Sparsity Control for Robustness and Social Data Analysis

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Gonzalo Mateos Buckstein

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Professor Georgios B. Giannakis, Advisor

May 2012

Acknowledgments

First and foremost, my deepest gratitude goes to my Ph. D. advisor Prof. Georgios B. Giannakis. I would like to thank him for giving me the opportunity to embark on this journey as a graduate student, a real privilege for which I am really honored. His guidance and constant encouragement has made me become not only a better researcher, but also a better person. This thesis would not have been possible without all his insightful suggestions.

Due thanks go to Profs. Mos Kaveh, Guillermo Sapiro, Nikos Sidiropoulos, and Niels Waller for agreeing to serve on my committee.

Throughout my graduate studies, I had the opportunity to collaborate with several individuals and I greatly benefited from their vision, ideas, and insights. I would like to extend my gratitude to Prof. Arindam Banerjee, Juan-Andrés Bazerque, Dr. Shahrokh Farahmand, Dr. Vassilis Kekatos, Morteza Mardani, Prof. Yannis Schizas, Prof. Nikos Sidiropoulos, and Hao Zhu. I would also like to acknowledge the grants that support financially our research.

The material in this thesis benefited from discussions with current and former members of SPiNCOM: Dr. Daniele Angelosante, Brian Baingana, Dr. Alfonso Cano, Dr. Emiliano Dall'Anese, Yannis Delis, Pedro Forero, Nikos Gatsis, Dr. Seung-Jun Kim, Guobing Li, Prof. Geert Leus, Prof. Antonio G. Marques, Dr. Eric Msechu, Ketan Rajawat, Dr. Tairan Wang, Dr. Yuchen Wu, Nasim Yahya Soltani, Dr. Yingqun Yu, and Yu Zhang. I am not forgetting Prof. Alejandro Ribeiro and his family: I want to give them special thanks for – among many other things – their hospitality and help upon my arrival to Minneapolis. At this point I would also like to thank my colleagues at the Instituto de Ingeniería Eléctrica, Universidad de la República, who taught me what electrical engineering is all about; and specially Profs. Gregory Randall and Alicia Fernandez that encouraged me to pursue graduate studies abroad.

My family and friends, some of which I've already mentioned above, are the most important part in making my life a wonderful experience. For this reason I wish to thank those here in Minneapolis, and specially all of my long-time friends that are far way, but nonetheless feel as close to me as in the past. Por último, quiero agradecer a Cachito, Mamá, Fede y Agus por la familia que tengo. Otra vez, esta tesis va dedicada a ustedes.

Gonzalo Mateos, Minneapolis, November 30 2011.

Abstract

The information explosion propelled by the advent of personal computers, the Internet, and the global-scale communications has rendered *statistical learning* from data increasingly important for analysis and processing. The ability to mine valuable information from unprecedented volumes of data will facilitate preventing or limiting the spread of epidemics and diseases, identifying trends in global financial markets, protecting critical infrastructure including the smart grid, and understanding the social and behavioral dynamics of emergent social-computational systems. Along with data that adhere to postulated models, present in large volumes of data are also those that do not – the so-termed *outliers*. This thesis contributes in several issues that pertain to resilience against outliers, a fundamental aspect of statistical inference tasks such as estimation, model selection, prediction, classification, tracking, and dimensionality reduction, to name a few.

The recent upsurge of research toward compressive sampling and parsimonious signal representations hinges on signals being sparse, either naturally, or, after projecting them on a proper basis. The present thesis introduces a neat link between *sparsity* and *robustness* against outliers, even when the signals involved are not sparse. It is argued that controlling sparsity of model residuals leads to statistical learning algorithms that are computationally affordable and universally robust to outlier models. Even though focus is placed first on robustifying linear regression, the universality of the developed framework is highlighted through diverse generalizations that pertain to: i) the information used for selecting the sparsity-controlling parameters; ii) the nominal data model; and iii) the criterion adopted to fit the chosen model. Explored application domains include *preference measurement* for consumer utility function estimation in marketing, and *load curve cleansing* – a critical task in power systems engineering and management.

Finally, robust principal component analysis (PCA) algorithms are developed to extract the most informative low-dimensional structure, from (grossly corrupted) high-dimensional data. Beyond its ties to robust statistics, the developed outlier-aware PCA framework is versatile to accommodate novel and scalable algorithms to: i) track the low-rank signal subspace as new data are acquired in real time; and ii) determine principal components robustly in (possibly) infinite-dimensional feature spaces. Synthetic and real data tests corroborate the effectiveness of the proposed robust PCA schemes, when used to identify aberrant responses in personality assessment surveys, as well as unveil communities in social networks, and intruders from video surveillance data.

Contents

List of Figures

List of Tables

Chapter 1

Robust Statistical Learning from 'Big Data'

The information explosion propelled by the advent of personal computers, the Internet, and the global-scale communications has rendered *statistical learning* from data increasingly important for analysis and processing. At any given time instant and all around the globe, large volumes of data are being generated by today's ubiquitous communication and mobile sensing devices such as cell-phones, surveillance cameras and microphones, e-commerce sites, wireless sensor networks, medical devices, and social-networking sites; see e.g., Fig. 1.1. The term 'Big Data' has been coined to describe this data deluge phenomenon, and as mentioned in a recent article published in The Economist 'The effect (of Big Data) is being felt everywhere, from business to science, from government to the arts' [30]. The ability to mine valuable information from unprecedented volumes of data will facilitate preventing or limiting the spread of epidemics and diseases, identifying trends in global financial markets, combating crime, protecting critical infrastructure including the smart grid, understanding the social and behavioral dynamics of emergent social-computational systems, and the advancement of science as a whole. But the great promise comes with great as well as exciting research challenges; as Google's chief economist explains in the same article 'Data are widely available, what is scarce is the ability to extract wisdom from them'. While significant progress has been made in the last decade towards achieving the

Figure 1.1: North America's map of Flickr and Twitter locations obtained from [41]. The map was generated by drawing: i) a red dot at the location where a picture was taken and uploaded to Flickr (during an unspecified time horizon); ii) a blue dot at the location where a Twitter tweet was generated; and iii) a white dot at the location that has been posted to both Flickr and Twitter.

ultimate goal of 'making sense of it all', the consensus is that we are still quite not there.

1.1 Motivation and Context

Along with data that adhere to postulated models, present in large volumes of data are also those that do not – the so-termed *outliers*. Resilience to outliers is of paramount importance in a plethora of statistical learning tasks such as estimation, model selection, prediction, classification, tracking, and dimensionality reduction, to name a few. Due to its universal applicability, the method of least-squares (LS) is the workhorse of statistical learning. Unfortunately, LS is known to be very sensitive to outliers, since a single outlying datum can

be sufficient to negatively influence (bias) the fit [63,104]. Naturally, this undesirable property extends to most learning methods that minimize a residual sum of *squared* errors as part of their criterion. To illustrate this observation with the simplest of examples, consider data $\{x_n\}_{n=1}^N$ and form the sample mean estimator $\hat{x}_{\ell_2} = (x_1 + \ldots + x_N)/N$, obtained as solution to the ℓ_2-norm minimization problem

$$\hat{x}_{\ell_2} := \arg\min_{\theta} \sum_{n=1}^{N} (x_n - \theta)^2. \tag{1.1}$$

It is apparent that a single arbitrarily large observation x_n can result in an arbitrarily large estimate $\hat{x}_{\ell_2}$. In the robust statistics parlance, this means that the *breakdown point* of the sample mean estimator is zero; see e.g. [63]. The higher the breakdown point of an estimator, that is, the higher the fraction of large observations that the estimator can tolerate without yielding an arbitrarily large result, the more robust it is. In particular, the median estimator $\hat{x}_{\ell_1} = \text{med}(x_1, \ldots, x_N)$ given by

$$\hat{x}_{\ell_1} := \arg\min_{\theta} \sum_{n=1}^{N} |x_n - \theta| \tag{1.2}$$

attains the maximum possible breakdown point of 0.5. Relative to (1.1), ℓ_1-norm regression in (1.2) downweighs the effect of large residuals $r_n := x_n - \theta$.

Beyond ℓ_1 regression, robust alternatives to LS include the M-estimators, which are maximum-likelihood (ML) optimal for a class of ϵ-contaminanted outlier models [63]. Other options are least-trimmed squares (LTS) estimators, which remove outliers from the LS fit [104]. LTS estimators have high breakdown point, but prohibitive complexity except for small sample sizes [103]. Random sample consensus (RANSAC) provides a computationally tractable, near-LTS alternative, especially popular in computer vision for coping with a large number of outliers [42, 58]. From a high-level vantage point, RANSAC randomly draws subsets of a given training set of data samples, fits a model, and evaluates whether the number of samples consistent with the model is large enough to accept the fit.

In this dissertation, a universal sparsity-controlling outlier rejection framework is developed for robust learning from high-dimensional data. The novel framework is rooted at the crossroads of robust statistics [63, 104], the least-absolute shrinkage and selection operator

(Lasso) for sparse regression [59, 110], and convex optimization [11, 13]. Leveraging the attribute of *sparsity* has made headways across science and engineering in recent years, with well documented merits in terms of complexity control through variable selection (automatically single out the most important variables in high-dimensional feature space). A main contribution of this thesis is to show that controlling sparsity in model residuals, can be tantamount to controlling the number of outliers rejected. In addition, neat connections are established between the seemingly unrelated fields of robust statistics and sparsity-aware regression using the Lasso.

1.1.1 The Lasso

Consider the classical setup for linear regression, in which an input vector $\mathbf{x} := [x_1, \ldots, x_p]' \in \mathbb{R}^p$ is given, and the goal is to predict the real-valued scalar response y, where $'$ stands for transposition. A linear approximation to the regression function $E[y|\mathbf{x}]$ is adopted to this end, namely $f(\mathbf{x}) = \theta_0 + \mathbf{x}'\boldsymbol{\theta}$, where $\boldsymbol{\theta} := [\theta_1, \ldots, \theta_p]' \in \mathbb{R}^p$ is the vector of model coefficients, and the intercept is θ_0. Given a training data set $\{y_n, \mathbf{x}_n\}_{n=1}^N$, the model parameters $\{\theta_0, \boldsymbol{\theta}\}$ are to be estimated according to a suitable criterion. The long standing and most popular criterion is LS, which: i) often times yields unsatisfactory prediction accuracy; and ii) fails to provide a parsimonious model estimate whereby only the most relevant predictor variables are selected; see e.g. [59]. Parsimony is a particularly attractive feature for interpretation purposes, especially in high-dimensional problems commonly arising with 'Big Data', where p is large.

The least-absolute shrinkage and selection operator [110], abbreviated as *Lasso*, is a regularization technique capable of performing both estimation and variable selection. It combines the features of ridge regression and subset selection, the two popular techniques traditionally employed to improve the LS estimates by separately dealing with the aforementioned limitations i) and ii). Upon defining $\mathbf{y} := [y_1 \ldots y_N]' \in \mathbb{R}^N$ and the regression matrix $\mathbf{X} := [\mathbf{x}_1 \ldots \mathbf{x}_N]' \in \mathbb{R}^{N \times p}$, the Lasso estimator is the minimizer of the following nonsmooth convex optimization problem

$$\hat{\boldsymbol{\theta}}_{\text{Lasso}} = \arg \min_{\{\theta_0, \boldsymbol{\theta}\}} \frac{1}{2} \|\mathbf{y} - \mathbf{1}_N \theta_0 - \mathbf{X}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \tag{1.3}$$

where $\mathbf{1}_N$ denotes the $N \times 1$ vector of all ones, and $\|\theta\|_1 := \sum_{i=1}^{p} |[\boldsymbol{\theta}]_i|$ is the *sparsity-encouraging* ℓ_1-norm of vector $\boldsymbol{\theta}$. The nonnegative parameter λ controls the amount of sparsity (number of nonzero entries in $\hat{\boldsymbol{\theta}}_{\text{Lasso}}$), and is typically chosen via model selection techniques such as cross-validation (CV); see e.g., [59]. Problem (1.3) is also known as *basis pursuit denoising*, a term coined by [26] in the context of finding the best sparse signal expansion over an overcomplete basis.

Lasso is equivalent to a quadratic programming (QP) problem [110]; hence, an iterative procedure is required to determine $\hat{\boldsymbol{\theta}}_{\text{Lasso}}$ for a given value of λ. While standard QP solvers can be certainly invoked to this end, an increasing amount of effort has been put recently into developing fast algorithms that capitalize on the unique properties of the Lasso. The LARS-Lasso algorithm [34] is an efficient scheme for computing the entire path of solutions (corresponding to all values of λ), elsewhere referred to as homotopy paths [34, 48], or, regularization paths [44]. LARS capitalizes on the piecewise linearity of the Lasso path of solutions, while incurring the complexity of a single LS fit, i.e., when $\lambda = 0$. Homotopy algorithms have been also developed to solve the Lasso online, when data pairs $\{y_n, \mathbf{x}_n\}$ are collected sequentially in time [6, 48]; see also [77] and [5]. Coordinate descent algorithms have been shown competitive, even outperforming LARS when p is large, as demonstrated in [44]; see also [126], and the references therein. Coordinate descent solvers capitalize on the fact that Lasso can afford a very simple solution in the scalar case, which is given in closed form in terms of a soft-thresholding operator $\mathcal{S}(\cdot, \lambda)$. Specifically, the scalar Lasso problem gives the solution

$$\begin{aligned}\hat{\theta}_{\text{Lasso}} &= \arg\min_{\theta} \frac{1}{2}(y-\theta)^2 + \lambda|\theta|_1 \\ &= \begin{cases} y-\lambda, & y > \lambda \\ 0, & |y| < \lambda \\ y+\lambda, & y < -\lambda \end{cases} \\ &:= \mathcal{S}(y, \lambda)\end{aligned}$$

where the operator $\mathcal{S}(y, \lambda) := \text{sign}(y)\max(|y| - \lambda, 0)$ is shown in Fig. 1.2.

Other approaches based on variable decoupling have been proposed by [53] and [125],

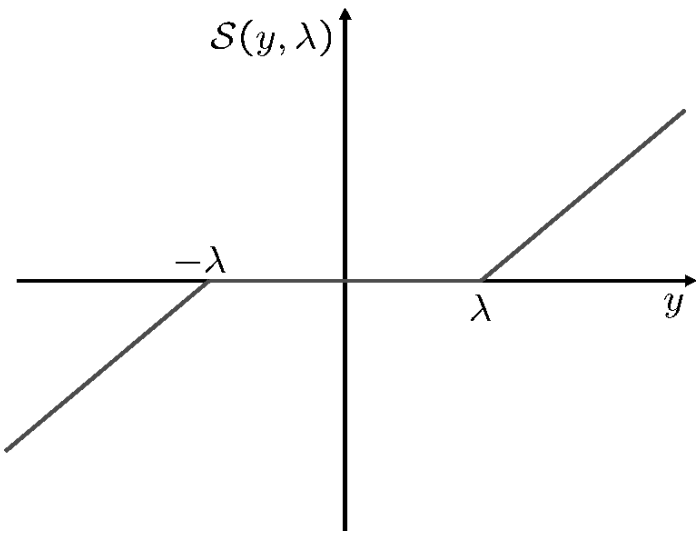


Figure 1.2: The soft-thresholding operator.

while Nesterov's accelerated proximal gradient algorithms have also enjoyed increasing popularity recently [9]. Since $\|\boldsymbol{\theta}\|_1$ is nondifferentiable, iterative subgradient methods are also applicable despite their generally slow convergence rate; see [105] for a survey.

1.1.2 A motivating application domain

The growing volume of consumer-generated media provides ample testament to the urgent need for understanding the complex interactions between people and computers. Contemporary examples include financial markets involving human and computer traders and regulators; computer-mediated bidding and auction systems such as Priceline or e-bay; massively multiplayer online role playing games (MMORPG); online retailing and recommendation systems; collective works such as open source software development and Wikipedia; online 'challenge' competitions; and cloud computing among many others. Yet as diverse and fascinating as these examples are, they can only allude to what will come next. Understanding the dynamics of the emergent *social computational systems* (SoCS), is a critical task to social and behavioral engineering towards desired collective objectives. SoCS involve human and computer 'actors' whose individual capabilities, values, and preferences determine modes of social engagement. Thus, a holistic approach to *preference measurement, analysis, and*

management (PM for short) holds the keys to understanding and engineering SoCS.

PM has a long history in marketing, retailing, product design, healthcare, and also psychology and behavioral sciences, where *conjoint analysis* (CA - the PM 'workhorse') is commonly used [55,60,90]. In a nutshell, the goal of PM is to learn the utility function of an individual or group of individuals from expressed preference data (buying patterns, surveys, ratings, recommendations, etc). The pioneering idea behind CA is to decompose consumer preferences, into weights (partworths) of judiciously selected product attributes [55]. This not only allows one to understand the preferences of existing products, but also to *predict* utilities generated by new products obtained as combinations of the studied attributes. Beyond profit-maximizing firms, the beneficiaries of PM have progressively expanded to include consumers (e.g., using e-recommender systems), policy makers, and academics from diverse fields with possibly altruistic and social welfare objectives [90]. Although the benefits of CA have been well appreciated in the marketing and healthcare sectors, only recently researchers have started to explore its links with SoCS under the general umbrella of PM – an area of markedly growing interest given the exponential increase of preference data (choices, rankings, surveys, questionnaires) generated through the web, and the associated challenges emerging with contemporary requirements for socially intelligent computing.

With few exceptions, PM has traditionally been an off-line task, assuming mostly 'rational' individuals, clean data collected via paper and pencil questionnaires, and linear utilities that depend on only a few product attributes. These are very restrictive for existing and forthcoming SoCS, which may involve thousands of underlying variables and include grossly inconsistent 'social liars' or even malicious actors. In addition, the desiderata of web-collected data come with challenges. Data collected online often include invalid protocols due to a respondent's linguistic incompetence, careless inattentiveness, or deliberate misrepresentation [66]. Recent research suggests that aberrant or otherwise invalid response data is higher when data are collected over the web rather than by more traditional means. This has led a recent panel of experts on online data collection [71, p. 108] to suggest that 'researchers should use exploratory data analysis and systematic data mining to identify and eliminate records with anomalous data patterns or to determine the need for statistics

robust to outliers.'

Towards overcoming the aforementioned challenges and limitations, research in this thesis aims at contributing to build the next generation of socially-intelligent computing systems.

1.2 Thesis Outline and Contributions

The research dealt with in this thesis contributes to the advancement of robust statistical learning theory and methods, by putting forth an universal sparsity-controlling outlier rejection (USPACOR) framework. It is shown that a sparsity tuning parameter (λ) in Lasso controls the degree of sparsity in the sought estimator, and the number of outliers rejected. Related approaches for robust linear regression can be found in [46, 64]. The major difference is that λ in these works is tied to a preselected outlier model, whereas here it is dictated by the data. This promotes universality and a systematic approach leveraging solvers for all *robustifaction* (a.k.a. homotopy) paths of Lasso; that is, for all values of λ_1 [34, 45, 130]. In this sense, USPACOR capitalizes on but *is not limited to* sparse settings (few outliers), since one can examine the gamut of sparsity levels along the robustification path. Beyond linear regression models, USPACOR's universality is highlighted through diverse generalizations pertaining to: i) the information used for selecting λ; ii) the nominal data model; and iii) the criterion adopted to fit the chosen model. Accordingly, we can divide the contributions of this thesis in three interrelated thrusts:

[T1] Robust learning for conjoint analysis. Driven by the explosion of web-collected metric and choice-based preference data, the objective of this thrust is to develop utility function (partworth) estimation algorithms under the linear regression and classification paradigms. Different from existing approaches, the robust algorithms sought are implementable in a distributed fashion to facilitate coping with massive amounts of choice data dispersed over the network, and account for consumer heterogeneity.

[T2] Robust nonparametric regression. In the dilemma of trusting a parametric model versus trusting the data, nonparametric regression methods favor the latter.

The goal in this thrust is to robustify nonparametric and kernel methods (such as smoothing splines) against outliers. Also useful in the context of PM, the nonparametric models investigated in this thrust can capture interdependencies among product attributes, an attractive feature lacking with linear utilities.

[T3] Robust principal component analysis. The goal here is to robustify principal component analysis (PCA), thus enabling the possibility of extracting informative low-dimensional structure from (grossly corrupted) high-dimensional data. Real-time algorithms are developed to process data as it is acquired on-the-fly, and a novel robust kernel PCA algorithm is shown effective in unveiling communities in social networks.

To gauge the effectiveness of the proposed robust methods, extensive experiments with computer generated data are reported throughout the thesis. These are important since they provide a 'ground truth', against which performance can be assessed by evaluating suitable figures of merit. Nevertheless, no effort of this kind can have impact without thorough testing, experimentation, and validation with real data. To this end, tests on real video surveillance, social network, electric grid load curve, and personality assessment data are included to compile a comprehensive validation package.

Elaborate discussion of [T1]-[T3] follows next along with a succinct literature review per thrust. Moreover, contributions of this thesis in each case are pointed out.

1.2.1 Robust learning for conjoint analysis

To address the challenges outlined in Section 1.1.2, Chapter 2 develops novel noise and outlier-robust partworth estimators for both metric and choice-based CA. For metric conjoint data, questionnaire responses (product ratings) are assumed generated from a linear regression model, which explicitly incorporates an unknown *sparse* vector of outliers. The proposed partworth estimator minimizes a tradeoff between fidelity to the training data, and sparsity of the outlier vector encouraged via a natural ℓ_0-(pseudo)norm regularization; or its convex ℓ_1-norm surrogate leading to the Lasso [34, 102]. While regularization for model complexity control in conjoint estimation has well-documented merits in terms of

generalization capability [29,35,37], the major innovative claim here is that *sparsity control* is tantamount to *robustness control*. This is indeed the case since a tunable parameter in Lasso, controls the degree of sparsity in the estimated vector of model outliers. Selection of tuning parameters could be at first thought as a mundane task. However, arguing on the importance of such task as well as devising principled methods to effectively carry out sparsity control, are at the heart of Chapter 2's contribution to the field of CA.

For choice-based CA, a novel sparsity-controlling classifier is developed that is capable of attaining desirable tradeoffs between model fit and complexity, while at the same time controlling robustness and revealing the outliers present. Simulated tests demonstrate that: i) USPACOR outperforms RANSAC in a linear regression setup, especially when the percentage of outliers is high; and ii) the proposed sparsity-controlling classifier for choice-based data consistently outperforms the SVM alternative of [35].

1.2.2 Robust nonparametric regression

Consider again the prototypical supervised learning problem, in which an input vector $\mathbf{x} := [x_1, \ldots, x_p]' \in \mathbb{R}^p$ is given, and the goal is to predict the real-valued scalar response $y = f(\mathbf{x})$. Function f is unknown, to be estimated from a training data set $\mathcal{T} := \{y_n, \mathbf{x}_n\}_{i=1}^N$. When f is assumed to be a member of a finitely-parameterized family of functions, standard (non-) linear regression techniques can be adopted. If on the other hand, one is only willing to assume that f belongs to a (possibly infinite dimensional) space of 'smooth' functions $\mathcal{H}$, then a *nonparametric* approach is in order, and this is the focus of Chapter 3.

Without further constraints beyond $f \in \mathcal{H}$, functional estimation from finite data is an ill-posed problem. To bypass this challenge, the problem is typically solved by minimizing appropriately regularized criteria, allowing one to control model complexity; see, e.g., [36, 111]. It is then further assumed that $\mathcal{H}$ has the structure of a reproducing kernel Hilbert space (RKHS), with corresponding positive definite reproducing kernel function $K(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, and norm denoted by $\|\cdot\|_{\mathcal{H}}$. Under the formalism of *regularization networks*,

one seeks $\hat{f}$ as the solution to the variational problem

$$\min_{f\in\mathcal{H}} \left[\sum_{n=1}^{N} V(y_n - f(\mathbf{x}_n)) + \mu \|f\|_{\mathcal{H}}^2 \right] \tag{1.4}$$

where $V(\cdot)$ is a convex loss function, and $\mu \geq 0$ controls complexity by weighting the effect of the smoothness functional $\|f\|_{\mathcal{H}}^2$. Interestingly, the Representer Theorem asserts that the unique solution of (1.4) is finitely parametrized and has the form $\hat{f}(\mathbf{x}) = \sum_{n=1}^{N} \theta_n K(\mathbf{x}, \mathbf{x}_n)$, where $\{\theta_n\}_{n=1}^{N}$ can be obtained from $\mathcal{T}$; see e.g., [92, 119]. Further details on RKHS, and in particular on the evaluation of $\|f\|_{\mathcal{H}}$, can be found in e.g., [119, Ch. 1]. A fundamental relationship between model complexity control and generalization capability, i.e., the predictive ability of $\hat{f}$ beyond the training set, was formalized in [118].

The generalization error performance of approaches that minimize the sum of squared model residuals [that is $V(u) = u^2$ in (1.4)] regularized by a term of the form $\|f\|_{\mathcal{H}}^2$, is degraded in the presence of outliers. This is because the LS part of the cost is not robust, and can result in severe overfitting of the (contaminanted) training data [63]. Recent efforts have considered replacing the squared loss with a robust counterpart such as Huber's function, or its variants, but lack a data-driven means of selecting the proper threshold that determines which datum is considered an outlier [132]; see also [78]. Other approaches have instead relied on the so-termed ϵ-insensitive loss function, originally proposed to solve function approximation problems using support vector machines (SVMs) [118]. These family of estimators often referred to as support vector regression (SVR), have been shown to enjoy robustness properties; see e.g., [73, 88, 107] and references therein. In [27], improved performance in the presence of outliers is achieved by refining the SVR solution through a subsequent robust learning phase.

The starting point in Chapter 3 is a variational least-trimmed squares (VLTS) estimator, suitable for robust function approximation in $\mathcal{H}$. It is established that VLTS is closely related to an (NP-hard) ℓ_0-(pseudo)norm-regularized estimator, adopted to fit a regression model that explicitly incorporates an unknown *sparse* vector of outliers [46]. As in compressive sampling (CS) [115], efficient (approximate) solvers are obtained by replacing the outlier vector's ℓ_0-norm with its closest convex approximant, the ℓ_1-norm. This leads

naturally to a variational M-type estimator of f, also shown equivalent to a Lasso [110] on the vector of outliers. A tunable parameter in Lasso *controls* the *sparsity* of the estimated vector, and the number of outliers as a byproduct. Hence, effective methods to select this parameter are of paramount importance.

The link between ℓ_1-norm regularization and robustness was also exploited for parameter (but not function) estimation in [46] and [64]; see also [124] for related ideas in the context of face recognition, and error correction codes [21, 22]. In [46] however, the selection of Lasso's tuning parameter is only justified for Gaussian training data; whereas a fixed value motivated by CS error bounds is adopted in the Bayesian formulation of [64]. Here instead, a more general and systematic approach is pursued, building on contemporary algorithms that can efficiently compute all *robustifaction* paths of Lasso solutions (also known as homotopy paths) obtained for all values of the tuning parameter [34, 45, 48, 125]. An estimator with reduced bias and improved generalization capability is also obtained in Chapter 3, after replacing the ℓ_0-norm with a nonconvex surrogate, instead of the ℓ_1-norm that introduces bias [110, 133]. Simulated tests demonstrate the effectiveness of the novel approaches in robustifying thin-plate smoothing splines [33], and in estimating the sinc function – a paradigm typically adopted to assess performance of robust function approximation approaches [27, 132]. The novel robust spline-based smoother is adopted to cleanse real *load curve* data, a key task aiding operational decisions in the envisioned smart grid system [1, 25].

1.2.3 Robust principal component analysis

Principal component analysis (PCA) is the workhorse of high-dimensional data analysis and dimensionality reduction, with numerous applications in statistics, engineering, and the biobehavioral sciences; see, e.g., [67]. Nowadays ubiquitous e-commerce sites, the Web, and urban traffic surveillance systems generate massive volumes of data. As a result, the problem of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [59]. To this end, PCA provides LS optimal linear approximants in $\mathbb{R}^q$ to a data set in $\mathbb{R}^p$, for $q \leq p$. The desired linear subspace is obtained

from the q dominant eigenvectors of the sample data covariance matrix [67].

Data obeying postulated low-rank models include also outliers, which are samples not adhering to those nominal models. Unfortunately, LS is known to be very sensitive to outliers [63, 104], and this undesirable property is inherited by PCA as well [67]. Early efforts to robustify PCA have relied on robust estimates of the data covariance matrix; see, e.g., [18]. Related approaches are driven from statistical physics [128], and also from M-estimators [31]. Recently, polynomial-time algorithms with remarkable performance guarantees have emerged for low-rank matrix recovery in the presence of sparse – but otherwise arbitrarily large – errors [20, 24]. This pertains to an 'idealized robust' PCA setup, since those entries not affected by outliers are assumed error free. Stability in reconstructing the low-rank and sparse matrix components in the presence of 'dense' noise have been reported in [127, 131]. A hierarchical Bayesian model was proposed to tackle the aforementioned low-rank plus sparse matrix decomposition problem in [32].

In Chapter 4, a robust PCA approach is pursued requiring minimal assumptions on the outlier model. A natural least-trimmed squares (LTS) PCA estimator is first shown closely related to an estimator obtained from an ℓ_0-(pseudo)norm-regularized criterion, adopted to fit a low-rank bilinear factor analysis model that explicitly incorporates an unknown *sparse* vector of outliers per datum. As in compressive sampling [115], efficient (approximate) solvers are obtained by surrogating the ℓ_0-norm of the outlier matrix with its closest convex approximant. This leads naturally to an M-type PCA estimator, which subsumes Huber's optimal choice as a special case [46]. Unlike Huber's formulation though, results here are not confined to an outlier contamination model. A tunable parameter controls the sparsity of the estimated matrix, and the number of outliers as a byproduct. Hence, effective data-driven methods to select this parameter are of paramount importance, and systematic approaches are pursued by efficiently exploring the entire *robustifaction* (a.k.a. homotopy) path of (group-) Lasso solutions [59, 130]. In this sense, the method here capitalizes on but *is not limited to* sparse settings where outliers are sporadic, since one can examine all sparsity levels along the robustification path. The outlier-aware generative data model and its sparsity-controlling estimator are quite general, since minor modifications

discussed in Chapter 4 enable robustifiying linear regression [49], dictionary learning [77, 114], and K-means clustering as well [43, 59]. Further modifications for bias reduction through nonconvex regularization, and automatic determination of the reduced dimension q, are also investigated.

Beyond its neat ties to robust statistics, the developed outlier-aware PCA framework is versatile to accommodate scalable *robust* algorithms to: i) track the low-rank signal subspace, as new data are acquired in real time; and ii) determine principal components in (possibly) infinite-dimensional feature spaces, thus robustifying kernel PCA [106], and spectral clustering as well [59, p. 544]. The vast literature on *non-robust* subspace tracking algorithms includes [77, 129], and [7]; see also [62] for a first-order algorithm that is robust to outliers and incomplete data. Relative to [62], the online robust (OR-) PCA algorithm of Chapter 4 is a second-order method, which minimizes an outlier-aware exponentially-weighted LS estimator of the low-rank factor analysis model. Since the outlier and subspace estimation tasks decouple nicely in OR-PCA, one can readily devise a first-order counterpart when minimal computational loads are at a premium. In terms of performance, online algorithms are known to be markedly faster than their batch alternatives [7, 62], e.g., in the timely context of low-rank matrix completion [95, 96]. While the focus here is not on incomplete data records, extensions to account for missing data are immediate and left as future work.

Numerical tests with synthetic and real data corroborate the effectiveness of the proposed robust PCA schemes, when used to identify aberrant responses from a questionnaire designed to measure the Big-Five dimensions of personality traits [65], as well as unveil communities in a (social) network of college football teams [50], and intruders from video surveillance data [31].

1.3 Published Results

The present Ph.D. work on sparsity-controlling outlier rejection algorithms has resulted in the pub- lication of 4 journals papers in the Institute of Electrical and Electronic Engineers (IEEE) Transactions on Signal Processing [81, 84, 85], and in the Institute for Operations

Research and the Management Sciences (INFORMS) Marketing Science [87]. The work has also been disseminated at pertinent conferences, where a total of 5 articles have been accepted for presentation [8, 49, 82, 83, 86].

1.4 Notational Conventions

The following notational conventions will be adopted throughout the subsequent chapters. Bold uppercase letters will denote matrices, whereas bold lowercase letters will stand for column vectors. Whenever the context makes it sufficiently clear, $[.]_{ij}$ will be used for a matrix to denote block matrix partitioning. Operators $\otimes$, $\odot$, $(.)'$, $(.)^{\dagger}$, $\lambda_{\max}(.)$, $\exp(.)$, $\mathrm{tr}(.)$, $E[.]$, $\mathrm{vec}[.]$, $\mathrm{med}(\cdot)$ will denote Kronecker product, Hadamard product, transposition, matrix pseudo-inverse, spectral radius, exponential function, matrix trace, expectation, matrix vectorization, and median, respectively. Vector $\mathrm{diag}(\mathbf{M})$ collects the diagonal entries of $\mathbf{M}$, whereas the diagonal matrix $\mathrm{diag}(\mathbf{v})$ has the entries of $\mathbf{v}$ on its diagonal. The ℓ_q norm of vector $\mathbf{x} \in \mathbb{R}^p$ is $\|\mathbf{x}\|_q := (\sum_{i=1}^{p} |x_i|^q)^{1/q}$ for $q \geq 1$; and $\|\mathbf{M}\|_F := \sqrt{\mathrm{tr}\,(\mathbf{M}\mathbf{M}')}$ is the matrix Frobenious norm. Positive-definite matrices will be denoted by $\mathbf{M} \succ \mathbf{0}$. The $p \times p$ identity matrix will be represented by $\mathbf{I}_p$, while $\mathbf{0}_p$ will denote the $p \times 1$ vector of all zeros, and $\mathbf{0}_{p\times q} := \mathbf{0}_p\mathbf{0}_q'$. Similar notation will be adopted for vectors (matrices) of all ones. The i-th vector of the canonical basis in $\mathbb{R}^n$ will be denoted by $\mathbf{b}_{n,i}$, $i = 1, \ldots, n$.

Chapter 2

Exploiting Sparsity in Model Residuals for Robust Conjoint Analysis

2.1 Introduction

Preference measurement (PM) has a long history in marketing, healthcare, and the biobehavioral sciences, where conjoint analysis (CA) is commonly used. In a nutshell, the goal of PM is to learn the utility function of an individual or group of individuals from expressed preference data (buying patterns, surveys, ratings, recommendations, etc). The pioneering idea behind CA is to decompose consumer preferences, into weights (partworths) of judiciously selected product attributes [55]. For metric conjoint data, an outlier-robust partworth estimator is developed in this chapter, on the basis of a neat connection between ℓ_0-(pseudo)norm-regularized regression, and the least-trimmed squared estimator [104]. This connection suggests efficient solvers based on convex relaxation, which lead naturally to a family of robust estimators subsuming Huber's optimal M-class. Outliers are identified by tuning a regularization parameter, which amounts to controlling the sparsity of an outlier vector along the entire robustification path of least-absolute shrinkage and selection

operator (Lasso) solutions.

For choice-based CA, a novel classifier is developed that is capable of attaining desirable tradeoffs between model fit and complexity, while at the same time controlling robustness and revealing the outliers present. Variants accounting for nonlinear utilities and consumer heterogeneity are also investigated.

2.2 Preliminaries and Robustness

Consider I respondents (e.g., consumers) indexed by $i \in \{1, \ldots, I\}$, each rating J_i profiles represented by the $p \times 1$ vectors $\mathbf{x}_{ij}$, $j \in \{1, \ldots, J_i\}$. Each $\mathbf{x}_{ij}$ comprises p attributes of the profile (or question) j presented to respondent i. As an example consider the CA of personal computers reported in [74]. With $p = 13$, the attributes considered are *'Telephone Service Hot Line'*, *'Amount of RAM'*, *'Screen Size'*, *'CPU Speed'*, *'Hard Disk Size'*, *'CD ROM/Multimedia'*, *'Cache'*, *'Color of Unit'*, *'Availability'*, *'Warranty'*, *'Bundled Productivity Software'*, *'Money Back Guarantee'*, and *'Price'*. All attributes are binary valued; for *'Amount of RAM'*, say, the corresponding entry in $\mathbf{x}_{ij}$ is coded as 1 to represent 16 Mb, and -1 to represent 8 Mb. Observe that one could in principle generate up to 2^{13} profiles to describe different candidate computers, but typically a few dozens are of real interest. In addition, fewer profiles naturally give rise to shorter questionnaires, which are attractive for both practical and theoretical reasons [74].

Parametric and linear utility functions $u(\mathbf{x})$ are typically adopted for modeling preference measurements [10, 108]. In these models responses $\{y_{ij}\}_{j=1}^{J_i}$ adhere to the linear regression $y_{ij} = \mathbf{x}_{ij}'\mathbf{w}_i + \varepsilon_{ij}$, $\mathbf{w}_i$ is the unknown $p \times 1$ vector of *partworths* for respondent i, and ε_{ij} captures random errors. Such a model describes the three most common types of conjoint data collection formats, namely:

(M1) *Full-profile* ratings, where one question per profile is presented to the respondent.

(M2) *Metric paired-comparison* ratings, where $\mathbf{x}_{ij}$ is replaced by the difference $\tilde{\mathbf{x}}_{ij} := \mathbf{x}_{ij}^{(1)} - \mathbf{x}_{ij}^{(2)}$ of a pair of profiles.

(M3) *Choice-based* conjoint data, where in addition to taking pairwise differences of profiles, the measurement is the sign of y_{ij} [113].

In words, question j under (M3) asks respondent i to choose between profiles $\mathbf{x}_{ij}^{(1)}$ and $\mathbf{x}_{ij}^{(2)}$; whereas under (M2), the surplus utility of the preferred profile over the other one is also quantified. For simplicity of exposition, focus will be placed first on individual partworth estimates; that is, each $\mathbf{w}_i$ will be estimated separately without fusing information from individual respondents. Subscript i can clearly be dropped in this case. Once the homogeneous case is addressed, approaches to account for consumer heterogeneities are possible along the lines of [37, 74, 113], as discussed in Section 2.4.3.

Given survey- or questionnaire-based training data $\mathcal{T} := \{y_j, \mathbf{x}_j\}_{j=1}^{J}$, modern statistical learning techniques have been developed to obtain $\mathbf{w}$. Under (M1) or (M2), the task amounts to parameter (or generally function) estimation, whereas under (M3) it boils down to a binary classification problem [29, 35, 37]. Following either deterministic or Bayesian formulations, these state-of-the-art techniques rely on suitably regularized loss functions to 'optimally' tradeoff complexity for error in the resultant model fit – an approach effecting the desirable generalization capability beyond $\mathcal{T}$ [113].

However, most existing partworth estimators have not accounted for outliers commonly present in large volumes of conjoint data. Outliers can be attributed to multiple factors, including: i) unintentional deviations from the adopted model of e.g., choice-based data; ii) behavioral effects of human respondents, e.g., response errors due to impatient or inattentive responders; and iii) intentional errors caused by malicious responders. Considering for simplicity (M1)[1], the starting point here is to develop a robust estimator of $\mathbf{w}$ that is universal with respect to the outlier model. One such approach is the least-trimmed squares (LTS) estimator given by [104]

$$\hat{\mathbf{w}}_{LTS} := \arg\min_{\mathbf{w}} \sum_{j=1}^{s} r_{[j]}^2(\mathbf{w}) \tag{2.1}$$

where $r_{[j]}^2(\mathbf{w})$ is the j-th order statistic among the squared residuals $r_1^2(\mathbf{w}), \ldots, r_J^2(\mathbf{w})$,

[1]Upon replacing $\mathbf{x}_{ij}$ with profile pair differences $\tilde{\mathbf{x}}_{ij} := \mathbf{x}_{ij}^{(1)} - \mathbf{x}_{ij}^{(2)}$, the estimators for model (M1) apply also to model (M2). A robust estimator for choice-based conjoint data (M3) is presented in Section 2.4.1.

and $r_j(\mathbf{w}) := y_j - \mathbf{x}_j'\mathbf{w}$. The so-termed *coverage* s determines the breakdown point of LTS [63, 104], since $J - s$ profile ratings resulting in the largest residuals are not present in (2.1). Beyond this universal outlier-rejection property, the LTS estimator is an attractive option due to its high breakdown point and desirable theoretical properties, namely $\sqrt{J}$-consistency and asymptotic normality under mild assumptions [104].

Even though (2.1) is nonconvex, existence of a minimizer $\hat{\mathbf{w}}_{LTS}$ can be established as follows: i) for each subset of $\{y_j, \mathbf{x}_j\}_{j=1}^J$ with cardinality s (there are $\binom{J}{s}$ such subsets), solve the corresponding ordinary least-squares (LS) problem to obtain a candidate estimator per subset; and ii) pick $\hat{\mathbf{w}}_{LTS}$ as the one among all $\binom{J}{s}$ candidates with the least cost. This solution procedure is combinatorially complex, and thus intractable except for small number of profiles J. Algorithms to obtain (approximate) LTS estimates are available [103].

2.3 Robust Linear Regression via Outlier Sparsity

Instead of discarding large residuals, the proposed approach is to model outliers explicitly and estimate them jointly with $\mathbf{w}$. To this end, consider introducing scalar auxiliary variables $\{o_j\}_{j=1}^J$ one per question (rated profile), which take values $o_j \neq 0$ whenever rating j is outlier contaminated, and $o_j = 0$ otherwise. This leads to the preference model

$$y_j = \mathbf{x}_j'\mathbf{w} + o_j + \varepsilon_j \tag{2.2}$$

where o_j can be deterministic or random with possibly unknown distribution. A similar model was advocated under different assumptions in [46] and [64]; see also [21] and [124]. In this *under-determined* linear regression model, both $\mathbf{w}$ as well as the $J \times 1$ vector $\mathbf{o} := [o_1, \ldots, o_J]'$ are unknown. The percentage of outliers dictates the degree of *sparsity* (number of zero entries) in $\mathbf{o}$. Sparsity control will prove instrumental in efficiently estimating $\mathbf{o}$, rejecting outliers as a byproduct, and consequently arriving at a robust estimate $\hat{\mathbf{w}}$. A natural criterion for controlling outlier sparsity is to seek the estimator

$$\{\hat{\mathbf{w}}, \hat{\mathbf{o}}\} = \arg\min_{\mathbf{w},\mathbf{o}} \sum_{j=1}^{J} (y_j - \mathbf{x}_j'\mathbf{w} - o_j)^2 + \lambda_0 \|\mathbf{o}\|_0 \tag{2.3}$$

where $\|\mathbf{o}\|_0$ denotes the nonconvex ℓ_0-norm (equal to the number of nonzero entries of $\mathbf{o}$). Tuning $\lambda_0 \geq 0$ controls sparsity in $\hat{\mathbf{o}}$.

As with compressive sampling and sparse modeling schemes that rely on the ℓ_0-norm, e.g., [115], (2.3) is also NP-hard [89]. In addition, the sparsity-controlling estimator (2.3) is intimately related to LTS, as asserted next (see Appendix 2.7.1 for a proof).

Proposition 2.1 *If* $\{\hat{\mathbf{w}}, \hat{\mathbf{o}}\}$ *solves (2.3) with* λ_0 *chosen such that* $\|\hat{\mathbf{o}}\|_0 = J-s$*, then* $\hat{\mathbf{w}}_{LTS} = \hat{\mathbf{w}}$ *in (2.1).*

Whenever (2.3) deems $J-s$ ratings as outliers, the obtained partworth estimate $\hat{\mathbf{w}}$ coincides with the LTS solution of (2.1). (Recall that LTS with trimming constant s, effectively discards the $J-s$ largest residuals.) The importance of Proposition 2.1 is threefold: i) it formally justifies the additive contamination model and its estimator for robust metric CA; ii) it provides a neat link between the seemingly unrelated fields of sparse linear regression and robust estimation; and iii) it lends itself naturally to efficient (approximate) LTS solvers based on convex relaxation.

Recalling that the ℓ_1-norm $\|\mathbf{o}\|_1$ is the closest convex approximation of $\|\mathbf{o}\|_0$ [115], motivates relaxing (2.3) to

$$\min_{\mathbf{w},\mathbf{o}} \sum_{j=1}^{J} (y_j - \mathbf{x}_j'\mathbf{w} - o_j)^2 + \lambda_1 \|\mathbf{o}\|_1. \tag{2.4}$$

This estimator is universally robust, and subsumes Huber's M-estimator for a specific choice of λ_1; details are given in Appendix 2.7.2. M-type estimators (including Huber's) adopt a fortiori an ϵ-contaminated probability distribution for the outliers, and rely on minimizing the *asymptotic* variance of the resultant estimator for the least favorable distribution of the ϵ-contaminated class (asymptotic min-max approach) [63]. The assumed degree of contamination specifies the tuning parameter λ_1 in Huber's robust loss function

$$\rho(r) := \begin{cases} r^2, & |r| \leq \lambda_1/2 \\ \lambda_1|r| - \lambda_1^2/4, & |r| > \lambda_1/2 \end{cases} \tag{2.5}$$

and thus the threshold for deciding the outliers in M-estimators. In contrast, the present approach is universal in the sense that it is not confined to any assumed class of outlier

distributions, and can afford a data-driven selection of the tuning parameter [cf. Section 2.3.1]. Before dwelling into algorithmic alternatives to solve (2.4), a remark is in order.

Remark 2.1 (Partworth regularization) In addition to $\mathbf{o}$ it is possible to also promote sparsity and/or smoothness of the partworth vector $\mathbf{w}$ by augmenting the cost in (2.4) with additional regularization terms entailing its ℓ_1-norm $\|\mathbf{w}\|_1$ and/or its ℓ_2-norm $\|\mathbf{w}\|_2^2$. The former promotes sparsifying the partworth vectors and retaining only the most critical attributes explaining the respondent's preferences. When the number of attributes p is large, parsimonious $u(\mathbf{x})$ can ease managerial decision-making. Ridge-type regularization allows to further control the (model) complexity of the solution, which is important when the responses J are few and p is considerably larger [37].

Albeit non-differentiable, (2.4) can be solved efficiently via e.g., alternating minimization (block-coordinate descent) iterations with guaranteed convergence to the global optimum. Iterations comprise a sequence of LS fits for $\mathbf{w}$, and coordinatewise soft-thresholded updates for $\mathbf{o}$; detailed iterations are tabulated under Algorithm 1. Alternatively, it is possible to show that the solutions $\{\hat{\mathbf{w}}, \hat{\mathbf{o}}\}$ of (2.4) are respectively given by $\hat{\mathbf{w}} := \mathbf{X}^\dagger(\mathbf{y} - \hat{\mathbf{o}}_{\text{Lasso}})$ and $\hat{\mathbf{o}} := \hat{\mathbf{o}}_{\text{Lasso}}$, where $\mathbf{y} := [y_1, \ldots, y_J]'$, $\mathbf{X}^\dagger := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ with $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_J]'$; and $\hat{\mathbf{o}}_{\text{Lasso}}$ is given by the Lasso esimator

$$\hat{\mathbf{o}}_{\text{Lasso}} := \arg\min_{\mathbf{o}} \|(\mathbf{I}_J - \mathbf{X}\mathbf{X}^\dagger)(\mathbf{y} - \mathbf{o})\|_2^2 + \lambda_1\|\mathbf{o}\|_1. \tag{2.6}$$

Selecting λ_1 along the *robustification* (a.k.a. homotpy) path of Lasso solutions controls the number of outliers rejected. But this choice is challenging because existing techniques such as cross-validation (CV) are not effective when outliers are present [104]. Interestingly, it is possible to devise a general and systematic approach to selecting λ_1, by leveraging recent convex optimization solvers that yield the entire path of Lasso solutions, i.e., $\hat{\mathbf{o}}_{\text{Lasso}}$ for all values of λ_1 in (2.6) [34, 44, 102]. Based on these robustification paths and prior knowledge possibly available on the model (2.2), one can effectively select λ_1 – the subject dealt with in the next section.

Algorithm 1 : Alternating-minimization solver

Initialize $\mathbf{o}(-1)=\mathbf{0}_J$.

Form $\mathbf{y}:=[y_1,\ldots,y_J]'$ and $\mathbf{X}:=[\mathbf{x}_1,\ldots,\mathbf{x}_J]'$

for $k=0,1,\ldots$ **do**

Update $\mathbf{w}(k)=(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'[\mathbf{y}-\mathbf{o}(k-1)]$.

Update the residuals $\mathbf{r}(k)=\mathbf{y}-\mathbf{X}\mathbf{w}(k)$.

Update $\mathbf{o}(k)$ via $o_j(k)=\text{sign}[r_j(k)]\max\left(|r_j(k)|-\lambda_1/2,0\right),\quad j=1,\ldots,J.$

end for

2.3.1 Selection of outlier sparsity

The ensuing methods for choosing λ_1 depend on prior information available about the outliers (number or statistics), and safely assume that the robustification path of (2.6) is efficiently obtained.

Number of outliers is known. By direct inspection of the robustification paths one can determine the range of values for λ_1, so that the degree of sparsity in $\hat{\mathbf{o}}$ equals the number of outliers. Specializing to the interval of interest, and after discarding the identified outliers, K-fold CV methods can be applied to determine the 'best' λ_1^*. Note that the number of outliers is also assumed known by RANSAC, in order to determine the number of random draws needed to attain a prescribed probability of success [42, 58].

Variance of the nominal noise is known. If the variance σ_ε^2 of the inlier noise ε_i in (2.2) is known, one can proceed as follows. Consider the estimates $\hat{\mathbf{w}}_g$ obtained using (2.4) or (2.6) after sampling the robustification path for each point $\{\lambda_g\}_{g=1}^G$ on a prescribed grid of size G. Based on $\{\hat{\mathbf{w}}_g\}_{g=1}^G$ and the data $\mathcal{T}$, find the sample variances $\{\hat{\sigma}_g^2\}_{g=1}^G$ after neglecting those training data $\{y_j,\mathbf{x}_j\}$ identified as outliers. The winner $\lambda_1^*:=\lambda_{g^*}$ corresponds to the grid point

$$g^*:=\arg\min_g|\hat{\sigma}_g^2-\sigma_\varepsilon^2|\,. \tag{2.7}$$

This is an absolute variance deviation (AVD) criterion for selecting λ_1^*. Knowledge of σ_ε^2 is also required by RANSAC; see also Sec. 2.5.

Variance of the nominal noise is unknown. If σ_ε^2 is unknown, one can still compute a robust estimate of the variance $\hat{\sigma}_\varepsilon^2$, and repeat the previous procedure after replacing σ_ε^2 with $\hat{\sigma}_\varepsilon^2$ in

(2.7). One simple option is based on the median absolute deviation (MAD) estimator, where $\hat{\sigma}_\varepsilon := 1.48 \times \text{med}_i \left(|\hat{r}_i - \text{med}_j \left(|\hat{r}_j| \right)| \right)$. The residuals $\hat{r}_i$ are formed based on a nonrobust estimate of $\mathbf{w}$, e.g., obtained via an LS fit using a small subset of the training data $\mathcal{T}$. The factor 1.48 provides an approximately unbiased estimate of σ_ε, when the nominal noise is Gaussian. In general, MAD requires knowledge of ε_j's symmetric pdf to determine the leading factor in $\hat{\sigma}_\varepsilon$ [104].

Contamination model. One may know a priori that the disturbances $\{o_j + \varepsilon_j\}$ in (2.2) adhere to Huber's contamination model [63]. Here ε_j can be thought of as nominal noise, and o_j as the contamination. If in this case λ_1 equals the threshold value in Huber's function, then $\hat{\mathbf{w}}$ enjoys asymptotic optimality in a well defined minimax sense [46].

Bayesian framework. Adopting a Bayesian perspective, one could model $\mathbf{w}$ as having i.i.d. entries obeying a non-informative (i.e., uniform) prior, independent of $\mathbf{o}$, which is assumed to have i.i.d. entries adhering to a common Laplacian distribution with parameter $2/\lambda_1^*$. Using $\lambda_1 = \lambda_1^*$ in (2.4), yields estimates $\hat{\mathbf{w}}$ (and $\hat{\mathbf{o}}$) which are optimal in the maximum a posteriori (MAP) sense; see also [64].

2.3.2 Estimator refinements

Nonconvex regularization. Instead of substituting $\|\mathbf{o}\|_0$ in (2.3) by its closest convex approximation, namely $\|\mathbf{o}\|_1$, letting the surrogate function to be nonconvex can yield tighter approximations. To this end, consider approximating (2.3) by the *nonconvex* formulation

$$\min_{\mathbf{w},\mathbf{o}} \sum_{j=1}^{J} (y_j - \mathbf{x}_j'\mathbf{w} - o_j)^2 + \lambda_0 \sum_{j=1}^{J} \log(|o_j| + \delta) \tag{2.8}$$

where $\delta \approx 0$ is introduced to avoid numerical instability.

Local methods based on iterative linearization of $\log(|o_j| + \delta)$ around the current iterate $o_j(k)$, can be adopted to minimize (2.8). Skipping details that can be found in [69], this procedure leads to the following iteration for $k = 0, 1, 2, \ldots$

$$\{\mathbf{w}(k), \mathbf{o}(k)\} = \arg\min_{\mathbf{w},\mathbf{o}} \sum_{j=1}^{J} \left[(y_j - \mathbf{x}_j'\mathbf{w} - o_j)^2 + \omega_j(k-1)|o_j| \right]$$

$$\omega_j(k) = \lambda_0 / \left(|o_j(k)| + \delta \right), \quad j = 1, \ldots, J$$

which altogether amounts to an iteratively reweighted version of (2.4). To avoid getting trapped in local minima, a good initialization for the iteration is the solution of (2.4). Numerical tests have shown that a couple iterations of this second-stage refinement suffices to yield improved partworth estimates $\hat{\mathbf{w}}$, in comparison to those obtained from (2.4). The improvements can be leveraged to bias reduction, also achieved by similar *weighted* norm regularizers [133].

Outlier rejection. From the equivalence between (2.4) and Huber's M-estimator, it follows that data $\{y_j, \mathbf{x}_j : j \text{ s.t. } \hat{o}_j \neq 0\}$ deemed as outliers are not completely discarded as with LTS. Instead, their effect is downweighted as per Huber's loss function (2.5); see also [63]. Nevertheless, explicitly accounting for the outliers in $\hat{\mathbf{o}}$ provides the means of identifying and removing the contaminated data altogether, and thus possibly re-estimating partworths using the 'clean' data.

2.4 Robust Conjoint Analysis Variants

2.4.1 Choice-based robust conjoint analysis

Over the last decade, choice-based CA has become a very popular alternative to metric analysis [60]. For the choice-based data model (M3) however, the approach to retrieve outliers and robustify the binary classifier for CA must be modified. Similar to [35] and for notational simplicity, assume without loss of generality that $\mathbf{x}_j^{(1)}$ is the preferred profile for all questions – otherwise profiles can be renamed accordingly. With this convention consumer responses become $y_j = 1$, $j = 1, \ldots, J$, and the proposed classifier is given by

$$\min_{\mathbf{w},\mathbf{o}} \sum_{j=1}^{J} \left[\left(1 - \tilde{\mathbf{x}}_j' \mathbf{w}\right)_+ - o_j \right]^2 + \lambda_o \|\mathbf{o}\|_1 + \lambda_w \|\mathbf{w}\|_2^2 \tag{2.9}$$

where $(\cdot)_+ := \max(\cdot, 0)$. To gain further intuition as to why (2.9) is a suitable robust estimator for stated-preference data, introduce *slack* variables $\xi_j \geq 0$ collected in the vector

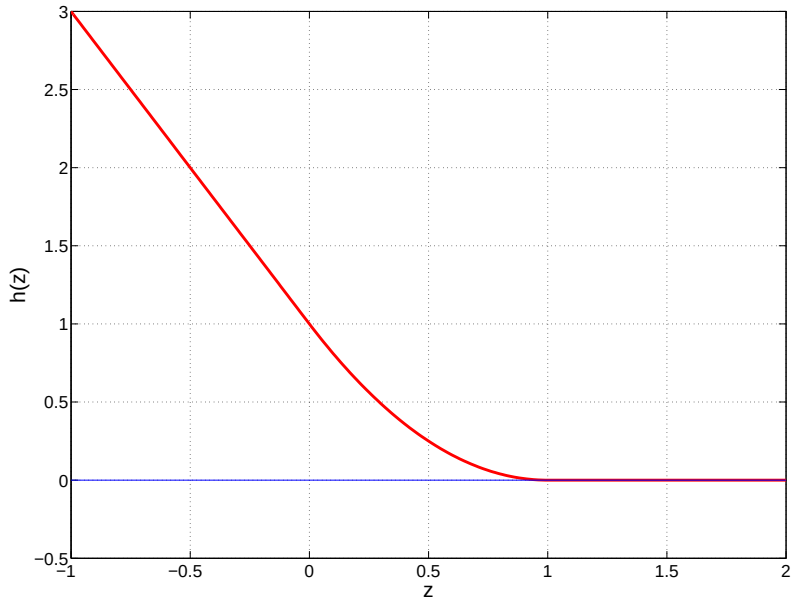


Figure 2.1: Huberized square hinge loss function for $\lambda_o = 2$.

$\boldsymbol{\xi} := [\xi_1, \ldots, \xi_J]'$, and note that (2.9) is equivalent to the linearly constrained formulation

$$\min_{\mathbf{w},\mathbf{o},\boldsymbol{\xi}} \sum_{j=1}^{J} (\xi_j - o_j)^2 + \lambda_o \|\mathbf{o}\|_1 + \lambda_w \|\mathbf{w}\|_2^2 \tag{2.10}$$

$$\text{s.t. } \tilde{\mathbf{x}}_j' \mathbf{w} \geq 1 - \xi_j, \quad \xi \geq 0, \quad j = 1, \ldots, J.$$

Because preference data can be contradictory (preferences change over time due to external factors, and unmodeled dynamics), it is often times impossible to find $\hat{\mathbf{w}}$ such that all inequalities $\tilde{\mathbf{x}}_j' \hat{\mathbf{w}} \geq 0$ are satisfied. It is thus prudent to allow for some 'slack', and try to minimize the inconsistencies ξ_j in the LS sense. This is exactly what (2.10) achieves in the absence of outliers. When outliers are present though, nonzero estimates $\hat{o}_j$ will ideally take values $\hat{o}_j \approx \hat{\xi}_j$, thus effectively removing the effect of the invalid responses in the estimation process. Note that 1 in the right-hand side of the first set of inequality constraints accounts for classifier margin; any other positive constant is equally good.

Problem (2.10) is a linearly-constrained quadratic program (QP), and is efficiently solved using general-purpose convex optimization software. In particular, it can be solved in the

primal domain (advisable when p is small but J is large), or, in the dual domain (preferable when p is large and J is small). A result with ramifications to the robustness properties and computational advantages of (2.9), is asserted in the following proposition.

Proposition 2.2 *The robust CA classifier (2.9) is equivalent to*

$$\min_{\mathbf{w}} \sum_{j=1}^{J} h\left(\tilde{\mathbf{x}}_j'\mathbf{w}\right) + \lambda_w \|\mathbf{w}\|_2^2 \tag{2.11}$$

where $h : \mathbb{R} \to \mathbb{R}$ is the 'Huberized' square hinge loss function [102]

$$h(z) := \begin{cases} \lambda_o(1-z) - \lambda_o^2/4, & z < 1 - \lambda_o/2, \\ (1-z)_+^2, & \textit{otherwise} \end{cases}. \tag{2.12}$$

Problem (2.11) is obtained after eliminating from (2.9), the optimized outlier variables $\hat{\mathbf{o}}(\mathbf{w})$. The derivation is based on similar arguments to those in Appendix 2.7.2. Examination of (2.12) (see also Fig. 2.1) reveals that (2.9) gives rise to three classification regions: r1) containing 'consistent' data for which $\tilde{\mathbf{x}}_j'\mathbf{w} \geq 1$; r2) comprising support vectors for which $1 - \lambda_o/2 \leq \tilde{\mathbf{x}}_j'\mathbf{w} \leq 1$; and r3) over which data satisfy $-\infty < \tilde{\mathbf{x}}_j'\mathbf{w} \leq 1 - \lambda_o/2$, and are deemed as contaminated with outliers. For $\lambda_0 = \infty$, $\hat{\mathbf{o}} = \mathbf{0}$ and h becomes the squared hinge loss function used in SVM variants.

When compared to the SVM used for CA [35,37,113], the key advantage of the classifier obtained via (2.9) is its ability to attain desirable tradeoffs between model fit and complexity, while at the same time controlling robustness and revealing the outliers present. Furthermore, convexity of the cost in (2.9) is not affected even when one chooses a different regularizer such as, e.g., $\lambda_w\|\mathbf{w}\|_1$ to encourage sparse partworth vectors and effect model complexity control. In fact, this could also be a wise choice from a computational standpoint, since the ℓ_1-norm regularized counterpart of (2.11) attains piecewise-linear solution paths as λ_w varies [102]. By capitalizing on this property, [102] shows that the entire path of solutions is efficiently obtained, using an algorithm that generalizes the LARS solver developed for Lasso [34]. An elastic net penalty was used in conjunction with a 'Huberized' square hinge loss in [122], for microarray selection and classification.

2.4.2 Nonparametric utility function estimation

The linear utility function $u(\mathbf{x}) = \mathbf{x}'\mathbf{w}$ considered so far falls short in capturing *interdependencies* among the attributes of each profile (entries of vector $\mathbf{x}_j$) – customers preferring cell-phones with mp3 players, will also value highly those models with memory capacity above 4Gb, say. As these interdependencies are driven by complex mechanisms that are typically hard to model a priori, it is prudent to let the data dictate the form of the $u(\mathbf{x})$ sought. This motivates the *nonparametric regression* methods for PM modeling briefly outlined in this section, and which are the main focus of Chapter 3.

To ensure versatility, u is only assumed to belong to a (possibly infinite dimensional) space of e.g., 'smooth' functions $\mathcal{H}$ [119]. As estimating $u \in \mathcal{H}$ from finite data is inherently ill-posed, one typically invokes properly regularized criteria [111]. Accordingly, u is robustly estimated from data adhering to (M1) by solving

$$\{\hat{u}, \hat{\mathbf{o}}\} := \arg \min_{u \in \mathcal{H}, \mathbf{o}} \sum_{j=1}^{J} (y_j - u(\mathbf{x}_j) - o_j)^2 + \mu R(u) + \lambda_o \|\mathbf{o}\|_1 \tag{2.13}$$

where $R : \mathcal{H} \to \mathbb{R}$ is a convex smoothing regularization functional, and $\mu \geq 0$ is chosen to tradeoff fidelity to the (outlier compensated) data for the degree of smoothness measured by $R(u)$. Problem (2.13) is variational in nature, and in principle requires searching over the infinite-dimensional space $\mathcal{H}$.

There is a neat workaround however, if one lets $R(u) := \|u\|^2_{\mathcal{H}}$ in (2.13), and endows $\mathcal{H}$ with the structure of a reproducing kernel Hilbert space [119]; with corresponding positive definite reproducing kernel function $K(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. The following proposition asserts that in this case, the unique solution of (2.13) is finitely parametrized, and it suffices to solve a single instance of Lasso to determine $\hat{u}$ along with the outliers $\hat{\mathbf{o}}$. Before stating the result, recall the conjoint data model (M1), the definition $\mathbf{y} := [y_1, \ldots, y_J]'$, and introduce the kernel matrix $\mathbf{K} \in \mathbb{R}^{J \times J}$ with ij-th entry $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$. The proof relies on the Representer Theorem; see e.g., [119], and can be found in Section 3.3.1.

Proposition 2.3 *Consider* $\hat{\mathbf{o}}_{Lasso}$ *defined as*

$$\hat{\mathbf{o}}_{Lasso} := \arg \min_{\mathbf{o}} \|\mathbf{X}_\mu \mathbf{y} - \mathbf{X}_\mu \mathbf{o}\|_2^2 + \lambda_o \|\mathbf{o}\|_1 \tag{2.14}$$

where

$$\mathbf{X}_{\mu} := \left[\begin{array}{c} \mathbf{I}_J - \mathbf{K}\left(\mathbf{K}+\mu\mathbf{I}_J\right)^{-1} \\ (\mu\mathbf{K})^{1/2}\left(\mathbf{K}+\mu\mathbf{I}_J\right)^{-1} \end{array} \right]. \tag{2.15}$$

Then the minimizers $\{\hat{u}, \hat{\mathbf{o}}\}$ *of (2.13) with* $R(u) := \|u\|^2_{\mathcal{H}}$ *are fully determined given* $\hat{\mathbf{o}}_{Lasso}$*, as* $\hat{\mathbf{o}} := \hat{\mathbf{o}}_{Lasso}$ *and* $\hat{u}(\mathbf{x}) = \sum_{j=1}^{J} \hat{\beta}_j K(\mathbf{x}, \mathbf{x}_j)$*, with* $\hat{\boldsymbol{\beta}} = \left(\mathbf{K}+\mu\mathbf{I}_J\right)^{-1}\left(\mathbf{y} - \hat{\mathbf{o}}_{Lasso}\right)$.

Joint outlier sparsity and function complexity control mechanisms identify the best (μ^*, λ_o^*) in (2.14), trading-off optimally the number of outliers rejected and the predictive capability of $\hat{u}$. These methods extend naturally those outlined in Section 2.3.1 [cf. the similarity between (2.14) and (2.6)], and require searching over a collection of robustifaction paths – one per μ value in a prescribed μ-grid. The end result yields estimates $\hat{u}$ with enhanced *ecological rationality*, yielding preference models better adapted to the shopping environment in which customers operate.

2.4.3 Distributed conjoint analysis

So far a single $\mathbf{w}$ was estimated, but multiple $\{\mathbf{w}_i\}$s are often needed to capture consumer heterogeneity, while improving estimation performance by fusing data from multiple respondents [37, 60, 113]. Traditional approaches have relied on hierarchical Bayes (HB) [4, 74], and share with convex optimization based ones [37] the idea of shrinking the individual estimates $\{\hat{\mathbf{w}}_i\}_{i=1}^{I}$ towards the population mean $\bar{\mathbf{w}}$. Specifically for (M1), [37] suggests

$$\min_{\{\mathbf{w}_i, \mathbf{D}, \bar{\mathbf{w}}\}} \sum_{i=1}^{I}\sum_{j=1}^{J}(y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i)^2 + \mu \sum_{i=1}^{I} \|\mathbf{w}_i - \bar{\mathbf{w}}\|^2_{\mathbf{D}} \tag{2.16}$$

which is jointly convex in $\{\mathbf{w}_i, \mathbf{D}, \bar{\mathbf{w}}\}$, while $\mathbf{D} \succ \mathbf{O}$ is normalized to have $\mathrm{tr}(\mathbf{D}) = 1$; and $\|\mathbf{v}\|^2_{\mathbf{M}} = \mathbf{v}'\mathbf{M}^{-1}\mathbf{v}$. Matrix $\mathbf{D}$ is related to the covariance matrix of the partworth estimators, so that pronounced shrinkage is effected to those $\mathbf{w}_i$'s far away from the mean $\bar{\mathbf{w}}$. MAP optimality is also apparent under a Gaussian nominal noise assumption, and identical Gaussian priors on the $\mathbf{w}_i$; see [37] for a detailed comparison between (2.16) and HB in [4]. Extension to choice-based data (M3) is possible by replacing the ℓ_2-error loss in (2.16) with e.g., the logistic error [37].

Algorithm 2 : DRCA

Agents $i \in \mathcal{I}$ initialize $\{\mathbf{w}_i(0), \bar{\mathbf{w}}_i(0), \mathbf{p}_i(0)), \mathbf{P}_i(0)\}$ to zero, $\{\mathbf{D}_i(0)\}$ to random unit-trace positive-definite matrices , and locally run

for $k = 0, 1, \ldots$ **do**

Exchange $\{\bar{\mathbf{w}}_i(k), \mathbf{D}_i(k)\}$ with neighbors in $\mathcal{N}_i$.

Update $\{\mathbf{w}_i(k+1), \bar{\mathbf{w}}_i(k+1)\}$ using (2.18).

Update $\mathbf{D}_i(k+1)$ using (2.19).

$o_{ij}(k+1) = \mathcal{S}\left(y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i(k+1), \lambda_o/2\right), \quad j = 1, \ldots, J.$

$\mathbf{p}_i(k+1) = \mathbf{p}_i(k) + c\sum_{i' \in \mathcal{N}_i}[\bar{\mathbf{w}}_i(k+1) - \bar{\mathbf{w}}_{i'}(k+1)].$

$\mathbf{P}_i(k+1) = \mathbf{P}_i(k) + c\sum_{i' \in \mathcal{N}_i}[\mathbf{D}_i(k+1) - \mathbf{D}_{i'}(k+1)].$

end for

All existing works assume that the data $\{y_{ij}, \mathbf{x}_{ij}\}_{i,j=1}^{I,J}$ are available centrally to determine the estimates $\{\hat{\mathbf{w}}_i, \hat{\mathbf{D}}, \hat{\bar{\mathbf{w}}}\}$. However, collecting all data in a central location may be prohibitive in certain studies, simply because respondents are not collocated, or due to finite storage, limited complexity, or even privacy constraints. In CA-based healthcare studies carried out by pharmaceutical companies, physicians provide private patient information for the purpose of estimating partworth vectors. They may not be willing to share training (questionnaire) data but only the learning results $\hat{\mathbf{w}}_i$. These reasons motivate well the *distributed* partworth estimator developed in this section, which is implementable through a cooperating network of processing units (agents) $\mathcal{I} := \{1, \ldots, I\}$, that exchange messages with directly connected neighbors. In the sequel, the network of agents will be modeled as a connected graph, and $\mathcal{N}_i \subseteq \mathcal{I}$ will denote the set of neighbors of agent i.

Towards distributing the centralized problem (2.16), introduce *local* auxiliary copies $\{\mathbf{D}_i, \bar{\mathbf{w}}_i\}_{i=1}^{I}$ of the *global* variables $\{\mathbf{D}, \bar{\mathbf{w}}\}$ per agent, along with constraints $\bar{\mathbf{w}}_i = \bar{\mathbf{w}}_{i'}$, $\mathbf{D}_i = \mathbf{D}_{i'}$, $i \in \mathcal{I}$, $i' \in \mathcal{N}_i$ to ensure consensus of these variables per neighborhood. Introducing the local quantities $\mathbf{y}_i := [y_{i1}, \ldots, y_{iJ}]'$, $\mathbf{X}_i := [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{iJ}]'$, and likewise for $\mathbf{o}_i$; the proposed

approach to *distributed* and *robust* (DR) CA solves

$$\min_{\substack{\{\mathbf{w}_i,\bar{\mathbf{w}}_i,\\ \mathbf{D}_i,\mathbf{o}_i\}}} \sum_{i=1}^{I} \left[\|\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i - \mathbf{o}_i\|_2^2 + \lambda_o\|\mathbf{o}_i\|_1 + \mu\|\mathbf{w}_i - \bar{\mathbf{w}}_i\|_{\mathbf{D}_i}^2\right]$$

$$\text{s. t.} \qquad \bar{\mathbf{w}}_i = \bar{\mathbf{w}}_{i'},\ \mathbf{D}_i = \mathbf{D}_{i'},\ i \in \mathcal{I},\ i' \in \mathcal{N}_i \tag{2.17}$$

with constraints $\mathrm{tr}(\mathbf{D}_i) = 1$, $i \in \mathcal{I}$, left implicit. Leaving aside robustness (cf. $\lambda_o = \infty$), problems (2.17) and (2.16) are equivalent since the network is connected. This property is instrumental because it ensures that the optimal local estimates coincide with the *global* minimizer of (2.16). Interestingly, the structure of (2.17) lends itself naturally to distributed implementation via the alternating-direction method of multipliers (AD-MoM), an iterative augmented Lagrangian method especially well-suited for parallel processing [12, 81]. AD-MoM iterations for $k = 0, 1, 2, \ldots$ entail: i) local optimization tasks to be run per agent; and ii) exchanges of local estimates $\{\bar{\mathbf{w}}_i(k), \mathbf{D}_i(k)\}$ only within $\mathcal{N}_i$, $i \in \mathcal{I}$. The latter are critical to percolate the spatially distributed data in $\mathcal{T}$ throughout the network, thus enabling agents to attain consensus on $\{\hat{\mathbf{w}}, \hat{\mathbf{D}}\}$ – the optimal solution of the centralized problem (2.16).

A detailed derivation of the DRCA algorithm (tabulated under Algorithm 2) can be found in Appendix 2.7.3; see also [81]. At the beginning of iteration $k+1$, agent i collects its neighbors most up to date estimates $\{\bar{\mathbf{w}}_{i'}(k), \mathbf{D}_{i'}(k)\}_{i' \in \mathcal{N}_i}$, and updates its own ones by solving the following strictly convex optimization problems

$$\{\mathbf{w}_i(k+1), \bar{\mathbf{w}}_i(k+1)\} = \arg\min_{\{\mathbf{w},\bar{\mathbf{w}}\}} \Bigg[\|\mathbf{y}_i - \mathbf{X}_i\mathbf{w} - \mathbf{o}_i(k)\|_2^2 + \mu\|\mathbf{w} - \bar{\mathbf{w}}\|_{\mathbf{D}_i(k)}^2 + \mathbf{p}_i'(k)\bar{\mathbf{w}} + c\sum_{i' \in \mathcal{N}_i} \left\|\bar{\mathbf{w}} - \frac{\bar{\mathbf{w}}_i(k) + \bar{\mathbf{w}}_{i'}(k)}{2}\right\|_2^2\Bigg] \tag{2.18}$$

$$\mathbf{D}_i(k+1) = \arg\min_{\mathbf{D}} \Bigg[\mu\|\mathbf{w}_i(k+1) - \bar{\mathbf{w}}_i(k+1)\|_{\mathbf{D}}^2 + \mathrm{tr}(\mathbf{P}_i(k)\mathbf{D}) + c\sum_{i' \in \mathcal{N}_i} \left\|\mathbf{D} - \frac{\mathbf{D}_i(k) + \mathbf{D}_{i'}(k)}{2}\right\|_F^2\Bigg]. \tag{2.19}$$

While (2.18) is an unconstrained QP with solution given in closed form, solving (2.19) requires an extra iterative procedure. Outliers are updated by parallel soft-thresholding of local residuals, where $\mathcal{S}(z,u) := \text{sign}(z)(|z|-u)_+$ in Algorithm 2. Iteration $k+1$ is concluded after obtaining dual prices $\mathbf{p}(k+1)$ and $\mathbf{P}(k+1)$ through dual ascent updates (see Algorithm 2), where $c>0$ is a stepsize which affects the convergence rate of the DRCA algorithm.

To close this section, it is useful to mention that convergence of the DRCA algorithm to the minimizer of (2.16) is ensured – for any $c>0$ – by virtue of AD-MoM's convergence theory [12, Prop. 4.2].

2.5 Numerical Tests

2.5.1 Robustifying linear regression

A numerical experiment is carried out here, to compare the performance of the sparsity-controlling estimator of this chapter against RANSAC, in a linear regression setting. For $J=100$ and $p=10$, nominal data adhere to the linear Gaussian model $y_j = \mathbf{x}_j'\mathbf{w}_0 + \varepsilon_j$, where the 'true' vector of partworths is generated as $\mathbf{w}_0 \sim \mathcal{N}(10\times\mathbf{1}_p, \mathbf{I}_p)$. The i.i.d. product profiles and nominal disturbances are $\mathbf{x}_j \sim \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$ and $\varepsilon_j \sim \mathcal{N}(0,1)$, respectively. Outliers are Laplacian distributed with zero-mean and standard deviation $\sqrt{2}\times 10^3$, i.e., $y_j \sim \mathcal{L}(0, 10^3)$ and i.i.d.. Contamination levels ranging from 0% to 80% are examined. The nominal noise variance $\sigma_\varepsilon^2 = 1$ is assumed known.

When solving (2.6), the optimum tuning parameter λ_1^* is obtained using the AVD criterion in (2.7). Ten samples ($G=10$) of the robustification path are employed, equispaced on a logarithmic λ_1 scale. To further enhance the estimation performance, a single iteration is carried out to minimize the concave sum-of-logs surrogate of (2.3). The refinement step is initialized with the solution to (2.4), for $\lambda_1 = \lambda_1^*$. The number of RANSAC iterations is fixed to either 1000 or 10000; and the threshold used to decide whether a data point is an outlier is set to $3\times\sigma_\varepsilon$. RANSAC is enhanced with a follow-up Huber M-estimation step using the RANSAC-generated inlier set. The Huber function parameter is set to $1.345\times\sigma_\varepsilon$

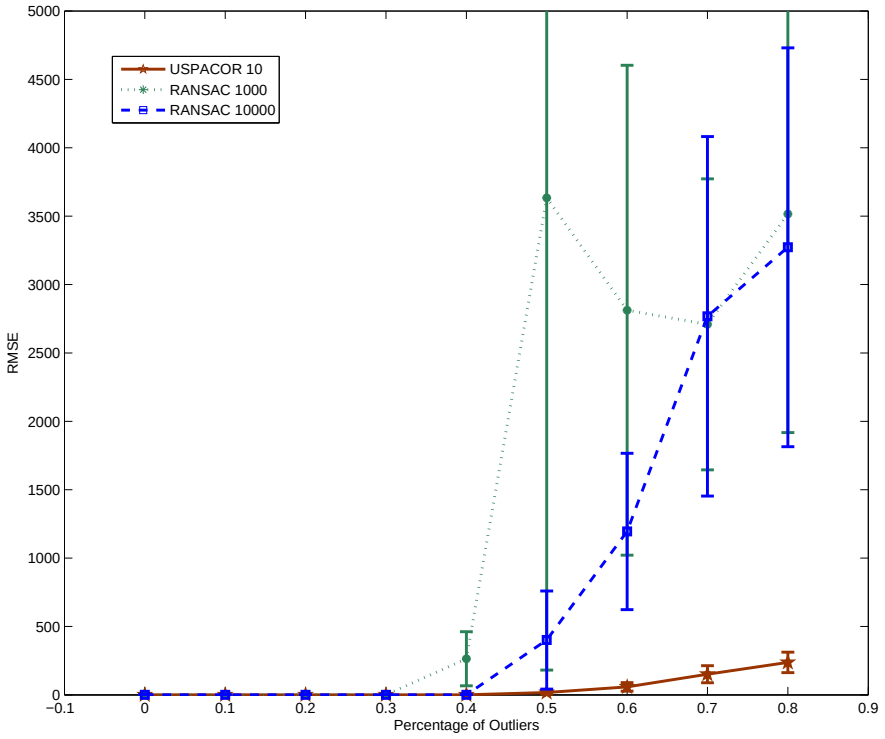


Figure 2.2: Sparsity-controlling outlier rejection vs. RANSAC: RMSE comparison.

as suggested in [46].

Fig. 2.2 compares RANSAC with the refined sparsity-controlling estimator (2.4) in terms of root mean square error (RMSE). The RMSE is defined as RMSE $:= E[\|\hat{\mathbf{w}} - \mathbf{w}_0\|_2]$, and approximated by sample averaging over 100 Monte Carlo runs. It is apparent that both methods generate very accurate results for small percentages of contamination. However, as the fraction of outliers increases, RANSAC breaks down resulting in large RMSEs with high variability. The proposed algorithm provides accurate results up to 40% contamination, and degrades gracefully beyond this level. In terms of complexity, (2.4) falls in between RANSAC 1000 and RANSAC 10000. These results corroborate that the novel methods of this chapter offer a competitive alternative for robust linear regression, and outperform state of the art RANSAC algorithms.

2.5.2 Choice-based conjoint analysis

The following simulated test case is used to corroborate the effectiveness of the proposed sparsity-controlling estimator for choice-based CA (cf. Section 2.4.1), and compare it with the SVM approach in [35].

The adopted simulation setup is standard for choice-based CA simulation studies under different (low-high) response-error levels, and (low-high) number of questions; see e.g. [35, 37]. Stated-preference questionnaires are simulated with $p = 10$ binary attributes per product profile, while the $\mathbf{x}_j^{(1)}$ were generated according to an orthogonal fractional-factorial design with $J = 16$. As per (M3), each of the questions comprises a pair of profiles to choose from, and given $\mathbf{x}_j^{(1)}$, the $\mathbf{x}_j^{(2)}$ were obtained through the shifting method of [16]. In the high number of questions setting, all $J = 16$ profiles pairs were presented to each respondent. For the reduced-size questionnaire condition, 8 profile pairs where randomly drawn from the complete set of 16. Each of the $I = 50$ respondents in a homogeneous population were given the same questionnaire, and 'true' partworths were drawn from a Gaussian distribution, i.e., $\mathbf{w}_i \sim \mathcal{N}(\mu \mathbf{1}_p, \sigma_{w_i}^2 \mathbf{I}_p)$, where $\mathbf{1}_p$ is the $p \times 1$ vector of all ones. The mean parameter μ takes the values 1.2 and 0.2, respectively in the low and high response error conditions. Since consumer heterogeneity is not considered here, values $\sigma_{w_i}^2 = \mu$ are adopted for $i = 1, \ldots, I$. Finally, logistic probabilities were used for the simulated nominal responses y_{ij}, i.e.,

$$\Pr(y_{ij} = 1) = \frac{\exp\left(\mathbf{w}_i' \mathbf{x}_j^{(1)}\right)}{\exp\left(\mathbf{w}_i' \mathbf{x}_j^{(1)}\right) + \exp\left(\mathbf{w}_i' \mathbf{x}_j^{(2)}\right)}, \quad \Pr(y_{ij} = -1) = 1 - \Pr(y_{ij} = 1)$$

whereas outliers were generated by simulating y_{i3}, $i = 1, \ldots, I$, as the outcome of an unbiased coin toss.

The results are summarized under Table 2.1, the figure of merit being the average partworth estimation error across respondents $\sum_{i=1}^{I} \|\hat{\mathbf{w}}_i - \mathbf{w}_i\|_2 / I$, after normalizing partworths to have unit ℓ_1 norm. Results for the method of [35] are shown under the column labeled SVM. Interestingly, the proposed sparsity-controlling estimator (2.9) consistently outperforms the SVM alternative of [35]. Regardless of the number of questions, the performance edge is more significant under the high response error condition. This is a manifestation of the robustness properties of the novel estimator, not only against outliers but also against

Table 2.1: Average partworth estimation errors.

Response error	Questions	SVM [35]	Proposed (2.9)
Low	8	0.3791	0.3730
Low	16	0.2472	0.2445
High	8	0.4023	0.3901
High	16	0.2922	0.2831

noisy (erroneous) responses. For all practical purposes, both schemes attain comparable estimation errors under the low response error regime.

2.6 Summary

Conjoint analysis has been a central problem in the marketing community, but recently researchers have started to explore its links with social computational systems under the general umbrella or preference measurement and modeling – an area of markedly growing interest given the explosion of preference data generated trough the Web. However, existing approaches have for the most part neglected that – online data collection cannot be controlled, and hence is prone to (un)intentional errors and inconsistencies, meaning outliers. These considerations motivate well the outlier-robust preference models developed in this chapter, for partworth estimation based on metric and choice-based conjoint data. Robust counterparts of estimators as fundamental as LS for linear regression, and the SVM for (binary) classification were developed under the proposed framework.

For the case of metric ratings, training samples from the (unknown) utility function were assumed generated from a linear regression model, which explicitly incorporates an unknown sparse vector of outliers. To fit such a model, the proposed regularized LS estimator minimizes a tradeoff between fidelity to the training data, and the sparsity level of the vector of outliers. The major innovative claim here is that *sparsity control* translates into *outlier-robustness control*. The LS partworth estimator was shown equivalent to Lasso,

and sparsity control can be carried out at affordable complexity by capitalizing on state-of-the-art algorithms, that return the whole path of Lasso solutions (i.e., for all values of the sparsity-controlling parameter). In this sense, the method here capitalizes on but is not limited to sparse settings where few outliers are present, since one can efficiently examine the gamut of sparsity levels along the robustification path. Computer simulations have shown that the novel sparsity-controlling algorithm of this chapter outperforms RANSAC, especially when the number of outliers is above 40% of the sample.

For choice-based CA, a novel classifier was developed that is capable of attaining desirable tradeoffs between model fit and complexity, while at the same time controlling robustness and revealing the outliers present. Numerical tests with synthetic stated-preference data show that the proposed sparsity-controlling partworth estimator consistently outperforms existing SVM alternatives. We also explored CA variants: i) entailing nonlinear utilities to capture interdependencies between the different product attributes (see also Chapter 3); and ii) accounting for consumer heterogeneity. In the context of ii), a distributed partworth estimator was developed, which is implementable through a cooperating network of processing units, that exchange messages with directly connected neighbors. Problem (2.16) is reformulated into an equivalent constrained form, whose structure lends itself naturally to distributed implementation via the AD-MoM. Interestingly, the distributed iterations are provably convergent to the centralized heterogeneity-aware estimator (2.16).

2.7 Appendices

2.7.1 Proof of Proposition 2.1

Given λ_0 such that $\|\hat{\mathbf{o}}\|_0 = J - s$, the goal is to characterize $\hat{\mathbf{w}}$ as well as the positions and values of the nonzero entries of the vector of outliers $\hat{\mathbf{o}}$. Upon defining the optimum residuals $\hat{r}_j := \mathbf{y}_j - \mathbf{x}_j'\hat{\mathbf{w}}$, the optimization with respect to $\mathbf{o}$ (given $\hat{\mathbf{w}}$) decouples into J scalar subproblems

$$\hat{o}_j := \arg\min_o (\hat{r}_j - o)^2 + \lambda_o \mathcal{I}_{\{o\neq 0\}}, \quad j = 1, \ldots, J \tag{2.20}$$

where $\mathcal{I}_{\{o\neq 0\}}$ denotes an indicator function taking the value 1 whenever $o \neq 0$, and 0 otherwise. It thus follows that the entries of $\hat{\mathbf{o}}$ are obtained by (hard-) thresholding residuals

$$\hat{o}_j = \begin{cases} 0, & |\hat{r}_j| \leq \sqrt{\lambda_0} \\ \hat{r}_j, & |\hat{r}_j| > \sqrt{\lambda_0} \end{cases}, \quad j = 1, \ldots, J. \tag{2.21}$$

This is intuitive, since for those nonzero $\hat{o}_j$ the best thing to do in terms of minimizing the overall cost is to set $\hat{o}_j = \hat{r}_j$, and thus null the corresponding squared-residual terms in (2.20). In conclusion, for the chosen value of λ_0 it holds that $J - s$ squared residuals effectively do not contribute to the cost in (2.3).

To determine $\hat{\mathbf{w}}$ and the support of $\hat{\mathbf{o}}$, one can in theory test all $\binom{J}{J-s} = \binom{J}{s}$ admissible support combinations. For each one of these combinations (indexed by i), let $\mathcal{S}_i \subset \{1, \ldots, J\}$ be the index set describing the support of $\hat{\mathbf{o}}^{(i)}$, i.e., $\hat{o}_j^{(i)} \neq 0$ if and only if $j \in \mathcal{S}_i$; and $|\mathcal{S}_i| = J - s$. By virtue of (2.21), the corresponding candidate $\hat{\mathbf{w}}^{(i)}$ solves $\min_{\mathbf{w}} \sum_{j\in\mathcal{S}_i} r_j^2(\mathbf{w})$, while $\hat{\mathbf{w}}$ is the one among all $\{\hat{\mathbf{w}}^{(i)}\}$ that yields the least cost. Recognizing the aforementioned solution procedure as the one described to solve LTS at the end of Section 2.2, it follows that $\hat{\mathbf{w}}_{LTS} = \hat{\mathbf{w}}$. ■

2.7.2 Equivalence between (2.4) and Huberized regression

It is established here that the outlier-aware estimator (2.4) is equivalent to

$$\min_{\mathbf{w}} \sum_{j=1}^{J} \rho(y_j - \mathbf{x}_j'\mathbf{w}) \tag{2.22}$$

where $\rho(r)$ is Huber's loss function defined in (2.5). As discussed in Section 2.3.1, if the disturbances $\{o_j + \varepsilon_j\}$ adhere to an ϵ-contaminated model, then there are optimal ways to select the tuning parameter λ_1 and (2.22) boils down to an M-estimator [46, 63].

Towards establishing the equivalence between both problem formulations, consider the pair $\{\hat{\mathbf{w}}, \hat{\mathbf{o}}\}$ that solves (2.4). Assume that $\hat{\mathbf{w}}$ is given, and the goal is to determine $\hat{\mathbf{o}}$. Upon defining the (optimum) residuals $\hat{r}_j := y_j - \mathbf{x}_j'\hat{\mathbf{w}}$ and because $\|\mathbf{o}\|_1 = \sum_{j=1}^{J} |o_j|$, the entries of $\hat{\mathbf{o}}$ are separately given by

$$\hat{o}_j = \arg\min_o \left[(\hat{r}_j - o)^2 + \lambda_1 |o|\right], \quad j = 1, \ldots, J. \tag{2.23}$$

For each $j = 1, \ldots, J$, because (2.23) is nondifferentiable at the origin one should consider three cases: i) if $\hat{o}_j = 0$, it follows that the minimum cost in (2.23) is $\hat{r}_j^2$; ii) if $\hat{o}_j > 0$, the first-order condition for optimality gives $\hat{o}_j = \hat{r}_j - \lambda_1/2$ provided $\hat{r}_j > \lambda_1/2$, and the minimum cost is $\lambda_1 \hat{r}_j - \lambda_1^2/4$; otherwise, iii) if $\hat{o}_j < 0$, it follows that $\hat{o}_j = \hat{r}_j + \lambda_1/2$ provided $\hat{r}_j < -\lambda_1/2$, and the minimum cost is $-\lambda_1 \hat{r}_j - \lambda_1^2/4$. In other words,

$$\hat{o}_j = \text{sign}(\hat{r}_j) \max(|\hat{r}_j| - \lambda_1/2, 0) = \begin{cases} \hat{r}_j - \lambda_1/2, & \hat{r}_j > \lambda_1/2 \\ 0, & |\hat{r}_j| \leq \lambda_1/2 \\ \hat{r}_j + \lambda_1/2, & \hat{r}_j < -\lambda_1/2 \end{cases}, \quad j = 1, \ldots, J. \tag{2.24}$$

Upon plugging (2.24) into (2.23), the minimum cost in (2.23) after minimizing with respect to o_j is $\rho(\hat{r}_j)$ [cf. (2.5) and the argument preceding (2.24)]. All in all, the conclusion is that $\hat{\mathbf{w}}$ is the minimizer of (2.22) – in addition to being the solution of (2.4) by definition.

2.7.3 Derivation of the DRCA algorithm

To tackle (2.17) using the alternating-direction method of multipliers (AD-MoM) [47, 51], consider adding to problem (2.17) the auxiliary local variables $\boldsymbol{\gamma} := \{\{\breve{\boldsymbol{\gamma}}_i^{i'}\}_{i' \in \mathcal{N}_i}, \{\bar{\boldsymbol{\gamma}}_i^{i'}\}_{i' \in \mathcal{N}_i}\}_{i \in \mathcal{I}}$ and $\boldsymbol{\Gamma} := \{\{\breve{\boldsymbol{\Gamma}}_i^{i'}\}_{i' \in \mathcal{N}_i}, \{\bar{\boldsymbol{\Gamma}}_i^{i'}\}_{i' \in \mathcal{N}_i}\}_{i \in \mathcal{I}}$, two pairs per neighbor. Introducing these new vari-

ables (2.17) is rewritten as

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \sum_{i=1}^{I} \left[\|\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i - \mathbf{o}_i\|_2^2 + \lambda_o\|\mathbf{o}_i\|_1 + \mu\|\mathbf{w}_i - \bar{\mathbf{w}}_i\|_{\mathbf{D}_i}^2\right] \tag{2.25}$$

$$\text{s. to} \quad \bar{\mathbf{w}}_i = \breve{\boldsymbol{\gamma}}_i^{i'},\ \bar{\mathbf{w}}_{i'} = \bar{\boldsymbol{\gamma}}_i^{i'},\ \breve{\boldsymbol{\gamma}}_i^{i'} = \bar{\boldsymbol{\gamma}}_i^{i'},\ i \in \mathcal{I},\ i' \in \mathcal{N}_i$$

$$\mathbf{D}_i = \breve{\boldsymbol{\Gamma}}_i^{i'},\ \mathbf{D}_{i'} = \bar{\boldsymbol{\Gamma}}_i^{i'},\ \breve{\boldsymbol{\Gamma}}_i^{i'} = \bar{\boldsymbol{\Gamma}}_i^{i'},\ i \in \mathcal{I},\ i' \in \mathcal{N}_i$$

where $\boldsymbol{\alpha} := \{\{\mathbf{w}_i\}_{i\in\mathcal{I}}, \{\bar{\mathbf{w}}_i\}_{i\in\mathcal{I}}, \boldsymbol{\Gamma}\}$, $\boldsymbol{\beta} := \{\{\mathbf{D}_i\}_{i\in\mathcal{I}}, \{\mathbf{o}_i\}_{i\in\mathcal{I}}, \boldsymbol{\gamma}\}$, and the constraints $\text{tr}(\mathbf{D}_i) = 1$, $i \in \mathcal{I}$ were left implicit. The equivalence of (2.16) and (2.25) is immediate because the latter only introduces the auxiliary variables in $\{\boldsymbol{\gamma}, \boldsymbol{\Gamma}\}$ to yield an alternative representation of the constraint set in (2.17).

Different from (2.16) however, (2.25) has a separable structure that facilitates distributed implementation. To capitalize on this favorable structure, associate Lagrange multipliers $\boldsymbol{v} := \{\{\breve{\mathbf{v}}_i^{i'}\}_{i'\in\mathcal{N}_i}, \{\bar{\mathbf{v}}_i^{i'}\}_{i'\in\mathcal{N}_i}\}_{i\in\mathcal{I}}$ and $\boldsymbol{V} := \{\{\breve{\mathbf{V}}_i^{i'}\}_{i'\in\mathcal{N}_i}, \{\bar{\mathbf{V}}_i^{i'}\}_{i'\in\mathcal{N}_i}\}_{i\in\mathcal{I}}$ with the constraints in (2.25), and form the quadratically augmented Lagrangian function

$$\begin{aligned}
\mathcal{L}_a\left[\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{V}\right] =& \sum_{i=1}^{I} \left[\|\mathbf{y}_i - \mathbf{X}_i\mathbf{w}_i - \mathbf{o}_i\|_2^2 + \lambda_o\|\mathbf{o}_i\|_1 + \mu\|\mathbf{w}_i - \bar{\mathbf{w}}_i\|_{\mathbf{D}_i}^2\right] \\
&+ \sum_{i=1}^{I}\sum_{i'\in\mathcal{N}_i} \left[(\breve{\mathbf{v}}_i^{i'})'(\bar{\mathbf{w}}_i - \breve{\boldsymbol{\gamma}}_i^{i'}) + (\bar{\mathbf{v}}_i^{i'})'(\bar{\mathbf{w}}_{i'} - \bar{\boldsymbol{\gamma}}_i^{i'})\right] \\
&+ \frac{c}{2}\sum_{i=1}^{I}\sum_{i'\in\mathcal{N}_i} \left[\|\bar{\mathbf{w}}_i - \breve{\boldsymbol{\gamma}}_i^{i'}\|_2^2 + \|\bar{\mathbf{w}}_{i'} - \bar{\boldsymbol{\gamma}}_i^{i'}\|_2^2\right] \\
&+ \sum_{i=1}^{I}\sum_{i'\in\mathcal{N}_i} \left[\text{tr}(\breve{\mathbf{V}}_i^{i'}(\mathbf{D}_i - \breve{\boldsymbol{\Gamma}}_i^{i'})) + \text{tr}(\bar{\mathbf{V}}_i^{i'}(\mathbf{D}_{i'} - \bar{\boldsymbol{\Gamma}}_i^{i'}))\right] \\
&+ \frac{c}{2}\sum_{i=1}^{I}\sum_{i'\in\mathcal{N}_i} \left[\|\mathbf{D}_i - \breve{\boldsymbol{\Gamma}}_i^{i'}\|_F^2 + \|\mathbf{D}_{i'} - \bar{\boldsymbol{\Gamma}}_i^{i'}\|_F^2\right].
\end{aligned} \tag{2.26}$$

where $c > 0$ is a preselected penalty coefficient. The constraints $\boldsymbol{\alpha} \in C_{\boldsymbol{\alpha}} := \{\boldsymbol{\Gamma} : \breve{\boldsymbol{\Gamma}}_i^{i'} = \bar{\boldsymbol{\Gamma}}_i^{i'},\ i \in \mathcal{I},\ i' \in \mathcal{N}_i\}$ and $\boldsymbol{\beta} \in C_{\boldsymbol{\beta}} := \{\boldsymbol{\gamma},\ \mathbf{D}_i : \breve{\boldsymbol{\Gamma}}_i^{i'} = \bar{\boldsymbol{\Gamma}}_i^{i'}, \text{tr}(\mathbf{D}_i) = 1,\ i \in \mathcal{I},\ i' \in \mathcal{N}_i\}$ have not been dualized. The AD-MoM entails three steps per iteration k of the algorithm:

[S1] Local estimate updates:

$$\boldsymbol{\alpha}(k+1) = \arg\min_{\boldsymbol{\alpha}\in C_{\boldsymbol{\alpha}}} \mathcal{L}_a\left[\boldsymbol{\alpha}, \boldsymbol{\beta}(k), \boldsymbol{v}(k), \boldsymbol{V}(k)\right]. \tag{2.27}$$

[S2] Local estimate updates:

$$\boldsymbol{\beta}(k+1) = \arg\min_{\boldsymbol{\beta}\in C_{\boldsymbol{\beta}}} \mathcal{L}_a\left[\boldsymbol{\alpha}(k+1), \boldsymbol{\beta}, \boldsymbol{v}(k), \boldsymbol{V}(k)\right] \tag{2.28}$$

[S3] Lagrange multiplier updates:

$$\breve{\mathbf{v}}_i^{i'}(k+1) = \breve{\mathbf{v}}_i^{i'}(k) + c[\bar{\mathbf{w}}_i(k+1) - \breve{\boldsymbol{\gamma}}_i^{i'}(k+1)] \tag{2.29}$$

$$\bar{\mathbf{v}}_i^{i'}(k+1) = \bar{\mathbf{v}}_i^{i'}(k) + c[\bar{\mathbf{w}}_{i'}(k+1) - \bar{\boldsymbol{\gamma}}_i^{i'}(k+1)] \tag{2.30}$$

$$\breve{\mathbf{V}}_i^{i'}(k+1) = \breve{\mathbf{V}}_i^{i'}(k) + c[\mathbf{D}_i(k+1) - \breve{\boldsymbol{\Gamma}}_i^{i'}(k+1)] \tag{2.31}$$

$$\bar{\mathbf{V}}_i^{i'}(k+1) = \bar{\mathbf{V}}_i^{i'}(k) + c[\mathbf{D}_{i'}(k+1) - \bar{\boldsymbol{\Gamma}}_i^{i'}(k+1)]. \tag{2.32}$$

The goal is to show that [S1]-[S3] can be simplified to the recursions tabulated under Algorithm 2. Focusing on [S2], in particular minimizing first with respect to $\boldsymbol{\gamma}$, from the decomposable structure of the augmented Lagrangian [cf. (2.26)] (2.28) decouples into $\sum_{i=1}^{I} |\mathcal{N}_i|$ quadratic sub-problems

$$\begin{aligned}\breve{\boldsymbol{\gamma}}_i^{i'}(k+1) = \bar{\boldsymbol{\gamma}}_i^{i'}(k+1) = \arg\min_{\breve{\boldsymbol{\gamma}}_i^{i'}} \Big\{ &-[\breve{\mathbf{v}}_i^{i'}(k) + \bar{\mathbf{v}}_i^{i'}(k)]'\breve{\boldsymbol{\gamma}}_i^{i'} \\ &+\frac{c}{2}\left[\|\bar{\mathbf{w}}_i(k+1) - \breve{\boldsymbol{\gamma}}_i^{i'}\|_2^2 + \|\bar{\mathbf{w}}_{i'}(k+1) - \breve{\boldsymbol{\gamma}}_i^{i'}\|_2^2\right]\Big\}\end{aligned} \tag{2.33}$$

which admit the closed-form solutions

$$\breve{\boldsymbol{\gamma}}_i^{i'}(k+1) = \bar{\boldsymbol{\gamma}}_i^{i'}(k+1) = \frac{1}{2c}[\breve{\mathbf{v}}_i^{i'}(k) + \bar{\mathbf{v}}_i^{i'}(k)] + \frac{1}{2}\left[\bar{\mathbf{w}}_i(k+1) + \bar{\mathbf{w}}_{i'}(k+1)\right]. \tag{2.34}$$

Note that in formulating (2.33), $\bar{\boldsymbol{\gamma}}_i^{i'}$ was eliminated using the constraint $\breve{\boldsymbol{\gamma}}_i^{i'} = \bar{\boldsymbol{\gamma}}_i^{i'}$. Using (2.34) to eliminate $\breve{\boldsymbol{\gamma}}_i^{i'}(k+1)$ and $\bar{\boldsymbol{\gamma}}_i^{i'}(k+1)$ from (2.29) and (2.30) respectively, a simple induction argument establishes that if the initial Lagrange multipliers obey $\breve{\mathbf{v}}_i^{i'}(0) = -\bar{\mathbf{v}}_i^{i'}(0) = \mathbf{0}$, then $\breve{\mathbf{v}}_i^{i'}(k) = -\bar{\mathbf{v}}_i^{i'}(k)$ for all $k \geq 0$ where $i \in \mathcal{I}$ and $i' \in \mathcal{N}_i$. The set $\{\bar{\mathbf{v}}_i^{i'}\}$ of multipliers has been shown redundant, and (2.34) readily simplifies to

$$\breve{\boldsymbol{\gamma}}_i^{i'}(k+1) = \bar{\boldsymbol{\gamma}}_i^{i'}(k+1) = \frac{1}{2}\left[\bar{\mathbf{w}}_i(k+1) + \bar{\mathbf{w}}_{i'}(k+1)\right], \quad i \in \mathcal{I}, \quad i' \in \mathcal{N}_i. \tag{2.35}$$

It then follows that $\breve{\boldsymbol{\gamma}}_i^{i'}(k) = \breve{\boldsymbol{\gamma}}_{i'}^{i}(k)$ for all $k \geq 0$, an identity that will be used later on. By plugging (2.35) in (2.29), the multiplier update becomes

$$\breve{\mathbf{v}}_i^{i'}(k+1) = \breve{\mathbf{v}}_i^{i'}(k) + \frac{c}{2}[\bar{\mathbf{w}}_i(k+1) - \bar{\mathbf{w}}_{i'}(k+1)], \quad i \in \mathcal{I},\ i' \in \mathcal{N}_i. \tag{2.36}$$

If $\breve{\mathbf{v}}_i^{i'}(0) = -\breve{\mathbf{v}}_{i'}^{i}(0) = \mathbf{0}$, then the structure of (2.36) reveals that $\breve{\mathbf{v}}_i^{i'}(k) = -\breve{\mathbf{v}}_{i'}^{i}(k)$ for all $k \geq 0$, where $i \in \mathcal{I}$ and $i' \in \mathcal{N}_i$.

Observe that when minimizing the augmented Lagrangian with respect to $\mathbf{\Gamma}$ in [S1], the role of $\{\breve{\mathbf{\Gamma}}_i^{i'}, \bar{\mathbf{\Gamma}}_i^{i'}, \breve{\mathbf{V}}_i^{i'}, \bar{\mathbf{V}}_i^{i'}\}$ is identical to the one of $\{\breve{\boldsymbol{\gamma}}_i^{i'}, \bar{\boldsymbol{\gamma}}_i^{i'}, \breve{\mathbf{v}}_i^{i'}, \bar{\mathbf{v}}_i^{i'}\}$ in the minimization just described. Thus, it follows immediately that

$$\breve{\mathbf{\Gamma}}_i^{i'}(k+1) = \bar{\mathbf{\Gamma}}_i^{i'}(k+1) = \frac{1}{2}\left[\mathbf{D}_i(k+1) + \mathbf{D}_{i'}(k+1)\right], \quad i \in \mathcal{I}, \;\; i' \in \mathcal{N}_i \tag{2.37}$$

$$\breve{\mathbf{V}}_i^{i'}(k+1) = \breve{\mathbf{V}}_i^{i'}(k) + \frac{c}{2}[\mathbf{D}_i(k+1) - \mathbf{D}_{i'}(k+1)], \quad i \in \mathcal{I}, \; i' \in \mathcal{N}_i. \tag{2.38}$$

while the multipliers $\bar{\mathbf{V}}_i^{i'}(k)$ are redundant since they can be expressed in terms of $\breve{\mathbf{V}}_i^{i'}(k)$. Moving on to the minimization with respect to $\mathbf{o}_i$ in [S2], observe that the augmented Lagrangian is separable with respect to the scalar entries of each $\mathbf{o}_i$, yielding the following $I \times J$ simpler subtasks

$$\begin{aligned} o_{ij}(k+1) &= \arg\min_o (y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i(k+1))^2 + \lambda_o|o| \\ &= \text{sign}(y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i(k+1)) \max(|y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i(k+1)| - \lambda_o/2, 0) \\ &= \mathcal{S}\left(y_{ij} - \mathbf{x}_{ij}'\mathbf{w}_i(k+1), \lambda_o/2\right), \quad j = 1, \ldots, J, \; i \in \mathcal{I}. \end{aligned} \tag{2.39}$$

The minimization with respect to $\{\mathbf{D}_i\}$ also decouples in I simpler sub-problems, namely

$$\begin{aligned} \mathbf{D}_i(k+1) = \arg\min_{\mathbf{D}} \Bigg\{ & \mu\|\mathbf{w}_i(k+1) - \bar{\mathbf{w}}_i(k+1)\|_{\mathbf{D}}^2 + \sum_{i' \in \mathcal{N}_i} \text{tr}((\breve{\mathbf{V}}_i^{i'}(k) - \bar{\mathbf{V}}_{i'}^{i}(k))\mathbf{D}) \\ & + \frac{c}{2} \sum_{i' \in \mathcal{N}_i} \left[\|\mathbf{D} - \breve{\mathbf{\Gamma}}_i^{i'}(k)\|_F^2 + \|\mathbf{D} - \bar{\mathbf{\Gamma}}_{i'}^{i}(k)\|_F^2\right] \Bigg\} \\ = \arg\min_{\mathbf{D}} \Bigg\{ & \mu\|\mathbf{w}_i(k+1) - \bar{\mathbf{w}}_i(k+1)\|_{\mathbf{D}}^2 + \text{tr}(\mathbf{P}_i(k)\mathbf{D}) \\ & + c \sum_{i' \in \mathcal{N}_i} \left\| \mathbf{D} - \frac{\mathbf{D}_i(k) + \mathbf{D}_{i'}(k)}{2} \right\|_F^2 \Bigg\} \end{aligned}$$

where in deriving the second equality we used that: i) $\breve{\mathbf{V}}_i^{i'}(k) = \bar{\mathbf{V}}_{i'}^{i}(k)$ which follows from the identities $\breve{\mathbf{V}}_i^{i'}(k) = -\bar{\mathbf{V}}_i^{i'}(k)$ and $\breve{\mathbf{V}}_i^{i'}(k) = -\breve{\mathbf{V}}_{i'}^{i}(k)$ established earlier; ii) the definition $\mathbf{P}_i(k) := 2\sum_{i' \in \mathcal{N}_i} \breve{\mathbf{V}}_i^{i'}(k)$; and iii) the identity $\breve{\mathbf{\Gamma}}_i^{i'}(k) = \bar{\mathbf{\Gamma}}_{i'}^{i}(k)$ which allows to merge the

identical quadratic penalty terms and eliminate both $\breve{\mathbf{\Gamma}}_i^{i'}(k)$ and $\bar{\mathbf{\Gamma}}_{i'}^{i}(k)$ using (2.37). This establishes (2.19), after recalling that the normalization constraint $\mathrm{tr}(\mathbf{D}_i) = 1$ has to be enforced for all $i \in \mathcal{I}$.

Finally, consider minimizing $\mathcal{L}_a\left[\boldsymbol{\alpha}, \boldsymbol{\beta}(k), \boldsymbol{v}(k), \boldsymbol{V}(k)\right]$ with respect to $\{\mathbf{w}_i, \bar{\mathbf{w}}_i\} \subset \boldsymbol{\alpha}$. The separable structure of the Lagrangian yields

$$\begin{aligned}
\{\mathbf{w}_i(k+1), \bar{\mathbf{w}}_i(k+1)\} = \arg \min_{\{\mathbf{w}, \bar{\mathbf{w}}\}} \Bigg\{ & \|\mathbf{y}_i - \mathbf{X}_i \mathbf{w}_i - \mathbf{o}_i(k)\|_2^2 + \mu \|\mathbf{w}_i - \bar{\mathbf{w}}_i\|_{\mathbf{D}_i(k)}^2 \\
& + \sum_{i' \in \mathcal{N}_i} (\breve{\mathbf{v}}_i^{i'}(k) - \bar{\mathbf{v}}_{i'}^{i}(k))' \bar{\mathbf{w}} \\
& + \frac{c}{2} \sum_{i' \in \mathcal{N}_i} \left[\|\bar{\mathbf{w}} - \breve{\boldsymbol{\gamma}}_i^{i'}(k)\|_F^2 + \|\bar{\mathbf{w}} - \bar{\boldsymbol{\gamma}}_{i'}^{i}(k)\|_2^2 \right] \Bigg\} \\
= \arg \min_{\{\mathbf{w}, \bar{\mathbf{w}}\}} \Bigg\{ & \|\mathbf{y}_i - \mathbf{X}_i \mathbf{w}_i - \mathbf{o}_i(k)\|_2^2 + \mu \|\mathbf{w}_i - \bar{\mathbf{w}}_i\|_{\mathbf{D}_i(k)}^2 + \mathbf{p}_i'(k) \bar{\mathbf{w}} \\
& + c \sum_{i' \in \mathcal{N}_i} \left\| \bar{\mathbf{w}} - \frac{\bar{\mathbf{w}}_i(k) + \bar{\mathbf{w}}_{i'}(k)}{2} \right\|_2^2 \Bigg\}
\end{aligned}$$

which is identical to (2.18), and in obtaining the second equality we have defined $\mathbf{p}_i(k) := 2 \sum_{i' \in \mathcal{N}_i} \breve{\mathbf{v}}_i^{i'}(k)$.

Chapter 3

Robust Nonparametric Regression via Sparsity Control

3.1 Introduction

Nonparametric methods are widely applicable to statistical inference problems, since they rely on a few modeling assumptions. In this context, the fresh look advocated in this chapter permeates benefits from variable selection and compressive sampling, to robustify nonparametric regression against outliers – that is, data markedly deviating from the postulated models. A variational counterpart to least-trimmed squares regression is proposed, and shown closely related to an ℓ_0-(pseudo)norm-regularized estimator, that encourages *sparsity* in a vector explicitly modeling the outliers. This connection suggests efficient solvers based on convex relaxation, which lead naturally to a variational M-type estimator equivalent to the least-absolute shrinkage and selection operator (Lasso). Outliers are identified by judiciously tuning regularization parameters, which amounts to controlling the sparsity of the outlier vector along the whole *robustification* path of Lasso solutions. Reduced bias and enhanced generalization capability are attractive features of an improved estimator obtained after replacing the ℓ_0-(pseudo)norm with a nonconvex surrogate.

The motivating application behind the robust nonparametric methods of this chapter is *load curve cleansing* [25] – a critical task in power systems engineering and management.

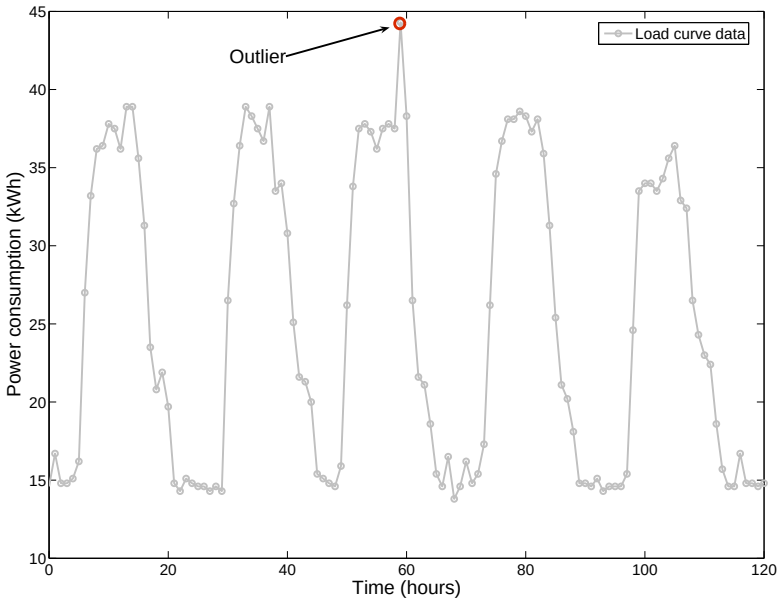


Figure 3.1: Example of load curve data with outliers.

Load curve data (also known as load profiles) refers to the electric energy consumption periodically recorded by meters at specific points across the power grid, e.g., end user-points and substations. Accurate load profiles are critical assets aiding operational decisions in the envisioned smart grid system [61]; see also [1, 2, 25]. However, in the process of acquiring and transmitting such massive volumes of information to a central processing unit, data is often noisy, corrupted, or lost altogether. This could be due to several reasons including meter misscalibration or outright failure, as well as communication errors due to noise, network congestion, and connectivity outages; see Fig. 3.1 for an example. In addition, data significantly deviating from nominal load models (outliers) are not uncommon, and could be attributed to unscheduled maintenance leading to shutdown of heavy industrial loads, weather constraints, holidays, strikes, and major sporting events, just to name a few.

In this context, it is critical to effectively reject outliers, and replace the contaminated data with 'healthy' load predictions, i.e., to cleanse the load data. While most utilities carry out this task manually based on their own personnel's know-how, a first scalable and prin-

cipled approach to load profile cleansing which is based on statistical learning methods was recently proposed in [25]; which also includes an extensive literature review on the related problem of outlier identification in time-series. After estimating the regression function f via either B-spline or Kernel smoothing, pointwise confidence intervals are constructed based on $\hat{f}$. A datum is deemed as an outlier whenever it falls outside its associated confidence interval. To control the degree of smoothing effected by the estimator, [25] requires the user to label the outliers present in a training subset of data, and in this sense the approach therein is not fully automatic. Here instead, a novel alternative to load curve cleansing is developed after specializing the robust estimators of Sections 3.3 and 3.4, to the case of cubic smoothing splines (Section 3.5.3). The smoothness-and outlier sparsity-controlling parameters are selected according to the guidelines in Section 3.3.2; hence, no input is required from the data analyst. The proposed spline-based method is tested on real load curve data from a government building.

3.2 Robust Estimation Problem

Consider the classical problem of function estimation, in which an input vector $\mathbf{x} := [x_1, \ldots, x_p]' \in \mathbb{R}^p$ is given, and the goal is to predict the real-valued scalar response $y = f(\mathbf{x})$. Function f is unknown, to be estimated from a training data set comprising N *noisy* samples of f taken at the input points $\{\mathbf{x}_i\}_{i=1}^N$ (also known as knots in the splines parlance), and in the present context they can be possibly contaminated with outliers. Building on the parametric least-trimmed squares (LTS) approach [104], the desired robust estimate $\hat{f}$ can be obtained as the solution of the following variational (V)LTS minimization problem

$$\min_{f \in \mathcal{H}} \left[\sum_{i=1}^{s} r_{[i]}^2(f) + \mu \|f\|_{\mathcal{H}}^2 \right] \tag{3.1}$$

where $r_{[i]}^2(f)$ is the i-th order statistic among the squared residuals $r_1^2(f), \ldots, r_N^2(f)$, and $r_i(f) := y_i - f(\mathbf{x}_i)$. In words, given a feasible $f \in \mathcal{H}$, to evaluate the sum of the cost in (3.1) one: i) computes all N squared residuals $\{r_i^2(f)\}_{i=1}^N$, ii) orders them to form the nondecreasing sequence $r_{[1]}^2(f) \leq \ldots \leq r_{[N]}^2(f)$; and iii) sums up the smallest s terms. As in the parametric LTS [104], the so-termed trimming constant s (also known as coverage)

determines the breakdown point of the VLTS estimator, since the largest $N - s$ residuals do not participate in (3.1). Ideally, one would like to make $N - s$ equal to the (typically unknown) number of outliers N_o in the training data. For most pragmatic scenaria where N_o is unknown, the LTS estimator is an attractive option due to its high breakdown point and desirable theoretical properties, namely $\sqrt{N}$-consistency and asymptotic normality [104].

The tuning parameter $\mu \geq 0$ in (3.1) controls the tradeoff between fidelity to the (trimmed) data, and the degree of 'smoothness' measured by $\|f\|^2_{\mathcal{H}}$. In particular, $\|f\|^2_{\mathcal{H}}$ can be interpreted as a generalized ridge regularization term penalizing more those functions with large coefficients in a basis expansion involving the eigenfunctions of the kernel K.

Given that the sum in (3.1) is a nonconvex functional, a nontrivial issue pertains to the existence of the proposed VLTS estimator, i.e., whether or not (3.1) attains a minimum in $\mathcal{H}$. Fortunately, a (conceptually) simple solution procedure suffices to show that a minimizer does indeed exist. Consider specifically a given subsample of s training data points, say $\{y_i, \mathbf{x}_i\}_{i=1}^s$, and solve

$$\min_{f \in \mathcal{H}} \left[\sum_{i=1}^{s} r_i^2(f) + \mu \|f\|^2_{\mathcal{H}} \right].$$

A unique minimizer of the form $\hat{f}^{(j)}(\mathbf{x}) = \sum_{i=1}^s \beta_i^{(j)} K(\mathbf{x}, \mathbf{x}_i)$ is guaranteed to exist, where j is used here to denote the chosen subsample, and the coefficients $\{\beta_i^{(j)}\}_{i=1}^s$ can be obtained by solving a particular linear system of equations [119, p. 11]. This procedure can be repeated for each subsample (there are $J := \binom{N}{s}$ of these), to obtain a collection $\{\hat{f}^{(j)}(\mathbf{x})\}_{j=1}^J$ of candidate solutions of (3.1). The winner(s) $\hat{f} := \hat{f}^{(j^*)}$ yielding the minimum cost, is the desired VLTS estimator.

Even though conceptually simple, the solution procedure just described guarantees existence of (at least) one solution, but entails a combinatorial search over all J subsamples which is intractable for moderate to large sample sizes N. In the context of linear regression, algorithms to obtain approximate LTS solutions are available; see e.g., [103].

3.2.1 Robust function approximation via ℓ_0-norm regularization

Instead of discarding large residuals, the alternative approach proposed here explicitly accounts for outliers in the regression model. To this end, consider the scalar variables $\{o_i\}_{i=1}^N$ one per training datum, taking the value $o_i = 0$ whenever datum i adheres to the postulated nominal model, and $o_i \neq 0$ otherwise. A regression model naturally accounting for the presence of outliers is

$$y_i = f(\mathbf{x}_i) + o_i + \varepsilon_i, \qquad i = 1, \ldots, N \tag{3.2}$$

where $\{\varepsilon_i\}_{i=1}^N$ are zero-mean independent and identically distributed (i.i.d.) random variables modeling the observation errors. A similar model was advocated under different assumptions in [46] and [64], in the context of robust parametric regression; see also [21] and [124]. For an outlier-free datum i, (3.2) reduces to $y_i = f(\mathbf{x}_i) + \varepsilon_i$; hence, ε_i will be often referred to as the nominal noise. Note that in (3.2), both $f \in \mathcal{H}$ as well as the $N \times 1$ vector $\mathbf{o} := [o_1, \ldots, o_N]'$ are unknown; thus, (3.2) is underdetermined. On the other hand, as outliers are expected to often comprise a small fraction of the training sample say, not exceeding 20% – vector $\mathbf{o}$ is typically *sparse*, i.e., most of its entries are zero; see also Remark 3.2. Sparsity compensates for underdeterminacy and provides valuable side-information when it comes to efficiently estimating $\mathbf{o}$, identifying outliers as a byproduct, and consequently performing *robust* estimation of the unknown function f.

A natural criterion for controlling outlier sparsity is to seek the desired estimate $\hat{f}$ as the solution of

$$\min_{\substack{f \in \mathcal{H} \\ \mathbf{o} \in \mathbb{R}^N}} \left[\sum_{i=1}^{N} (y_i - f(\mathbf{x}_i) - o_i)^2 + \mu \|f\|_{\mathcal{H}}^2 + \lambda_0 \|\mathbf{o}\|_0 \right] \tag{3.3}$$

where $\lambda_0 \geq 0$ is a preselected sparsity controlling parameter, and $\|\mathbf{o}\|_0$ denotes the ℓ_0-norm of $\mathbf{o}$, which equals the number of nonzero entries of its vector argument. Unfortunately, analogously to related ℓ_0-norm regularized formulations in compressive sampling and sparse signal representations, problem (3.3) is NP-hard [89].

To further motivate model (3.2) and the proposed criterion (3.3) for robust nonparametric regression, it is worth checking the structure of the minimizers $\{\hat{f}, \hat{\mathbf{o}}\}$ of the cost in (3.3). Consider for the sake of argument that λ_0 is given, and its value is such that $\|\hat{\mathbf{o}}\|_0 = \nu$, for

some $0 \leq \nu \leq N$. The goal is to characterize $\hat{f}$, as well as the positions and values of the nonzero entries of $\hat{\mathbf{o}}$. Note that because $\|\hat{\mathbf{o}}\|_0 = \nu$, the last term in (3.3) is constant, hence inconsequential to the minimization. Upon defining $\hat{r}_i := y_i - \hat{f}(\mathbf{x}_i)$, it is not hard to see that the entries of $\hat{\mathbf{o}}$ satisfy

$$\hat{o}_i = \begin{cases} 0, & |\hat{r}_i| \leq \sqrt{\lambda_0} \\ \hat{r}_i, & |\hat{r}_i| > \sqrt{\lambda_0} \end{cases}, \quad i = 1, \ldots, N \tag{3.4}$$

at the optimum. This is intuitive, since for those $\hat{o}_i \neq 0$ the best thing to do in terms of minimizing the overall cost is to set $\hat{o}_i = \hat{r}_i$, and thus null the corresponding squared-residual terms in (3.3). In conclusion, for the chosen value of λ_0 it holds that ν squared residuals effectively do not contribute to the cost in (3.3).

To determine the support of $\hat{\mathbf{o}}$ and $\hat{f}$, one alternative is to exhaustively test all $\binom{N}{\nu}$ admissible support combinations. For each one of these combinations (indexed by j), let $\mathcal{S}_j \subset \{1, \ldots, N\}$ be the index set describing the support of $\hat{\mathbf{o}}^{(j)}$, i.e., $\hat{o}_i^{(j)} \neq 0$ if and only if $i \in \mathcal{S}_j$; and $|\mathcal{S}_j| = \nu$. By virtue of (3.4), the corresponding candidate $\hat{f}^{(j)}$ minimizes

$$\min_{f \in \mathcal{H}} \left[\sum_{i \in \mathcal{S}_j} r_i^2(f) + \mu \|f\|_{\mathcal{H}}^2 \right]$$

while $\hat{f}$ is the one among all $\{\hat{f}^{(j)}\}$ that yields the least cost. The previous discussion, in conjunction with the one preceding Section 3.2.1 completes the argument required to establish the following result.

Proposition 3.1 *If $\{\hat{f}, \hat{\mathbf{o}}\}$ minimizes (3.3) with λ_0 chosen such that $\|\hat{\mathbf{o}}\|_0 = N - s$, then $\hat{f}$ also solves the VLTS problem (3.1).*

The importance of Proposition 3.1 is threefold. First, it formally justifies model (3.2) and its estimator (3.3) for robust function approximation, in light of the well documented merits of LTS regression [103]. Second, it further solidifies the connection between sparse linear regression and robust estimation. Third, the ℓ_0-norm regularized formulation in (3.3) lends itself naturally to efficient solvers based on convex relaxation, the subject dealt with next.

3.3 Sparsity Controlling Outlier Rejection

To overcome the complexity hurdle in solving the robust regression problem in (3.3), one can resort to a suitable relaxation of the objective function. The goal is to formulate an optimization problem which is tractable, and whose solution yields a satisfactory approximation to the minimizer of the original hard problem. To this end, it is useful to recall that the ℓ_1-norm $\|\mathbf{x}\|_1$ of vector $\mathbf{x}$ is the closest convex approximation of $\|\mathbf{x}\|_0$. This property also utilized in the context of compressive sampling [115], provides the motivation to relax the NP-hard problem (3.3) to

$$\min_{\substack{f\in\mathcal{H}\\ \mathbf{o}\in\mathbb{R}^N}} \left[\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i) - o_i)^2 + \mu\|f\|_{\mathcal{H}}^2 + \lambda_1\|\mathbf{o}\|_1\right]. \tag{3.5}$$

Being a convex optimization problem, (3.5) can be solved efficiently. The nondifferentiable ℓ_1-norm regularization term controls sparsity on the estimator of $\mathbf{o}$, a property that has been recently exploited in diverse problems in engineering, statistics and machine learning. A noteworthy representative is the least-absolute shrinkage and selection operator (Lasso) [110], a popular tool in statistics for joint estimation and continuous variable selection in linear regression problems. In its Lagrangian form, Lasso is also known as basis pursuit denoising in the signal processing literature, a term coined by [26] in the context of finding the best sparse signal expansion using an overcomplete basis.

It is pertinent to ponder on whether problem (3.5) has built-in ability to provide robust estimates $\hat{f}$ in the presence of outliers. The answer is in the affirmative, since a straightforward argument (details are deferred to the Appendix) shows that (3.5) is equivalent to a variational M-type estimator found by

$$\min_{f\in\mathcal{H}} \left[\sum_{i=1}^{N}\rho(y_i - f(\mathbf{x}_i)) + \mu\|f\|_{\mathcal{H}}^2\right] \tag{3.6}$$

where $\rho:\mathbb{R}\to\mathbb{R}$ is a scaled version of Huber's convex loss function [63]

$$\rho(u) := \begin{cases} u^2, & |u|\le\lambda_1/2 \\ \lambda_1|u| - \lambda_1^2/4, & |u|>\lambda_1/2 \end{cases}. \tag{3.7}$$

Remark 3.1 (Regularized regression and robustness) Existing works on linear regression have pointed out the equivalence between ℓ_1-norm regularized regression and M-type estimators, under specific assumptions on the distribution of the outliers (ϵ-contamination) [46, 69]. However, they have not recognized the link with LTS through the convex relaxation of (3.3), and the connection asserted by Proposition 3.1. Here, the treatment goes beyond linear regression by considering nonparametric functional approximation in RKHS. Linear regression is subsumed as a special case, when the linear kernel $K(\mathbf{x}, \mathbf{y}) := \mathbf{x}'\mathbf{y}$ is adopted. In addition, no assumption is imposed on the outlier vector.

It is interesting to compare the ℓ_0- and ℓ_1-norm formulations [cf. (3.3) and (3.5), respectively] in terms of their equivalent purely variational counterparts in (3.1) and (3.6), that entail robust loss functions. While the VLTS estimator completely discards large residuals, ρ still retains them, but downweighs their effect through a linear penalty. Moreover, while (3.6) is convex, (3.1) is not and this has a direct impact on the complexity to obtain either estimator. Regarding the trimming constant s in (3.1), it controls the number of residuals retained and hence the breakdown point of VLTS. Considering instead the threshold $\lambda_1/2$ in Huber's function ρ, when the outliers' distribution is known a-priori, its value is available in closed form so that the robust estimator is optimal in a well-defined sense [63]. Convergence in probability of M-type cubic smoothing splines estimators – a special problem subsumed by (3.6) – was studied in [28].

3.3.1 Solving the convex relaxation

Because (3.5) is jointly convex in f and $\mathbf{o}$, an alternating minimization (AM) algorithm can be adopted to solve (3.5), for fixed values of μ and λ_1. Selection of these parameters is a critical issue that will be discussed in Section 3.3.2. AM solvers are iterative procedures that fix one of the variables to its most up to date value, and minimize the resulting cost with respect to the other one. Then the roles are reversed to complete one cycle, and the overall two-step minimization procedure is repeated for a prescribed number of iterations, or, until a convergence criterion is met. Letting $k = 0, 1, \ldots$ denote iterations, consider that

$\mathbf{o} := \mathbf{o}^{(k-1)}$ is fixed in (3.5). The update for $f^{(k)}$ at the k-th iteration is given by

$$f^{(k)} := \arg\min_{f\in\mathcal{H}} \left[\sum_{i=1}^{N} \left((y_i - o_i^{(k-1)}) - f(\mathbf{x}_i)\right)^2 + \mu\|f\|_{\mathcal{H}}^2\right] \tag{3.8}$$

which corresponds to a standard regularization problem for functional approximation in $\mathcal{H}$ [36], but with *outlier-compensated* data $\left\{y_i - o_i^{(k-1)}, \mathbf{x}_i\right\}_{i=1}^{N}$. It is well known that the minimizer of the variational problem (3.8) is finitely parameterized, and given by the kernel expansion $f^{(k)}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i^{(k)} K(\mathbf{x}, \mathbf{x}_i)$ [119]. The vector $\boldsymbol{\beta} := [\beta_1, \ldots, \beta_N]'$ is found by solving the linear system of equations

$$[\mathbf{K} + \mu\mathbf{I}_N]\,\boldsymbol{\beta}^{(k)} = \mathbf{y} - \mathbf{o}^{(k-1)} \tag{3.9}$$

where $\mathbf{y} := [y_1, \ldots, y_N]'$, and the $N \times N$ matrix $\mathbf{K} \succ \mathbf{0}$ has entries $[\mathbf{K}]_{ij} := K(\mathbf{x}_i, \mathbf{x}_j)$.

In a nutshell, updating $f^{(k)}$ is equivalent to updating vector $\boldsymbol{\beta}^{(k)}$ as per (3.9), where only the independent vector variable $\mathbf{y} - \mathbf{o}^{(k-1)}$ changes across iterations. Because the system matrix is positive definite, the per iteration systems of linear equations (3.9) can be efficiently solved after computing once, the Cholesky factorization of $\mathbf{K} + \mu\mathbf{I}_N$.

For fixed $f := f^{(k)}$ in (3.5), the outlier vector update $\mathbf{o}^{(k)}$ at iteration k is obtained as

$$\mathbf{o}^{(k)} := \arg\min_{\mathbf{o}\in\mathbb{R}^N} \left[\sum_{i=1}^{N} \left(r_i^{(k)} - o_i\right)^2 + \lambda_1\|\mathbf{o}\|_1\right] \tag{3.10}$$

where $r_i^{(k)} := y_i - \sum_{j=1}^{N} \beta_j^{(k)} K(\mathbf{x}_i, \mathbf{x}_j)$. Problem (3.10) can be recognized as an instance of Lasso for the so-termed orthonormal case, in particular for an identity regression matrix. The solution of such Lasso problems is readily obtained via soft-thresholding [44], in the form of

$$o_i^{(k)} := \mathcal{S}\left(r_i^{(k)}, \lambda_1/2\right), \quad i = 1, \ldots, N \tag{3.11}$$

where $\mathcal{S}(z, \gamma) := \text{sign}(z)(|z| - \gamma)_+$ is the soft-thresholding operator, and $(\cdot)_+ := \max(0, \cdot)$ denotes the projection onto the nonnegative reals. The coordinatewise updates in (3.11) are in par with the sparsifying property of the ℓ_1 norm, since for 'small' residuals, i.e., $r_i^{(k)} \leq \lambda_1/2$, it follows that $o_i^{(k)} = 0$, and the i-th training datum is deemed outlier free. Updates (3.9) and (3.11) comprise the iterative AM solver of the ℓ_1-norm regularized problem (3.5),

Algorithm 3 : AM solver

Initialize $\mathbf{o}^{(-1)} = \mathbf{0}$, and run till convergence

for $k = 0, 1, \ldots$ **do**

Update $\boldsymbol{\beta}^{(k)}$ solving $[\mathbf{K} + \mu \mathbf{I}_N]\, \boldsymbol{\beta}^{(k)} = \mathbf{y} - \mathbf{o}^{(k-1)}$.

Update $\mathbf{o}^{(k)}$ via $o_i^{(k)} = \mathcal{S}\left(y_i - \sum_{j=1}^{N} \beta_j^{(k)} K(\mathbf{x}_i, \mathbf{x}_j), \lambda_1/2\right), \quad i = 1, \ldots, N.$

end for

return $f(\mathbf{x}) = \sum_{i=1}^{N} \beta_i^{(\infty)} K(\mathbf{x}, \mathbf{x}_i)$

which is tabulated as Algorithm 4. Convexity ensures convergence to the global optimum solution regardless of the initial condition; see e.g., [11].

Algorithm 4 is also conceptually interesting, since it explicitly reveals the intertwining between the outlier identification process, and the estimation of the regression function with the appropriate outlier-compensated data. An additional point is worth mentioning after inspection of (3.11) in the limit as $k \to \infty$. From the definition of the soft-thresholding operator $\mathcal{S}$, for those 'large' residuals $\hat{r}_i := \lim_{k\to\infty} r_i^{(k)}$ exceeding $\lambda_1/2$ in magnitude, $\hat{o}_i = \hat{r}_i - \lambda_1/2$ when $\hat{r}_i > 0$, and $\hat{o}_i = \hat{r}_i + \lambda_1/2$ otherwise. In other words, larger residuals that the method identifies as corresponding to outlier-contaminated data are shrunk, but not completely discarded. By plugging $\hat{\mathbf{o}}$ back into (3.5), these 'large' residuals cancel out in the squared error term, but still contribute linearly through the ℓ_1-norm regularizer. This is exactly what one would expect, in light of the equivalence established with the variational M-type estimator in (3.6).

Next, it is established that an alternative to solving a sequence of linear systems and scalar Lasso problems, is to solve a single instance of the Lasso with specific response vector and (non-orthonormal) regression matrix.

Proposition 3.2 *Consider* $\hat{\mathbf{o}}_{Lasso}$ *defined as*

$$\hat{\mathbf{o}}_{Lasso} := \arg\min_{\mathbf{o}\in\mathbb{R}^N} \|\mathbf{X}_\mu \mathbf{y} - \mathbf{X}_\mu \mathbf{o}\|_2^2 + \lambda_1 \|\mathbf{o}\|_1 \tag{3.12}$$

where

$$\mathbf{X}_\mu := \begin{bmatrix} \mathbf{I}_N - \mathbf{K}\,(\mathbf{K} + \mu \mathbf{I}_N)^{-1} \\ (\mu \mathbf{K})^{1/2}\,(\mathbf{K} + \mu \mathbf{I}_N)^{-1} \end{bmatrix}. \tag{3.13}$$

Then the minimizers $\{\hat{f}, \hat{\mathbf{o}}\}$ *of (3.5) are fully determined given* $\hat{\mathbf{o}}_{Lasso}$*, as* $\hat{\mathbf{o}} := \hat{\mathbf{o}}_{Lasso}$ *and* $\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \hat{\beta}_i K(\mathbf{x}, \mathbf{x}_i)$*, with* $\hat{\boldsymbol{\beta}} = (\mathbf{K} + \mu \mathbf{I}_N)^{-1} (\mathbf{y} - \hat{\mathbf{o}}_{Lasso})$*.*

Proof: For notational convenience introduce the $N \times 1$ vectors $\mathbf{f} := [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]'$ and $\hat{\mathbf{f}} := [\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_N)]'$, where $\hat{f} \in \mathcal{H}$ is the minimizer of (3.5). Next, consider rewriting (3.5) as

$$\min_{\mathbf{o} \in \mathbb{R}^N} \left[\min_{f \in \mathcal{H}} \|(\mathbf{y} - \mathbf{o}) - \mathbf{f}\|_2^2 + \mu \|f\|_{\mathcal{H}}^2 \right] + \lambda_1 \|\mathbf{o}\|_1. \tag{3.14}$$

The quantity inside the square brackets is a function of $\mathbf{o}$, and can be written explicitly after carrying out the minimization with respect to $f \in \mathcal{H}$. From the results in [119], it follows that the vector of optimum predicted values at the points $\{\mathbf{x}_i\}_{i=1}^N$ is given by $\hat{\mathbf{f}} = \mathbf{K}\hat{\boldsymbol{\beta}} = \mathbf{K} (\mathbf{K} + \mu \mathbf{I}_N)^{-1} (\mathbf{y} - \mathbf{o})$; see also the discussion after (3.8). Similarly, one finds that $\|\hat{f}\|_{\mathcal{H}}^2 = \hat{\boldsymbol{\beta}}' \mathbf{K} \hat{\boldsymbol{\beta}} = (\mathbf{y} - \mathbf{o})' (\mathbf{K} + \mu \mathbf{I}_N)^{-1} \mathbf{K} (\mathbf{K} + \mu \mathbf{I}_N)^{-1} (\mathbf{y} - \mathbf{o})$. Having minimized (3.14) with respect to f, the quantity inside the square brackets is ($\boldsymbol{\Gamma}_\mu := (\mathbf{K} + \mu \mathbf{I}_N)^{-1}$)

$$\begin{aligned} \min_{f \in \mathcal{H}} \left[\|(\mathbf{y} - \mathbf{o}) - \mathbf{f}\|_2^2 + \mu \|f\|_{\mathcal{H}}^2 \right] &= \left\| (\mathbf{y} - \mathbf{o}) - \hat{\mathbf{f}} \right\|_2^2 + \mu \|\hat{f}\|_{\mathcal{H}}^2 \\ &= \|(\mathbf{y} - \mathbf{o}) - \mathbf{K}\boldsymbol{\Gamma}_\mu (\mathbf{y} - \mathbf{o})\|_2^2 + \mu (\mathbf{y} - \mathbf{o})' \boldsymbol{\Gamma}_\mu \mathbf{K} \boldsymbol{\Gamma}_\mu (\mathbf{y} - \mathbf{o}) \\ &= \|(\mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu)\mathbf{y} - (\mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu)\mathbf{o}\|_2^2 + \mu (\mathbf{y} - \mathbf{o})' \boldsymbol{\Gamma}_\mu \mathbf{K} \boldsymbol{\Gamma}_\mu (\mathbf{y} - \mathbf{o}). \end{aligned} \tag{3.15}$$

After expanding the quadratic form in the right-hand side of (3.15), and eliminating the term that does not depend on $\mathbf{o}$, problem (3.14) becomes

$$\min_{\mathbf{o} \in \mathbb{R}^N} \left[\|(\mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu)\mathbf{y} - (\mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu)\mathbf{o}\|_2^2 - 2\mu \mathbf{y}' \boldsymbol{\Gamma}_\mu \mathbf{K} \boldsymbol{\Gamma}_\mu \mathbf{o} + \mu \mathbf{o}' \boldsymbol{\Gamma}_\mu \mathbf{K} \boldsymbol{\Gamma}_\mu \mathbf{o} + \lambda_1 \|\mathbf{o}\|_1 \right].$$

Completing the square one arrives at

$$\min_{\mathbf{o} \in \mathbb{R}^N} \left[\left\| \begin{bmatrix} \mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu \\ (\mu \mathbf{K})^{1/2} \boldsymbol{\Gamma}_\mu \end{bmatrix} \mathbf{y} - \begin{bmatrix} \mathbf{I}_N - \mathbf{K}\boldsymbol{\Gamma}_\mu \\ (\mu \mathbf{K})^{1/2} \boldsymbol{\Gamma}_\mu \end{bmatrix} \mathbf{o} \right\|_2^2 + \lambda_1 \|\mathbf{o}\|_1 \right]$$

which completes the proof. ■

The result in Proposition 3.2 opens the possibility for effective methods to select λ_1. These methods to be described in detail in the ensuing section, capitalize on recent algorithmic advances on Lasso solvers, which allow one to efficiently compute $\hat{\mathbf{o}}_{\text{Lasso}}$ for all values

of the tuning parameter λ_1. This is crucial for obtaining satisfactory robust estimates $\hat{f}$, since *controlling the sparsity* in $\mathbf{o}$ by tuning λ_1 is tantamount to controlling the number of outliers in model (3.2).

3.3.2 Selection of the tuning parameters: robustification paths

As argued before, the tuning parameters μ and λ_1 in (3.5) control the degree of smoothness in $\hat{f}$ and the number of outliers (nonzero entries in $\hat{\mathbf{o}}_{\text{Lasso}}$), respectively. From a statistical learning theory standpoint, μ and λ_1 control the amount of regularization and model complexity, thus capturing the so-termed effective degrees of freedom [59]. Complex models tend to have worse generalization capability, even though the prediction error over the training set $\mathcal{T}$ may be small (overfitting). In the contexts of regularization networks [36] and Lasso estimation for regression [110], corresponding tuning parameters are typically selected via model selection techniques such as cross-validation, or, by minimizing the prediction error over an independent test set, if available [59]. However, these simple methods are severely challenged in the presence of multiple outliers. For example, the *swamping* effect refers to a very large value of the residual r_i corresponding to a left out clean datum $\{y_i, \mathbf{x}_i\}$, because of an unsatisfactory model estimation based on all data except i; data which contain outliers.

The idea here offers an alternative method to overcome the aforementioned challenges, and the possibility to efficiently compute $\hat{\mathbf{o}}_{\text{Lasso}}$ for all values of λ_1, given μ. A brief overview of the state-of-the-art in Lasso solvers is given first. Several methods for selecting μ and λ_1 are then described, which differ on the assumptions of what is known regarding the outlier model (3.2).

Lasso amounts to solving a quadratic programming (QP) problem [110]; hence, an iterative procedure is required to determine $\hat{\mathbf{o}}_{\text{Lasso}}$ in (3.12) for a given value of λ_1. While standard QP solvers can be certainly invoked to this end, an increasing amount of effort has been put recently toward developing fast algorithms that capitalize on the unique properties of Lasso. The Lasso variation of the LARS algorithm [34, Sec. 3.1] is an efficient scheme for computing the entire path of solutions (corresponding to all values of λ_1), elsewhere

referred to as homotopy paths [34, 48], or, regularization paths [44]. LARS capitalizes on piecewise linearity of the Lasso path of solutions, while incurring the complexity of a single LS fit, i.e., when $\lambda_1 = 0$. Homotopy algorithms have been also developed to solve the Lasso online, when data pairs $\{y_i, \mathbf{x}_i\}$ are collected sequentially in time [6, 48]. Coordinate descent algorithms have been shown competitive, even outperforming LARS when p is large, as demonstrated in [45]; see also [44, 126], and the references therein. Coordinate descent solvers capitalize on the fact that Lasso can afford a very simple solution in the scalar case, which is given in closed form in terms of a soft-thresholding operation [cf. (3.11)]. Further computational savings are attained through the use of *warm starts* [44], when computing the Lasso path of solutions over a grid of decreasing values of λ_1. An efficient solver capitalizing on variable separability has been proposed in [125], while a semismooth Newton method was put forth in [56].

Consider then a grid of G_μ values of μ in the interval $[\mu_{\min}, \mu_{\max}]$, evenly spaced in a logarithmic scale. Likewise, for each μ consider a similar type of grid consisting of G_λ values of λ_1, where $\lambda_{\max} := 2 \min_i |\mathbf{y}' \mathbf{X}'_\mu \mathbf{x}_{\mu,i}|$ is the minimum λ_1 value such that $\hat{\mathbf{o}}_{\text{Lasso}} \neq \mathbf{0}_N$ [45], and $\mathbf{X}_\mu := [\mathbf{x}_{\mu,1} \ldots \mathbf{x}_{\mu,N}]$ in (3.12). Typically, $\lambda_{\min} = \epsilon \lambda_{\max}$ with $\epsilon = 10^{-4}$, say. Note that each of the G_μ values of μ gives rise to a different λ grid, since $\lambda_{\max}$ depends on μ through $\mathbf{X}_\mu$. Given the previously surveyed algorithmic alternatives to tackle the Lasso, it is safe to assume that (3.12) can be efficiently solved over the (nonuniform) $G_\mu \times G_\lambda$ grid of values of the tuning parameters. This way, for each value of μ one obtains G_λ samples of the Lasso homotopy paths, henceforth referred to as *robustification paths* as a means of highlighting the connection between robustness and sparsity in the nonparametric context of the present work. As λ_1 decreases, more variables $\hat{o}_{\text{Lasso},i}$ enter the model signifying that more of the training data are deemed to contain outliers. An example of the robustification path is given in Fig. 3.3.

Based on the robustification paths and the prior knowledge available on the outlier model (3.2), several alternatives are given next to select the 'best' pair $\{\mu, \lambda_1\}$ in the grid $G_\mu \times G_\lambda$.

Number of outliers is known: For each value of μ in the grid G_μ, by direct inspection of

the robustification paths one can determine the range of values for λ_1, such that $\hat{\mathbf{o}}_{\text{Lasso}}$ has exactly N_o nonzero entries. This procedure yields a reduced grid $G_\mu \times \tilde{G}_\lambda$ of candidate tuning parameter pairs, which is again nonuniform since the obtained λ_1-intervals may differ per μ. Focusing on the reduced grid, and after discarding outliers which are now fixed and known, K-fold cross-validation can be applied to determine $\{\mu^*, \lambda_1^*\}$; see e.g., [59, Ch. 7].

Variance of the nominal noise is known: Supposing that the variance σ_ε^2 of the i.i.d. nominal noise variables ε_i in (3.2) is known, one can proceed as follows. Using the solution $\hat{f}$ obtained for each pair $\{\mu_i, \lambda_j\}$ on the grid, form the $G_\mu \times G_\lambda$ sample variance matrix $\bar{\mathbf{\Sigma}}$ with ij-th entry

$$[\bar{\mathbf{\Sigma}}]_{ij} := \sum_{u|\hat{o}_{\text{Lasso},u}=0} \hat{r}_u^2/\hat{N}_o = \sum_{u|\hat{o}_{\text{Lasso},u}=0} (y_u - \hat{f}(\mathbf{x}_u))^2/\hat{N}_o \tag{3.16}$$

where $\hat{N}_o$ stands for the number of nonzero entries in $\hat{\mathbf{o}}_{\text{Lasso}}$. Although not made explicit, the right-hand side of (3.16) depends on $\{\mu_i, \lambda_j\}$ through the estimate $\hat{f}$, $\hat{\mathbf{o}}_{\text{Lasso}}$ and $\hat{N}_o$. The entries $[\bar{\mathbf{\Sigma}}]_{ij}$ correspond to a sample estimate of σ_ε^2, without considering those training data $\{y_i, \mathbf{x}_i\}$ that the method determined to be contaminated with outliers, i.e., those indices i for which $\hat{o}_{\text{Lasso},i} \neq 0$. The 'winner' tuning parameters $\{\mu^*, \lambda_1^*\} := \{\mu_{i^*}, \lambda_{j^*}\}$ are such that

$$[i^*, j^*] := \arg\min_{i,j} |[\bar{\mathbf{\Sigma}}]_{ij} - \sigma_\varepsilon^2| \tag{3.17}$$

which is an absolute variance deviation (AVD) criterion.

Variance of the nominal noise is unknown: If σ_ε^2 is unknown, one can still compute a robust estimate of the variance $\hat{\sigma}_\varepsilon^2$, and repeat the previous procedure (with known nominal noise variance) after replacing σ_ε^2 with $\hat{\sigma}_\varepsilon^2$ in (3.17). One option is based on the median absolute deviation (MAD) estimator, namely

$$\hat{\sigma}_\varepsilon := 1.4826 \times \text{median}_i \left(|\hat{r}_i - \text{median}_j \left(\hat{r}_j \right)| \right) \tag{3.18}$$

where the residuals $\hat{r}_i = y_i - \hat{f}(\mathbf{x}_i)$ are formed based on a nonrobust estimate of f, obtained e.g., after solving (3.5) with $\lambda_1 = 0$ and using a small subset of the training dataset $\mathcal{T}$. The factor 1.4826 provides an approximately unbiased estimate of the standard deviation when the nominal noise is Gaussian. Typically, $\hat{\sigma}_\varepsilon$ in (3.18) is used as an estimate for the scale of the errors in general M-type robust estimators; see e.g., [28] and [78].

Remark 3.2 (How sparse is sparse) Even though the very nature of outliers dictates that N_o is typically a small fraction of N – and thus $\mathbf{o}$ in (3.2) is sparse – the method here capitalizes on, but *is not limited* to sparse settings. For instance, choosing $\lambda_1 \in [\lambda_{\min} \approx 0, \lambda_{\max}]$ along the robustification paths allows one to continuously control the sparsity level, and potentially select the right value of λ_1 for any given $N_o \in \{1, \ldots, N\}$. Admittedly, if N_o is large relative to N, then even if it is possible to identify and discard the outliers, the estimate $\hat{f}$ may not be accurate due to the lack of outlier-free data. Interestingly, simulation results in [49] demonstrate that the performance of this chapter's sparsity-controlling outlier rejection methods degrade gracefully, as $N_o \to N$.

3.4 Refinement via Nonconvex Regularization

Instead of substituting $\|\mathbf{o}\|_0$ in (3.3) by its closest convex approximation, namely $\|\mathbf{o}\|_1$, letting the surrogate function to be non-convex can yield tighter approximations. For example, the ℓ_0-norm of a vector $\mathbf{x} \in \mathbb{R}^n$ was surrogated in [23] by the logarithm of the geometric mean of its elements, or by $\sum_{i=1}^{n} \log|x_i|$. In rank minimization problems, apart from the nuclear norm relaxation, minimizing the logarithm of the determinant of the unknown matrix has been proposed as an alternative surrogate [39]. Adopting related ideas in the present nonparametric context, consider approximating (3.3) by

$$\min_{\substack{f\in\mathcal{H}\\ \mathbf{o}\in\mathbb{R}^N}} \left[\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i) - o_i)^2 + \mu\|f\|_{\mathcal{H}}^2 + \lambda_0 \sum_{i=1}^{N} \log(|o_i| + \delta)\right] \tag{3.19}$$

where δ is a sufficiently small positive offset introduced to avoid numerical instability.

Since the surrogate term in (3.19) is concave, the overall problem is nonconvex. Still, local methods based on iterative linearization of $\log(|o_i| + \delta)$, around the current iterate $o_i^{(k)}$, can be adopted to minimize (3.19). From the concavity of the logarithm, its local linear approximation serves as a global overestimator. Standard majorization-minimization algorithms motivate minimizing the global linear overestimator instead. This leads to the

following iteration for $k = 0, 1, \ldots$ (see e.g., [72] for further details)

$$[f^{(k)}, \mathbf{o}^{(k)}] := \arg \min_{\substack{f \in \mathcal{H} \\ \mathbf{o} \in \mathbb{R}^N}} \left[\sum_{i=1}^{N} (y_i - f(\mathbf{x}_i) - o_i)^2 + \mu \|f\|_{\mathcal{H}}^2 + \lambda_0 \sum_{i=1}^{N} w_i^{(k)} |o_i| \right] \tag{3.20}$$

$$w_i^{(k)} := \left(|o_i^{(k-1)}| + \delta \right)^{-1}, \quad i = 1, \ldots, N. \tag{3.21}$$

It is possible to eliminate the optimization variable $f \in \mathcal{H}$ from (3.20), by direct application of the result in Proposition 3.2. The equivalent update for $\mathbf{o}$ at iteration k is then given by

$$\mathbf{o}^{(k)} := \arg \min_{\mathbf{o} \in \mathbb{R}^N} \left[\|\mathbf{X}_\mu \mathbf{y} - \mathbf{X}_\mu \mathbf{o}\|_2^2 + \lambda_0 \sum_{i=1}^{N} w_i^{(k)} |o_i| \right] \tag{3.22}$$

which amounts to an iteratively reweighted version of (3.12). If the value of $|o_i^{(k-1)}|$ is small, then in the next iteration the corresponding regularization term $\lambda_0 w_i^{(k)} |o_i|$ has a large weight, thus promoting shrinkage of that coordinate to zero. On the other hand when $|o_i^{(k-1)}|$ is significant, the cost in the next iteration downweighs the regularization, and places more importance to the LS component of the fit. For small δ, analysis of the limiting point $\mathbf{o}^*$ of (3.22) reveals that

$$\lambda_0 w_i^* |o_i^*| \approx \begin{cases} \lambda_0, & |o_i^*| \neq 0 \\ 0, & |o_i^*| = 0 \end{cases}$$

and hence, $\lambda_0 \sum_{i=1}^{N} w_i^* |o_i^*| \approx \lambda_0 \|\mathbf{o}^*\|_0$.

A good initialization for the iteration in (3.22) and (3.21) is $\hat{\mathbf{o}}_{\text{Lasso}}$, which corresponds to the solution of (3.12) [and (3.5)] for $\lambda_0 = \lambda_1^*$ and $\mu = \mu^*$. This is equivalent to a single iteration of (3.22) with all weights equal to unity. The numerical tests in Section 3.5 will indicate that even a single iteration of (3.22) suffices to obtain improved estimates $\hat{f}$, in comparison to those obtained from (3.12). The following remark sheds further light towards understanding why this should be expected.

Remark 3.3 (Refinement through bias reduction) Uniformly weighted ℓ_1-norm regularized estimators such as (3.5) are biased [133], due to the shrinkage effected on the estimated coefficients. It will be argued next that the improvements due to (3.22) can be leveraged to bias reduction. Several workarounds have been proposed to correct the bias in

sparse regression, that could as well be applied here. A first possibility is to retain only the support of (3.12) and re-estimate the amplitudes via, e.g., the unbiased LS estimator [34]. An alternative approach to reducing bias is through nonconvex regularization using e.g., the smoothly clipped absolute deviation (SCAD) scheme [38]. The SCAD penalty could replace the sum of logarithms in (3.19), still leading to a nonconvex problem. To retain the efficiency of convex optimization solvers while simultaneously limiting the bias, suitably *weighted* ℓ_1-norm regularizers have been proposed instead [133]. The constant weights in [133] play a role similar to those in (3.21); hence, bias reduction is expected.

3.5 Numerical Experiments

3.5.1 Robust thin-plate smoothing splines

To validate the proposed approach to robust nonparametric regression, a simulated test is carried out here in the context of thin-plate smoothing spline approximation [33, 120]. Specializing (3.5) to this setup, the robust thin-plate splines estimator can be formulated as

$$\min_{\substack{f\in\mathcal{S}\\ \mathbf{o}\in\mathbb{R}^N}} \left[\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i) - o_i)^2 + \mu\int_{\mathbb{R}^2}\|\nabla^2 f\|_F^2 d\mathbf{x} + \lambda_1\|\mathbf{o}\|_1\right] \tag{3.23}$$

where $||\nabla^2 f||_F$ denotes the Frobenius norm of the Hessian of $f:\mathbb{R}^2\to\mathbb{R}$. The penalty functional

$$J[f] := \int_{\mathbb{R}^2}\|\nabla^2 f\|_F^2 d\mathbf{x} = \int_{\mathbb{R}^2}\left[\left(\frac{\partial^2 f}{\partial x_1^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_1\partial x_2}\right)^2 + \left(\frac{\partial^2 f}{\partial x_2^2}\right)^2\right] d\mathbf{x} \tag{3.24}$$

extends to $\mathbb{R}^2$ the one-dimensional roughness regularization used in smoothing spline models. For $\mu = 0$, the (non-unique) estimate in (3.23) corresponds to a *rough* function interpolating the outlier compensated data; while as $\mu\to\infty$ the estimate is linear (cf. $\nabla^2\hat{f}(\mathbf{x})\equiv\mathbf{0}_{2\times 2}$). The optimization is over $\mathcal{S}$, the space of Sobolev functions, for which $J[f]$ is well defined [33, p. 85]. Reproducing kernel Hilbert spaces such as $\mathcal{S}$, with inner-products (and norms) involving derivatives are studied in detail in [119].

Different from the cases considered so far, the smoothing penalty in (3.24) is only a seminorm, since first-order polynomials vanish under $J[\cdot]$. Omitting details than can be

found in [119, p. 30], a unique minimizer of (3.23) exists provided the input vectors $\{\mathbf{x}_i \in \mathbb{R}^2\}_{i=1}^N$ do not fall on a straight line. The solution admits the finitely parametrized form $\hat{f}(\mathbf{x}) = \sum_{i=1}^N \beta_i K(\mathbf{x}, \mathbf{x}_i) + \boldsymbol{\alpha}_1' \mathbf{x} + \alpha_0$, where in this case $K(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|^2 \log \|\mathbf{x} - \mathbf{y}\|$ is a radial basis function. In simple terms, the solution as a kernel expansion is augmented with a member of the null space of $J[\cdot]$. The unknown parameters $\{\boldsymbol{\beta}, \boldsymbol{\alpha}_1, \alpha_0\}$ are obtained in closed form, as solutions to a constrained, regularized LS problem; see [119, p. 33]. As a result, Proposition 3.2 still holds with minor modifications on the structure of $\mathbf{X}_\mu$.

Remark 3.4 (Bayesian framework) Adopting a Bayesian perspective, one could model $f(\mathbf{x})$ in (3.2) as a sample function of a zero mean Gaussian stationary process, with covariance function $K(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2 \log \|\mathbf{x} - \mathbf{y}\|$ [70]. Consider as well that $\{f(\mathbf{x}), \{o_i, \varepsilon_i\}_{i=1}^N\}$ are mutually independent, while $\varepsilon_i \sim \mathcal{N}(0, \mu^*/2)$ and $o_i \sim \mathcal{L}(0, \mu^*/\lambda_1^*)$ in (3.2) are i.i.d. Gaussian and Laplace distributed, respectively. From the results in [70] and a straightforward calculation, it follows that setting $\lambda_1 = \lambda_1^*$ and $\mu = \mu^*$ in (3.23) yields estimates $\hat{f}$ (and $\hat{\mathbf{o}}$) which are optimal in a maximum a posteriori sense. This provides yet another means of selecting the parameters μ and λ_1, further expanding the options presented in Section 3.3.2.

The simulation setup is as follows. Noisy samples of the true function $f_o : \mathbb{R}^2 \to \mathbb{R}$ comprise the training set $\mathcal{T}$. Function f_o is generated as a Gaussian mixture with two components, with respective mean vectors and covariance matrices given by

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.2295 \\ 0.4996 \end{bmatrix}, \; \boldsymbol{\Sigma}_1 = \begin{bmatrix} 2.2431 & 0.4577 \\ 0.4577 & 1.0037 \end{bmatrix},$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 2.4566 \\ 2.9461 \end{bmatrix}, \; \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2.9069 & 0.5236 \\ 0.5236 & 1.7299 \end{bmatrix}.$$

Function $f_o(\mathbf{x})$ is depicted in Fig. 3.4 (a). The training data set comprises $N = 200$ examples, with inputs $\{\mathbf{x}_i\}_{i=1}^N$ drawn from a uniform distribution in the square $[0, 3] \times [0, 3]$. Several values ranging from 5% to 25% of the data are generated contaminated with outliers. Without loss of generality, the corrupted data correspond to the first N_o training samples with $N_o = \{10, 20, 30, 40, 50\}$, for which the response values $\{y_i\}_{i=1}^{N_o}$ are independently

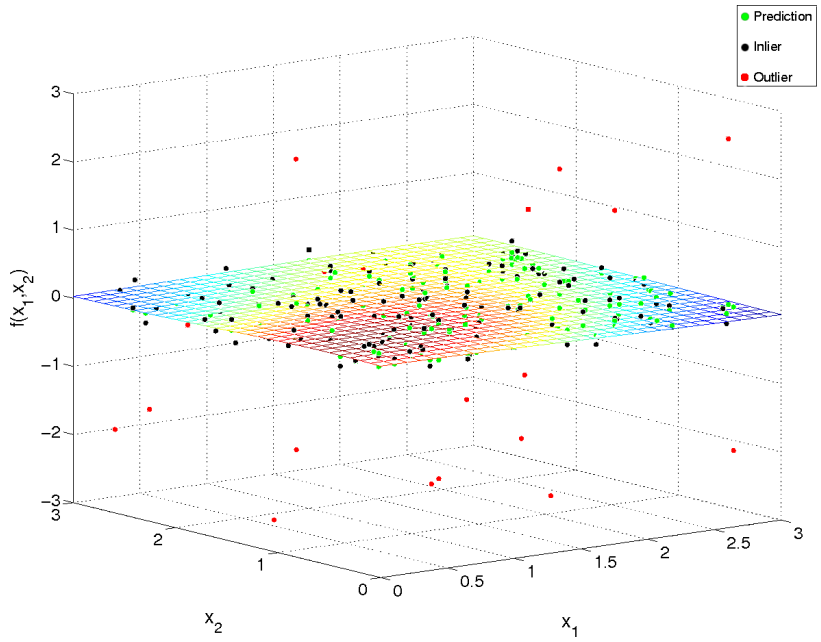


Figure 3.2: True Gaussian mixture function $f_o(\mathbf{x})$, and its 180 noisy samples taken over $[0,3]\times[0,3]$ shown as black dots. The red dots indicate the $N_o = 20$ outliers in the training data set $\mathcal{T}$. The green points indicate the predicted responses $\hat{y}_i$ at the sampling points $\mathbf{x}_i$, from the estimate $\hat{f}$ obtained after solving (3.23). Note how all green points are close to the surface f_o.

drawn from a uniform distribution over $[-3,3]$. Outlier-free data are generated according to the model $y_i = f_o(\mathbf{x}_i) + \varepsilon_i$, where the independent additive noise terms $\varepsilon_i \sim \mathcal{N}(0, 10^{-1})$ are Gaussian distributed, for $i = N_o + 1, \ldots, 200$. For the case where $N_o = 20$, the data used in the experiment is shown in Fig. 3.2. Superimposed to the true function f_o are 180 black points corresponding to data drawn from the nominal model, as well as 20 red outlier points.

For this experiment, the nominal noise variance $\sigma_\varepsilon^2 = 10^{-1}$ is assumed known. A nonuniform grid of μ and λ_1 values is constructed, as described in Section 3.3.2. The relevant parameters are $G_\mu = G_\lambda = 200$, $\mu_{\min} = 10^{-9}$ and $\mu_{\max} = 1$. For each value of μ, the λ_1 grid spans the interval defined by $\lambda_{\max} := 2\min_i |\mathbf{y}'\mathbf{X}'_\mu \mathbf{x}_{\mu,i}|$ and $\lambda_{\min} = \epsilon\lambda_{\max}$, where $\epsilon = 10^{-4}$. Each of the G_μ robustification paths corresponding to the solution of (3.12) is obtained using the SpaRSA toolbox in [125], exploiting warm starts for faster convergence.

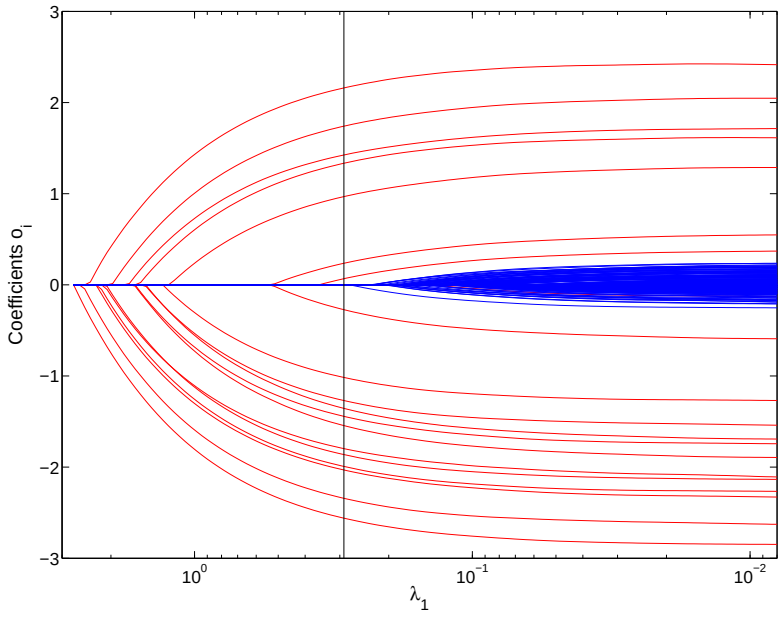


Figure 3.3: Robustification path with optimum smoothing parameter $\mu^* = 3.53 \times 10^{-1}$. The data is corrupted with $N_o = 20$ outliers. The coefficients $\hat{o}_i$ corresponding to the outliers are shown in red, while the rest are shown in blue. The vertical line indicates the selection of $\lambda_1^* = 2.90 \times 10^{-1}$, and shows that the outliers were correctly identified.

Fig. 3.3 depicts an example with $N_o = 20$ and $\mu^* = 3.53 \times 10^{-1}$. With the robustification paths at hand, it is possible to form the sample variance matrix $\bar{\mathbf{\Sigma}}$ [cf. (3.16)], and select the optimum tuning parameters $\{\mu^*, \lambda_1^*\}$ based on the criterion (3.17). Finally, the robust estimates are refined by running a single iteration of (3.22) as described in Section 3.4. The value $\delta = 10^{-5}$ was utilized, and several experiments indicated that the results are quite insensitive to the selection of this parameter.

The same experiment was conducted for a variable number of outliers N_o, and the results are listed in Table 3.1. In all cases, a 100% outlier identification success rate was obtained, for the chosen value of the tuning parameters. This even happened at the first stage of the method, i.e., $\hat{\mathbf{o}}_{\text{Lasso}}$ in (3.12) had the correct support in all cases. It has been observed in some other setups that (3.12) may select a larger support than $[1, N_o]$, but after running a

Table 3.1: Results for the thin-plate splines simulated test.

N_o	λ_1^*	μ^*	**e$\bar{\text{r}}$r (3.5)**	**e$\bar{\text{r}}$r (3.19)**	**Err$_{\mathcal{T}}$ (3.5)**	**Err$_{\mathcal{T}}$ (3.19)**
10	2.72×10^{-1}	9.72×10^{-2}	1.00×10^{-2}	9.92×10^{-3}	1.92×10^{-2}	1.47×10^{-2}
20	2.90×10^{-1}	3.53×10^{-1}	1.02×10^{-2}	1.03×10^{-2}	5.79×10^{-2}	4.86×10^{-2}
30	2.75×10^{-1}	4.33×10^{-2}	1.00×10^{-2}	9.80×10^{-3}	1.60×10^{-2}	1.32×10^{-2}
40	2.58×10^{-1}	9.90×10^{-1}	9.90×10^{-3}	1.07×10^{-2}	5.13×10^{-2}	2.90×10^{-2}
50	2.36×10^{-1}	5.34×10^{-1}	1.04×10^{-2}	1.03×10^{-2}	6.89×10^{-2}	4.53×10^{-2}

few iterations of (3.22) the true support was typically identified. To assess quality of the estimated function $\hat{f}$, two figures of merit were considered. First, the *training error* e$\bar{\text{r}}$r was evaluated as

$$\text{e}\bar{\text{r}}\text{r} = \frac{1}{N - N_o} \sum_{i=N_o}^{N} \left(y_i - \hat{f}(\mathbf{x}_i) \right)^2$$

i.e., the average loss over the training sample $\mathcal{T}$ after excluding outliers. Second, to assess the generalization capability of $\hat{f}$, an approximation to the *generalization error* Err$_{\mathcal{T}}$ was computed as

$$\text{Err}_{\mathcal{T}} = E\left[\left(y - \hat{f}(\mathbf{x}) \right)^2 | \mathcal{T} \right] \approx \frac{1}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \left(\tilde{y}_i - \hat{f}(\tilde{\mathbf{x}}_i) \right)^2 \tag{3.25}$$

where $\{\tilde{y}_i, \tilde{\mathbf{x}}_i\}_{i=1}^{\tilde{N}}$ is an independent test set generated from the model $\tilde{y}_i = f_o(\tilde{\mathbf{x}}_i) + \varepsilon_i$. For the results in Table 3.1, $\tilde{N} = 961$ was adopted corresponding to a uniform rectangular grid of 31×31 points $\tilde{\mathbf{x}}_i$ in $[0, 3] \times [0, 3]$. Inspection of Table 3.1 reveals that the training errors e$\bar{\text{r}}$r are comparable for the function estimates obtained after solving (3.5) or its nonconvex refinement (3.19). Interestingly, when it comes to the more pragmatic generalization error Err$_{\mathcal{T}}$, the refined estimator (3.19) has an edge for all values of N_o. As expected, the bias reduction effected by the iteratively reweighting procedure of Section 3.4 improves considerably the generalization capability of the method; see also Remark 3.3.

A pictorial summary of the results is given in Fig. 3.4, for $N_o = 20$ outliers. Fig. 3.4 (a) depicts the true Gaussian mixture $f_o(\mathbf{x})$, whereas Fig. 3.4 (b) shows the nonrobust

thin-plate splines estimate obtained after solving

$$\min_{f\in\mathcal{S}} \left[\sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 + \mu \int_{\mathbb{R}^2} \|\nabla^2 f\|_F^2 d\mathbf{x} \right]. \tag{3.26}$$

Even though the thin-plate penalty enforces some degree of smoothness, the estimate is severely disrupted by the presence of outliers [cf. the difference on the z-axis ranges]. On the other hand, Figs. 3.4 (c) and (d), respectively, show the robust estimate $\hat{f}$ with $\lambda_1^* = 2.90 \times 10^{-1}$, and its bias reducing refinement for which the improvement is apparent.

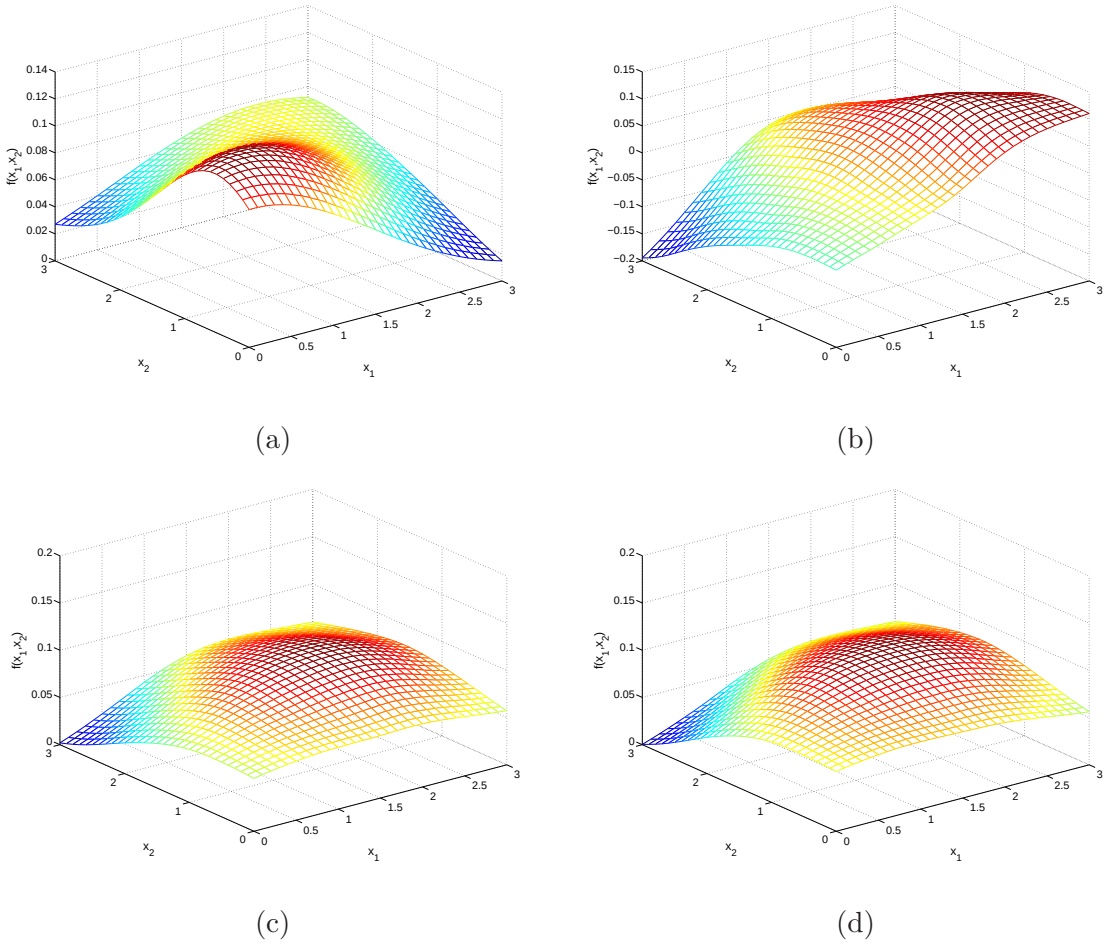


Figure 3.4: Robust estimation of a Gaussian mixture using thin-plate splines. The data is corrupted with $N_o = 20$ outliers. (a) True function $f_o(\mathbf{x})$; (b) nonrobust predicted function obtained after solving (3.26); (c) predicted function after solving (3.23) with the optimum tuning parameters; (d) refined predicted function using the nonconvex regularization in (3.19).

3.5.2 Sinc function estimation

The univariate function $\text{sinc}(x) := \sin(\pi x)/(\pi x)$ is commonly adopted to evaluate the performance of nonparametric regression methods [27, 132]. Given noisy training examples with a small fraction of outliers, approximating $\text{sinc}(x)$ over the interval $[-5, 5]$ is considered in the present simulated test. The sparsity-controlling robust nonparametric regression methods of this chapter are compared with the SVR [118] and robust SVR in [27], for the case of the ϵ-insensitve loss function with values $\epsilon = 0.1$ and $\epsilon = 0.01$. In order to implement (R)SVR, routines from a publicly available SVM Matlab toolbox were utilized [57]. Results for the nonrobust regularization network approach in (1.4) (with $V(u) = u^2$) are reported as well, to assess the performance degradation incurred when compared to the aforementioned robust alternatives. Because the fraction of outliers (N_o/N) in the training data is assumed known to the method of [27], the same will be assumed towards selecting the tuning parameters λ_1 and μ in (3.5), as described in Section 3.3.2. The $\{\mu, \lambda_1\}$-grid parameters selected for the experiment in Section 3.5.1 were used here as well, except for $\mu_{\min} = 10^{-5}$. Space $\mathcal{H}$ is chosen to be the RKHS induced by the positive definite Gaussian kernel function $K(u, v) = \exp\left[-(u - v)^2/(2\eta^2)\right]$, with parameter $\eta = 0.1$ for all cases.

The training set comprises $N = 50$ examples, with scalar inputs $\{x_i\}_{i=1}^N$ drawn from a uniform distribution over $[-5, 5]$. Uniformly distributed outliers $\{y_i\}_{i=1}^{N_o} \sim \mathcal{U}[-5, 5]$ are artificially added in $\mathcal{T}$, with $N_o = 3$ resulting in 6% contamination. Nominal data in $\mathcal{T}$ adheres to the model $y_i = \text{sinc}(x_i) + \varepsilon_i$ for $i = N_o + 1, \ldots, N$, where the independent additive noise terms ε_i are zero-mean Gaussian distributed. Three different values are considered for the nominal noise variance, namely $\sigma_\varepsilon^2 = 1 \times 10^{-l}$ for $l = 2, 3, 4$. For the case where $\sigma_\varepsilon^2 = 1 \times 10^{-4}$, the data used in the experiment are shown in Fig. 3.5 (a). Superimposed to the true function $\text{sinc}(x)$ (shown in blue) are 47 black points corresponding to the noisy data obeying the nominal model, as well as 3 outliers depicted as red points.

The results are summarized in Table 3.2, which lists the generalization errors $\text{Err}_{\mathcal{T}}$ attained by the different methods tested, and for varying σ_ε^2. The independent test set $\{\tilde{y}_i, \tilde{x}_i\}_{i=1}^{\tilde{N}}$ used to evaluate (3.25) was generated from the model $\tilde{y}_i = \text{sinc}(\tilde{x}_i) + \varepsilon_i$, where the $\tilde{x}_i$ define a $\tilde{N} = 101$-element uniform grid over $[-5, 5]$. A first (expected) observation

Table 3.2: Generalization error ($\mathrm{Err}_{\mathcal{T}}$) results for the sinc function estimation experiment.

Method	$\sigma_\varepsilon^2 = 1 \times 10^{-4}$	$\sigma_\varepsilon^2 = 1 \times 10^{-3}$	$\sigma_\varepsilon^2 = 1 \times 10^{-2}$
Nonrobust [(1.4) with $V(u) = u^2$]	5.67×10^{-2}	8.28×10^{-2}	1.13×10^{-1}
SVR with $\epsilon = 0.1$	5.00×10^{-3}	6.42×10^{-4}	6.15×10^{-3}
RSVR with $\epsilon = 0.1$	1.10×10^{-3}	5.10×10^{-4}	4.47×10^{-3}
SVR with $\epsilon = 0.01$	8.24×10^{-5}	4.79×10^{-4}	5.60×10^{-3}
RSVR with $\epsilon = 0.01$	7.75×10^{-5}	3.90×10^{-4}	3.32×10^{-3}
Sparsity-controlling in (3.5)	1.47×10^{-4}	6.56×10^{-4}	4.60×10^{-3}
Refinement in (3.19)	7.46×10^{-5}	3.59×10^{-4}	3.21×10^{-3}

is that all robust alternatives markedly outperform the nonrobust regularization network approach in (1.4), by an order of magnitude or even more, regardless of the value of σ_ε^2. As reported in [27], RSVR uniformly outperforms SVR. For the case $\epsilon = 0.01$, RSVR also uniformly outperforms the sparsity-controlling method in (3.5). Interestingly, after refining the estimate obtained via (3.5) through a couple iterations of (3.22) (cf. Section 3.4), the lowest generalization errors are obtained, uniformly across all simulated values of the nominal noise variance. Results for the RSVR with $\epsilon = 0.01$ come sufficiently close, and are equally satisfactory for all practical purposes; see also Fig. 3.5 for a pictorial summary of the results when $\sigma_\varepsilon^2 = 1 \times 10^{-4}$.

While specific error values or method rankings are arguably anecdotal, two conclusions stand out: (i) model (3.2) and its sparsity-controlling estimators (3.5) and (3.19) are effective approaches to nonparametric regression in the presence of outliers; and (ii) when initialized with $\hat{\mathbf{o}}_{\mathrm{Lasso}}$ the refined estimator (3.19) can considerably improve the performance of (3.5), at the price of a modest increase in computational complexity. While (3.5) endowed with the sparsity-controlling mechanisms of Section 3.3.2 tends to overestimate the 'true' support of $\mathbf{o}$, numerical results have consistently shown that the refinement in Section 3.4 is more effective when it comes to support recovery.

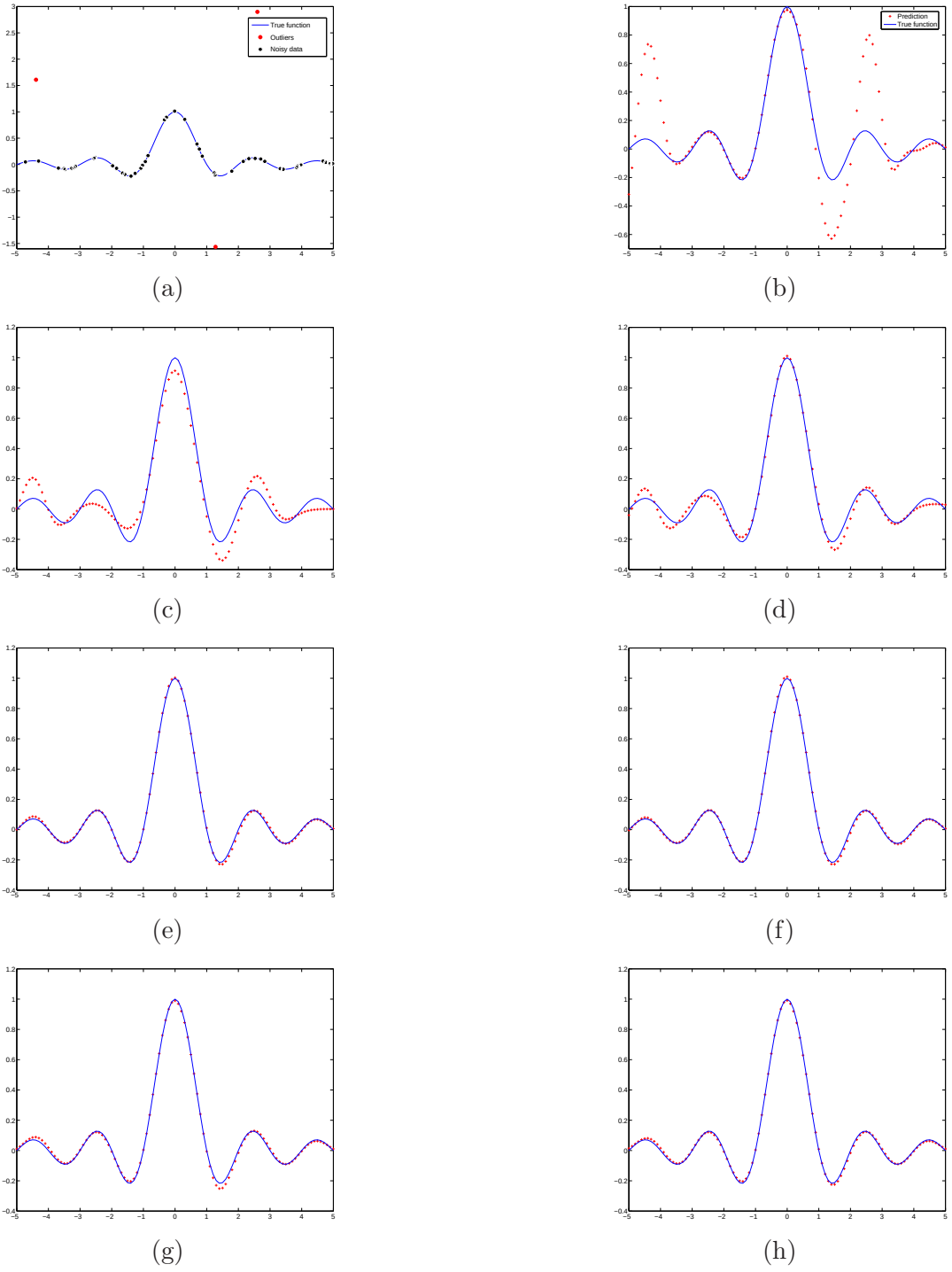


Figure 3.5: Robust estimation of the sinc function. (a) Noisy training data and outliers; (b) predicted values obtained after solving (1.4) with $V(u) = u^2$; (c) SVR predictions for $\epsilon = 0.1$; (d) RSVR predictions for $\epsilon = 0.1$; (e) SVR predictions for $\epsilon = 0.01$; (f) RSVR predictions for $\epsilon = 0.01$; (g) predicted values obtained after solving (3.5); (h) refined predictions using the (3.19).

3.5.3 Load curve data cleansing

In this section, the robust nonparametric methods described so far are applied to the problem of load curve cleansing outlined in Section 3.1. Given load data $\mathcal{T} := \{y_i, t_i\}_{i=1}^N$ corresponding to a building's power consumption measurements y_i, acquired at time instants t_i, $i = 1, \ldots, N$, the proposed approach to load curve cleansing minimizes

$$\min_{\substack{f \in \mathcal{S} \\ \mathbf{o} \in \mathbb{R}^N}} \left[\sum_{i=1}^{N} (y_i - f(t_i) - o_i)^2 + \mu \int_{\mathbb{R}} f''(t) dt + \lambda_1 \|\mathbf{o}\|_1 \right] \tag{3.27}$$

where $f''(t)$ denotes the second-order derivative of $f : \mathbb{R} \to \mathbb{R}$. This way, the solution $\hat{f}$ provides a cleansed estimate of the load profile, and the support of $\hat{\mathbf{o}}$ indicates the instants where significant load deviations, or, meter failures occurred. Estimator (3.27) specializes (3.5) to the so-termed *cubic smoothing splines*; see e.g., [59, 119]. It is also subsumed as a special case of the robust thin-plate splines estimator (3.23), when the target function f has domain in $\mathbb{R}$ [cf. how the smoothing penalty (3.24) simplifies to the one in (3.27) in the one-dimensional case].

In light of the aforementioned connection, it should not be surprising that $\hat{f}$ admits a unique, finite-dimensional minimizer, which corresponds to a *natural spline* with knots at $\{t_i\}_{i=1}^N$; see e.g., [59, p. 151]. Specifically, it follows that $\hat{f}(t) = \sum_{i=1}^N \hat{\theta}_i b_i(t)$, where $\{b_i(t)\}_{i=1}^N$ is the basis set of natural spline functions, and the vector of expansion coefficients $\hat{\boldsymbol{\theta}} := [\hat{\theta}_1, \ldots, \hat{\theta}_N]'$ is given by

$$\hat{\boldsymbol{\theta}} = \left(\mathbf{B}'\mathbf{B} + \mu\boldsymbol{\Psi}\right)^{-1} \mathbf{B}'(\mathbf{y} - \hat{\mathbf{o}})$$

where matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$ has ij-th entry $[\mathbf{B}]_{ij} = b_j(t_i)$; while $\boldsymbol{\Psi} \in \mathbb{R}^{N \times N}$ has ij-th entry $[\boldsymbol{\Psi}]_{ij} = \int b_i''(t) b_j''(t) dt$. Spline coefficients can be computed more efficiently if the basis of B-splines is adopted instead; details can be found in [59, p. 189] and [117].

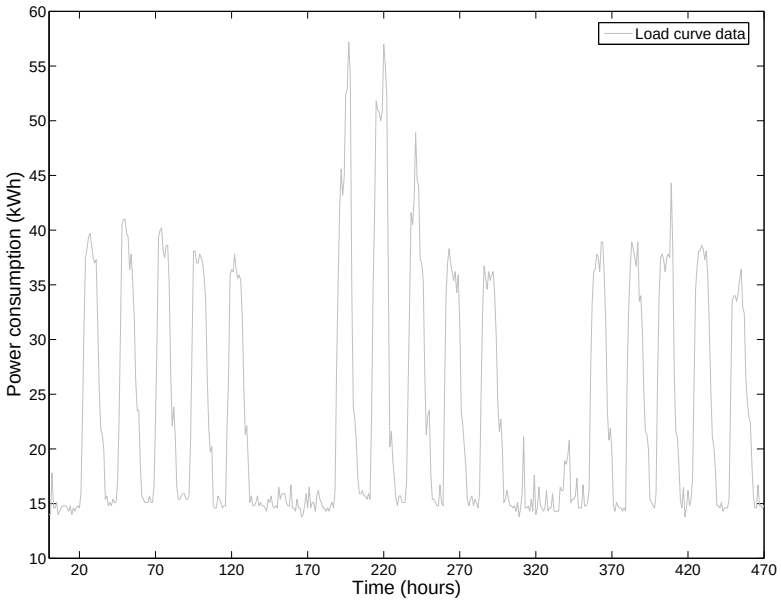
Without considering the outlier variables in (3.27), a B-spline estimator for load curve cleansing was put forth in [25]. An alternative Nadaraya-Watson estimator from the Kernel smoothing family was considered as well. In any case, outliers are identified during a post-processing stage, after the load curve has been estimated nonrobustly. Supposing for instance that the approach in [25] correctly identifies outliers most of the time, it still does

not yield a cleansed estimate $\hat{f}$. This should be contrasted with the estimator (3.27), which accounts for the outlier compensated data to yield a cleansed estimate at once. Moreover, to select the 'optimum' smoothing parameter μ, the approach of [25] requires the user to manually label the outliers present in a training subset of data, during a pre-processing stage. This subjective component makes it challenging to reproduce the results of [25], and for this reason comparisons with the aforementioned scheme are not included in the sequel.

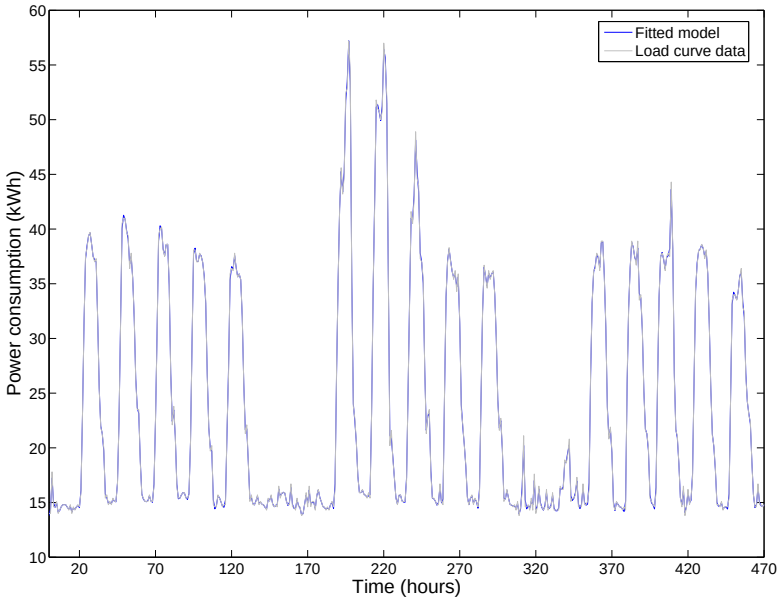
Next, estimator (3.27) is tested on real load curve data provided by the NorthWrite Energy Group. The dataset consists of power consumption measurements (in kWh) for a government building, collected every fifteen minutes during a period of more than five years, ranging from July 2005 to October 2010. Data is downsampled by a factor of four, to yield one measurement per hour. For the present experiment, only a subset of the whole data is utilized for concreteness, where $N = 501$ was chosen corresponding to a 501 hour period. A snapshot of this training load curve data in $\mathcal{T}$, spanning a particular three-week period is shown in Fig. 3.6 (a). Weekday activity patterns can be clearly discerned from those corresponding to weekends, as expected for most government buildings; but different, e.g., for the load profile of a grocery store. Fig. 3.6 (b) shows the nonrobust smoothing spline fit to the training data in $\mathcal{T}$ (also shown for comparison purposes), obtained after solving

$$\min_{f\in\mathcal{S}} \left[\sum_{i=1}^{N} (y_i - f(t_i))^2 + \mu \int_{\mathbb{R}} f''(t) dt \right] \tag{3.28}$$

using Matlab's built-in spline toolbox. Parameter μ was chosen based on leave-one-out cross-validation, and it is apparent that no cleansing of the load profile takes place. Indeed, the resulting fitted function follows very closely the training data, even during the abnormal energy peaks observed on the so-termed 'building operational transition shoulder periods.'



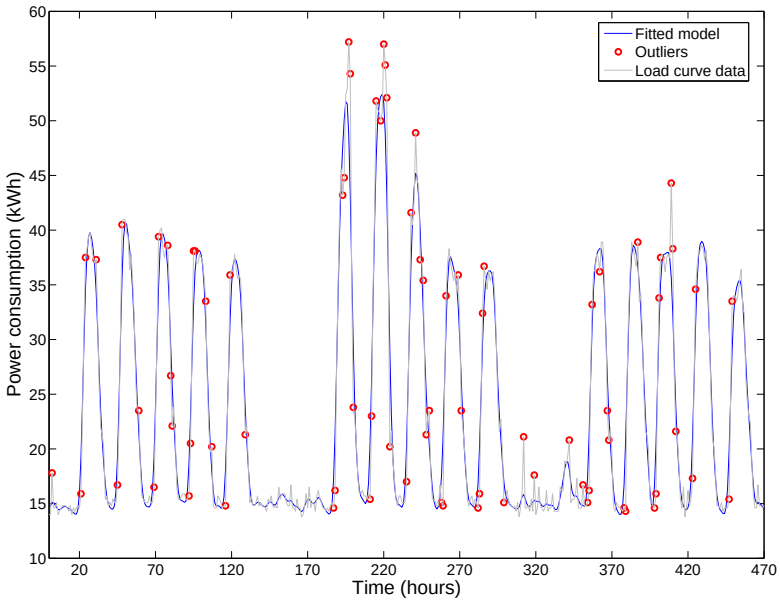
(a)



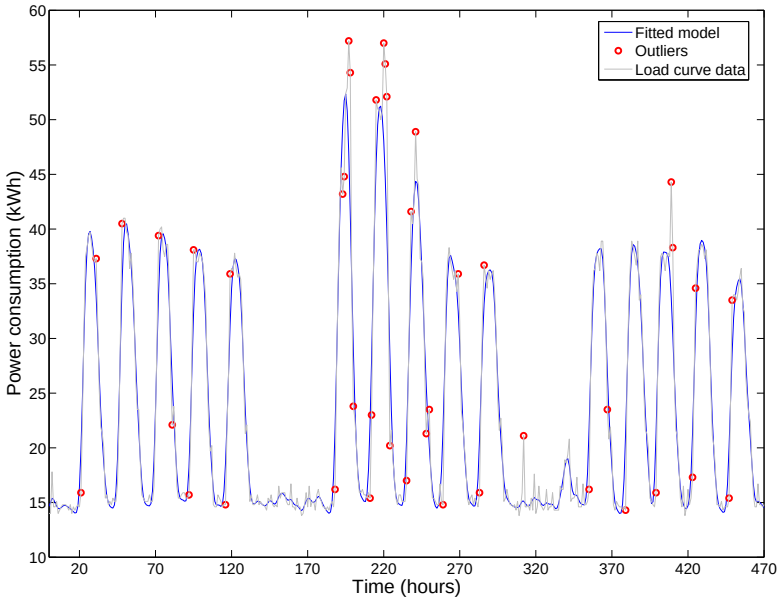
(b)

Figure 3.6: Load curve data cleansing. (a) Noisy training data and outliers; (b) fitted load profile obtained after solving (3.28).

Because with real load curve data the nominal noise variance σ_ε^2 in (3.2) is unknown, selection of the tuning parameters $\{\mu, \lambda_1\}$ in (3.27) requires a robust estimate of the variance $\hat{\sigma}_\varepsilon^2$ such as the MAD [cf. Section 3.3.2]. Similar to [25], it is assumed that the nominal errors are zero mean Gaussian distributed, so that (3.18) can be applied yielding the value $\hat{\sigma}_\varepsilon^2 = 0.6964$. To form the residuals in (3.18), (3.28) is solved first using a small subset of $\mathcal{T}$ that comprises 126 measurements. A nonuniform grid of μ and λ_1 values is constructed, as described in Section 3.3.2. Relevant parameters are $G_\mu = 100$, $G_\lambda = 200$, $\mu_{\min} = 10^{-3}$, $\mu_{\max} = 10$, and $\epsilon = 10^{-4}$. The robustification paths (one per μ value in the grid) were obtained using the SpaRSA toolbox in [125], with the sample variance matrix $\bar{\mathbf{\Sigma}}$ formed as in (3.16). The optimum tuning parameters $\mu^* = 1.637$ and $\lambda_1^* = 3.6841$ are finally determined based on the criterion (3.17), where the unknown σ_ε^2 is replaced with $\hat{\sigma}_\varepsilon^2$. Finally, the cleansed load curve is refined by running four iterations of (3.22) as described in Section 3.4, with a value of $\delta = 10^{-5}$. Results are depicted in Fig. 3.7, where the cleansed load curves are superimposed to the training data in $\mathcal{T}$. Red circles indicate those data points deemed as outliers, information that is readily obtained from the support of $\hat{\mathbf{o}}$. By inspection of Fig. 3.7, it is apparent that the proposed sparsity-controlling estimator has the desired cleansing capability. The cleansed load curves closely follow the training data, but are smooth enough to avoid overfitting the abnormal energy peaks on the 'shoulders.' Indeed, these peaks are in most cases identified as outliers. As seen from Fig. 3.7 (a), the solution of (3.27) tends to overestimate the support of $\mathbf{o}$, since one could argue that some of the red circles in Fig. 3.7 (a) do not correspond to outliers. Again, the nonconvex regularization in Section 3.4 prunes the outlier support obtained via (3.27), resulting in a more accurate result in terms of the residual fit to the data and reducing the number of outliers identified from 77 to 41.



(a)



(b)

Figure 3.7: Load curve data cleansing. (a) Cleansed load profile obtained after solving (3.27); (b) refined load profile obtained after using the nonconvex regularization in (3.19).

3.6 Summary

Outlier-robust nonparametric regression methods were developed in this chapter for function approximation in RKHS. Building on a neat link between the seemingly unrelated fields of robust statistics and sparse regression, the novel estimators were found rooted at the crossroads of outlier-resilient estimation, the Lasso, and convex optimization. Estimators as fundamental as LS for linear regression, regularization networks, and (thin-plate) smoothing splines, can be robustified under the proposed framework.

Training samples from the (unknown) target function were assumed generated from a regression model, which explicitly incorporates an unknown sparse vector of outliers. To fit such a model, the proposed variational estimator minimizes a tradeoff between fidelity to the training data, the degree of 'smoothness' of the regression function, and the sparsity level of the vector of outliers. While model complexity control effected through a smoothing penalty has quite well understood ramifications in terms of generalization capability, the major innovative claim here is that sparsity control is tantamount to robustness control. This is indeed the case since a tunable parameter in a Lasso reformulation of the variational estimator, controls the degree of sparsity in the estimated vector of model outliers. Selection of tuning parameters could be at first thought as a mundane task. However, arguing on the importance of such task in the context of robust nonparametric regression, as well as devising principled methods to effectively carry out smoothness and sparsity control, are at the heart of this chapters novelty. Sparsity control can be carried out at affordable complexity, by capitalizing on state-of-the-art algorithms that can efficiently compute the whole path of Lasso solutions. In this sense, the method here capitalizes on but is not limited to sparse settings where few outliers are present, since one can efficiently examine the gamut of sparsity levels along the robustification path. Computer simulations have shown that the novel methods of this chapter outperform existing alternatives including SVR, and one if its robust variants.

As an application domain relevant to robust nonparametric regression, the problem of load curve cleansing for power systems engineering was also considered along with a solution proposed based on robust cubic spline smoothing. Numerical tests on real load curve data

demonstrated that the smoothness and sparsity controlling methods of this chapter are effective in cleansing load profiles, without user intervention to aid the learning process.

3.7 Appendices

3.7.1 Proof of equivalence of (3.5) and (3.6)

Towards establishing the equivalence between problems (3.5) and (3.6), consider the pair $\{\hat{f}, \hat{\mathbf{o}}\}$ that solves (3.5). Assume that $\hat{f}$ is given, and the goal is to determine $\hat{\mathbf{o}}$. Upon defining the residuals $\hat{r}_i := y_i - \hat{f}(\mathbf{x}_i)$ and because $\|\mathbf{o}\|_1 = \sum_{i=1}^N |o_i|$, the entries of $\hat{\mathbf{o}}$ are separately given by

$$\hat{o}_i := \arg\min_{o_i \in \mathbb{R}} \left[(\hat{r}_i - o_i)^2 + \lambda_1 |o_i| \right], \quad i = 1, \ldots, N, \tag{3.29}$$

where the term $\mu \|\hat{f}\|_{\mathcal{H}}^2$ in (3.5) has been omitted, since it is inconsequential for the minimization with respect to $\mathbf{o}$. For each $i = 1, \ldots, N$, because (3.29) is nondifferentiable at the origin one should consider three cases: i) if $\hat{o}_i = 0$, it follows that the minimum cost in (3.29) is $\hat{r}_i^2$; ii) if $\hat{o}_i > 0$, the first-order condition for optimality gives $\hat{o}_i = \hat{r}_i - \lambda_1/2$ provided $\hat{r}_i > \lambda_1/2$, and the minimum cost is $\lambda_1 \hat{r}_i - \lambda_1^2/4$; otherwise, iii) if $\hat{o}_i < 0$, it follows that $\hat{o}_i = \hat{r}_i + \lambda_1/2$ provided $\hat{r}_i < -\lambda_1/2$, and the minimum cost is $-\lambda_1 \hat{r}_i - \lambda_1^2/4$. In other words,

$$\hat{o}_i = \begin{cases} \hat{r}_i - \lambda_1/2, & \hat{r}_i > \lambda_1/2 \\ 0, & |\hat{r}_i| \leq \lambda_1/2 \\ \hat{r}_i + \lambda_1/2, & \hat{r}_i < -\lambda_1/2 \end{cases}, \quad i = 1, \ldots, N. \tag{3.30}$$

Upon plugging (3.30) into (3.29), the minimum cost in (3.29) after minimizing with respect to o_i is $\rho(\hat{r}_i)$ [cf. (3.7) and the argument preceding (3.30)]. All in all, the conclusion is that $\hat{f}$ is the minimizer of (3.6) – in addition to being the solution of (3.5) by definition – completing the proof. ■

Chapter 4

Robust PCA as Bilinear Decomposition with Outlier-Sparsity Regularization

4.1 Introduction

Principal component analysis (PCA) is widely used for dimensionality reduction, with well-documented merits in various applications involving high-dimensional data, including computer vision, preference measurement, and bioinformatics. A least-trimmed squares estimator of a low-rank bilinear factor analysis model is shown closely related to that obtained from an ℓ_0-(pseudo)norm-regularized criterion encouraging *sparsity* in a matrix explicitly modeling the outliers. This connection suggests robust PCA schemes based on convex relaxation, which lead naturally to a family of robust estimators encompassing Huber's optimal M-class as a special case. Outliers are identified by tuning a regularization parameter, which amounts to controlling sparsity of the outlier matrix along the whole *robustification* path of (group) least-absolute shrinkage and selection operator (Lasso) solutions. Beyond its neat ties to robust statistics, the outlier-aware PCA framework of this chapter is versatile to accommodate novel and scalable algorithms to: i) track the low-rank signal subspace

robustly, as new data are acquired in real time; and ii) determine principal components robustly in (possibly) infinite-dimensional feature spaces. Synthetic and real data tests corroborate the effectiveness of the proposed robust PCA schemes, when used to identify aberrant responses in personality assessment surveys, as well as unveil communities in social networks, and intruders from video surveillance data.

4.2 Robustifying PCA

Consider the standard PCA formulation, in which a set of data $\mathcal{T}_y := \{\mathbf{y}_n\}_{n=1}^N$ in the p-dimensional Euclidean *input* space is given, and the goal is to find the best q-rank ($q \leq p$) linear approximation to the data in $\mathcal{T}_y$; see e.g., [67]. Unless otherwise stated, it is assumed throughout that the value of q is given. One approach to solving this problem, is to adopt a low-rank bilinear (factor analysis) model

$$\mathbf{y}_n = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n, \quad n = 1, \ldots, N \tag{4.1}$$

where $\mathbf{m} \in \mathbb{R}^p$ is a location (mean) vector; matrix $\mathbf{U} \in \mathbb{R}^{p \times q}$ has orthonormal columns spanning the signal subspace; $\{\mathbf{s}_n\}_{n=1}^N$ are the so-termed *principal components*, and $\{\mathbf{e}_n\}_{n=1}^N$ are zero-mean i.i.d. random errors. The unknown variables in (4.1) can be collected in $\mathcal{V} := \{\mathbf{m}, \mathbf{U}, \{\mathbf{s}_n\}_{n=1}^N\}$, and they are estimated using the LS criterion as

$$\min_{\mathcal{V}} \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2^2, \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \tag{4.2}$$

PCA in (4.2) is a nonconvex optimization problem due to the bilinear terms $\mathbf{U}\mathbf{s}_n$, yet a global optimum $\hat{\mathcal{V}}$ can be shown to exist; see e.g., [129]. The resulting estimates are $\hat{\mathbf{m}} = \sum_{n=1}^N \mathbf{y}_n/N$ and $\hat{\mathbf{s}}_n = \hat{\mathbf{U}}'(\mathbf{y}_n - \hat{\mathbf{m}})$, $n = 1, \ldots, N$; while $\hat{\mathbf{U}}$ is formed with columns equal to the q-dominant right singular vectors of the $N \times p$ data matrix $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]'$ [59, p. 535]. The principal components (entries of) $\mathbf{s}_n$ are the projections of the centered data points $\{\mathbf{y}_n - \hat{\mathbf{m}}\}_{n=1}^N$ onto the signal subspace. Equivalently, PCA can be formulated based on maximum variance, or, minimum reconstruction error criteria; see e.g., [67].

4.2.1 Least-trimmed squares PCA

Given training data $\mathcal{T}_x := \{\mathbf{x}_n\}_{n=1}^N$ possibly contaminated with outliers, the goal here is to develop a robust estimator of $\mathcal{V}$ that requires minimal assumptions on the outlier model. Note that there is an explicit notational differentiation between: i) the data in $\mathcal{T}_y$ which adhere to the nominal model (4.1); and ii) the given data in $\mathcal{T}_x$ that may also contain outliers, i.e., those $\mathbf{x}_n$ not adhering to (4.1). Building on LTS regression [104], the desired robust estimate $\hat{\mathcal{V}}_{LTS} := \{\hat{\mathbf{m}}, \hat{\mathbf{U}}, \{\hat{\mathbf{s}}_n\}_{n=1}^N\}$ for a prescribed $\nu < N$ can be obtained via the following LTS PCA estimator [cf. (4.2)]

$$\hat{\mathcal{V}}_{LTS} := \arg\min_{\mathcal{V}} \sum_{n=1}^{\nu} r_{[n]}^2(\mathcal{V}), \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{4.3}$$

where $r_{[n]}^2(\mathcal{V})$ is the n-th order statistic among the squared residual norms $r_1^2(\mathcal{V}), \ldots, r_N^2(\mathcal{V})$, and $r_n(\mathcal{V}) := \|\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2$. The so-termed *coverage* ν determines the breakdown point of the LTS PCA estimator [104], since the $N - \nu$ largest residuals are absent from the estimation criterion in (4.3). Beyond this universal outlier-rejection property, the LTS-based estimation offers an attractive alternative to robust linear regression due to its high breakdown point and desirable analytical properties, namely $\sqrt{N}$-consistency and asymptotic normality under mild assumptions [104].

Remark 4.1 (Robust estimation of the mean) In most applications of PCA, data in $\mathcal{T}_y$ are typically assumed zero mean. This is without loss of generality, since nonzero-mean training data can always be rendered zero mean, by subtracting the sample mean $\sum_{n=1}^N \mathbf{y}_n/N$ from each $\mathbf{y}_n$. In modeling zero-mean data, the known vector $\mathbf{m}$ in (4.1) can obviously be neglected. When outliers are present however, data in $\mathcal{T}_x$ are not necessarily zero mean, and it is unwise to center them using the non-robust sample mean estimator which has a breakdown point equal to zero [104]. Towards robustifying PCA, a more sensible approach is to estimate $\mathbf{m}$ robustly, and jointly with $\mathbf{U}$ and the principal components $\{\mathbf{s}_n\}_{n=1}^N$.

Because (4.3) is a nonconvex optimization problem, a nontrivial issue pertains to the existence of the proposed LTS PCA estimator, i.e., whether or not (4.3) attains a minimum.

Fortunately, the answer is in the affirmative as asserted next.

Property 4.1 *The LTS PCA estimator is well defined, since (4.3) has (at least) one solution.*

Existence of $\hat{\mathcal{V}}_{LTS}$ can be readily established as follows: i) for each subset of $\mathcal{T}$ with cardinality ν (there are $\binom{N}{\nu}$ such subsets), solve the corresponding PCA problem to obtain a unique candidate estimator per subset; and ii) pick $\hat{\mathcal{V}}_{LTS}$ as the one among all $\binom{N}{\nu}$ candidates with the minimum cost.

Albeit conceptually simple, the solution procedure outlined under Property 4.1 is combinatorially complex, and thus intractable except for small sample sizes N. Algorithms to obtain approximate LTS solutions in large-scale linear regression problems are available; see e.g., [103].

4.2.2 ℓ_0-norm regularization for robustness

Instead of discarding large residuals, the alternative approach here explicitly accounts for outliers in the low-rank data model (4.1). This becomes possible through the vector variables $\{\mathbf{o}_n\}_{n=1}^N$ one per training datum $\mathbf{x}_n$, which take the value $\mathbf{o}_n \neq \mathbf{0}_p$ whenever datum n is an outlier, and $\mathbf{o}_n = \mathbf{0}_p$ otherwise. Thus, the novel outlier-aware factor analysis model is

$$\mathbf{x}_n = \mathbf{y}_n + \mathbf{o}_n = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n + \mathbf{o}_n, \qquad n = 1, \ldots, N \tag{4.4}$$

where $\mathbf{o}_n$ can be deterministic or random with unspecified distribution. In the *underdetermined* linear system of equations (4.4), both $\mathcal{V}$ as well as the $N \times p$ matrix $\mathbf{O} := [\mathbf{o}_1, \ldots, \mathbf{o}_N]'$ are unknown. The percentage of outliers dictates the degree of *sparsity* (number of zero rows) in $\mathbf{O}$. Sparsity control will prove instrumental in efficiently estimating $\mathbf{O}$, rejecting outliers as a byproduct, and consequently arriving at a *robust* estimator of $\mathcal{V}$. To this end, a natural criterion for controlling outlier sparsity is to seek the estimator [cf. (4.2)]

$$\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\} = \arg\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X} - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \lambda_0\|\mathbf{O}\|_0, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{4.5}$$

where $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]' \in \mathbb{R}^{N \times p}$, $\mathbf{S} := [\mathbf{s}_1, \ldots, \mathbf{s}_N]' \in \mathbb{R}^{N \times q}$, and $\|\mathbf{O}\|_0$ denotes the nonconvex ℓ_0-norm that is equal to the number of nonzero rows of $\mathbf{O}$. Vector (group) sparsity in the rows $\hat{\mathbf{o}}_n$ of $\hat{\mathbf{O}}$ can be directly controlled by tuning the parameter $\lambda_0 \geq 0$.

As with compressive sampling and sparse modeling schemes that rely on the ℓ_0-norm [115], the robust PCA problem (4.5) is NP-hard [89]. In addition, the sparsity-controlling estimator (4.5) is intimately related to LTS PCA, as asserted next.

Proposition 4.1 *If* $\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\}$ *minimizes (4.5) with* λ_0 *chosen such that* $\|\hat{\mathbf{O}}\|_0 = N - \nu$*, then* $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$.

Proof: Given λ_0 such that $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the goal is to characterize $\hat{\mathcal{V}}$ as well as the positions and values of the nonzero rows of $\hat{\mathbf{O}}$. Note that because $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the last term in the cost of (4.5) is constant, hence inconsequential to the minimization. Upon defining $\hat{\mathbf{r}}_n := \mathbf{x}_n - \hat{\mathbf{m}} - \hat{\mathbf{U}}\hat{\mathbf{s}}_n$, it is not hard to see from the optimality conditions that the rows of $\hat{\mathbf{O}}$ satisfy

$$\hat{\mathbf{o}}_n = \begin{cases} \mathbf{0}_p, & \|\hat{\mathbf{r}}_n\|_2 \leq \sqrt{\lambda_0} \\ \hat{\mathbf{r}}_n, & \|\hat{\mathbf{r}}_n\|_2 > \sqrt{\lambda_0} \end{cases}, \quad n = 1, \ldots, N. \tag{4.6}$$

This is intuitive, since for those nonzero $\hat{\mathbf{o}}_n$ the best thing to do in terms of minimizing the overall cost is to set $\hat{\mathbf{o}}_n = \hat{\mathbf{r}}_n$, and thus null the corresponding squared-residual terms in (4.5). In conclusion, for the chosen value of λ_0 it holds that $N - \nu$ squared residuals effectively do not contribute to the cost in (4.5).

To determine $\hat{\mathcal{V}}$ and the row support of $\hat{\mathbf{O}}$, one alternative is to exhaustively test all $\binom{N}{N-\nu} = \binom{N}{\nu}$ admissible row-support combinations. For each one of these combinations (indexed by j), let $\mathcal{S}_j \subset \{1, \ldots, N\}$ be the index set describing the row support of $\hat{\mathbf{O}}^{(j)}$, i.e., $\hat{\mathbf{o}}_n^{(j)} \neq \mathbf{0}_p$ if and only if $n \in \mathcal{S}_j$; and $|\mathcal{S}_j| = N - \nu$. By virtue of (4.6), the corresponding candidate $\hat{\mathcal{V}}^{(j)}$ solves $\min_{\mathcal{V}} \sum_{n \in \mathcal{S}_j} r_n^2(\mathcal{V})$ subject to $\mathbf{U}'\mathbf{U} = \mathbf{I}_q$, while $\hat{\mathcal{V}}$ is the one among all $\{\hat{\mathcal{V}}^{(j)}\}$ that yields the least cost. Recognizing the aforementioned solution procedure as the one for LTS PCA outlined under Property 4.1, it follows that $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$. ■

The importance of Proposition 4.1 is threefold. First, it formally justifies model (4.4) and its estimator (4.5) for robust PCA, in light of the well documented merits of LTS [104].

Second, it further solidifies the connection between sparsity-aware learning and robust estimation. Third, problem (4.5) lends itself naturally to efficient (approximate) solvers based on convex relaxation, the subject dealt with next.

4.3 Sparsity-Controlling Outlier Rejection

Recall that the row-wise ℓ_2-norm sum $\|\mathbf{B}\|_{2,r} := \sum_{n=1}^{N} \|\mathbf{b}_n\|_2$ of matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_N]' \in \mathbb{R}^{N\times p}$ is the closest convex approximation of $\|\mathbf{B}\|_0$. This property motivates relaxing problem (4.5) to

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X} - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \lambda_2\|\mathbf{O}\|_{2,r}, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \tag{4.7}$$

The nondifferentiable ℓ_2-norm regularization term encourages row-wise (vector) sparsity on the estimator of $\mathbf{O}$, a property that has been exploited in diverse problems in engineering, statistics, and machine learning [59]. A noteworthy representative is the group Lasso [130], a popular tool for joint estimation and selection of grouped variables in linear regression.

It is pertinent to ponder on whether problem (4.7) still has the potential of providing robust estimates $\hat{\mathcal{V}}$ in the presence of outliers. The answer is positive, since it is shown in the Appendix that (4.7) is equivalent to an M-type estimator

$$\min_{\mathcal{V}} \sum_{n=1}^{N} \rho_v(\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n), \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{4.8}$$

where $\rho_v : \mathbb{R}^p \to \mathbb{R}$ is a vector extension to Huber's convex loss function [63]; see also [69], and

$$\rho_v(\mathbf{r}) := \begin{cases} \|\mathbf{r}\|_2^2, & \|\mathbf{r}\ \|_2 \le \lambda_2/2 \\ \lambda_2\|\mathbf{r}\|_2 - \lambda_2^2/4, & \|\mathbf{r}\|_2 > \lambda_2/2 \end{cases}. \tag{4.9}$$

M-type estimators (including Huber's) adopt a fortiori an ϵ-contaminated probability distribution for the outliers, and rely on minimizing the *asymptotic* variance of the resultant estimator for the least favorable distribution of the ϵ-contaminated class (asymptotic min-max approach) [63]. The assumed degree of contamination specifies the tuning parameter λ_2 in (4.9) (and thus the threshold for deciding the outliers in M-estimators). In contrast, the present approach is universal in the sense that it is not confined to any assumed class

of outlier distributions, and can afford a data-driven selection of the tuning parameter. In a nutshell, M-estimators can be viewed as a special case of the present formulation only for a specific choice of λ_2, which is not obtained via a data-driven approach, but from distributional assumptions instead.

All in all, the sparsity-controlling role of the tuning parameter $\lambda_2 \geq 0$ in (4.7) is central, since model (4.4) and the equivalence of (4.7) with (4.8) suggest that λ_2 is a robustness-controlling constant. Data-driven approaches to select λ_2 are described in detail under Section 4.3.2. Before dwelling into algorithmic issues to solve (4.7), a couple of remarks are in order.

Remark 4.2 (ℓ_1-norm regularization for entry-wise outliers) In computer vision applications where robust PCA schemes are particularly attractive, one may not wish to discard the entire (vectorized) images $\mathbf{x}_n$, but only specific pixels deemed as outliers [31]. This can be accomplished by replacing $\|\mathbf{O}\|_{2,r}$ in (4.7) with $\|\mathbf{O}\|_1 := \sum_{n=1}^{N} \|\mathbf{o}_n\|_1$, a Lasso-type regularization that encourages entry-wise sparsity in $\hat{\mathbf{O}}$.

Remark 4.3 (Outlier rejection) From the equivalence between problems (4.7) and (4.8), it follows that those data points $\mathbf{x}_n$ deemed as containing outliers ($\hat{\mathbf{o}}_n \neq \mathbf{0}_p$) are not completely discarded from the estimation process. Instead, their effect is downweighted as per Huber's loss function [cf. (4.9)]. Nevertheless, explicitly accounting for the outliers in $\hat{\mathbf{O}}$ provides the means of identifying and removing the contaminated data altogether, and thus possibly re-running PCA on the outlier-free data.

4.3.1 Solving the relaxed problem

To optimize (4.7) iteratively for a given value of λ_2, an alternating minimization (AM) algorithm is adopted which cyclically updates $\mathbf{m}(k) \to \mathbf{S}(k) \to \mathbf{U}(k) \to \mathbf{O}(k)$ per iteration $k = 1, 2, \ldots$. AM algorithms are also known as block-coordinate-descent methods in the optimization parlance; see e.g., [11, 116]. To update each of the variable groups, (4.7) is minimized while fixing the rest of the variables to their most up-to-date values. While the overall problem (4.7) is not jointly convex with respect to (w.r.t.) $\{\mathbf{S}, \mathbf{U}, \mathbf{O}, \mathbf{m}\}$, fixing all

but one of the variable groups yields subproblems that are efficiently solved, and attain a unique solution.

Towards deriving the updates at iteration k and arriving at the desired algorithm, note first that the mean update is $\mathbf{m}(k) = (\mathbf{X}-\mathbf{O}(k))'\mathbf{1}_N/N$. Next, form the centered and outlier-compensated data matrix $\mathbf{X}_o(k) := \mathbf{X} - \mathbf{1}_N\mathbf{m}(k)' - \mathbf{O}(k-1)$. The principal components are readily given by

$$\mathbf{S}(k) = \arg\min_{\mathbf{S}} \|\mathbf{X}_o(k) - \mathbf{S}\mathbf{U}(k-1)'\|_F^2 = \mathbf{X}_o(k)\mathbf{U}(k-1).$$

Continuing the cycle, $\mathbf{U}(k)$ solves

$$\min_{\mathbf{U}} \|\mathbf{X}_o(k) - \mathbf{S}(k)\mathbf{U}'\|_F^2, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q$$

a constrained LS problem also known as reduced-rank *Procrustes rotation* [134]. The minimizer is given in analytical form in terms of the left and right singular vectors of $\mathbf{X}_o'(k)\mathbf{S}(k)$ [134, Thm. 4]. In detail, one computes the SVD of $\mathbf{X}_o'(k)\mathbf{S}(k) = \mathbf{L}(k)\mathbf{D}(k)\mathbf{R}'(k)$ and updates $\mathbf{U}(k) = \mathbf{L}(k)\mathbf{R}'(k)$. Next, the minimization of (4.7) w.r.t. $\mathbf{O}$ is an orthonormal group Lasso problem. As such, it decouples across rows $\mathbf{o}_n$ giving rise to N ℓ_2-norm regularized subproblems, namely

$$\mathbf{o}_n(k) = \arg\min_{\mathbf{o}} \|\mathbf{r}_n(k) - \mathbf{o}\|_2^2 + \lambda_2\|\mathbf{o}\|_2, \quad n = 1, \ldots, N$$

where $\mathbf{r}_n(k) := \mathbf{x}_n - \mathbf{m}(k) - \mathbf{U}(k)\mathbf{s}_n(k)$. The respective solutions are given by (see e.g., [93])

$$\mathbf{o}_n(k) = \frac{\mathbf{r}_n(k)(\|\mathbf{r}_n(k)\|_2 - \lambda_2/2)_+}{\|\mathbf{r}_n(k)\|_2}, \quad n = 1, \ldots, N \tag{4.10}$$

where $(\cdot)_+ := \max(\cdot, 0)$. For notational convenience, these N parallel vector soft-thresholded updates are denoted as $\mathbf{O}(k) = \mathcal{S}\left[\mathbf{X} - \mathbf{1}_N\mathbf{m}'(k-1) - \mathbf{S}(k)\mathbf{U}'(k), (\lambda_2/2)\mathbf{I}_N\right]$ under Algorithm 4, where the thresholding operator $\mathcal{S}$ sets the entire outlier vector $\mathbf{o}_n(k)$ to zero whenever $\|\mathbf{r}_n(k)\|_2$ does not exceed $\lambda_2/2$, in par with the group sparsifying property of group Lasso. Interestingly, this is the same rule used to decide if datum $\mathbf{x}_n$ is deemed an outlier, in the equivalent formulation (4.8) which involves Huber's loss function. Whenever an ℓ_1-norm regularizer is adopted as discussed in Remark 4.2, the only difference is that updates (4.10) boil down to soft-thresholding the scalar entries of $\mathbf{r}_n(k)$.

Algorithm 4 : Batch robust PCA solver

Set $\mathbf{U}(0) = \mathbf{I}_p(:, 1:q)$ and $\mathbf{O}(0) = \mathbf{0}_{N\times p}$.

for $k = 1, 2, \ldots$ **do**

Update $\mathbf{m}(k) = (\mathbf{X} - \mathbf{O}(k-1))'\mathbf{1}_N/N$.

Form $\mathbf{X}_o(k) = \mathbf{X} - \mathbf{1}_N\mathbf{m}'(k) - \mathbf{O}(k-1)$.

Update $\mathbf{S}(k) = \mathbf{X}_o(k)\mathbf{U}(k-1)$.

Obtain $\mathbf{L}(k)\mathbf{D}(k)\mathbf{R}(k)' = \text{svd}[\mathbf{X}_o'(k)\mathbf{S}(k)]$ and update $\mathbf{U}(k) = \mathbf{L}(k)\mathbf{R}'(k)$.

Update $\mathbf{O}(k) = \mathcal{S}\left[\mathbf{X} - \mathbf{1}_N\mathbf{m}'(k) - \mathbf{S}(k)\mathbf{U}'(k), (\lambda_2/2)\mathbf{I}_N\right]$.

end for

The entire AM solver is tabulated under Algorithm 4, indicating also the recommended initialization. Algorithm 4 is conceptually interesting, since it explicitly reveals the intertwining between the outlier identification process, and the PCA low-rank model fitting based on the outlier compensated data $\mathbf{X}_o(k)$.

The AM solver is also computationally efficient. Computing the $N \times q$ matrix $\mathbf{S}(k) = \mathbf{X}_o(k)\mathbf{U}(k-1)$ requires Npq operations per iteration, and equally costly is to obtain $\mathbf{X}_o'(k)\mathbf{S}(k) \in \mathbb{R}^{p\times q}$. The cost of computing the SVD of $\mathbf{X}_o'(k)\mathbf{S}(k)$ is of order $\mathcal{O}(pq^2)$, while the rest of the operations including the row-wise soft-thresholdings to yield $\mathbf{O}(k)$ are linear in both N and p. In summary, the total cost of Algorithm 4 is roughly $k_{\max}\mathcal{O}(Np + pq^2)$, where $k_{\max}$ is the number of iterations required for convergence (typically $k_{\max} = 5$ to 10 iterations suffice). Because $q \leq p$ is typically small, Algorithm 4 is attractive computationally both under the classic setting where $N > p$, and p is not large; as well as in high-dimensional data settings where $p \gg N$, a situation typically arising e.g., in microarray data analysis.

Because each of the optimization problems in the per-iteration cycles has a unique minimizer, and the nondifferentiable regularization only affects one of the variable groups ($\mathbf{O}$), the general results of [116] apply to establish convergence of Algorithm 4 as follows.

Proposition 4.2 *As $k \to \infty$, the iterates generated by Algorithm 4 converge to a stationary point of (4.7).*

4.3.2 Selection of λ_2: robustification paths

Selecting λ_2 controls the number of outliers rejected. But this choice is challenging because existing techniques such as cross-validation are not effective when outliers are present [104]. To this end, systematic data-driven approaches were devised in [49], which e.g., require a rough estimate of the percentage of outliers, or, robust estimates $\hat{\sigma}_e^2$ of the nominal noise variance that can be obtained using median absolute deviation (MAD) schemes [63]. These approaches can be adapted to the robust PCA setting considered here, and leverage the *robustification paths* of (group-)Lasso solutions [cf. (4.7)], which are defined as the solution paths corresponding to $\|\hat{\mathbf{o}}_n\|_2$, $n = 1, \ldots, N$, for all values of λ_2. As λ_2 decreases, more vectors $\hat{\mathbf{o}}_n$ enter the model signifying that more of the training data are deemed to contain outliers.

Consider then a grid of G_λ values of λ_2 in the interval $[\lambda_{\min}, \lambda_{\max}]$, evenly spaced on a logarithmic scale. Typically, $\lambda_{\max}$ is chosen as the minimum λ_2 value such that $\hat{\mathbf{O}} \neq \mathbf{0}_{N\times p}$, while $\lambda_{\min} = \epsilon\lambda_{\max}$ with $\epsilon = 10^{-4}$, say. Because Algorithm 4 converges quite fast, (4.7) can be efficiently solved over the grid of G_λ values for λ_2. In the order of hundreds of grid points can be easily handled by initializing each instance of Algorithm 1 (per value of λ_2) using *warm starts* [59]. This means that multiple instances of (4.7) are solved for a sequence of decreasing λ_2 values, and the initialization of Algorithm 4 per grid point corresponds to the solution obtained for the immediately preceding value of λ_2 in the grid. For sufficiently close values of λ_2, one expects that the respective solutions will also be close (the row support of $\hat{\mathbf{O}}$ will most likely not change), and hence Algorithm 1 will converge after few iterations.

Based on the G_λ samples of the robustification paths and the prior knowledge available on the outlier model (4.4), a couple of alternatives are also possible for selecting the 'best' value of λ_2 in the grid. A comprehensive survey of options can be found in [49].

Number of outliers is known: By direct inspection of the robustification paths one can determine the range of values for λ_2, such that the number of nonzero rows in $\hat{\mathbf{O}}$ equals the known number of outliers sought. Zooming-in to the interval of interest, and after discarding the identified outliers, K-fold cross-validation methods can be applied to determine the 'best' λ_2^*.

Nominal noise covariance matrix is known: Given $\mathbf{\Sigma}_e := E[\mathbf{e}_n\mathbf{e}_n']$, one can proceed as follows. Consider the estimates $\hat{\mathcal{V}}_g$ obtained using (4.7) after sampling the robustification path for each point $\{\lambda_{2,g}\}_{g=1}^G$. Next, pre-whiten those residuals corresponding to training data not deemed as containing outliers; i.e., form $\hat{\mathcal{R}}_g := \{\bar{\mathbf{r}}_{n,g} = \mathbf{\Sigma}_e^{-1/2}(\mathbf{x}_n - b\hat{b}m_g - \hat{\mathbf{U}}_g\hat{\mathbf{s}}_{n,g}) : n \text{ s. to } \hat{\mathbf{o}}_n = \mathbf{0}\}$, and find the sample covariance matrices $\{\hat{\mathbf{\Sigma}}_{\bar{r},g}\}_{g=1}^G$. The winner $\lambda_2^* := \lambda_{2,g^*}$ corresponds to the grid point minimizing an absolute variance deviation criterion, namely $g^* := \arg\min_g |\mathrm{tr}[\hat{\mathbf{\Sigma}}_{\bar{r},g}] - p|$.

4.3.3 Connections with robust linear regression, dictionary learning, and clustering

Previous efforts towards robustifying linear regression have pointed out the equivalence between M-type estimators and ℓ_1-norm regularized regression [46], and capitalized on this neat connection under a Bayesian framework [64]. However, they have not recognized the link to LTS via convex relaxation of the ℓ_0-norm in (4.5). The treatment here goes beyond linear regression by considering the PCA framework, which entails a more challenging bilinear factor analysis model. Linear regression is subsumed as a special case, when matrix $\mathbf{U}$ is not necessarily tall but *assumed known*, while $\mathbf{s}_n = \mathbf{s}$, $\forall\, n = 1, \ldots, N$.

As an alternative to PCA, it is possible to device dimensionality reduction schemes when the data admit a sparse representation over a perhaps *unknown* basis. Such sparse representations comprise only a few elements (atoms) of the overcomplete basis (a.k.a. dictionary) to reconstruct the original data record. Thus, each datum is represented by a coefficient vector whose effective dimensionality (number of nonzero coefficients) is smaller than that of the original data vector. Recently, the *dictionary learning* paradigm offers techniques to design a dictionary over which the data assume a sparse representation; see e.g., [114] for a tutorial treatment. Dictionary learning schemes are flexible, in the sense that they utilize training data to learn an appropriate overcomplete basis customized for the data at hand [77, 114].

However, as in PCA the criteria adopted typically rely on a squared-error loss function as a measure of fit, which is known to be very sensitive to outliers [63, 104]. Interestingly,

one can conceivably think of robustifying dictionary learning via minor modifications to the framework described so far. For instance, with the same matrix notation used in e.g., (4.5), one seeks to minimize

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X}-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2+\lambda_1\|\mathbf{S}\|_1+\lambda_2\|\mathbf{O}\|_{2,r}. \tag{4.11}$$

Different from the low-rank outlier-aware model adopted for PCA [cf. (4.4)], here the dictionary $\mathbf{U}\in\mathbb{R}^{p\times q}$ is fat ($q\gg p$), with column vectors that are no longer orthogonal but still constrained to have unit ℓ_2-norm. (This constraint is left implicit in (4.11) for simplicity.) Moreover, one seeks a sparse vector $\mathbf{s}_n$ to represent each datum $\mathbf{x}_n$, in terms of a few atoms of the learnt dictionary $\hat{\mathbf{U}}$. This is why (4.11) includes an additional sparsity-promoting ℓ_1-norm regularization on $\mathbf{S}$, that is not present in (4.7). Sparsity is thus present both in the representation coefficients $\mathbf{S}$, as well as in the outliers $\mathbf{O}$.

Finally, it is shown here that a generative data model for K-means clustering [59] can share striking similarities with the bilinear model (4.1). Consequently, the sparsity-controlling estimator (4.7) can be adapted to robustify the K-means clustering task too [43]. Consider for instance that the data in $\mathcal{T}_x$ come from q clusters, each of which is represented by a centroid $\mathbf{u}_i\in\mathbb{R}^p$, $i=1,\ldots,q$. Moreover, for each input vector $\mathbf{x}_n$, K-means introduces the unknown membership variables $s_{ni}\in\{0,1\}$, $i=1,\ldots,q$, where $s_{ni}=1$ whenever $\mathbf{x}_n$ comes from cluster i, and $s_{ni}=0$ otherwise. Typically, the membership variables are also constrained to satisfy $\sum_{n=1}^N s_{ni}>0\ \forall\, i$ (no empty clusters), and $\sum_{i=1}^q s_{ni}=1\ \forall\, n$ (single cluster membership). Upon defining $\mathbf{U}:=[\mathbf{u}_1,\ldots,\mathbf{u}_q]\in\mathbb{R}^{p\times q}$ and the membership vectors $\mathbf{s}_n:=[s_{n1},\ldots,s_{nq}]'\in\mathbb{R}^q$, a pertinent model for hard K-means clustering assumes that input vectors can be expressed as $\mathbf{x}_n=\mathbf{U}\mathbf{s}_n+\mathbf{e}_n+\mathbf{o}_n$, where $\mathbf{e}_n$ and $\mathbf{o}_n$ are as in (4.4). Because the aforementioned constraints imply $\|\mathbf{s}_n\|_0=\|\mathbf{s}_n\|_1=1\ \forall n$, if $\mathbf{x}_n$ belongs to cluster i, then $s_{ni}=1$ and in the absence of outliers one effectively has $\mathbf{x}_n=\mathbf{u}_i+\mathbf{e}_n$. Based on this data model, a natural approach towards robustifying K-means clustering solves [43]

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X}-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2+\lambda_2\|\mathbf{O}\|_{2,r},\quad \text{s. to } s_{ni}\in\{0,1\},\ \sum_{n=1}^N s_{ni}>0,\ \sum_{i=1}^q s_{ni}=1. \tag{4.12}$$

Recall that in the robust PCA estimator (4.7), the subspace matrix is required to be orthonormal and the principal components are unrestrained. In the clustering context how-

ever, the centroid columns of $\mathbf{U}$ are free optimization variables, whereas the cluster membership variables adhere to the constraints in (4.12). Suitable relaxations to tackle the NP-hard problem (4.12) have been investigated in [43].

4.4 Further Algorithmic Issues

4.4.1 Bias reduction through nonconvex regularization

Instead of substituting $\|\mathbf{O}\|_0$ in (4.5) by its closest convex approximation, namely $\|\mathbf{O}\|_{2,r}$, letting the surrogate function to be nonconvex can yield tighter approximations, and improve the statistical properties of the estimator. In rank minimization problems for instance, the logarithm of the determinant of the unknown matrix has been proposed as a smooth surrogate to the rank [40]; an alternative to the convex nuclear norm in e.g., [95]. Nonconvex penalties such as the smoothly clipped absolute deviation (SCAD) have been also adopted to reduce bias [38], present in uniformly weighted ℓ_1-norm regularized estimators such as (4.7) [59, p. 92]. In the context of sparse signal reconstruction, the ℓ_0-norm of a vector was surrogated in [23] by the logarithm of the geometric mean of its elements; see also [94].

Building on this last idea, consider approximating (4.5) by the *nonconvex* formulation

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X}-\mathbf{1}_N\mathbf{m}'-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2+\lambda_0\sum_{n=1}^{N}\log(\|\mathbf{o}_n\|_2+\delta), \quad \text{s. to } \mathbf{U}'\mathbf{U}=\mathbf{I}_q \tag{4.13}$$

where the small positive constant δ is introduced to avoid numerical instability. Since the surrogate term in (4.13) is concave, the overall minimization problem is nonconvex and admittedly more complex to solve than (4.7). Local methods based on iterative linearization of $\log(\|\mathbf{o}_n\|_2+\delta)$ around the current iterate $\mathbf{o}_n(k)$, are adopted to minimize (4.13). Skipping details that can be found in [69], application of the majorization-minimization technique to (4.13) leads to an iteratively-reweighted version of (4.7), whereby $\lambda_2 \leftarrow \lambda_0 w_n(k)$ is used for updating $\mathbf{o}_n(k)$ in Algorithm 4. Specifically, per $k=1,2,\ldots$ one updates

$$\mathbf{O}(k)=\mathcal{S}\left[\mathbf{X}-\mathbf{1}_N\mathbf{m}'(k-1)-\mathbf{S}(k)\mathbf{U}'(k),(\lambda_0/2)\text{diag}(w_1(k),\ldots,w_N(k))\right]$$

where the weights are given by $w_n(k) = (\|\mathbf{o}_n(k-1)\|_2 + \delta)^{-1}$, $n = 1, \ldots, N$. Note that the thresholds vary both across rows (indexed by n), and across iterations. If the value of $\|\mathbf{o}_n(k-1)\|_2$ is small, then in the next iteration the regularization term $\lambda_0 w_n(k)\|\mathbf{o}_n\|_2$ has a large weight, thus promoting shrinkage of that entire row vector to zero. If $\|\mathbf{o}_n(k-1)\|_2$ is large, the cost in the next iteration downweighs the regularization, and places more importance to the LS component of the fit.

All in all, the idea is to start from the solution of (4.7) for the 'best' λ_2, which is obtained using Algorithm 4. This initial estimate is refined after runnning a few iterations of the iteratively-reweighted counterpart to Algorithm 4. Extensive numerical tests suggest that even a couple iterations of this second stage refinement suffices to yield improved estimates $\hat{\mathcal{V}}$, in comparison to those obtained from (4.7). The improvements can be leveraged to bias reduction – and its positive effect with regards to outlier support estimation – also achieved by similar *weighted* norm regularizers proposed for linear regression [59, p. 92].

4.4.2 Automatic rank determination: from nuclear- to Frobenius-norm regularization

Recall that $q \leq p$ is the dimensionality of the subspace where the outlier-free data (4.1) are assumed to live in, or equivalently, $q = \text{rank}[\mathbf{Y}]$ in the absence of noise. So far, q was assumed known and fixed. This is reasonable in e.g., compression/quantization, where a target distortion-rate tradeoff dictates the maximum q. In other cases, the physics of the problem may render q known. This is indeed the case in array processing for direction-of-arrival estimation, where q is the dimensionality of the so-termed *signal subspace*, and is given by the number of plane waves impinging on a uniform linear array; see e.g., [129].

Other applications however, call for signal processing tools that can determine the 'best' q, as well as robustly estimate the underlying low-dimensional subspace $\mathbf{U}$ from data $\mathbf{X}$. Noteworthy representatives for this last kind of problems include unveiling traffic volume anomalies in large-scale networks [79], and automatic intrusion detection from video surveillance frames [20, 31], just to name a few. A related approach in this context is (stable)

principal components pursuit (PCP) [127, 131], which solves

$$\min_{\mathbf{L},\mathbf{O}} \|\mathbf{X}-\mathbf{L}-\mathbf{O}\|_F^2 + \lambda_*\|\mathbf{L}\|_* + \lambda_2\|\mathbf{O}\|_{2,r} \tag{4.14}$$

with the objective of reconstructing the low-rank matrix $\mathbf{L} \in \mathbb{R}^{N\times p}$, as well as the sparse matrix of outliers $\mathbf{O}$ in the presence of dense noise with known variance.[1] Note that $\|\mathbf{L}\|_*$ denotes the matrix nuclear norm, defined as the sum of the singular values of $\mathbf{L}$. The same way that the ℓ_2-norm regularization promotes sparsity in the rows of $\hat{\mathbf{O}}$, the nuclear norm encourages a low-rank $\hat{\mathbf{L}}$ since it effects sparsity in the vector of singular values of $\mathbf{L}$. Upon solving the convex optimization problem (4.14), it is possible to obtain $\hat{\mathbf{L}} = \hat{\mathbf{S}}\hat{\mathbf{U}}'$ using the SVD. Interestingly, (4.14) does not fix (or require the knowledge of) rank[$\mathbf{L}$] a fortiori, but controls it through the tuning parameter λ_*. Adopting a Bayesian framework, a similar problem was considered in [32].

Instead of assuming that q is known, suppose that only an upper bound $\bar{q}$ is given. Then, the class of feasible noise-free low-rank matrix components of $\mathbf{Y}$ in (4.1) admit a factorization $\mathbf{L} = \mathbf{S}\mathbf{U}'$, where $\mathbf{S}$ and $\mathbf{U}$ are $N \times \bar{q}$ and $p \times \bar{q}$ matrices, respectively. Building on the ideas used in the context of finding minimum rank solutions of linear matrix equations [95], a novel alternative approach to robustifying PCA is to solve

$$\min_{\mathbf{U},\mathbf{S},\mathbf{O}} \|\mathbf{X}-\mathbf{S}\mathbf{U}'-\mathbf{O}\|_F^2 + \frac{\lambda_*}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2) + \lambda_2\|\mathbf{O}\|_{2,r}. \tag{4.15}$$

Different from (4.14) and (4.7), a Frobenius-norm regularization on both $\mathbf{U}$ and $\mathbf{S}$ is adopted to control the dimensionality of the estimated subspace $\hat{\mathbf{U}}$. Relative to (4.7), $\mathbf{U}$ in (4.15) is not constrained to be orthonormal. It is certainly possible to include the mean vector $\mathbf{m}$ in the cost of (4.15), as well as an ℓ_1-norm regularization for entrywise outliers. The main motivation behind choosing the Frobenius-norm regularization comes from the equivalence of (4.14) with (4.15), as asserted in the ensuing result which adapts [95, Lemma 5.1] to the problem formulation considered here.

Lemma 4.1 *If* $\{\hat{\mathbf{L}}, \hat{\mathbf{O}}\}$ *minimizes (4.14) and rank*$[\hat{\mathbf{L}}] \leq \bar{q}$*, then (4.14) and (4.15) are equivalent.*

[1] Actually, [131] considers entrywise outliers and adopts an ℓ_1-norm regularization on $\mathbf{O}$.

Algorithm 5 : Batch robust PCA solver with controllable rank

Set $\mathbf{O}(0)=\mathbf{0}_{N\times p}$, and randomly initialize $\mathbf{S}(0)$.

for $k=1,2,\ldots$ **do**

Update $\mathbf{m}(k)=[\mathbf{X}-\mathbf{O}(k-1)]'\mathbf{1}_N/N$.

Form $\mathbf{X}_o(k)=\mathbf{X}-\mathbf{1}_N\mathbf{m}'(k)-\mathbf{O}(k-1)$.

Update $\mathbf{U}(k)=\mathbf{X}_o(k)'\mathbf{S}(k-1)[\mathbf{S}'(k-1)\mathbf{S}(k-1)+(\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.

Update $\mathbf{S}(k)=\mathbf{X}_o(k)\mathbf{U}(k)[\mathbf{U}'(k)\mathbf{U}(k)+(\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.

Update $\mathbf{O}(k)=\mathcal{S}\left[\mathbf{X}-\mathbf{S}(k)\mathbf{U}'(k),\lambda_2/2\right]$.

end for

Proof: Because $\mathrm{rank}[\hat{\mathbf{L}}]\leq\bar{q}$, the relevant feasible subset of (4.14) can be re-parametrized as $\{\mathbf{S}\mathbf{U}',\mathbf{O}\}$, where $\mathbf{S}$ and $\mathbf{U}$ are $N\times\bar{q}$ and $p\times\bar{q}$ matrices, respectively. For every triplet $\{\mathbf{U},\mathbf{S},\mathbf{O}\}$ the objective of (4.15) is no smaller than the one of (4.14), since it holds that [95]

$$\|\mathbf{L}\|_*=\min_{\mathbf{U},\mathbf{S}}\frac{1}{2}(\|\mathbf{U}\|_F^2+\|\mathbf{S}\|_F^2),\quad \text{s. to } \mathbf{L}=\mathbf{S}\mathbf{U}'. \tag{4.16}$$

One can show that the gap between the objectives of (4.14) and (4.15) vanishes at $\mathbf{O}^*:=\hat{\mathbf{O}}$, $\mathbf{S}^*:=\mathbf{U}_L\mathbf{\Sigma}^{1/2}$, and $\mathbf{U}^*:=\mathbf{V}_L\mathbf{\Sigma}^{1/2}$; where $\hat{\mathbf{L}}=\mathbf{U}_L\mathbf{\Sigma}\mathbf{V}_L'$ is the SVD of $\hat{\mathbf{L}}$. Therefore, from the previous arguments it follows that (4.14) and (4.15) attain the same global minimum objective, which completes the proof. ■

Even though problem (4.15) is nonconvex, the number of optimization variables is reduced from $2Np$ to $Np+(N+p)\bar{q}$, which becomes significant when $\bar{q}$ is in the order of a few dozens and both N and p are large. Also note that the dominant Np-term in the variable count of (4.15) is due to $\mathbf{O}$, which is sparse and can be efficiently handled. While the factorization $\mathbf{L}=\mathbf{S}\mathbf{U}'$ could have also been introduced in (4.14) to reduce the number of unknowns, the cost in (4.15) is separable and much simpler to optimize using e.g., an AM solver comprising the iterations tabulated as Algorithm 5. The decomposability of the Frobenius-norm regularizer has been recently exploited for parallel processing across multiple processors when solving large-scale matrix completion problems [96], or to unveil network anomalies [79].

Because (4.15) is a nonconvex optimization problem, most solvers one can think of will at most provide convergence guarantees to a stationary point that may not be globally

optimum. Nevertheless, simulation results in Section 4.7 demonstrate that Algorithm 5 is effective in providing good solutions most of the time, which is somehow expected since there is quite a bit of structure in (4.15). Formally, the next proposition adapted from [79, Prop. 1] provides a sufficient condition under which Algorithm 5 yields an optimal solution of (4.14). For a proof of a slightly more general result, see [79].

Proposition 4.3 *If* $\{\bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{O}}\}$ *is a stationary point of (4.15) and* $\|\mathbf{X} - \bar{\mathbf{S}}\bar{\mathbf{U}}' - \bar{\mathbf{O}}\|_2 \leq \lambda_*/2$, *then* $\{\hat{\mathbf{L}} := \bar{\mathbf{S}}\bar{\mathbf{U}}', \hat{\mathbf{O}} := \bar{\mathbf{O}}\}$ *is the optimal solution of (4.14).*

4.5 Robust Subspace Tracking

E-commerce and Internet-based retailing sites, the World Wide Web, and video surveillance systems generate huge volumes of data, which far outweigh the ability of modern computers to analyze them in real time. Furthermore, data are generated sequentially in time, which motivates updating previously obtained learning results rather than re-computing new ones from scratch each time a new datum becomes available. This calls for low-complexity real-time (adaptive) algorithms for robust subspace tracking.

One possible adaptive counterpart to (4.7) is the exponentially-weighted LS (EWLS) estimator found by

$$\min_{\{\mathcal{V},\mathbf{O}\}} \sum_{n=1}^{N} \beta^{N-n} \left[\|\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n - \mathbf{o}_n\|_2^2 + \lambda_2 \|\mathbf{o}_n\|_2\right] \tag{4.17}$$

where $\beta \in (0, 1]$ is a forgetting factor. In this context, n should be understood as a temporal variable, indexing the instants of data acquisition. Note that in forming the EWLS estimator (4.17) at time N, the entire history of data $\{\mathbf{x}_n\}_{n=1}^N$ is incorporated in the real-time estimation process. Whenever $\beta < 1$, past data are exponentially discarded thus enabling operation in nonstationary environments. Adaptive estimation of sparse signals has been considered in e.g., [5] and [77].

Towards deriving a real-time, computationally efficient, and recursive (approximate) solver of (4.17), an AM scheme will be adopted in which iterations k coincide with the time scale $n = 1, 2, \ldots$ of data acquisition. Per time instant n, a new datum $\mathbf{x}_n$ is drawn and the

corresponding pair of decision variables $\{\mathbf{s}(n), \mathbf{o}(n)\}$ are updated via

$$\{\mathbf{s}(n), \mathbf{o}(n)\} := \arg\min_{\{\mathbf{s},\mathbf{o}\}} \|\mathbf{x}_n - \mathbf{m}(n-1) - \mathbf{U}(n-1)\mathbf{s} - \mathbf{o}\|_2^2 + \lambda_2\|\mathbf{o}\|_2. \tag{4.18}$$

As per (4.18), only $\mathbf{o}(n)$ is updated at time n, rather than the whole (growing with time) matrix $\mathbf{O}$ that minimization of (4.17) would dictate; see also [77] for a similar approximation.

Because (4.18) is a smooth optimization problem w.r.t. $\mathbf{s}$, from the first-order optimality condition the principal component update is $\mathbf{s}(n) = \mathbf{U}'(n-1)[\mathbf{x}_n - \mathbf{m}(n-1) - \mathbf{o}(n)]$. Interestingly, this resembles the projection approximation adopted in [129], and can only be evaluated after $\mathbf{o}(n)$ is obtained. To this end, plug $\mathbf{s}(n)$ in (4.18) to obtain $\mathbf{o}(n)$ via a particular instance of the group Lasso estimator

$$\mathbf{o}(n) = \arg\min_{\mathbf{o}} \|[\mathbf{I}_p - \mathbf{U}(n-1)\mathbf{U}'(n-1)](\mathbf{x}_n - \mathbf{m}(n-1) - \mathbf{o})\|_2^2 + \lambda_2\|\mathbf{o}\|_2 \tag{4.19}$$

with a single group of size equal to p. The cost in (4.19) is non-differentiable at the origin, and different from e.g., ridge regression, it does not admit a closed-form solution. Upon defining

$$\mathbf{H}(n) := 2[\mathbf{I}_p - \mathbf{U}(n-1)\mathbf{U}'(n-1)]'[\mathbf{I}_p - \mathbf{U}(n-1)\mathbf{U}'(n-1)] \in \mathbb{R}^{p\times p} \tag{4.20}$$

$$\mathbf{g}(n) := -\mathbf{H}(n)[\mathbf{x}_n - \mathbf{m}(n-1)] \in \mathbb{R}^p \tag{4.21}$$

one can recognize (4.19) as the multidimensional shrinkage-thresholding operator $\mathcal{T}_{\mathbf{H}(n),\lambda_2}(\mathbf{g}(n))$ introduced in [93]. In particular, as per [93, Corollary 2] it follows that

$$\mathbf{o}(n) = \mathcal{T}_{\mathbf{H}(n),\lambda_2}(\mathbf{g}(n)) = \begin{cases} -(\mathbf{H}(n) + \gamma\mathbf{I}_p)^{-1}\mathbf{g}(n), & \text{if } \|\mathbf{g}(n)\|_2 > \lambda_2 \\ \mathbf{0}_p, & \text{otherwise} \end{cases} \tag{4.22}$$

where parameter $\gamma := \lambda_2^2/(2\eta)$ is such that $\eta > 0$ solves the scalar optimization

$$\min_{\eta>0} \left(1 - \mathbf{g}'(n)\left(2\eta\mathbf{H}(n) + \lambda_2^2\right)^{-1}\mathbf{g}(n)\right)\eta. \tag{4.23}$$

Remarkably, one can easily determine if $\mathbf{o}(n) = \mathbf{0}_p$, by forming $\mathbf{g}(n)$ and checking whether $\|\mathbf{g}(n)\|_2 \leq \lambda_2$. This will be the computational burden incurred to solve (4.19) for most n, since outliers are typically sporadic and one would expect to obtain $\mathbf{o}(n) = \mathbf{0}_p$ most of the time. When datum $\mathbf{x}_n$ is deemed an outlier, $\|\mathbf{g}(n)\|_2 > \lambda_2$, and one needs to carry out the

Algorithm 6 : Online robust (OR-)PCA

* Batch initialization phase

Determine λ_2 and $\mathbf{U}(n_0)$ from $\{\mathbf{x}_n\}_{n=1}^{n_0}$, as in Section 4.3.2.

Initialize $\mathbf{P}(n_0) = 10^3\mathbf{I}_p$ and $\mathbf{s}(n_0) = \mathbf{0}_q$.

* Online phase

for $n = n_0 + 1, n_0 + 2, \ldots$ **do**

Form $\mathbf{H}(n)$ and $\mathbf{g}(n)$ using (4.20) and (4.21).

Update $\mathbf{o}(n) = \mathcal{T}_{\mathbf{H}(n),\lambda_2}(\mathbf{g}(n))$ via (4.22).

Update $\mathbf{s}(n) = \mathbf{U}'(n-1)[\mathbf{x}_n - \mathbf{o}(n)]$.

* RLS subspace update

Update $\mathbf{k}(n) = \mathbf{P}(n-1)\mathbf{s}(n)/[\beta + \mathbf{s}'(n)\mathbf{P}(n-1)\mathbf{s}(n)]$.

Update $\mathbf{P}(n) = (1/\beta)[\mathbf{P}(n-1) - \mathbf{k}(n)(\mathbf{P}(n-1)\mathbf{s}(n))']$.

Update $\mathbf{U}(n) = \mathbf{U}(n-1) + [\mathbf{x}_n - \mathbf{U}(n-1)\mathbf{s}(n) - \mathbf{o}(n)]\mathbf{k}'(n)$.

end for

extra line search in (4.23) to determine $\mathbf{o}(n)$ as per (4.22); further details can be found in in [93]. Whenever an ℓ_1-norm outlier regularization is adopted, the resulting counterpart of (4.19) can be solved using e.g., coordinate descent [5], or, the Lasso variant of least-angle regression (LARS) [77].

Moving on, the subspace update is given by

$$\mathbf{U}(n) = \arg\min_{\mathbf{U}} \sum_{i=1}^{n} \beta^{n-i} \|\mathbf{x}_i - \mathbf{m}(i-1) - \mathbf{U}\mathbf{s}(i) - \mathbf{o}(i)\|_2^2$$

and can be efficiently obtained from $\mathbf{U}(n-1)$, via a recursive LS update leveraging the matrix inversion lemma; see e.g., [129]. Note that the orthonormality constraint on $\mathbf{U}$ is not enforced here, yet the deviation from orthonormality is typically small as observed in [129]. Still, if orthonormal principal directions are required, an extra orthonormalization step can be carried out per iteration, or, once at the end of the process. Finally, $\mathbf{m}(n)$ is obtained recursively as the exponentially-weighted average of the outlier-compensated data $\{\mathbf{x}_i - \mathbf{o}(i)\}_{i=1}^{n}$. The resulting online robust (OR-)PCA algorithm and its initialization are summarized under Algorithm 6, where $\mathbf{m}$ and its update have been omitted for brevity.

For the batch case where all data in $\mathcal{T}_x$ are available for joint processing, two data-driven

criteria to select λ_2 have been outlined in Section 4.3.2. However, none of these sparsity-controlling mechanisms can be run in real-time, and selecting λ_2 for subspace tracking via OR-PCA is challenging. One possibility to circumvent this problem is to select λ_2 once during a short initialization (batch) phase of OR-PCA, and retain its value for the subsequent time instants. Specifically, the initialization phase of OR-PCA entails solving (4.7) using Algorithm 4, with a typically small batch of data $\{\mathbf{x}_n\}_{n=1}^{n_0}$. At time n_0, the criteria in Section 4.3.2 are adopted to find the 'best' λ_2, and thus obtain the subspace estimate $\hat{\mathbf{U}}(n_0)$ required to initialize the OR-PCA iterations.

Convergence analysis of OR-PCA algorithm is beyond the scope of this dissertation, and is only confirmed via simulations. The numerical tests in Section 4.7 also show that in the presence of outliers, the novel adaptive algorithm outperforms existing non-robust alternatives for subspace tracking.

4.6 Robustifying Kernel PCA

Kernel (K)PCA is a generalization to (linear) PCA, seeking principal components in a *feature space* nonlinearly related to the *input space* where the data in $\mathcal{T}_x$ live [106]. KPCA has been shown effective in performing nonlinear feature extraction for pattern recognition [106]. In addition, connections between KPCA and spectral clustering [59, p. 548] motivate well the novel KPCA method developed in this section, to robustly identify cohesive subgroups (communities) from social network data.

Consider a nonlinear function $\boldsymbol{\phi} : \mathbb{R}^p \to \mathcal{H}$, that maps elements from the input space $\mathbb{R}^p$ to a feature space $\mathcal{H}$ of arbitrarily large – possibly infinite – dimensionality. Given transformed data $\mathcal{T}_{\mathcal{H}} := \{\boldsymbol{\phi}(\mathbf{x}_n)\}_{n=1}^N$, the proposed approach to robust KPCA fits the model

$$\boldsymbol{\phi}(\mathbf{x}_n) = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n + \mathbf{o}_n, \quad n = 1, \ldots, N \tag{4.24}$$

by solving ($\boldsymbol{\Phi} := [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]$)

$$\min_{\mathbf{U},\mathbf{S},\mathbf{O}} \|\boldsymbol{\Phi}' - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \frac{\lambda_*}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2) + \lambda_2\|\mathbf{O}\|_{2,r}. \tag{4.25}$$

It is certainly possible to adopt the criterion (4.7) as well, but (4.25) is chosen here for simplicity in exposition. Except for the principal components' matrix $\mathbf{S} \in \mathbb{R}^{N\times\bar{q}}$, both the data and the unknowns in (4.25) are now vectors/matrices of generally infinite dimension. In principle, this challenges the optimization task since it is impossible to store, or, perform updates of such quantities directly. For these reasons, assuming zero-mean data $\boldsymbol{\phi}(\mathbf{x}_n)$, or, the possibility of mean compensation for that matter, cannot be taken for granted here [cf. Remark 4.1]. Thus, it is important to explicitly consider the estimation of $\mathbf{m}$.

Interestingly, this hurdle can be overcome by endowing $\mathcal{H}$ with the structure of a reproducing kernel Hilbert space (RKHS), where inner products between any two members of $\mathcal{H}$ boil down to evaluations of the reproducing kernel $K_{\mathcal{H}} : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, i.e., $\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle_{\mathcal{H}} = K_{\mathcal{H}}(\mathbf{x}_i, \mathbf{x}_j)$. Specifically, it is possible to form the kernel matrix $\mathbf{K} := \boldsymbol{\Phi}'\boldsymbol{\Phi} \in \mathbb{R}^{N\times N}$, without directly working with the vectors in $\mathcal{H}$. This so-termed *kernel trick* is the crux of most kernel methods in machine learning [59], including kernel PCA [106]. The problem of selecting $K_{\mathcal{H}}$ (and $\boldsymbol{\phi}$ indirectly) will not be considered here.

Building on these ideas, it is shown in the sequel that Algorithm 5 can be *kernelized*, to solve (4.25) at affordable computational complexity and memory storage requirements that do not depend on the dimensionality of $\mathcal{H}$.

Proposition 4.4 *For $k \geq 1$, the sequence of iterates generated by Algorithm 5 when applied to solve (4.25) can be written as $\mathbf{m}(k) = \boldsymbol{\Phi}\boldsymbol{\mu}(k)$, $\mathbf{U}(k) = \boldsymbol{\Phi}\boldsymbol{\Upsilon}(k)$, and $\mathbf{O}'(k) = \boldsymbol{\Phi}\boldsymbol{\Omega}(k)$. The quantities $\boldsymbol{\mu}(k) \in \mathbb{R}^N$, $\boldsymbol{\Upsilon}(k) \in \mathbb{R}^{N\times\bar{q}}$, and $\boldsymbol{\Omega}(k) \in \mathbb{R}^{N\times N}$ are recursively updated as in Algorithm 7, without the need of operating with vectors in $\mathcal{H}$.*

Proof: The proof relies on an inductive argument. Suppose that at iteration $k-1$, there exists a matrix $\boldsymbol{\Omega}(k-1) \in \mathbb{R}^{N\times N}$ such that the outliers can be expressed as $\mathbf{O}'(k-1) = \boldsymbol{\Phi}\boldsymbol{\Omega}(k-1)$. From Algorithm 5, the update for the mean vector is $\mathbf{m}(k) = [\boldsymbol{\Phi}' - \mathbf{O}(k-1)]'\mathbf{1}_N/N = [\boldsymbol{\Phi} - \boldsymbol{\Phi}\boldsymbol{\Omega}(k-1)]\mathbf{1}_N/N = \boldsymbol{\Phi}\boldsymbol{\mu}(k)$ where $\boldsymbol{\mu}(k) := [\mathbf{I}_n - \boldsymbol{\Omega}(k-1)]\mathbf{1}_N/N$. Likewise, $\mathbf{X}_o(k) = \boldsymbol{\Phi}' - \mathbf{1}_N\boldsymbol{\mu}'(k)\boldsymbol{\Phi}' - \boldsymbol{\Omega}'(k-1)\boldsymbol{\Phi}'$ so that one can write the subspace update as $\mathbf{U}(k) = \boldsymbol{\Phi}\boldsymbol{\Upsilon}(k)$, upon defining

$$\boldsymbol{\Upsilon}(k) := [\mathbf{I}_N - \boldsymbol{\mu}(k)\mathbf{1}_N' - \boldsymbol{\Omega}(k-1)]\mathbf{S}(k-1)[\mathbf{S}'(k-1)\mathbf{S}(k-1) + (\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}.$$

With regards to the principal components, it follows that (cf. Algorithm 5)

$$\begin{aligned}\mathbf{S}(k) &= [\mathbf{I}_N - \mathbf{1}_N\boldsymbol{\mu}'(k) - \boldsymbol{\Omega}'(k-1)]\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\Upsilon}(k)[\boldsymbol{\Upsilon}(k)'\boldsymbol{\Phi}'\boldsymbol{\Phi}\boldsymbol{\Upsilon}(k) + (\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1} \\ &= [\mathbf{I}_N - \mathbf{1}_N\boldsymbol{\mu}'(k) - \boldsymbol{\Omega}'(k-1)]\mathbf{K}\boldsymbol{\Upsilon}(k)[\boldsymbol{\Upsilon}(k)'\mathbf{K}\boldsymbol{\Upsilon}(k) + (\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1} \qquad (4.26)\end{aligned}$$

which is expressible in terms of the kernel matrix $\mathbf{K} := \boldsymbol{\Phi}'\boldsymbol{\Phi}$. Finally, the columns $\mathbf{o}_n(k)$ are given by the vector soft-thresholding operation (4.10), where the residuals are

$$\mathbf{r}_n(k) = \boldsymbol{\phi}(\mathbf{x}_n) - \mathbf{m}(k) - \mathbf{U}(k)\mathbf{s}_n(k) = \boldsymbol{\Phi}[\mathbf{b}_{N,n} - \boldsymbol{\mu}(k) - \boldsymbol{\Upsilon}(\boldsymbol{k})\mathbf{s}_n(k)] := \boldsymbol{\Phi}\boldsymbol{\rho}_n(k).$$

Upon stacking all columns $\mathbf{o}_n(k)$, $n = 1, \ldots, N$, one readily obtains [cf. (4.10)]

$$\mathbf{O}'(k) = \boldsymbol{\Phi}[\mathbf{I}_N - \boldsymbol{\mu}(k)\mathbf{1}_N' - \boldsymbol{\Upsilon}(\boldsymbol{k})\mathbf{S}'(k)]\boldsymbol{\Lambda}(k) \qquad (4.27)$$

where $\boldsymbol{\Lambda}(k) := \mathrm{diag}((\|\mathbf{r}_1(k)\|_2 - \lambda_2/2)_+/\|\mathbf{r}_1(k)\|_2, \ldots, (\|\mathbf{r}_N(k)\|_2 - \lambda_2/2)_+/\|\mathbf{r}_N(k)\|_2)$. Interestingly, the diagonal elements of $\boldsymbol{\Lambda}(k)$ can be computed using the kernel matrix, since $\|\mathbf{r}_n(k)\|_2 = \sqrt{\boldsymbol{\rho}_n'(k)\mathbf{K}\boldsymbol{\rho}_n(k)}$, $n = 1, \ldots, N$. From (4.27) it is apparent that one can write $\mathbf{O}'(k) = \boldsymbol{\Phi}\boldsymbol{\Omega}(k)$, after defining

$$\boldsymbol{\Omega}(k) := [\mathbf{I}_N - \boldsymbol{\mu}(k)\mathbf{1}_N' - \boldsymbol{\Upsilon}(\boldsymbol{k})\mathbf{S}'(k)]\boldsymbol{\Lambda}(k).$$

The proof is concluded by noting that for $k = 0$, Algorithm 5 is initialized with $\mathbf{O}'(0) = \mathbf{0}_{p\times N}$. One can thus satisfy the inductive base case $\mathbf{O}'(0) = \boldsymbol{\Phi}\boldsymbol{\Omega}(0)$, by letting $\boldsymbol{\Omega}(0) = \mathbf{0}_{N\times N}$. ■

In order to run the novel robust KPCA algorithm (tabulated as Algorithm 7), one does not have to store or process the quantities $\mathbf{m}(k)$, $\mathbf{U}(k)$, and $\mathbf{O}(k)$. As per Proposition 4.4, the iterations of the provably convergent AM solver in Section 4.4.2 can be equivalently carried out by cycling through *finite-dimensional* 'sufficient statistics' $\boldsymbol{\mu}(k) \to \boldsymbol{\Upsilon}(k) \to \mathbf{S}(k) \to \boldsymbol{\Omega}(k)$. In other words, the iterations of the robust kernel PCA algorithm are devoid of algebraic operations among vectors in $\mathcal{H}$. Recall that the size of matrix $\mathbf{S}$ is independent of the dimensionality of $\mathcal{H}$. Nevertheless, its update in Algorithm 5 cannot be carried out verbatim in the high-dimensional setting here, and is instead kernelized to yield the update rule (4.26).

Algorithm 7 : Robust KPCA solver

Initialize $\mathbf{\Omega}(0)=\mathbf{0}_{N\times N}$, $\mathbf{S}(0)$ randomly, and form $\mathbf{K}=\mathbf{\Phi}'\mathbf{\Phi}$.

for $k=1,2,\ldots$ **do**

Update $\boldsymbol{\mu}(k)=[\mathbf{I}_n-\mathbf{\Omega}(k-1)]\mathbf{1}_N/N$.

Form $\mathbf{\Phi}_o(k)=\mathbf{I}_N-\boldsymbol{\mu}(k)\mathbf{1}_N'-\mathbf{\Omega}(k-1)$.

Update $\mathbf{\Upsilon}(k)=\mathbf{\Phi}_o(k)\mathbf{S}(k-1)[\mathbf{S}'(k-1)\mathbf{S}(k-1)+(\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.

Update $\mathbf{S}(k)=\mathbf{\Phi}_o'(k)\mathbf{K}\mathbf{\Upsilon}(k)[\mathbf{\Upsilon}(k)'\mathbf{K}\mathbf{\Upsilon}(k)+(\lambda_*/2)\mathbf{I}_{\bar{q}}]^{-1}$.

Form $\boldsymbol{\rho}_n(k)=\mathbf{b}_{N,n}-\boldsymbol{\mu}(k)-\mathbf{\Upsilon}(\boldsymbol{k})\mathbf{s}_n(k)$, $n=1,\ldots,N$, and update $\mathbf{\Lambda}(k)$.

Update $\mathbf{\Omega}(k)=[\mathbf{I}_N-\boldsymbol{\mu}(k)\mathbf{1}_N'-\mathbf{\Upsilon}(\boldsymbol{k})\mathbf{S}'(k)]\mathbf{\Lambda}(k)$.

end for

Because $\mathbf{O}'(k)=\mathbf{\Phi\Omega}(k)$ and upon convergence of the algorithm, the outlier vector norms are computable in terms of $\mathbf{K}$, i.e., $[\|\mathbf{o}_1(\infty)\|_2^2,\ldots,\|\mathbf{o}_N(\infty)\|_2^2]'=\mathrm{diag}[\mathbf{\Omega}'(\infty)\mathbf{K}\mathbf{\Omega}(\infty)]$. These are critical to determine the robustification paths needed to carry out the outlier sparsity control methods in Section 4.3.2. Moreover, the principal component corresponding to any given new data point $\mathbf{x}$ is obtained through the projection $\mathbf{s}=\mathbf{U}(\infty)'[\boldsymbol{\phi}(\mathbf{x})-\mathbf{m}(\infty)]=\mathbf{\Upsilon}'(\infty)\mathbf{\Phi}'\boldsymbol{\phi}(\mathbf{x})-\mathbf{\Upsilon}'(\infty)\mathbf{K}\boldsymbol{\mu}(\infty)$, which is again computable after N evaluations the kernel function $K_{\mathcal{H}}$.

4.7 Numerical Tests

4.7.1 Synthetic data tests

To corroborate the effectiveness of the proposed robust methods, experiments with computer generated data are carried out first. These are important since they provide a 'ground truth', against which performance can be assessed by evaluating suitable figures of merit.

Outlier-sparsity control. To generate the data (4.4), a similar setting as in [131, Sec. V] is considered here with $N=p$ and $\mathbf{m}=\mathbf{0}_p$. For $n=1,\ldots,N$, the errors are $\mathbf{e}_n\sim\mathcal{N}(\mathbf{0}_p,\sigma_e^2\mathbf{I}_p)$ (multivariate normal distribution) and i.i.d. The entries of $\mathbf{U}$ and $\{\mathbf{s}_n\}_{n=1}^N$ are i.i.d. zero-mean Gaussian distributed, with variance $\sigma_{U,s}^2=10\sigma_e/\sqrt{N}$. Outliers are generated as $\mathbf{o}_n=\mathbf{p}_n\odot\mathbf{q}_n$, where the entries of $\mathbf{p}_n$ are i.i.d. Bernoulli distributed with parameter ρ_p, and $\mathbf{q}_n$ has i.i.d. entries drawn from a uniform distribution supported on $[-5,5]$. The

Table 4.1: Results for the first synthetic data test.

σ_e^2	λ_2^* **in (4.7)**	**ēr̄r for (4.7) (refined)**	**ēr̄r for (4.14)**	**ēr̄r for PCA**
0.01	0.7142	0.0622	0.0682	0.4679
0.05	1.7207	0.1288	0.1519	1.0122
0.1	2.4348	0.1742	0.2150	1.4141
0.25	3.6084	0.2525	0.3403	2.2480
0.5	6.1442	0.3361	0.4783	3.1601

chosen values of the parameters are $N = p = 200$, $q = 20$, $\rho_p = 0.01$, and varying noise levels $\sigma_e^2 = \{0.01, 0.05, 0.1, 0.25, 0.5\}$.

In this setup, the ability to recover the low-rank component of the data $\mathbf{L} := \mathbf{S}\mathbf{U}'$ is tested for the sparsity-controlling robust PCA method of this chapter [cf. (4.7)], stable PCP (4.14), and (non-robust) PCA. The ℓ_1-norm regularized counterparts of (4.7) and (4.14) are adopted to deal with entry-wise outliers. Both values of q and σ_e^2 are assumed known to obtain $\hat{\mathbf{L}} := \hat{\mathbf{S}}\hat{\mathbf{U}}'$ and $\hat{\mathbf{O}}$ via (4.7). This way, λ_2 is chosen using the sparsity-controlling algorithm of Section 4.3.2, searching over a grid where $G_\lambda = 200$, $\lambda_{\min} = 10^{-2}\lambda_{\max}$, and $\lambda_{\max} = 20$. In addition, the solutions of (4.7) are refined by running two iterations of the iteratively reweighted algorithm in Section 4.4.1, where $\delta = 10^{-5}$. Regarding SPCP, only the knowledge of σ_e^2 is required to select the tuning parameters $\lambda_* = 2\sqrt{2N\sigma_e^2}$ and $\lambda_2 = 2\sqrt{2\sigma_e^2}$ in (4.14), as suggested in [131]. Finally, the best rank q approximation to the data $\mathbf{X}$ is obtained using standard PCA.

The results are summarized in Table 4.1, which shows the estimation errors $\bar{\text{er}}\text{r} := \|\mathbf{L} - \hat{\mathbf{L}}\|_F/N$ attained by the aforementioned schemes, averaged over 15 runs of the experiment. The 'best' tuning parameters λ_2^* used in (4.7) are also shown. Both robust schemes attain an error which is approximately an order of magnitude smaller than PCA. With the additional knowledge of the true data rank q, the sparsity-controlling algorithm of this chapter outperforms stable PCP in terms of $\bar{\text{er}}\text{r}$. This numerical test is used to validate Proposition 4.3 as well. For the same values of the tuning parameters chosen for (4.14) and

the rank upper-bound set to $\bar{q} = 2q$, Algorithm 5 is run to obtain the solution $\{\bar{\mathbf{U}}, \bar{\mathbf{S}}, \bar{\mathbf{O}}\}$ of the nonconvex problem (4.15). The average (across realizations and values of σ_e^2) errors obtained are $\|\hat{\mathbf{L}} - \bar{\mathbf{S}}\bar{\mathbf{U}}'\|_F/N = 0.15 \times 10^{-6}$ and $\|\hat{\mathbf{O}} - \bar{\mathbf{O}}\|_F/N = 0.78 \times 10^{-7}$, where $\{\hat{\mathbf{L}}, \hat{\mathbf{O}}\}$ is the solution of stable PCP [cf. (4.14)]. Thus, the solutions are identical for all practical purposes.

Identification of invalid survey protocols. Robust PCA is tested here to identify invalid or otherwise aberrant item response (questionnaire) data in surveys, that is, to flag and hold in abeyance data that may negatively influence (i.e., bias) subsequent data summaries and statistical analyses. In recent years, item response theory (IRT) has become the dominant paradigm for constructing and evaluating questionnaires in the biobehavioral and health sciences and in high-stakes testing (e.g., in the development of college admission tests); see e.g., [121]. IRT entails a class of nonlinear models characterizing an individual's item response behavior by one or more latent traits, and one or more item parameters. An increasingly popular IRT model for survey data is the 2-parameter logistic IRT model (2PLM) [97]. 2PLM characterizes the probability of a keyed (endorsed) response y_{nm}, as a nonlinear function of a weighted difference between a person parameter θ_n and an item parameter b_m

$$\Pr(y_{nm} = 1|\theta_n) = \frac{e^{1.7a_m(\theta_n - b_m)}}{1 + e^{1.7a_m(\theta_n - b_m)}} \tag{4.28}$$

where θ_n is a latent trait value for individual n; a_m is an item discrimination parameter (similar to a factor loading) for item m; and b_m is an item difficulty (or extremity) parameter for item m. One reason for the popularity of 2PLM is that under certain assumptions, its parameters can be transformed into the person and item parameters of the maximum likelihood factor analysis model [109].

Binary item responses ('agree/disagree' response format) were generated for $N = 1,000$ hypothetical subjects who were administered $p = 200$ items (questions). The 2PLM function (4.28) was used to generate the underlying item response probabilities, which were converted into binary item responses as follows: a response was coded 1 if $\Pr(y_{nm}|\theta_n) \geq \mathcal{U}(0,1)$, and coded 0 otherwise, where $\mathcal{U}[0,1]$ denotes a uniform random deviate over $[0,1]$. Model parameters were randomly drawn as $\{a_m\}_{m=1}^{200} \sim \mathcal{U}[1, 1.5]$, $\{b_m\}_{m=1}^{200} \sim \mathcal{U}[-2, 2]$, and

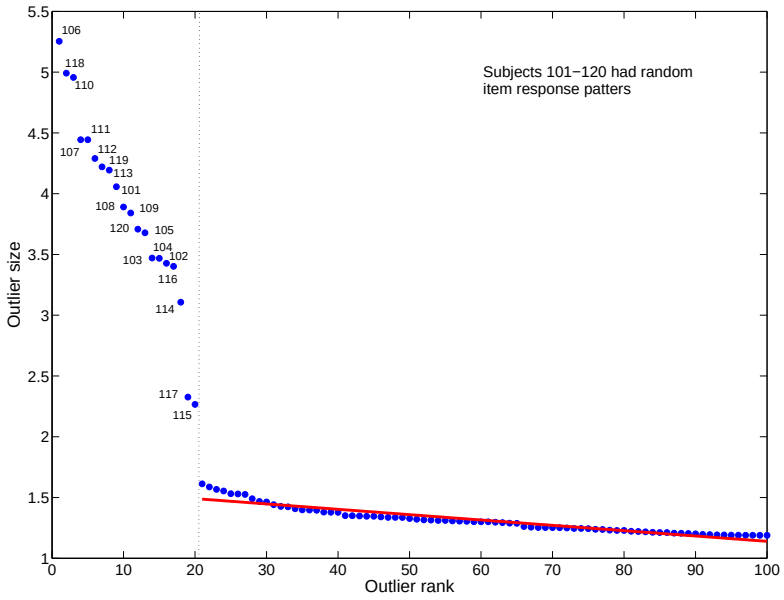


Figure 4.1: Pseudo scree plot of outlier size ($\|\hat{\mathbf{o}}_n\|_2$); the 100 largest outliers are shown.

$\{\boldsymbol{\theta}_l\}_{l=1}^{200} \sim \mathcal{N}(\mathbf{0}_5, \mathbf{I}_5)$. Each of the 200 items loaded on one of $q = 5$ latent factors. To simulate random responding – a prevalent form of aberrancy in e.g., web-collected data – rows 101-120 of the item response matrix $\mathbf{Y}$ were modified by (re)drawing each of the entries from a Bernoulli distribution with parameter 0.5, thus yielding the corrupted matrix $\mathbf{X}$.

Robust PCA in (4.7) was adopted to identify invalid survey data, with $q = 5$, and λ_2 chosen such that $\|\hat{\mathbf{O}}\|_0 = 150$, a safe overestimate of the number of outliers. Results of this study are summarized in Fig. 4.1, which displays the 100 largest outliers ($\|\hat{\mathbf{o}}_n\|_2$) from the robust PCA analysis of the $N = 1,000$ simulated response vectors. When the outliers are plotted against their ranks, there is an unmistakable break between the 20th and 21st ordered value indicating that the method correctly identified the *number* of aberrant response patterns in $\mathbf{X}$. Perhaps more impressively, the method also correctly identified rows 101-to-120 as containing the invalid data.

Online robust subspace estimation. A simulated test is carried out here to corroborate

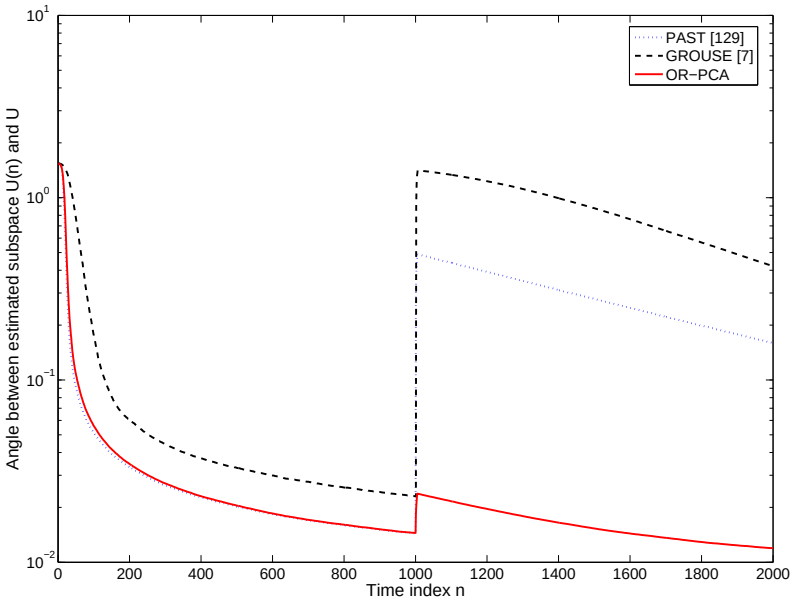


Figure 4.2: Time evolution of the angle between the learnt subspace $\mathbf{U}(n)$, and the true $\mathbf{U}$ used to generate the data ($\beta = 0.99$ and $\lambda_2 = 1.65$). Outlier contaminated data is introduced at time $n = 1001$.

the convergence and effectiveness of the OR-PCA algorithm in Section 4.5. For $N = 2,000$, $p = 150$, and $q = 5$, nominal data in $\mathcal{T}_y$ are generated according to the stationary model (4.1), where $\mathbf{e}_n \sim \mathcal{N}(\mathbf{0}_p, 10^{-3}\mathbf{I}_p)$. Vectors $\mathbf{x}_{1001}, \ldots, \mathbf{x}_{1005}$ are outliers, uniformly i.i.d. over $[-0.5, 0.5]$. The results depicted in Figs. 4.2 and 4.3 are obtained after averaging over 50 runs. Fig. 4.2 depicts the time evolution of the angle between the learnt subspace (spanned by the columns of) $\hat{\mathbf{U}}(n)$ and the true subspace $\mathbf{U}$ generating $\mathcal{T}_y$, where $\lambda_2 = 1.65$ and $\beta = 0.99$. The convergent trend of Algorithm 6 to $\mathbf{U}$ is apparent; and markedly outperforms the non-robust subspace tracking method in [129], and the first-order GROUSE algorithm in [7]. Note that even though $\mathbf{U}$ is time-invariant, it is meaningful to select $0 \ll \beta < 1$ to quickly 'forget' and recover from the outliers. A similar trend can be observed in Fig. 4.3, which depicts the time evolution of the reconstruction error $\|\mathbf{y}_n - \hat{\mathbf{U}}(n)\hat{\mathbf{U}}(n)'\mathbf{y}_n\|_2^2/p$.

Robust spectral clustering. The following simulated test demonstrates that robust

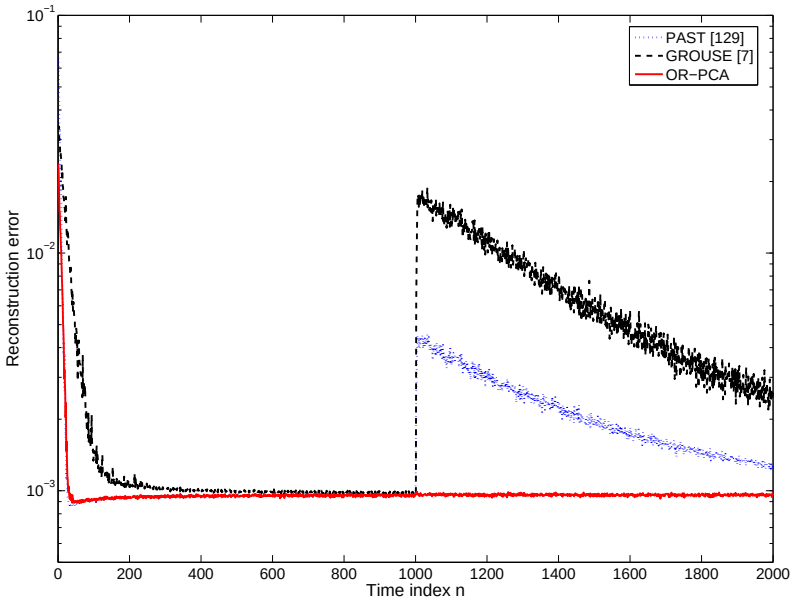


Figure 4.3: Time evolution of the reconstruction error. ($\beta = 0.99$ and $\lambda_2 = 1.65$). Outlier contaminated data is introduced at time $n = 1001$.

KPCA in Section 4.6 can be effectively used to robustify spectral clustering (cf. the connection between both non-robust methods in e.g., [59, p. 548]). Adopting the data setting from [59, p. 546]), $N = 450$ points in $\mathbb{R}^2$ are generated from three circular concentric clusters, with respective radii of 1, 2.8, and 5. The points are uniformly distributed in angle, and additive noise $\mathbf{e}_n \sim \mathcal{N}(\mathbf{0}_2, 0.15\mathbf{I}_2)$ is added to each datum. Five outliers $\{\mathbf{x}_n\}_{n=451}^{455}$ uniformly distributed in the square $[-7, 7]^2$ complete the training data $\mathcal{T}_x$; see Fig. 4.4 (left). To unveil the cluster structure from the data, Algorithm 7 is run using the Gaussian radial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/c)$, with $c = 10$. The sparsity-controlling parameter is set to $\lambda_2 = 1.85$ so that $\|\hat{\mathbf{O}}\|_0 = 5$, while $\lambda_* = 1$, and $\bar{q} = 2$. Upon convergence, the vector of estimated outlier norms is $[\|\mathbf{o}_1(\infty)\|_2^2, \ldots, \|\mathbf{o}_{N+5}(\infty)\|_2^2]' = [0, \ldots, 0, 10^{-4}, 1.3 \times 10^{-3}, 1.5 \times 10^{-2}, 10^{-2}, 1.7 \times 10^{-2}]'$, which shows that the outliers are correctly identified. Estimates of the (rotated) first two dominant eigenvectors of the kernel matrix $\mathbf{K}$ are obtained as the columns of $\hat{\boldsymbol{\Upsilon}}$, and are

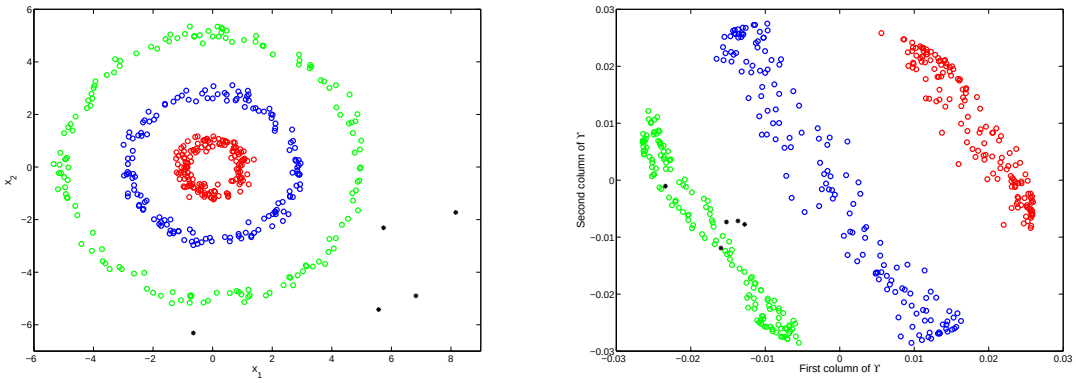


Figure 4.4: (Left) Data in three concentric clusters, in addition to five outliers shown in black. (Right) Coordinates of the first two columns of $\mathbf{\Upsilon}$, obtained by running Algorithm 7. The five outlying points are correctly identified, and thus can be discarded. Non-robust methods will assign them to the green cluster.

depicted in Fig. 4.4 (right). After removing the rows of $\hat{\mathbf{\Upsilon}}$ corresponding to the outliers [black points in Fig. 4.4 (right)], e.g., K-means clustering of the remaining points in Fig. 4.4 (right) will easily reveal the three clusters sought. From Fig. 4.4 (right) it is apparent that a non-robust KPCA method will incorrectly assign the outliers to the outer (green) cluster.

4.7.2 Real data tests

Video surveillance. To validate the proposed approach to robust PCA, Algorithm 4 was tested to perform background modeling from a sequence of video frames; an approach that has found widespread applicability for intrusion detection in video surveillance systems. The experiments were carried out using the dataset studied in [31], which consists of $N = 520$ images ($p = 120 \times 160$) acquired from a static camera during two days. The illumination changes considerably over the two day span, while approximately 40% of the training images contain people in various locations. For $q = 10$, both standard PCA and the robust PCA of Section 4.3 were applied to build a low-rank background model of the environment captured by the camera. For robust PCA, ℓ_1-norm regularization on $\mathbf{O}$ was adopted to

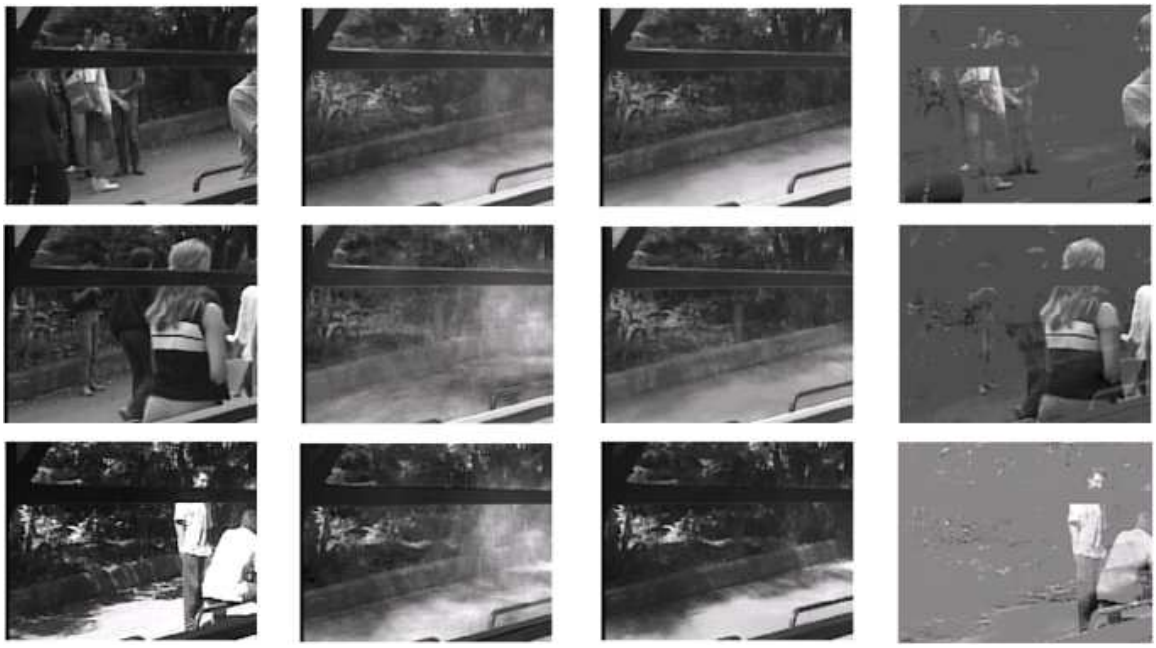


Figure 4.5: Background modeling for video surveillance. First column: original frames. Second column: PCA reconstructions, where the presence of undesirable 'ghostly' artifacts is apparent, since PCA is not able to completely separate the people from the background. Third column: robust PCA reconstructions, which recover the illumination changes while successfully subtracting the people. Fourth column: outliers in $\hat{\mathbf{o}}$, which mostly capture the people and abrupt changes in illumination.

identify outliers at a pixel level. The outlier sparsity-controlling parameter was chosen as $\lambda_2 = 9.69 \times 10^{-4}$, whereas a single iteration of the reweighted scheme in Section 4.4.1 was run to reduce the bias in $\hat{\mathbf{O}}$.

Results are shown in Fig. 4.5, for three representative images. The first column comprises the original frames from the training set, while the second column shows the corresponding PCA image reconstructions. The presence of undesirable 'ghostly' artifacts is apparent, since PCA is unable to completely separate the people from the background. The third column illustrates the robust PCA reconstructions, which recover the illumination changes while successfully subtracting the people. The fourth column shows the reshaped outlier vectors $\hat{\mathbf{o}}_n$, which mostly capture the people and abrupt changes in illumination.

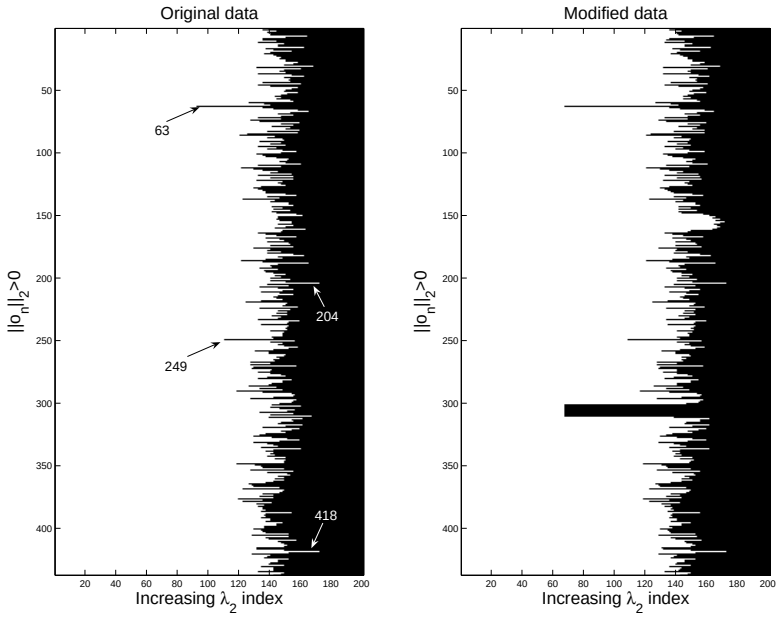


Figure 4.6: Evolution of $\hat{\mathbf{O}}$'s row support as a function of λ_2 – black pixels along the nth row indicate that $\|\hat{\mathbf{o}}_n\|_2 = 0$, whereas white ones reflect that the responses from subject n are deemed as outliers for given λ_2. The results for both the original and modified (introducing random and constant item responses) BFI datasets are shown.

Robust measurement of the Big Five personality factors. The 'Big Five' are five factors ($q = 5$) of personality traits, namely extraversion, agreeableness, conscientiousness, neuroticism, and openness; see e.g., [65]. The Big Five inventory (BFI) on the other hand, is a brief questionnaire (44 items in total) tailored to measure the Big Five dimensions. Subjects taking the questionnaire are asked to rate in a scale from 1 (disagree strongly) to 5 (agree strongly), items of the form 'I see myself as someone who is talkative'. Each item consists of a short phrase correlating (positively or negatively) with one factor; see e.g., [65, pp. 157-58] for a copy of the BFI and scoring instructions.

Robust PCA is used to identify aberrant responses from real BFI data comprising the Eugene-Springfield community sample [52]. The rows of $\mathbf{X}$ contain the $p = 44$ item responses for each one of the $N = 437$ subjects under study. For $q = 5$, (4.7) is solved over grid of

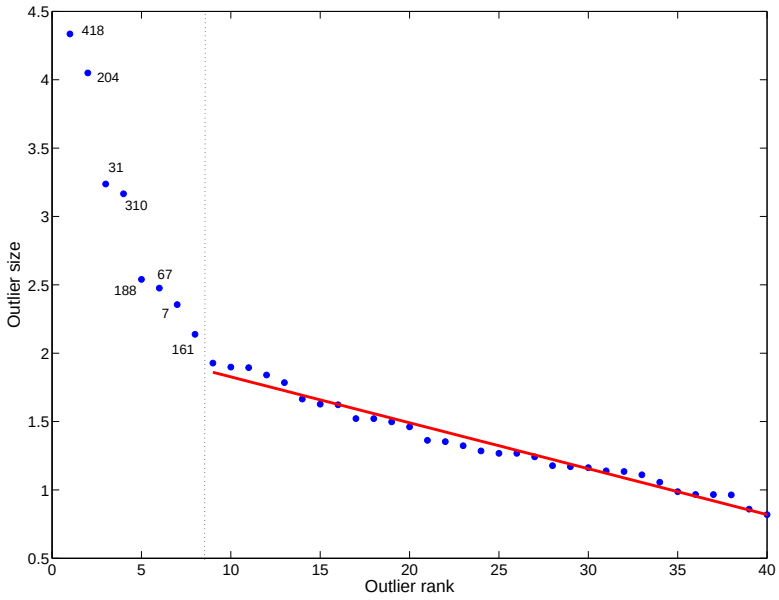


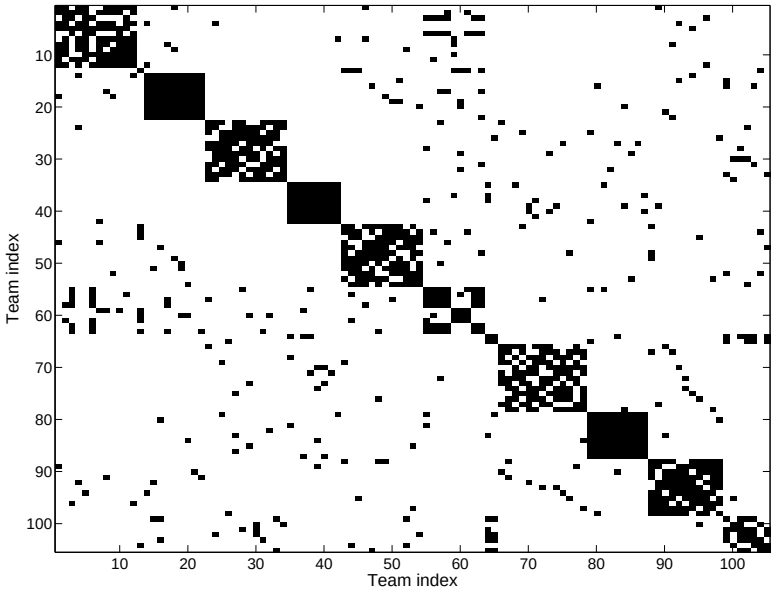
Figure 4.7: Pseudo scree plot of outlier size ($\|\hat{\mathbf{o}}_n\|_2$); the 40 largest outliers are shown. Robust PCA declares the largest 8 as aberrant responses.

$G_\lambda = 200$ values of λ_2, where $\lambda_{\min} = 10^{-2}\lambda_{\max}$, and $\lambda_{\max} = 20$. The first plot of Fig. 4.6 shows the evolution of $\hat{\mathbf{O}}$'s row support as a function of λ_2 with black pixels along the nth row indicating that $\|\hat{\mathbf{o}}_n\|_2 = 0$, and white ones reflecting that the responses from subject n are deemed as outliers for the given λ_2. For example subjects $n = 418$ and 204 are strong outlier candidates due to random responding, since they enter the model ($\|\hat{\mathbf{o}}_n\|_2 > 0$) for relatively large values of λ_2. The responses of e.g., subjects $n = 63$ (all items rated '3') and 249 (41 items rated '3' and 3 items rated '4') are also undesirable, but are well modeled by (4.1) and are only deemed as outliers when λ_2 is quite small. These two observations are corroborated by the second plot of Fig. 4.6, which shows the robust PCA results on a corrupted dataset, obtained from $\mathbf{X}$ by overwriting: (i) rows $151 - 160$ with random item responses drawn from a uniform distribution over $\{1, 2, 3, 4, 5\}$; and (ii) rows $301 - 310$ with constant item responses of value 3.

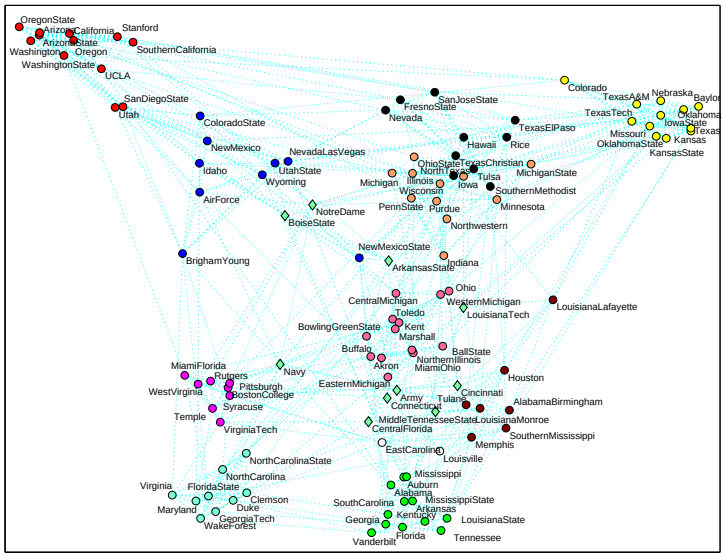
For $\lambda_2 = 5.6107$ corresponding to $\|\hat{\mathbf{O}}\|_0 = 100$, Fig. 4.7 depicts the norm of the 40

largest outliers. Following the methodology outlined in Section 4.7.1, 8 subjects including $n = 418$ and 204 are declared as outliers by robust PCA. As a means of validating these results, the following procedure is adopted. Based on the BFI scoring key [65], a list of all pairs of items hypothesized to yield positively correlated responses is formed. For each n, one counts the 'inconsistencies' defined as the number of times that subject n's ratings for these pairs differ in more than four, in absolute value. Interestingly, after rank-ordering all subjects in terms of this inconsistency score, one finds that $n = 418$ ranks highest with a count of 17, $n = 204$ ranks second (10), and overall the eight outliers found rank in the top twenty.

Unveiling communities in social networks. Next, robust KPCA is used to identify communities and outliers in a network of $N = 115$ college football teams, by capitalizing on the connection between KPCA and spectral clustering [59, p. 548]. Nodes in the network graph represent teams belonging to eleven conferences (plus five independent teams), whereas (unweighted) edges joining pairs of nodes indicate that both teams played against each other during the Fall 2000 Division I season [50]. The kernel matrix used to run robust KPCA is $\mathbf{K} = \zeta\mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{A}$ and $\mathbf{D}$ denote the graph adjacency and degree matrices, respectively; while $\zeta > 0$ is chosen to render $\mathbf{K}$ positive semi-definite. The tuning parameters are chosen as $\lambda_2 = 1.297$ so that $\|\hat{\mathbf{O}}\|_0 = 10$, while $\lambda_* = 1$, and $\bar{q} = 3$. Fig. 4.8 (a) shows the entries of $\mathbf{K}$, where rows and columns are permuted to reveal the clustering structure found by robust KPCA (after removing the outliers); see also Fig. 4.8 (b). The quality of the clustering is assessed through the adjusted rand index (ARI) after excluding outliers [43], which yielded the value 0.8967. Four of the teams deemed as outliers are Connecticut, Central Florida, Navy, and Notre Dame, which are indeed teams not belonging to any major conference. The community structure of traditional powerhouse conferences such as Big Ten, Big 12, ACC, Big East, and SEC was identified exactly.



(a)



(b)

Figure 4.8: (a) Entries of $\mathbf{K}$ after removing the outliers, where rows and columns are permuted to reveal the clustering structure found by robust KPCA. (b) Graph depiction of the clustered network. Teams belonging to the same estimated conference (cluster) are colored identically. The outliers are represented as diamond-shaped nodes.

4.8 Summary

Outlier-robust PCA methods were developed in this chapter, to obtain low-dimensional representations of (corrupted) data. Bringing together the seemingly unrelated fields of robust statistics and sparse regression, the novel robust PCA framework was found rooted at the crossroads of outlier-resilient estimation, learning via (group-) Lasso and kernel methods, and real-time adaptive signal processing. Social network analysis, video surveillance, and psychometrics, were highlighted as relevant application domains.

4.9 Appendices

4.9.1 Proof of equivalence of (4.7) and (4.8)

Towards establishing the equivalence between problems (4.7) and (4.8), consider the pair $\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\}$ that solves (4.7). Assume that $\hat{\mathcal{V}}$ is given, and the goal is to determine $\hat{\mathbf{O}}$. Upon defining the residuals $\hat{\mathbf{r}}_n := \mathbf{x}_n - \hat{\mathbf{m}} - \hat{\mathbf{U}}\hat{\mathbf{s}}_n$ and from the row-wise decomposability of $\|\cdot\|_{2,r}$, the rows of $\hat{\mathbf{O}}$ are separately given by

$$\hat{\mathbf{o}}_n := \arg \min_{\mathbf{o}_n \in \mathbb{R}^p} \left[\|\hat{\mathbf{r}}_n - \mathbf{o}_n\|_2^2 + \lambda_2 \|\mathbf{o}_n\|_2 \right], \quad n = 1, \ldots, N. \tag{4.29}$$

For each $n = 1, \ldots, N$, because (4.29) is nondifferentiable at the origin one should consider two cases: i) if $\hat{\mathbf{o}}_n = \mathbf{0}_p$, it follows that the minimum cost in (4.29) is $\|\hat{\mathbf{r}}_n\|_2^2$; otherwise, ii) if $\|\hat{\mathbf{o}}_n\|_2 > 0$, the first-order condition for optimality gives $\hat{\mathbf{o}}_n = \hat{\mathbf{r}}_n - (\lambda_2/2)\hat{\mathbf{r}}_n/\|\hat{\mathbf{r}}_n\|_2$ provided $\|\hat{\mathbf{r}}_n\|_2 > \lambda_2/2$, and the minimum cost is $\lambda_2\|\hat{\mathbf{r}}_n\|_2 - \lambda_2^2/4$. Compactly, the solution of (4.29) is given by $\hat{\mathbf{o}}_n = \hat{\mathbf{r}}_n(\|\hat{\mathbf{r}}_n\|_2 - \lambda_2/2)_+/\|\hat{\mathbf{r}}_n\|_2$, while the minimum cost in (4.29) after minimizing w.r.t. $\mathbf{o}_n$ is $\rho_v(\hat{\mathbf{r}}_n)$ [cf. (4.9) and the argument following (4.29)]. The conclusion is that $\hat{\mathcal{V}}$ is the minimizer of (4.8), in addition to being the solution of (4.7) by definition.■

Chapter 5

Future Work

This dissertation dealt with sparsity-controlling outlier rejection methods for statistical learning from high-dimensional data. In this final chapter, we point out possible directions for future research, and additional experimental validation using the largest repository of online-assessed personality and preference data.

5.1 Robust Canonical Correlation Analysis

While PCA can perform dimensionality reduction of sources that are directly observable at the encoder, oftentimes the compressed data are used to reconstruct a *remote* source presented to the encoder input. Such hidden, remote sources, arise due to sensing noise, or non-ideal channels between the source of interest and the sensing devices [68]. Dimensionality reduction in this case aims at compressing data to render them as much correlated with the hidden source of interest. A pertinent framework to tackle such a problem is canonical correlation analysis (CCA) [15]. CCA has been traditionally employed to reveal linear relationships between two correlated vectors [15]. Recently, this task has been successfully applied to genomic data interpretation [91, 123]. Similar to PCA, CCA is very sensitive to outlying observations [14]. We are planning on investigating *doubly robust* CCA formulations whereby outliers are explicitly accounted for both in the remote source training data, and the observations to be compressed. We envision application of the novel robust

methods to bioinformatics and system identification tasks [15].

5.2 Parametric Model Generalizations

This section shows how the USPACOR framework of this thesis can be generalized to other parametric models beyond linear regression (cf. Chapter 2).

5.2.1 Errors-in-variables and total least-squares

Total least-squares (TLS) extends ordinary LS to fully-perturbed linear models, such as the errors-in-variables one; see e.g., [80]. With $\hat{\mathbf{\Sigma}}$ denoting the sample covariance of the data vectors $\{[\mathbf{x}_n' \; y_n]'\}_{n=1}^N$, the TLS estimator corresponds to the eigenvector associated with the smallest eigenvalue of $\hat{\mathbf{\Sigma}}$. As such, TLS performs 'orthogonal regression,' which minimizes the sum of squared *orthogonal* distances from $[\mathbf{x}_n' \; y_n]'$ to the fitting hyperplane, as opposed to the *vertical* distance minimized by LS [80]. To robustify TLS against outliers, USPACOR can be applied to yield the desired robust estimator $\hat{\boldsymbol{\theta}}$ as solution of

$$\min_{\boldsymbol{\theta},\mathbf{o}} \sum_{i=1}^{N} \frac{(y_n - \mathbf{x}_n'\boldsymbol{\theta} - o_n)^2}{1 + \|\boldsymbol{\theta}\|_2^2} + \lambda_1 \|\mathbf{o}\|_1 \; . \tag{5.1}$$

Alternating minimization between variables $\boldsymbol{\theta}$ and $\mathbf{o}$ can converge to a stationary point of this nonconvex criterion. Each sub-problem per iteration reduces to either TLS or a scalar Lasso, and in both cases the solutions admit analytical forms.

5.2.2 Generalized linear models

The MSE-optimal regression function $E[y|\mathbf{x}]$ is modeled here by the so-termed activation function $f(\mathbf{x}'\boldsymbol{\theta})$. A special case popular for (say binary) classification leads to logistic regression, where $f(u) := (1 + e^{-u})^{-1}$, and the response y_n equals 1 when input vector $\mathbf{x}_n$ belongs to the first class, and 0 otherwise [59, p. 119]. To robustify logistic regression USPACOR estimates $\boldsymbol{\theta}$ by

$$\min_{\boldsymbol{\theta},\mathbf{o}} - \sum_{n=1}^{N} y_n \log z_n + (1 - y_n) \log(1 - z_n) + \lambda \|\mathbf{o}\|_1 \tag{5.2}$$

where $z_n := f(\mathbf{x}_n'\boldsymbol{\theta}+o_n)$. Problem (5.2) is convex and can be efficiently solved by reweighted LS iterations [59, p. 120]. The result can be extended readily to: i) multiclass classification; and ii) probit regression, where $f(u)$ is replaced by the standard Gaussian cumulative distribution function.

5.3 Distributed Algorithms for Matrix Completion

The popular Netflix prize competition (`http://www.netflixprize.com/`) has stirred a great deal of interest and research on (low-rank) *matrix completion*; see e.g., [7, 19, 24]. This problem is closely related to PM and *online recommendation systems* (ORS) [3]. ORS predict preferences of a consumer for products (Amazon books, Netflix movies), based on his/her previously revealed preferences, as well as the preferences of other consumers. Because each user only rates a small subset of products, there is an inherent under-determinacy in predicting based on limited ratings (most values $y_{ij} := [\mathbf{Y}]_{ij}$ are missing). This under-determinacy can only be resolved by relying on structural assumptions on the ratings matrix $\mathbf{Y}$ (consumers $\times$ products) sought.

Arguably the most common choice is to assume that the ratings matrix has low rank. A remarkable result asserts that under some technical incoherence conditions, the nuclear norm minimization problem

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*, \quad \text{s. to } [\mathbf{X}]_{ij} = [\mathbf{Y}]_{ij},\ (i,j) \in \Omega \tag{5.3}$$

can recover a highly incomplete low rank matrix $\mathbf{Y}$ with overwhelming probability [19]. Note that (5.3) is a convex problem, and Ω denotes the set of indices of observed preferences. While a great body of work has been proposed to solve (5.3) [17,75,76,112], it appears that none of these methods can currently operate on the scale of data commonly acquired by Internet retailers and social networking sites. This calls for custom-made low-complexity real-time (adaptive) matrix algorithms, ideally also distributable for *Hadoop* or large-scale grid computation.

Because the nuclear norm is non-separable, it is challenging to develop distributed algorithms directly from (5.3). Recently, an incremental stochastic gradient algorithm was

put forth in [96]. Building on the Frobenius norm characterization of the nuclear norm (cf. Chapter 4), we are currently working on developing a distributed matrix completion algorithm. The idea is solve the separable problem

$$\min_{\{\mathbf{U},\mathbf{S}\}} \frac{1}{2}\left(\|\mathbf{U}\|_F^2 + \|\mathbf{S}\|_F^2\right), \quad \text{s. to } [\mathbf{S}\mathbf{U}']_{ij} = [\mathbf{Y}]_{ij},\ (i,j) \in \Omega \tag{5.4}$$

which is equivalent to (5.3) under mild assumptions. The AD-MoM framework used in this thesis for distributed CA can be applied in this context as well. Since (5.4) is a non-convex optimization problem, convergence is not guaranteed by the existing theory. However, we believe that the convergence results can be extended to this non-convex setting, since there may be sufficient structure in the bilinear factorization $\mathbf{X} = \mathbf{S}\mathbf{U}'$.

5.4 Validation Using GPIPP Psychological Ratings

Since its inception in 1997, the Gosling-Potter Internet Personality Project (GPIPP) has generated the largest repository of online-assessed personality and preference data. Among several other inventories, the most popular GPIPP test is the BFI (`http://www.outofservice.com/bigfive/`) studied in Chapter 4. Relative to the Eugene-Springfield community sample [52], the GPIPP repository is much richer in terms of subject diversity and sample size.

To date, the family of related GPIPP Web sites (`http://www.outofservice.com/`) has attracted more than 8 million visitors interested in taking online personality and social attitude tests. These heterogeneous volunteers are aged 9-to-90, represent diverse ethnicities and cultures, and come from more than 100 countries. Much contemporary social science research is focused on individuals who are decidedly WEIRD (Western, Educated, Industrialized, Rich, and Democratic). The GPIPP data are 'WIRED' (i.e., collected over the Internet) but not WEIRD. For instance, in a recent study of $564,502$ cases '19% of the sample were not from advanced economies, 20% were from non-Western societies; 35% of the Western-society sample were not from the United States; and 66% of the U.S. sample were not in the 18-22 (college) age group' [54]. The GPIPP data have been used to study personality correlates [100] and measurement of self-esteem across the life span [101]; cross-

cultural and geographic variation in personality and attitudes [98]; and statewide political preferences and voting patterns [99]. We have contacted Prof. S. Gosling who has agreed to give us full access to the GPIPP repository. In analyzing the GPIPP data, Prof. Gosling and his colleagues have often struggled to find principled algorithmic means of identifying invalid protocols from among the sizable number of cases in the database. We have strong convictions that the methods in this dissertation (e.g., robust PCA in Chapter 4) will therefore help solve a pressing problem in the personality analysis community.

Bibliography

[1] Energy Independence and Security Act of 2007, an Act of the Congress of the United States of America Publ. L. No. 110-140, H.R. 6, Dec. 2007.

[2] The Smart Grid: An Introduction, United States Department of Energy, Office of Electricity Delivery and Energy Reliability, Jan. 2010.

[3] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. of Knowledge Data Eng.*, vol. 15, no. 6, pp. 733–749, 2005.

[4] G. M. Allenby and P. E. Rossi, "Marketing models of consumer heterogeneity," *J. Econometrics*, vol. 89, pp. 57–58, 1999.

[5] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, pp. 3436–3447, July 2010.

[6] M. S. Asif and J. Romberg, "Dynamic updating for l1 minimization," *IEEE Sel. Topics in Signal Process.*, vol. 4, no. 2, pp. 421–434, 2010.

[7] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. of 48th Allerton Conference*, Monticello, IL, pp. 704–711, Sept./Oct. 2010.

[8] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Distributed Lasso for in-network linear Regression," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Dallas, TX, Mar. 2010.

[9] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, Jan. 2009.

[10] M. Ben-Akiva and S. R. Lerman, *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA: MIT Press, 1985.

[11] D. P. Bertsekas, *Nonlinear Programming*. Athena-Scientific, second ed., 1999.

[12] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena-Scientific, second ed., 1999.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, 2004.

[14] J. A. Branco, C. Croux, P. Filzmoser, and M. R. Oliveira, "Robust canonical correlations: A comparative study," *Computational Statistics*, vol. 20, pp. 203–229, 2005.

[15] D. R. Brillinger, *Time Series: Data Analysis and Theory*. Holden Day, 1981.

[16] D. S. Bunch, J. J. Louviere, and D. Anderson, "A comparison of experimental design strategies for multinomial logit models: The case of generic attributes," tech. rep., Graduate School of Management, University of California at Davis, 1994.

[17] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, pp. 1956–1982, 2008.

[18] N. A. Campbell, "Robust procedures in multivariate analysis I: Robust covariance estimation," *Applied Stat.*, vol. 29, pp. 231–237, 1980.

[19] E. J. Candeas and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–722, 2009.

[20] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58. Article No. 11, Mar. 2011.

[21] E. J. Candes and P. A. Randall, "Highly robust error correction by convex programming," *IEEE Trans. on Inf. Theory*, vol. 54, no. 7, pp. 2829–2840, 2008.

[22] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. on Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[23] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, Dec. 2008.

[24] V. Chandrasekaran, S. Sanghavi, P. A. Parillo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, pp. 572–596, 2011.

[25] J. Chen, W. Li, A. Lau, J. Cao, and K. Eang, "Automated load curve data cleansing in power systems," *IEEE Trans. Smart Grid*, vol. 1, pp. 213–221, Sept. 2010.

[26] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[27] C. C. Chuang, S. F. Fu, J. T. Jeng, and C. C. Hsiao, "Robust support vector regression networks for function approximation with outliers," *IEEE Trans. Neural Netw.*, vol. 13, pp. 1322–1330, June 2002.

[28] D. D. Cox, "Asymptotics for M-type smoothing splines," *Ann. Statist.*, vol. 11, pp. 530–551, 1983.

[29] D. Cui and D. Curry, "Prediction in marketing using the support vector machines," *Marketing Science*, vol. 24, no. 4, pp. 595–615, 2005.

[30] K. Cukier, "Data, data everywhere," *The Economist*, Feb. 2010. [Online]. Available: http://www.economist.com/specialreports/displaystory.cfm?story-id=15557443.

[31] F. de la Torre and M. J. Black, "A framework for robust subspace learning," *Int. Jrnl. of Computer Vision*, vol. 54, pp. 1183–209, 2003.

[32] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Trans. Image Process.*, vol. 20, pp. 3419–3430, Dec. 2011.

[33] J. Duchon, *Splines Minimizing Rotation-Invariant Semi-norms in Sobolev Spaces*. Springer-Verlag, 1977.

[34] B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, pp. 407–499, 2004.

[35] T. Evgeniou, C. Boussios, and G. Zacharia, "Generalized robust conjoint analysis," *Marketing Science*, vol. 24, no. 3, pp. 415–129, 2005.

[36] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, pp. 1–50, 2000.

[37] T. Evgeniou, M. Pontil, and O. Toubia, "A convex optimization approach to modeling consumer heterogeneity in conjoint analysis," *Marketing Science*, vol. 26, no. 6, pp. 805–818, 2007.

[38] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 96, pp. 1348–1360, 2001.

[39] M. Fazel, *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[40] M. Fazel, H. Hindi, and S. Boyd, "Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proc. of the American Control Conf.*, Denver, CO, pp. 2156–2162, June 2003.

[41] E. Fischer, "North American detail map of Flickr and Twitter locations," July 2011. [Online]. Available: http://www.flickr.com/photos/walkingsf/5912385701/in/set-72157627140310742.

[42] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comms. of the ACM*, pp. 381–395, 1981.

[43] P. Forero, V. Kekatos, and G. B. Giannakis, "Outlier-aware robust clustering," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, pp. 2244–2247, May 2011.

[44] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Stat.*, vol. 1, pp. 302–332, 2007.

[45] J. Friedman, T. Hastie, and R. Tibshirani, "Regularized paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, 2010.

[46] J. J. Fuchs, "An inverse problem approach to robust regression," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Phoeniz, AZ, pp. 180–188, Mar. 1999.

[47] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comp. Math. Appl.*, vol. 2, pp. 17–40, 1976.

[48] P. Garrigues and L. El Ghaoui, "Recursive Lasso: A homotopy algorithm for Lasso with online observations," in *Proc. of Conf. on Neural Information Processing Systems*, Vancouver, Canada, Dec. 2008.

[49] G. B. Giannakis, G. Mateos, S. Farahmand, V. Kekatos, and H. Zhu, "USPACOR: Universal sparsity-controlling outlier rejection," in *Proc. of Intl. Conf. on Acoust., Speech and Signal Proc.*, Prague, Czech Republic, pp. 1952–1955, May 2011.

[50] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 7821–7826, 2002.

[51] R. Glowinski and A. Marrocco, "Sur l'approximation, par éléments finis d'orde un, et la résolution par pénalisation-dualité d'une classe de problèmes de Dirichlet non-linéaires," *Rev. Francaise d'Aut. Inf. Rech. Oper.*, vol. 2, pp. 41–76, 1975.

[52] L. R. Goldberg, "The Eugene-Springfield community sample: Information available from the research participants," Tech. Rep. vol. 48, no. 1, Oregon Research Institue, 2008.

[53] T. Goldstein and S. Osher, "The split Bregman method for L1 regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 323–343, 2009.

[54] S. Gosling, C. Sandy, O. John, and J. Potter, "Wired but not WEIRD: The promise of the Internet in reaching more diverse samples," *Behavioral and Brain Sciences*, vol. 33, pp. 94–95, 2010.

[55] P. E. Green and V. R. Rao, "Conjoint measurement for quantifying judgmental data," *J. of Marketing Research*, vol. 8, pp. 355–363, 1971.

[56] R. Griesse and D. A. Lorenz, "A semismooth Newton method for Tikhonov functionals with sparsity constraints," *Inverse Problems*, vol. 24, pp. 1–19, 2008.

[57] S. R. Gunn Matlab SVM Toolbox, 1997. [Online]. Available: http://www.isis.ecs.soton.ac.uk/resources/svminfo/.

[58] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, second ed., 2003.

[59] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, second ed., 2009.

[60] J. R. Hauser and V. R. Rao, "Conjoint analysis, related modeling, and applications," in *Marketing Research and Modeling: Progress and Prospects* (Y. Wind and P. E. Green, eds.), New York, NY: Springer, 2005.

[61] S. G. Hauser "Vision for the smart grid", presented at the U.S. Department of Energy Smart Grid R&D Roundtable Meeting, Dec. 9, 2009.

[62] J. He, L. Balzano, and J. C. S. Lui, "Online robust subspace tracking from partial information," 2011. (Submitted; see also arXiv:1109.3827v2 [cs.IT]).

[63] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. New York: Wiley, 2009.

[64] Y. Jin and B. D. Rao, "Algorithms for robust linear regression by exploiting the connection to sparse signal recovery," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Processing*, Dallas, TX, pp. 3830–3833, Mar. 2010.

[65] O. P. John, L. P. Naumann, and C. J. Soto, "Paradigm shift to the integrative Big-Five trait taxonomy: History, measurement, and conceptual issues," in *Handbook of personality: Theory and research* (O. P. John, R. W. Robins, and L. A. Pervin, eds.), New York, NY: Guilford Press, 2008.

[66] J. Johnson, "Ascertaining the validity of individual protocols from web-based personality inventories," *Journal of Research in Personality*, vol. 39, no. 1, pp. 103–129, 2005.

[67] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer, 2002.

[68] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[69] V. Kekatos and G. B. Giannakis, "From sparse signals to sparse residuals for robust sensing," *IEEE Trans. Signal Process.*, vol. 59, pp. 3355–3368, July 2010.

[70] G. Kimbeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *Ann. Math. Statist.*, vol. 41, pp. 495–502, 1970.

[71] R. Kraut, J. Olson, M. Banaji, A. Bruckman, J. Cohen, and M. Couper, "Psychological research online: Opportunities and challenges," *American Psychologist*, vol. 59, no. 2, pp. 105–107, 2004.

[72] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions (with discussion)," *J. Computat. Graph. Statist.*, vol. 9, pp. 1–59, 2000.

[73] Y. J. Lee, W. F. Heisch, and C. M. Huang, "ϵ-SSVR: A smooth support vector machine for ϵ-insensitive regression," *IEEE Trans. Knowl. Data Eng.*, vol. 17, pp. 678–685, 2005.

[74] P. J. Lenk, W. S. DeSarbo, P. E. Green, and M. R. Young, "Hierarchical Bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs," *Marketing Science*, vol. 15, pp. 173–191, 1996.

[75] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," UIUC Technical Report UILU-ENG-09-2214, July 2009.

[76] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, pp. 1235–1256, 2009.

[77] J. Mairal, J. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Jrnl. of Machine Learning Research*, vol. 11, pp. 19–60, Jan. 2010.

[78] O. L. Mangasarian and D. R. Musicant, "Robust linear and support vector regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 950–955, Sept. 2000.

[79] M. Mardani, G. Mateos, and G. B. Giannakis, "Unveiling network anomalies in large-scale networks via sparsity and low rank," in *Proc. of 44th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2011.

[80] I. Markovsky and S. V. Huffel, "Overview of total least-squares methods," *Signal Processing*, vol. 87, pp. 2283–2302, 2007.

[81] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, pp. 5262–5276, Oct. 2010.

[82] G. Mateos and G. B. Giannakis, "Sparsity control for robust principal component analysis," in *Proc. of 44th Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, pp. 1925–1929, Nov. 2010.

[83] G. Mateos and G. B. Giannakis, "Robust conjoint analysis by controlling outlier sparsity," in *Proc. of European Signal. Process. Conf.*, pp. 1914–1918, Barcelona, Spain, Aug./Sept. 2011.

[84] G. Mateos and G. B. Giannakis, "Robust nonparametric regression via sparsity control with application to load curve data cleansing," *IEEE Trans. Signal Process.*, 2011. (Revised; see also arXiv:1104.0455v1 [stat.ML]).

[85] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier sparsity regularization," *IEEE Trans. Signal Process.*, 2011. (Submitted; see also arXiv:1111.1788v1 [stat.ML]).

[86] G. Mateos and G. B. Giannakis, "Robust nonparametric regression by controlling sparsity," in *Proc. of Intl. Conf. on Acoust., Speech and Signal Proc.*, pp. 3880–3883, Prague, Czech Republic, May 22-27, 2011.

[87] G. Mateos, V. Kekatos, and G. B. Giannakis, "Exploiting sparsity in model residuals for robust conjoint analysis," *Marketing Science*, Dec. 2011. (Submitted).

[88] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear prediction of chaotic time series using a support vector machine," in *Proc. of Wrkshp. Neural Networks for Signal Proces.*, Amelia Island, FL, pp. 24–26, 97.

[89] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, 1995.

[90] O. Netzer, O. Toubia, E. T. Bradlow, E. Dahan, T. Evgeniou, F. M. Feinberg, E. M. Feit, S. K. Hui, J. Johnson, J. C. Liechty, J. B. Orlin, and V. R. Rao, "Beyond conjoint analysis: Advances in preference measurement," *Marketing Letters*, vol. 19, pp. 337–354, 2008.

[91] E. Parkhomenko, D. Tritcher, and J. Beyene, "Sparse canonical correlation analysis with application to genomic data integration," *Statistical Applications in Genetics and Molecular Biology*, vol. 8. Article 1, 2009.

[92] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," *A. I. Memo No. 1140.* Artificial Intelligence Laboratory, Massachussets Institute of Technology, 1989.

[93] A. T. Puig, A. Wiesel, and A. O. Hero, "A multidimensional shrinkage-thresholding operator," in *Proc. of 15th Workshop on Statistical Signal Processing*, Cardiff, Wales, pp. 113 –116, Aug./Sept. 2009.

[94] I. Ramirez, F. Lecumberry, and G. Sapiro, "Universal priors for sparse modeling," in *Proc. of 3rd Intl. Workshop on Comp. Advances in Multi-Sensor Adapt. Process.*, Aruba, Dutch Antilles, pp. 197–200, Dec. 2009.

[95] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, pp. 471–501, 2010.

[96] B. Recht and C. Re, "Parallel stochastic gradient algorithms for large-scale matrix completion," 2011. [Online]. Available: http://pages.cs.wisc.edu/ brecht/papers/11.Rec.Re.IPGM.pdf.

[97] S. P. Reise and N. G. Waller, "Traitedness and the assessment of response pattern scalability," *Journal of Personality and Social Psychology*, vol. 65, pp. 143–151, 1993.

[98] P. Rentfrow, S. Gosling, and J. Potter, "A theory of the emergence, persistence, and expression of geographic variation in psychological characteristics," *Perspectives on Psychological Science*, vol. 3, pp. 339–369, 2008.

[99] P. Rentfrow, J. Jost, S. Gosling, and J. Potter, "Statewide differences in personality predict voting patterns in 1996–2004 US presidential elections," *Social and Psychological Bases of Ideology and System Justification*, vol. 1, pp. 314–349, 2009.

[100] R. Robins, J. Tracy, K. Trzesniewski, J. Potter, and S. Gosling, "Personality correlates of self-esteem," *Journal of Research in Personality*, vol. 35, pp. 463–482, 2001.

[101] R. Robins, K. Trzesniewski, J. Tracy, S. Gosling, and J. Potter, "Global self-esteem across the life span," *Psychology and Aging*, vol. 17, pp. 423–434, 2002.

[102] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *Ann. Statist.*, vol. 35, pp. 1012–1030, 2007.

[103] P. J. Rousseeuw and K. V. Driessen, "Computing LTS regression for large data sets," *Data Mining and Knowledge Discovery*, vol. 12, pp. 29–45, 2006.

[104] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 1987.

[105] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Proc. of Euro. Conf. on Machine Learning*, Warsaw, Poland, pp. 286–297, 2007.

[106] B. Scholkopf, A. Smola, and K.-R. Muller, "Kernel principal component analysis," *Artificial Neural Networks: Lec. Notes in Computer Science*, vol. 1327, pp. 583–588, 1997.

[107] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Neuro COLT Technical Report TR-1998-030*. Royal Holloway College, London, 1998.

[108] V. Srinivasan and A. D. Shockern, "Linear programming techniques for multidimensional analysis of preferences," *Psychometrica*, vol. 38, no. 3, pp. 337–369, 1973.

[109] Y. Takane and J. D. Leeuw, "On the relationship between item response theory and factor analysis of discretized variables," *Psychometrika*, vol. 52, pp. 393–408, 1987.

[110] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc B*, vol. 58, pp. 267–288, 1996.

[111] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. Washington, DC: W. H. Winston, 1977.

[112] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Opt.*, vol. 6, p. 615640, 2010.

[113] O. Toubia, T. Evgeniou, and J. Hauser, "Optimization-based and machine-learning methods for conjoint analysis: Estimation and question design," in *Conjoint Measurement: Methods and Applications* (A. Gustafsson, A. Herrmann, and F. Huber, eds.), New York, NY: Springer, 2007.

[114] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, pp. 27–38, Mar. 2010.

[115] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. Inf. Theory*, vol. 51, pp. 1030–1051, Mar. 2006.

[116] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable maximization," *J. Optim. Theory Appl.*, vol. 109, pp. 473–492, 2001.

[117] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, pp. 22–38, Nov. 1999.

[118] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[119] G. Wahba, *Spline Models for Observational Data*. Philadelphia: SIAM, 1990.

[120] G. Wahba and J. Wendelberger, "Some new mathematical methods for variational objective analysis using splines and cross validation," *Monthly Weather Review*, vol. 108, pp. 1122–1145, 1980.

[121] N. Waller and S. Reise, "Measuring psychopathology with non-standard IRT models: Fitting the four parameter model to the MMPI," in *New Directions in Psychological Measurement with Model-Based Approaches* (S. Embretson, ed.), Washington, DC: Amer. Psych. Assoc., 2010.

[122] L. Wang, J. Zhu, and H. Zou, "Hybrid huberized support vector machines for microarray classification and gene selection," *Bioinformatics*, pp. 412–419, 2008.

[123] D. M. Witten and R. J. Tibshirani, "Extensions of sparse canonical correlation analysis with applications to genomic data," *Statistical Applications in Genetics and Molecular Biology*, vol. 8. Article 28, 2009.

[124] J. Wright and Y. Ma, "Dense error correction via ℓ^1-minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3540–3560, 2010.

[125] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, pp. 2479–2493, 2009.

[126] T. Wu and K. Lange, "Coordinate descent algorithms for Lasso penalized regression," *Ann. Appl. Stat.*, vol. 2, pp. 224–244, 2008.

[127] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," 2010. (Submitted; see also arXiv:1010.4237v2 [cs.LG]).

[128] L. Xu and A. L. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Trans. Neural Nets.*, vol. 6, pp. 131–143, Jan. 1995.

[129] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, pp. 95–107, Jan. 1995.

[130] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal. Statist. Soc B*, vol. 68, pp. 49–67, 2006.

[131] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, "Stable principal component pursuit," in *Proc. of Intl. Symp. on Information Theory*, Austin, TX, pp. 1518–1522, June 2010.

[132] J. Zhu, S. C. H. Hoi, and M. R. T. Lyu, "Robust regularized kernel regression," *IEEE Trans. Syst., Man, Cybern. B Cybern.*, vol. 38, pp. 1639–1644, Dec. 2008.

[133] H. Zou, "The adaptive Lasso and its oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[134] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Jrnl. of Comp. and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.