arguments for covariance
                  nearPD = FALSE,
                  alpha_x = NA,
                  alpha_y = NA,
                  k = 1,
                  tol = 1e-5,
                  lr = 1e-3,
                  epochs = 2000,
                  lr_decay = 1,
                  criterion = "TPO",
                  penalties = NA)
```

The function expects the x and y datasets as the first two arguments: The data should be formatted such that the variables are in the columns and the samples are in the rows, and the number of rows for both needs to be the same. Using the argument `method`, the user can decide on the type of covariance estimator to be used. The default setting, `method`

= `Pearson`, corresponds to the sample covariance (and the Pearson correlation for CCA). Other options are: `Spearman, Kendall, MCD, MRCD, OGK, pairhuber, quadrant`, and `Ledoit-Wolf`. The fourth argument is a keyword argument to provide options for the covariance estimator, for example, setting the size of the subset for the MCD. `Spearman` and `Kendall` are based on the pairwise rank-based correlation matrices, which are then scaled robustly to derive a covariance. When more variables than observations are present, the resulting covariance matrices are not positive definite and can be corrected using the `nearPD` algorithm (Higham, 2002). While positive definiteness is not strictly necessary for the proposed procedure, it can improve the performance. The α_x and α_y parameters correspond to the elastic net parameters for the x and y side and should be numeric vectors of length k, containing numbers between 0 and 1 (0 for Ridge, 1 for LASSO penalty), where k is the number of directions that should be computed. The next four parameters in Lines 8-11 are settings for the gradient descent algorithm, it is possible to adjust the tolerance for convergence (`tol`), the learning rate (`lr`), the maximum number of training epochs (`epochs`), and the exponential learning rate decay (`lr_decay`, a positive number lower or equal to 1 the learning rate is multiplied by before the next iteration). The `criterion` argument refers to the type of criterion to be used for hyperparameter optimization, the default (and currently only) option is "TPO", corresponding to "Tradeoff Product Optimization". Finally, the `penalties` argument can be used to provide the used penalties manually. This argument should be given as a list with the vectors `pen_x` and `pen_y` as entries, containing the penalty for each order of direction for both sides.

Let us now demonstrate how to use the function in practice. In Listing 6.2, a simple simulated dataset is generated.

Listing 6.2: Simulated example for CCA.

```
1  set.seed(123)
2  library(mvtnorm) # only needed for simulated example
3  p <- 10
4  q <- 10
5  n <- 100
6
7  cov_xx <- matrix(0, ncol = p, nrow = p)
8  cov_yy <- matrix(0, ncol = q, nrow = q)
9  cov_xy <- matrix(0, nrow = p, ncol = q)
10
11 diag(cov_xx) <- 1
12 diag(cov_yy) <- 1
13 cov_xy <- matrix(0, nrow = p, ncol = q)
14 cov_xy[1, 1] <- 0.9
15 cov_xy[2, 2] <- 0.7
16
17 sigma <- rbind(cbind(cov_xx, cov_xy),
18                 cbind(Matrix::t(cov_xy), cov_yy)
19                 )
20
21 data <- mvtnorm::rmvnorm(floor(n),
22                          mean = rep(0, p + q),
```

```
                          sigma = sigma, checkSymmetry = F
                          )

x <-  as.matrix(data[, 1:p])
y <-  as.matrix(data[, (p + 1):(p + q)])
```

We can now compute the classical and a sparse CCA solution by using the default parameters and setting the elastic net parameters to 0 and 1, respectively. Example code is shown in Listing 6.3.

Listing 6.3: Classical and sparse CCA (non-robust)

```
n_dir <- 2 # we want to derive the first two directions
res_classic <- ccaMM(x, y,
                     k = n_dir,
                     alpha_x = rep(0, n_dir),
                     alpha_y = rep(0, n_dir))
res_sparse <- ccaMM(x, y,
                    k = n_dir,
                    alpha_x = rep(1, n_dir),
                    alpha_y = rep(1, n_dir))
```

When no hyperparameter optimization is needed, the penalties to be used can be set directly. In addition, the robustness can be controlled by changing the `method` argument, as demonstrated in Listing 6.4.

Listing 6.4: Classical and sparse CCA (non-robust)

```
res_spearman <- ccaMM(x, y,
                      k = n_dir,
                      method = "Spearman",
                      alpha_x = rep(1, n_dir),
                      alpha_y = rep(1, n_dir),
                      penalties = list(pen_x = rep(1, n_dir),
                                       pen_y = rep(1, n_dir)))
```

6.1.1 Looking at the results

CCA leads to projections of the data we can look at. The results object of the function `ccaMM()` returns the determined directions (linear combinations) in `a` for the `x` dataset and `b` for the `y` dataset. The rows of the matrices `a` and `b` correspond to the number of variables in each dataset, while the columns correspond to the number of directions that were computed (we can also call this the order). The vector `measure` returns the computed canonical correlation measure for each order. The resulting projections are returned in the matrices `phi` and `eta`; here, the number of rows corresponds to the number of observations, and the number of columns again corresponds to the order.

The final used penalties are returned in `pen_x` and `pen_y`, and if hyperparameter optimization for sparsity was done, a summary is returned in `summary`.

The `plot` function applied to the result object plots the values that were tested for sparsity parameters, TPO tradeoff curve, and the projections for the respective order. This series of plots is provided for all orders. For the example above, the plots are shown in Figure 6.1.

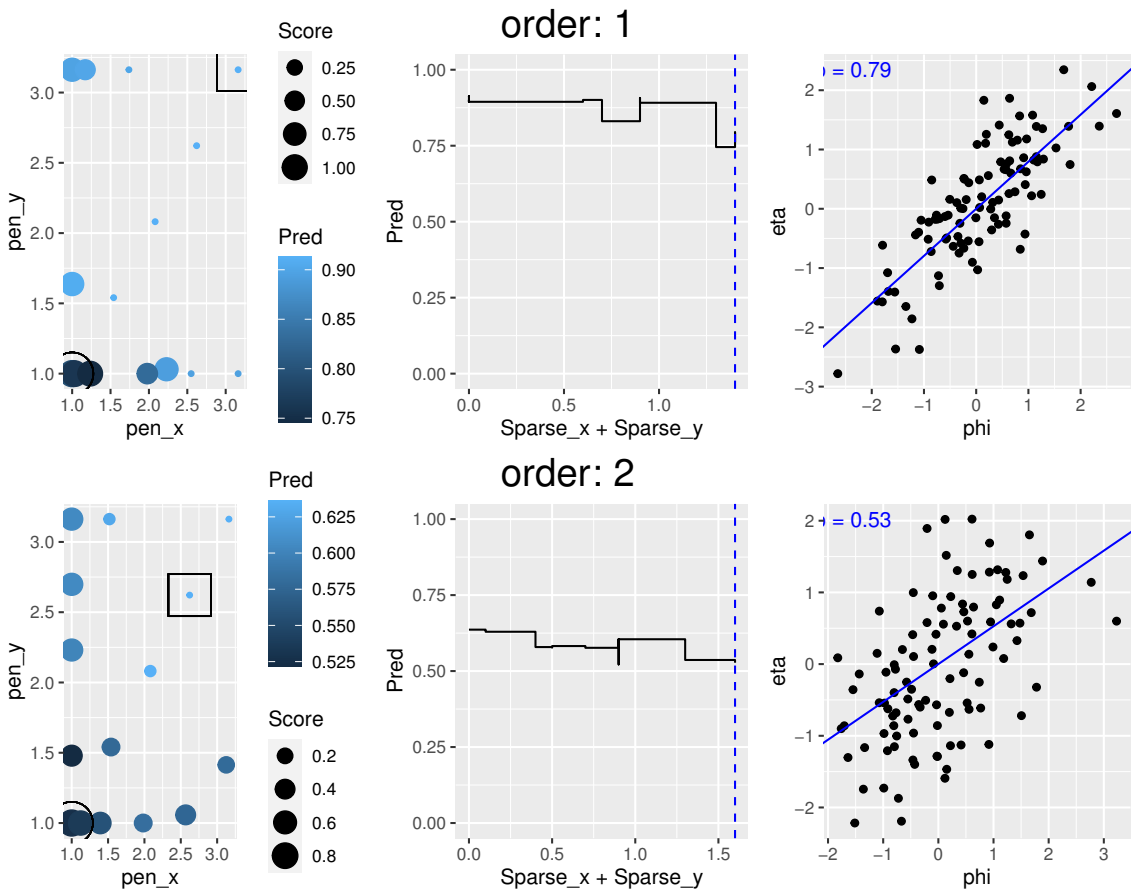


Figure 6.1: Plots generated with the `plot()` function implemented for the result object from `ccaMM()`. Following plots are generated for each order: On the left, the process of the hyperparameter optimization is visualized. The plot shows the sampled combinations of `pen_x` and `pen_y` and the corresponding score and prediction values. In the middle plot, the value of the prediction, depending on the combined sparsity value of `x` and `y`, is shown, and on the right, the projections are plotted.

6.1.2 Customization

In the ccaMM() function, the different covariance estimators are called with the default settings proposed by the authors of the respective packages and functions. Their parameters and options can be adjusted by supplying keyword arguments in the fourth argument. It is also possible to provide the covariance matrix directly and use another algorithm for hyperparameter optimization by using the train_CCA() function. This function was applied to the true covariance matrix in Chapter 4 for the evaluation of the precision of the algorithm. In addition, this function can be used to compute additional projection directions. Unlike the ccaMM() function, the order k refers to the specific order to be computed, and the lower-order projection directions, as well as the penalty values need to be provided. In Listing 6.5, we show a possible application in combination with Bayesian hyperparameter optimization, as implemented in bayesian_optimization_CCA, but it could be exchanged with a grid search or the like as well. The functions expect the covariance instead of the data matrices, and the lower-order directions need to be provided for the computation of directions of order larger than 1. The other arguments correspond to the ones needed for ccaMM().

Listing 6.5: Training loop using any covariance matrix Cov, using Bayesian hyperparameter optimization and train_CCA() function

```
CORR <- rep(NA, k)
PEN_X <- rep(NA, k)
PEN_Y <- rep(NA, k)

alpha_x <- rep(1, k)
alpha_y <- rep(1, k)

for (i in 1:k) {
    res_param <- bayesian_optimization_CCA(
                    C = Cov,
                    p = p, q = q, n = n, order = i,
                    alpha_x = alpha_x[i], alpha_y = alpha_y[i],
                    low_a = as.matrix(A[,1:max(1,(i-1))]),
                    low_b = as.matrix(B[,1:max(1,(i-1))]),
                    tol = 1e-5, lr = 1e-2, epochs = 1000,
                    lr_decay = 1,
                    criterion = "TPO")

    PEN_X[i] <- res_param$best_params$pen_x
    PEN_Y[i] <- res_param$best_params$pen_y

    SUMMARY[[i]] <- res_param$summary

  res <- train_CCA(C = Cov, p = p, q = q,
                   pen_x = PEN_X[i], pen_y = PEN_Y[i],
                   order = i,
                   alpha_x = alpha_x[i], alpha_y = alpha_y[i],
                   tol = 1e-5, lr = 1e-2, epochs = 1000,
                   lr_decay = 1,
```

```
                    low_a = as.matrix(A[,1:max(1,(i-1))]),
                    low_b = as.matrix(B[,1:max(1,(i-1))]))

  A[, i] <- res$best_a
  B[, i] <- res$best_b

  CORR[i] <- res$best_measure
}
```

Note that as this function only outputs the best measure and directions (`best_a, best_b`), the projections have to be computed manually. Many more details can be changed in the functions themselves. Customization of the gradient optimization algorithm can be done in the file `train_CCA_MM.R`, and the settings for Bayesian hyperparameter optimization can be changed in `hp_optim.R`.

6.2 Cellwise robust and sparse PCA - `pcaSCRAMBLE()`

The algorithm proposed in Chapter 5 is also implemented in the R package `RobSparseMVA`. The function `pcaSCRAMBLE()` implements the data transformation, provides a wrapper for hyperparameter optimization, and performs the optimization of the objective using Riemannian gradient descent. For the computation of the gradient steps on the manifold, a new optimizer has been implemented for the usage with `torch` for R (Falbel and Luraschi, 2023), contained in the file `optimizer_SGD_Stiefel.R`.

Let us first create a simulated example, shown in Listing 6.6.

Listing 6.6: Simulated example for PCA

```
p <- 100
n <- 50
R <- matrix(0, ncol = p, nrow = p)
R[1:4, 1:4] <- 0.9
R[5:8, 5:8] <- 0.5
diag(R) <- 1
V <- diag(c(100, 100, 100, 100, 25, 25, 25, 25, rep(4, p - 8)))
C <- sqrt(V) %*% R %*% sqrt(V)

data <- mvtnorm::rmvnorm(floor(n),
                         mean = rep(0, p),
                         sigma = C)
```

In Listing 6.7, the usage of the main function `pcaSCRAMBLE()` is demonstrated.

Listing 6.7: Usage of the function `pcaSCRAMBLE()`: arguments and options.

```
res_scramble <- pcaSCRAMBLE(data_x = data,
                            groups = NA,
                            transformation = "identity",
                            loss_type = "L2",
                            param = NA,
                            center = TRUE,
                            scale = TRUE,
```

```
8                                   alpha_x = 0,
9                                   k = NA,
10                                  tol = 1e-5,
11                                  lr = 1e-3,
12                                  epochs = 2000,
13                                  lr_decay = 0.99,
14                                  criterion = "TPO",
15                                  bounds = c(1e-3, 10),
16                                  penalties = NA)
```

In the first argument, the function expects the data matrix with p variables in the columns and n observations in the rows. Then, options regarding group structure, the data transformation to be applied, and the loss type can be specified. The second argument refers to a possible extension to group PCA, which has been implemented, but not investigated in detail yet. In this `group` argument, a vector of the same length as the data and containing the groupings has to be supplied. If left blank, no grouping is considered.

For the `transformation` argument, the following options are available: `identity` does not perform any transformation, `spearman` refers to the rank-based transformation, and `wrapping` to the wrapping transformation described in Chapter 5. The transformed data is used to obtain a robust initial estimate of the principal components via SVD, and if robustness is desired, it is recommended to use either wrapping or the rank-based transformation. In addition, the loss function is important for obtaining a robust result. The type of loss function (see Section 1.3 for an overview) is defined via the `loss_type` argument. Possible values are `L2`, corresponding to a least squares loss, `LTS` for a least trimmed squares loss, and `Huber` and `Tukey` referring to the well-known robust loss functions of the same name. The argument `param` controls the parameters of the loss functions and uses the default values suggested in Chapter 5 if left blank.

The arguments `center` and `scale` control whether the data will be robustly centered and scaled using the median and Qn estimator, respectively. The elastic net parameter `alpha_x` can be set to any value between 0 (Ridge penalty) and 1 (LASSO penalty) and `k` corresponds to the number of principal components to be computed, if left blank, it is set to the minimum of the number of observations and variables.

The arguments in Lines 10-13 of Listing 6.7 are parameters for the training of the gradient descent algorithm, with a tolerance (`tol`) for convergence, a learning rate (`lr`), the maximum number of epochs (`epochs`), and the learning rate decay (`lr_decay`, see previous Section).

The last two arguments concern the hyperparameters: the `bounds` argument defines the range for the penalty, and if no hyperparameter optimization is desired, an exact penalty can be provided as a list to `penalties`.

6.2.1 Output, fine-tuning, and diagnostics

The algorithm outputs a list containing the estimated loadings in `a` and the values before thresholding in `a_ortho`. The explained variance for each component is returned in `measure` and `measure_ortho`, respectively, and the proportion of explained variance can directly be seen in `explained_var`. The returned object also includes the values of the projected data

in `phi` and the used penalty in `pen_x`. Regarding the optimization, the `loss` is returned to analyze the convergence of the algorithm. If hyperparameter optimization was done via the Bayesian optimization procedure, a `summary` object is returned as well.

As the proposed algorithm estimates a subspace of dimension `k` directly, it is necessary to determine the desired number of components beforehand. A straightforward way is to consider the minimum number of principal components needed to explain a certain proportion of variance, common choices are an 80% or 90% threshold. Using the simulated example from Listing 6.6, we determine the necessary number of components from the full and non-sparse model, where the number of estimated components corresponds to the rank of the data matrix. In Figure 6.2, the covariance structure for this dataset is shown. With this block-diagonal structure, we would expect two important principal components, with a sparsity pattern in a way that the loadings corresponding to the first PC contains non-zero elements in the first 20 positions, and the second PC corresponding to the second block of size 20.

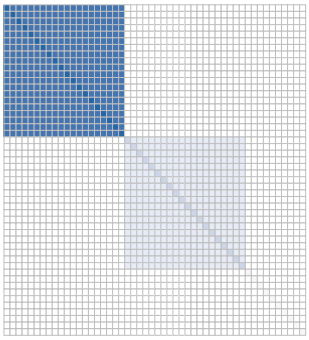


Figure 6.2: Covariance matrix of dataset simulated in Listing 6.6.

The `plot` function for the results class for PCA outputs a barplot for the cumulative explained variance and the variance per component, as shown in Figure 6.3.

After we have determined the desired number of principal components, we can change the elastic net parameter to `alpha_x = 1`, and run the algorithm with the hyperparameter optimization loop. We can again use the `plot` function to check the score dependent on the penalty parameter. It can also be useful to check the convergence of the algorithm by looking at the loss function and adjusting the parameters for the learning rate, learning rate decay, or maximum number of epochs if needed (Figure 6.4).

In addition, the first two principal components are shown, and a diagnostic plot for the identification of outlying observations is provided; see Figure 6.5.

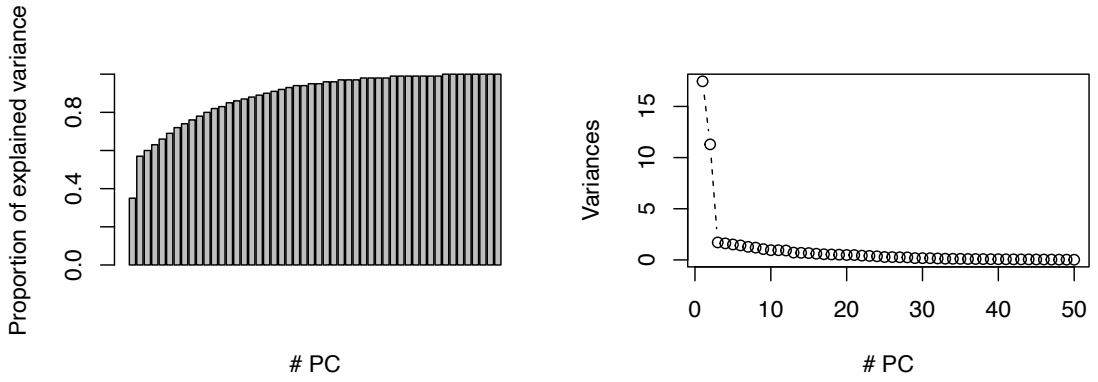


Figure 6.3: Plot of proportion of explained variance and variances for principal components.

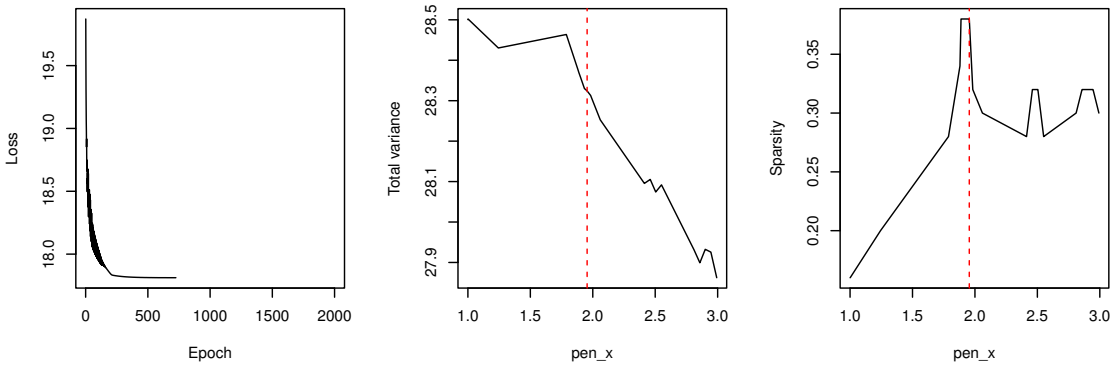


Figure 6.4: Left: Plot of loss function for the training of PCA objective using manifold optimization. Middle and right: A summary of the hyperparameter optimization, showing the estimated variance and resulting sparsity depending on the penalty parameter.

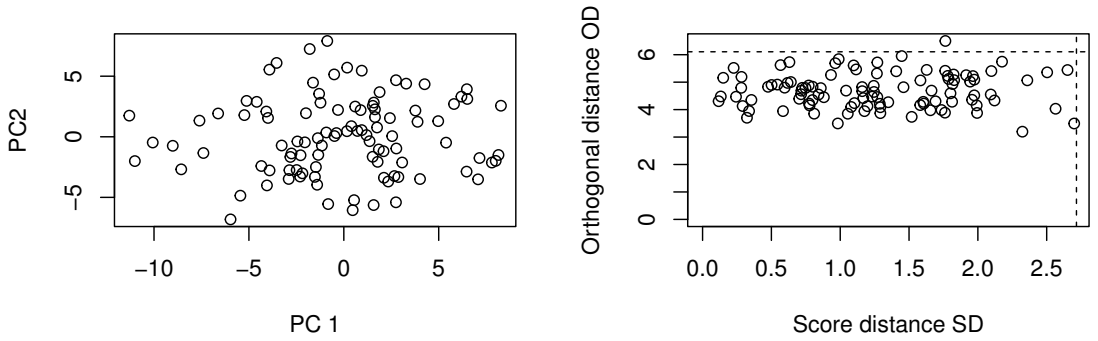


Figure 6.5: Left: Plot of first 2 principal component scores. Right: Diagnostic plot, showing the score distance plotted against the orthogonal distance.

7 Conclusions

The thesis makes contributions to the field of chemometrics and statistical methodology by proposing robust approaches for the analysis of high-dimensional datasets, particularly focusing on FTIR spectra and wear scar images of engine oils. Through the development and validation of novel methodologies, it addresses challenges associated with outlier detection, variable selection, and parameter estimation by applying modern optimization algorithms, offering practical solutions for enhancing the reliability and efficiency of data analysis techniques in complex real-world settings.

In Chapter 2, a methodology for quantifying the relationships between various artificial oil alteration methods and engine oils obtained from a passenger car through the analysis of FTIR spectroscopic data was developed. The approach involves reconstructing the spectra to filter out non-informative variables, followed by simultaneous variable selection and parameter estimation using weighted LASSO. Post-selection inference is then applied to determine confidence intervals for the selected model coefficients. This procedure, demonstrated and validated on real-world data, eliminates the need for manual selection of FTIR absorption bands, allowing for objective filtering of non-informative variables. Integration of expert knowledge is facilitated by the weighted LASSO, resulting in a robust pipeline for FTIR spectroscopic analysis.

Chapter 3 addressed the challenges posed by high-dimensional data sets derived from practical experiments, which often contain outliers and variables that deviate from the majority structure. Robust regression and classification methods developed for low-dimensional data are inadequate for high-dimensional cases due to numerical problems. We provided an overview of robust methods and their implementations, demonstrating their application with two high-dimensional data sets from tribology. Through appropriate pre-processing and sampling strategies, robust statistical methods were demonstrated to enhance prediction performance.

In Chapter 4, we explored the computational challenges associated with robust statistical estimators, particularly in high-dimensional settings. Leveraging optimization procedures from computer science, the chapter investigates their application to robust sparse association estimators. A robust estimation step is followed by an optimization process to solve the decoupled, biconvex problem, incorporating constraints for inducing sparsity. The augmented Lagrangian algorithm and adaptive gradient descent are combined to improve precision and efficiency, with empirical examples highlighting the effectiveness of this approach.

Chapter 5 introduced a cellwise robust method for sparse PCA, enhancing robustness by substituting the squared loss function with a robust version in low-rank matrix approximation. Integration of sparsity-inducing penalties offers modeling flexibility, and an algorithm based on Riemannian stochastic gradient descent enables scalability to high-dimensional data. Named SCRAMBLE (Sparse Cellwise Robust Algorithm for Manifold-based Learning and Estimation), the method demonstrates superiority over established approaches in

terms of casewise and cellwise robustness through simulations and application to real tribology datasets.

The innovative combination of robust estimators and modern optimization techniques offers a versatile toolbox for addressing statistical challenges. By decoupling robust estimation and optimization, the proposed methodology shows promise for extending to other statistical problems, such as robust linear discriminant analysis. Additionally, modifications could enable applications like groupwise PCA or robust data imputation. Investigating theoretical properties like the influence function and breakdown point presents intriguing avenues for further research.

Furthermore, the findings described in the thesis have broader implications, particularly in evaluating lubricant performance. By associating oil condition with lubricating performance using FTIR spectroscopic data, the proposed technique streamlines and potentially replaces time-consuming experiments. This advancement accelerates evaluation and reduces costs. Leveraging the methodology's robustness and predictive power, future research can extend its utility to various statistical modeling tasks.

Bibliography

A. Agocs, S. Budnyk, C. Besser, A. Ristic, M. Frauscher, B. Ronai, and N. Dörr. Production of used engine oils with defined degree of degradation in a large-scale device. *Acta Technica Jaurinensis*, 13:131–150, 05 2020. doi: 10.14513/actatechjaur.v13.n2.546.

A. Agocs, A. Nagy, Z. Tabakov, J. Perger, J. Rohde-Brandenburger, M. Schandl, C. Besser, and N. Dörr. Comprehensive assessment of oil degradation patterns in petrol and diesel engines observed in a field test with passenger cars – conventional oil analysis and fuel dilution. *Tribology International*, 161:107079, 05 2021. doi: 10.1016/j.triboint.2021.107079.

A. Agocs, C. Besser, J. Brenner, S. Budnyk, M. Frauscher, and N. Dörr. Engine oils in the field: A comprehensive tribological assessment of engine oil degradation in a passenger car. *Tribology Letters*, 70, 03 2022. doi: 10.1007/s11249-022-01566-7.

C. Agostinelli, A. Leung, V. J. Yohai, and R. H. Zamar. Robust estimation of multivariate location and scatter in the presence of cellwise and casewise contamination. *Test*, 24: 441–461, 2015.

M. A. Al-Ghouti, Y. S. Al-Degs, and M. Amer. Application of chemometrics and FTIR for determination of viscosity index and base number of motor oils. *Talanta*, 81(3):1096–1101, 2010. ISSN 00399140. doi: 10.1016/j.talanta.2010.02.003.

A. Alfons. *robustHD: Robust methods for high dimensional data*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `http://CRAN.R-project.org/package=robustHD`. R package version 0.4.0.

A. Alfons, C. Croux, and S. Gelper. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013.

A. Alfons, C. Croux, and P. Filzmoser. Robust maximum association estimators. *Journal of the American Statistical Association*, 112:1–29, 02 2016a. doi: 10.1080/01621459.2016.1148609.

A. Alfons, C. Croux, and P. Filzmoser. Robust maximum association between data sets: the R package ccaPP. *Austrian Journal of Statistics*, 45(1):71–79, 2016b.

F. Alqallaf, S. Van Aelst, V. J. Yohai, and R. H. Zamar. Propagation of outliers in multivariate data. *The Annals of Statistics*, pages 311–331, 2009.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. Wiley, New York, NY, 1958. ISBN 0471026409.

ASTM D7484. *Standard Test Method for Evaluation of Automotive Engine Oils for Valve-Train Wear Performance in Cummins ISB Medium-Duty Diesel Engine*. ASTM International, West Conshohocken, PA, USA, 2021.

M. Avella-Medina, H. S. Battey, J. Fan, and Q. Li. Robust estimation of high-dimensional covariance and precision matrices. *Biometrika*, 105(2):271–284, 2018.

V. Barnett, T. Lewis, et al. *Outliers in Statistical Data*. Wiley New York, 3 edition, 1994.

M. Bassbasi, A. Hafid, S. Platikanov, R. Tauler, and A. Oussama. Study of motor oil adulteration by infrared spectroscopy and chemometrics methods. *Fuel*, 104:798–804, 2013. ISSN 00162361. doi: 10.1016/j.fuel.2012.05.058.

D. Bates, M. Maechler, and M. Jagan. *Matrix: Sparse and Dense Matrix Classes and Methods*, 2023. URL `https://CRAN.R-project.org/package=Matrix`. R package version 1.5-4.1.

G. Bécigneul and O. Ganea. Riemannian adaptive optimization methods. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

R. Berk, L. Brown, A. Buja, K. Zhang, and L. Zhao. Valid post-selection inference. *Annals of Statistics*, 41(2):802–837, 2013. ISSN 00905364. doi: 10.1214/12-AOS1077.

D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, Belmont, Massachusetts, 1996. First published by Academic Press, Inc., in 1982.

C. Besser, C. Schneidhofer, N. Dörr, F. Novotny-Farkas, and G. Allmaier. Investigation of long-term engine oil performance using lab-based artificial ageing illustrated by the impact of ethanol as fuel component. *Tribology International*, 46:174–182, 2012. ISSN 0301679X. doi: 10.1016/j.triboint.2011.06.026.

C. Besser, N. Dörr, F. Novotny-Farkas, K. Varmuza, and G. Allmaier. Comparison of engine oil degradation observed in laboratory alteration and in the engine by chemometric data evaluation. *Tribology International*, 65:37–47, 2013. ISSN 0301679X. doi: 10.1016/j.triboint.2013.01.006.

C. Besser, A. Agocs, B. Ronai, A. Ristic, M. Repka, E. Jankes, C. McAleese, and N. Dörr. Generation of engine oils with defined degree of degradation by means of a large scale artificial alteration method. *Tribology International*, 132:39–49, 2019. ISSN 0301679X. doi: 10.1016/j.triboint.2018.12.003.

B. Bhushan. *Principles and Applications of Tribology*. John Wiley & Sons, New York, 2013.

P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577 – 2604, 2008. doi: 10.1214/08-AOS600.

Å. Björck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

S. Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

L. Bottmer, C. Croux, and I. Wilms. Sparse regression for large data sets with outliers. *European Journal of Operational Research*, 297(2):782–794, 2022. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2021.05.049.

K. Boudt, P. J. Rousseeuw, S. Vanduffel, and T. Verdonck. The minimum regularized covariance determinant estimator. *Statistics and Computing*, 30(1):113–128, 2020.

G. E. Box. Non-normality and tests on variances. *Biometrika*, 40(3/4):318–335, 1953.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 01 2011. doi: 10.1561/2200000016.

S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

J. Branco, C. Croux, P. Filzmoser, and M. R. Oliveira. Robust canonical correlations: A comparative study. *Katholieke Universiteit Leuven, Open Access publications from Katholieke Universiteit Leuven*, 20, 01 2003. doi: 10.1007/BF02789700.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

M. Chen, C. Gao, Z. Ren, and H. H. Zhou. Sparse CCA via precision adjusted iterative thresholding. *arXiv preprint arXiv:1311.6186*, 2013.

C. Chimeno-Trinchet, C. Murru, M. E. Díaz-García, A. Fernández-González, and R. Badía-Laíño. Artificial intelligence and fourier-transform infrared spectroscopy for evaluating water-mediated degradation of lubricant oils. *Talanta*, 219:121312, 2020.

H. Chun and S. Keleş. Sparse partial least squares for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 72:3–25, 01 2010. doi: 10.1111/j.1467-9868.2009.00723.x.

D. Chung, H. Chun, and S. Keles. *spls: Sparse Partial Least Squares (SPLS) Regression and Classification*, 2019. URL `https://CRAN.R-project.org/package=spls`. R package version 2.2-3.

C. Croux and C. Dehon. Robust linear discriminant analysis using S-estimators. *The Canadian Journal of Statistics*, 29, 02 2001. doi: 10.2307/3316042.

C. Croux and C. Dehon. Analyse canonique basée sur des estimateurs robustes de la matrice de covariance. *La Revue de Statistique Appliquee*, 2, 01 2002.

C. Croux and C. Dehon. Influence functions of the Spearman and Kendall correlation measures. *Tilburg University, Center for Economic Research, Discussion Paper*, 19, 01 2010. doi: 10.1007/s10260-010-0142-z.

C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226, 2005.

C. Croux, P. Filzmoser, and H. Fritz. Robust sparse principal component analysis. *Technometrics*, 55(2):202–214, 2013.

D. J. Cummins and C. W. Andrews. Iteratively reweighted partial least squares: A performance analysis by Monte Carlo simulation. *Journal of Chemometrics*, 9(6):489–507, 1995. doi: https://doi.org/10.1002/cem.1180090607.

J. P. Cunningham and Z. Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, 2015.

M. De Feo, C. Minfray, M. D. B. Bouchet, B. Thiebaut, and J.-M. Martin. Modtc friction modifier additive degradation: Correlation between tribological performance and chemical changes. *RSC Advances*, 5(114):93786–93796, 2015.

M. Debruyne, S. Höppner, S. Serneels, and T. Verdonck. Outlyingness: Which variables contribute most? *Statistics and Computing*, 29:707–723, 2019.

R. Dezeure, P. Bühlmann, L. Meier, and N. Meinshausen. High-dimensional inference: Confidence intervals, p-values and R-software hdi. *Statistical Science*, 30(4):533–558, 2015.

DIN (Deutsches Institut für Normung). *DIN 51453: Testing of lubricants - Determination of oxidation and nitration of used motor oils - Infrared spectrometric method*, 2004.

A. Douik and B. Hassibi. Manifold optimization over the set of doubly stochastic matrices: A second-order geometry. *IEEE Transactions on Signal Processing*, 67(22):5761–5774, 2019.

N. Dörr, A. Agocs, C. Besser, A. Ristić, and M. Frauscher. Engine oils in the field: A comprehensive chemical assessment of engine oil degradation in a passenger car. *Tribology Letters*, 67:68, 2019a. ISSN 1023-8883. doi: 10.1007/s11249-019-1182-7.

N. Dörr, J. Brenner, A. Ristić, B. Ronai, C. Besser, V. Pejaković, and M. Frauscher. Correlation between engine oil degradation, tribochemistry, and tribological behavior with focus on ZDDP deterioration. *Tribology Letters*, 67(2):62, 2019b. doi: 10.1007/s11249-019-1176-5.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

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

D. Falbel and J. Luraschi. *torch: Tensors and Neural Networks with 'GPU' Acceleration*, 2023. URL `https://CRAN.R-project.org/package=torch`. R package version 0.12.0.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 02 2001. doi: 10.1198/016214501753382273.

Y. Felkel, N. Dörr, F. Glatz, and K. Varmuza. Determination of the total acid number (TAN) of used gas engine oils by IR and chemometrics applying a combined strategy for variable selection. *Chemometrics and Intelligent Laboratory Systems*, 101(1):14–22, 2010. ISSN 01697439. doi: 10.1016/j.chemolab.2009.11.011.

P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Computational Statistics & Data Analysis*, 52(3):1694–1711, 2008.

P. Filzmoser, S. Höppner, I. Ortner, S. Serneels, and T. Verdonck. *crmReg: Cellwise Robust M-Regression and SPADIMO*, 2020a. URL `https://CRAN.R-project.org/package=crmReg`. R package version 1.0.2.

P. Filzmoser, S. Serneels, R. Maronna, and C. Croux. Robust multivariate methods in chemometrics. In S. Brown, R. Tauler, and B. Walczak, editors, *Comprehensive Chemometrics (Second Edition)*, pages 393–430. Elsevier, Oxford, second edition edition, 2020b. ISBN 978-0-444-64166-3. doi: https://doi.org/10.1016/B978-0-12-409547-2.14642-6.

P. Filzmoser, H. Fritz, and K. Kalcher. *pcaPP: Robust PCA by Projection Pursuit*, 2022. URL `https://CRAN.R-project.org/package=pcaPP`. R package version 2.0-3.

P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010a.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010b. URL `https://www.jstatsoft.org/v33/i01/`.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010c. doi: 10.18637/jss.v033.i01. URL `https://www.jstatsoft.org/v33/i01/`.

J. A. Gil and R. Romera. On robust partial least squares (PLS) methods. *Journal of Chemometrics*, 12, 1998.

R. Gnanadesikan and J. R. Kettenring. Robust estimates, residuals, and outlier detection with multiresponse data. *Biometrics*, pages 81–124, 1972.

I. González and S. Déjean. *CCA: Canonical Correlation Analysis*, 2021. URL `https://CRAN.R-project.org/package=CCA`. R package version 1.2.1.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

L. Greco and A. Farcomeni. A plug-in approach to sparse and robust principal component analysis. *Test*, 25:449–481, 2016.

P. R. Griffiths and J. A. de Haseth. *Fourier Transform Infrared Spectrometry*. John Wiley & Sons, Ltd, 2007. ISBN 9780470106310. doi: https://doi.org/10.1002/9780470106310.fmatter.

X. Gu and Q. Wang. Sparse canonical correlation analysis algorithm with alternating direction method of multipliers. *Communications in Statistics - Simulation and Computation*, 49(9):2372–2388, 2020. doi: 10.1080/03610918.2018.1520867.

F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley Series in Probability and Statistics. Wiley, 1986. ISBN 9781118150689.

F. R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848846.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 1st edition edition, 2015. ISBN 9780429171581. doi: https://doi.org/10.1201/b18401.

N. J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343, 2002.

A. Hirri, S. Tagourmate, A. Benamar, F. Kzaiber, and A. Oussama. Prediction of kinematic viscosity in motor oil using FTIR coupled with partial least squares regression. *International Journal of Chemical, Material and Environmental Research*, 4, 2017.

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

I. Hoffmann, S. Serneels, P. Filzmoser, and C. Croux. Sparse partial robust M regression. *Chemometrics and Intelligent Laboratory Systems*, 149:50–59, 2015.

K. Holmberg and A. Erdemir. Influence of tribology on global energy consumption, costs and emissions. *Friction*, 5:263–284, 2017.

K. Holmberg and A. Erdemir. The impact of tribology on energy use and CO2 emission globally and in combustion engine and electric cars. *Tribology International*, 135:389–396, 2019.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964. doi: 10.1214/aoms/1177703732.

M. Hubert and K. Van Driessen. Fast and robust discriminant analysis. *Computational Statistics & Data Analysis*, 45(2):301–320, 2004. ISSN 0167-9473. doi: https://doi.org/10.1016/S0167-9473(02)00299-2.

M. Hubert and K. Vanden Branden. Robust methods for partial least squares regression. *Journal of Chemometrics*, 17(10):537 – 549, 2003. doi: 10.1002/cem.822.

M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

M. Hubert, T. Reynkens, E. Schmitt, and T. Verdonck. Sparse PCA for high-dimensional data with outliers. *Technometrics*, 58:424–434, 10 2016. doi: 10.1080/00401706.2015.1093962.

M. Hubert, P. J. Rousseeuw, and W. Van den Bossche. MacroPCA: An all-in-one PCA method allowing for missing values as well as cellwise and rowwise outliers. *Technometrics*, 61(4):459–473, 2019.

A. Humeau-Heurtier. Texture feature extraction methods: A survey. *IEEE Access*, PP:1–1, 01 2019. doi: 10.1109/ACCESS.2018.2890743.

I. Hutchings and P. Shipway. *Tribology: Friction and Wear of Engineering Materials*. Butterworth-Heinemann, 2017.

IEA. *Global EV Outlook 2023*, 2023. URL `https://www.iea.org/reports/global-ev-outlook-2023`. License: CC BY 4.0.

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, Upper Saddle River, 6th edition, 2007.

I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

P. H. Jost. *Lubrication (tribology), Education and Research: A Report on the Present Position and Industry's Needs*. H.M. Stationery Office, 1966.

J. A. Khan, S. Van Aelst, and R. H. Zamar. Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, 102(480):1289–1299, 2007.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*. ICLR 2015, San Diego, California, May 7 - 9, 2015, 2015.

F. S. Kurnaz, I. Hoffmann, and P. Filzmoser. Robust and sparse estimation methods for high-dimensional linear and logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 172:211–222, 2017.

F. S. Kurnaz, I. Hoffmann, and P. Filzmoser. *enetLTS: Robust and Sparse Methods for High Dimensional Linear and Binary and Multinomial Regression*, 2022. URL `https://CRAN.R-project.org/package=enetLTS`. R package version 1.1.0.

B. Langworthy, R. Stephens, J. Gilmore, and J. Fine. Canonical correlation analysis for elliptical copulas. *Journal of Multivariate Analysis*, 183:104715, 12 2020. doi: 10.1016/j.jmva.2020.104715.

O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. ISSN 0047-259X. doi: https://doi.org/10.1016/S0047-259X(03)00096-4.

J. D. Lee, D. L. Sun, Y. Sun, and J. E. Taylor. Exact post-selection inference, with application to the lasso. *Annals of Statistics*, 44(3):907–927, 2016. ISSN 00905364. doi: 10.1214/15-AOS1371.

K. H. Liland, B.-H. Mevik, and R. Wehrens. *pls: Partial Least Squares and Principal Component Regression*, 2021. URL `https://CRAN.R-project.org/package=pls`. R package version 2.8-0.

J. Machkour, M. Muma, B. Alt, and A. M. Zoubir. A robust adaptive Lasso estimator for the independent contamination model. *Signal Process.*, 174:107608, 2020.

V. Macian, B. Tormos, A. Garcia-Barbera, and A. Tsolakis. Applying chemometric procedures for correlation the FTIR spectroscopy with the new thermometric evaluation of total acid number and total basic number in engine oils. *Chemometrics and Intelligent Laboratory Systems*, 208:104215, 12 2020. doi: 10.1016/j.chemolab.2020.104215.

M. Maechler, P. Rousseeuw, C. Croux, V. Todorov, A. Ruckstuhl, M. Salibian-Barrera, T. Verbeke, M. Koller, E. L. T. Conceicao, and M. Anna di Palma. *robustbase: Basic Robust Statistics*, 2024. URL `http://robustbase.r-forge.r-project.org/`. R package version 0.99-2.

T. Mang and W. Dresel, editors. *Lubricants and Lubrication*, volume 1. John Wiley & Sons, Ltd, 3rd edition, 2017. ISBN 9783527645565. doi: https://doi.org/10.1002/9783527645565.fmatter.

R. Maronna, D. Martin, and V. Yohai. *Robust Statistics: Theory and Methods*. Wiley, New York, 2006.

R. A. Maronna. Robust M-Estimators of Multivariate Location and Scatter. *The Annals of Statistics*, 4(1):51 – 67, 1976. doi: 10.1214/aos/1176343347.

R. A. Maronna and V. J. Yohai. Robust low-rank approximation of data matrices with elementwise contamination. *Technometrics*, pages 295–304, 2008.

R. A. Maronna and R. H. Zamar. Robust estimates of location and dispersion for high-dimensional datasets. *Technometrics*, 44(4):307–317, 2002.

R. A. Maronna, R. D. Martin, V. J. Yohai, and M. Salibián-Barrera. *Robust Statistics: Theory and Methods (with R)*. John Wiley & Sons, 2019.

P. G. Martin, H. Guillou, F. Lasserre, S. Déjean, A. Lan, J.-M. Pascussi, M. SanCristobal, P. Legrand, P. Besse, and T. Pineau. Novel aspects of PPARα-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. *Hepatology*, 45 (3):767–777, 2007.

M. Mayrhofer and P. Filzmoser. Multivariate outlier explanations using Shapley values and Mahalanobis distances. *Econometrics and Statistics*, 2023.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the LASSO. *The Annals of Statistics*, 34, 09 2006. doi: 10.1214/009053606000000281.

N. Meinshausen, L. Meier, and P. Bühlmann. p-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681, 2009. ISSN 01621459. doi: 10.1198/jasa.2009.tm08647.

E. J. Menvouta, S. Serneels, and T. Verdonck. direpack: A Python 3 package for state-of-the-art statistical dimensionality reduction methods. *SoftwareX*, 21:101282, 2023.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $\mathcal{O}\left(1/k^2\right)$. *Doklady Akademii Nauk SSSR*, 269(3):543, 1983.

V. Öllerer, C. Croux, and A. Alfons. The influence function of penalized regression estimators. *Statistics*, 49(4):741–765, 2015.

E. Ollila, D. P. Palomar, and F. Pascal. Shrinking the eigenvalues of M-estimators of covariance matrix. *IEEE Transactions on Signal Processing*, 69:256–269, 2020.

N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

P. Pfeiffer and P. Filzmoser. Robust statistical methods for high-dimensional data, with applications in tribology. *Analytica Chimica Acta*, page 341762, 2023. ISSN 0003-2670. doi: https://doi.org/10.1016/j.aca.2023.341762.

P. Pfeiffer, B. Ronai, G. Vorlaufer, N. Dörr, and P. Filzmoser. Weighted LASSO variable selection for the analysis of FTIR spectra applied to the prediction of engine oil degradation. *Chemometrics and Intelligent Laboratory Systems*, 228:104617, 2022. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2022.104617.

P. Pfeiffer, P. Filzmoser, and I. Wilms. *RobSparseMVA: Robust and Sparse Multivariate Analysis*, 2024. R package version 0.1.0.

E. Polat. The effects of different weight functions on partial robust M-regression performance: A simulation study. *Communications in Statistics-Simulation and Computation*, 49(4):1089–1104, 2020.

A. Ponjavic, T. Lemaigre, M. Southby, and H. Spikes. Influence of NO_x and air on the ageing behaviour of MoDTC. *Tribology Letters*, 65:1–7, 2017.

S. J. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

M. Prizer and J. Sawatzki. Method and device for correcting a spectrum, 2008. US Patent US007359815B2.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2023. URL `https://www.R-project.org/`.

J. Raymaekers and P. Rousseeuw. *cellWise: Analyzing Data with Cellwise Outliers*, 2023a. URL `https://CRAN.R-project.org/package=cellWise`. R package version 2.5.3.

J. Raymaekers and P. J. Rousseeuw. Fast robust correlation for high-dimensional data. *Technometrics*, 63(2):184–198, 2021.

J. Raymaekers and P. J. Rousseeuw. The cellwise minimum covariance determinant estimator. *Journal of the American Statistical Association*, pages 1–12, 2023b.

S. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *6th International Conference on Learning Representations*. ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, 2018.

T. Reynkens. *rospca: Robust Sparse PCA using the ROSPCA Algorithm*, 2018. URL `https://CRAN.R-project.org/package=rospca`. R package version 1.0.4.

D. Rivera-Barrera, H. Rueda-Chacón, and D. Molina V. Prediction of the total acid number (TAN) of colombian crude oils via ATR–FTIR spectroscopy and chemometric methods. *Talanta*, 206(July 2019):120186, 2020. ISSN 00399140. doi: 10.1016/j.talanta.2019.120186.

B. Ronai. Evaluation of chemical and tribometrical data of engine oils by selected multivariate statistics. Master's thesis, TU Wien, 2021.

P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

P. J. Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8(283-297):37, 1985.

P. J. Rousseeuw and W. V. D. Bossche. Detecting deviating data cells. *Technometrics*, 60 (2):135–145, 2018.

P. J. Rousseeuw and C. Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley & Sons, 2005.

P. J. Rousseeuw and G. Molenberghs. Transformation of non positive semidefinite correlation matrices. *Communications in Statistics–Theory and Methods*, 22(4):965–984, 1993.

P. J. Rousseeuw and K. Van Driessen. Computing LTS regression for large data sets. *Data Mining and Knowledge Discovery*, 12(1):29–45, 2006.

M. Sejkorová. Application of FTIR spectrometry using multivariate analysis for prediction fuel in engine oil. *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, 65(3):933–938, 2017. ISSN 12118516. doi: 10.11118/actaun201765030933.

S. Serneels and I. Hoffmann. *sprm: Sparse and Non-Sparse Partial Robust M Regression and Classification*, 2015. URL `https://CRAN.R-project.org/package=sprm`. R package version 1.2.

S. Serneels, C. Croux, P. Filzmoser, and P. J. V. Espen. Partial robust M-regression. *Chemometrics and Intelligent Laboratory Systems*, 79(1-2):55–64, 2005.

Y. She, S. Li, and D. Wu. Robust orthogonal complement principal component analysis. *Journal of the American Statistical Association*, 111(514):763–771, 2016.

H. Shu, X. Wang, and H. Zhu. D-CCA: A decomposition-based canonical correlation analysis for high-dimensional datasets. *Journal of the American Statistical Association*, 115 (529):292–306, 2020. doi: 10.1080/01621459.2018.1543599.

S. Taskinen, C. Croux, A. Kankainen, E. Ollila, and H. Oja. Influence functions and efficiencies of the canonical correlation and vector estimates based on scatter and shape matrices. *Journal of Multivariate Analysis*, 97:359–384, 02 2006. doi: 10.1016/j.jmva.2005.03.005.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246.

R. Tibshirani, R. Tibshirani, J. Taylor, J. Loftus, S. Reid, and J. Markovic. *selectiveInference: Tools for Post-Selection Inference*, 2019. URL `https://CRAN.R-project.org/package=selectiveInference`. R package version 1.2.5.

R. J. Tibshirani. The Lasso problem and uniqueness. *Electronic Journal of Statistics*, 7(1): 1456–1490, 2013. ISSN 19357524. doi: 10.1214/13-EJS815.

V. Todorov and P. Filzmoser. An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software*, 32(3):1–47, 2009a. doi: 10.18637/jss.v032.i03.

V. Todorov and P. Filzmoser. An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software*, 32(3):1–47, 2009b. doi: 10.18637/jss.v032.i03.

V. Todorov and P. Filzmoser. Comparing classical and robust sparse PCA. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, pages 283–291. Springer, 2013.

V. Todorov and A. Pires. Comparative performance of several robust linear discriminant analysis methods. *Revstat - Statistical Journal*, 5:63–83, 04 2007. doi: 10.57805/revstat.v5i1.42.

J. W. Tukey. A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics*, pages 448–485, 1960.

K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, 1 edition, 2009. doi: https://doi.org/10.1201/9781420059496.

S. Waaijenborg, P. C. Verselewel de Witt Hamer, and A. H. Zwinderman. Quantifying the association between gene expressions and dna-markers by penalized canonical correlation analysis. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008.

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

I. N. Wakeling and H. Macfie. A robust PLS procedure. *Journal of Chemometrics*, 6(4): 189 – 198, 1992.

S. Wartewig. *IR and Raman Spectroscopy: Fundamental Processing*. Spectroscopic Techniques: An Interactive Course. Wiley, 2006. ISBN 9783527606436.

R. D. Whitby, editor. *Lubricant Analysis and Condition Monitoring*. CRC Press, 1st edition, 2021. ISBN 9781003245254. doi: https://doi.org/10.1201/9781003245254.

H. Wickham, J. Hester, W. Chang, and J. Bryan. *devtools: Tools to Make Developing R Packages Easier*, 2022. URL `https://CRAN.R-project.org/package=devtools`. R package version 2.4.5.

I. Wilms and C. Croux. Sparse canonical correlation analysis from a predictive point of view. *SSRN Electronic Journal*, 57, 01 2015a. doi: 10.2139/ssrn.2381968.

I. Wilms and C. Croux. Robust sparse canonical correlation analysis. *BMC Systems Biology*, 10, 01 2015b. doi: 10.1186/s12918-016-0317-9.

S. Wilson. *ParBayesianOptimization: Parallel Bayesian Optimization of Hyperparameters*, 2022. URL `https://CRAN.R-project.org/package=ParBayesianOptimization`. R package version 1.2.6.

D. Witten and R. Tibshirani. Penalized classification using fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 2011.

D. Witten and R. Tibshirani. *PMA: Penalized Multivariate Analysis*, 2020. URL `https://CRAN.R-project.org/package=PMA`. R package version 1.2.1.

D. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics (Oxford, England)*, 10:515–34, 05 2009. doi: 10.1093/biostatistics/kxp008.

A. Wolak, W. Krasodomski, and G. Zając. FTIR analysis and monitoring of used synthetic oils operated under similar driving conditions. *Friction*, 8:995–1006, 10 2020. ISSN 22237704. doi: 10.1007/s40544-019-0344-9.

A. Wolak, J. Molenda, G. Zając, and P. Janocha. Identifying and modelling changes in chemical properties of engine oils by use of infrared spectroscopy. *Measurement*, 186, 12 2021. ISSN 02632241. doi: 10.1016/j.measurement.2021.110141.

Z. Xie, X. Feng, and X. Chen. Partial least trimmed squares regression. *Chemometrics and Intelligent Laboratory Systems*, 221:104486, 2022. ISSN 0169-7439. doi: https://doi.org/10.1016/j.chemolab.2021.104486.

V. J. Yohai and R. H. Zamar. High breakdown-point estimates of regression by means of the minimization of an efficient scale. *Journal of the American Statistical Association*, 83 (402):406–413, 1988.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006. ISSN 01621459. doi: 10.1198/016214506000000735.

H. Zou and T. Hastie. Regression shrinkage and selection via the elastic net, with applications to microarrays. *JR Stat Soc Ser B*, 67:301–20, 2003.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 13697412, 14679868.

Curriculum Vitae

Personal Data

Name	**Pia Pfeiffer**
Date of Birth	April 12th, 1994
Nationality	Austrian
E-mail	pia.pfeiffer@tuwien.ac.at

Professional Experience

06/2021–present	Project assistant at the CSTAT group, TU Wien, Austria
12/2018–05/2021	Data Scientist / Visualization & Finance, Hutchison Drei Austria GmbH, Vienna, Austria
07/2015–06/2017	Project assistant, Robert Bosch AG, Vienna, Austria
07/2013–06/2015	Intern, OMV Exploration & Production GmbH, Vienna, Austria

Education

06/2021–present	Doctoral program in Technical Sciences Technical Mathematics, TU Wien, Austria
12/2016–01/2019	Master's program Technical Mathematics, TU Wien, Austria
07/2017–11/2017	University Exchange Semester, Queensland University of Technology, Brisbane, Australia
10/2012–11/2016	Bachelor's program Technical Mathematics, TU Wien, Austria
06/2012	Matura (high school diploma), Konrad Lorenz Gymnasium Gänserndorf, Austria

List of Presentations

Pfeiffer, P., Alfons, A., & Filzmoser, P. (2024). Robust maximum association for high-dimensional data. Austrian Statistical Days, Vienna, Austria.

Pfeiffer, P. & Filzmoser, P. (2023). Robust penalized multivariate analysis for high-dimensional data. 14-th Scientific Meeting of the Classification and Data Analysis Group (CLADAG) of the SIS, Salerno, Italy.

Pfeiffer, P., Alfons, A., & Filzmoser, P. (2023). Robust and Sparse CCA: An Algorithm for Dimension Reduction via Sparsity Inducing Penalties. International Conference on Robust Statistics, Toulouse, France.

Pfeiffer, P., Ronai, B., Vorlaufer, G., Dörr, N., & Filzmoser, P. (2022). Prediction of engine oil degradation based on FTIR spectroscopic data. Symposium 2022 der Österreichischen Tribologischen Gesellschaft (ÖTG), Wr. Neustadt, Austria.

Pfeiffer, P., Alfons, A., & Filzmoser, P. (2022). Efficient computation of robust multivariate maximum association. 24th International Conference on Computational Statistics, Bologna, Italy.

Pfeiffer, P., Ronai, B., Vorlaufer, G., Dörr, N., & Filzmoser, P. (2022). Prediction of engine oil degradation based on FTIR spectra and weighted LASSO regression. 5th Young Tribological Researcher Symposium (YTRS), Karlsruhe, Germany.

Pfeiffer, P., Ronai, B., Vorlaufer, G., Dörr, N., & Filzmoser, P. (2022). Weighted LASSO feature selection for the analysis of FT-IR spectra applied to relate engine oil degradation patterns. Tribology International Conference 2022, Barcelona, Spain.

List of Publications

Pfeiffer, P., Ronai, B., Vorlaufer, G., Dörr, N., & Filzmoser, P. (2022). Weighted LASSO variable selection for the analysis of FTIR spectra applied to the prediction of engine oil degradation. Chemometrics and Intelligent Laboratory Systems, 228, 104617. `http://dx.doi.org/10.1016/j.chemolab.2022.104617.`

Pfeiffer, P., & Filzmoser, P. (2023). Robust statistical methods for high-dimensional data, with applications in tribology. Analytica Chimica Acta (2023): 341762. `https://doi.org/10.1016/j.aca.2023.341762`

Wien, am 29. April 2024

Pia Pfeiffer