Scaled sparse linear regression with the elastic net

Elias Raninen

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Otaniemi 18.4.2017

Thesis supervisor and advisor:

Prof. Esa Ollila

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY SCHOOL OF ELECTRICAL ENGINEERING	ABSTRACT OF THE MASTER'S THESIS

Author: Elias Raninen

Title: Scaled sparse linear regression with the elastic net

Date: 18.4.2017	Language: English	Number of pages: 8+52

Department of Signal Processing and Acoustics

Professorship: Signal Processing

Supervisor and advisor: Prof. Esa Ollila

Scaled linear regression is a form of penalized linear regression in which the penalty level is automatically scaled in proportion to the estimated noise level in the data. This makes the penalty parameter independent of the noise scale enabling an analytical approach for choosing an optimal penalty level for a given problem. In this thesis, we first review conventional penalized regression methods, such as ridge regression, lasso, and the elastic net. Then, we review some scaled sparse linear regression methods, the most relevant of which is the scaled lasso, also known as square-root lasso. As an original contribution, we propose two elastic net formulations, which extend the scaled lasso to the elastic net framework. We demonstrate by numerical examples that the proposed estimators improve upon the scaled lasso in the presence of high correlations in the feature space. As a real-world application example, we apply the proposed estimators in a simulated single snapshot direction-of-arrival (DOA) estimation problem, where we show that the proposed estimators perform better, especially when the angles of incidence of the DOAs are oblique with respect to the uniform linear array (ULA) axis.

Keywords: Signal processing, penalized linear regression, regularization, scaled lasso, square-root lasso, elastic net, sparse methods.

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Elias Raninen		
Työn nimi: Skaalattu harva lineaarinen regressio elastisella verkolla		
Päivämäärä: 18.4.2017	Kieli: Englanti	Sivumäärä: 8+52
Signaalinkäsittelyn ja akustiikan laitos		
Professuuri: Signaalinkäsittely		
Työn valvoja: Prof. Esa Ollila		
Työn ohjaaja: Prof. Esa Ollila		

Skaalattu lineaarinen regressio käsittää regularisointimenetelmiä, joissa regularisointitermin painoa skaalataan datasta estimoidun kohinatason perusteella. Tämä poistaa optimaalisen regularisointitermin riippuvuuden tuntemattomasta kohinatasosta, mikä mahdollistaa analyyttisesti johdettujen regularisointitermien käytön. Diplomityössä tarkasteltiin ridge, lasso ja elastinen verkko -regressiomenetelmien ominaisuuksia sekä skaalattuja regressiomenetelmiä, kuten skaalattua lasso- eli neliöjuurilassomenetelmää. Diplomityössä kehitettiin täysin uudet estimaattorit: skaalattu elastinen verkko ja neliöjuuri elastinen verkko, jotka toimivat paremmin kuin skaalattu lasso multikollineaarisissa tilanteissa, mikä osoitettiin numeerisilla simulaatioilla. Esimerkkinä käytännön sovelluksesta, uusia estimaattoreita sovellettiin DOA-estimoinnissa, jossa pyritään antenniryhmän avulla määrittämään signaalin tulosuunta. Saatujen tulosten perusteella voitiin päätellä, että diplomityössä ehdotetut estimaattorit pystyivät määrittämään tulosuunnan paremmin kuin skaalattu lasso etenkin, kun signaalin tulokulma oli suuri antenniryhmän akselin suhteen.

Avainsanat: Signaalinkäsittely, regularisoitu lineaarinen regressio, regularisointi, skaalattu lasso, neliöjuurilasso, elastinen verkko, harvat menetelmät.

Preface

This thesis was carried out in the Department of Signal Processing and Acoustics, Aalto University, during 2016–2017. The research work done for this thesis lead to an original contribution, which was presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'17), New Orleans, U.S.A.

First of all, I want to express deep gratitude to my supervisor Prof. Esa Ollila for giving me valuable support and guidance in this work. The making of this thesis has been a rewarding learning experience, which has consolidated my understanding of signal processing and given me a glimpse into the academic world. I want to collectively and separately thank everyone in the department for pleasant lunch and coffee breaks; and for creating a nice working environment.

Most importantly, I want thank my wife Miia and my son Jaakko for being in my life.

Espoo, April 2017 Elias Raninen

Contents

Symbols and abbreviations

Symbols

$\mathbb{R}$	field of real numbers
$\mathbb{C}$	field of complex numbers
$\jmath$	imaginary unit
x	generic scalar
x^*	complex conjugate of x
$\mathbf{x}$	generic vector
$\hat{\mathbf{x}}$	estimate of $\mathbf{x}$
$\mathbf{x}_S$	subvector of $\mathbf{x}$ consisting of elements indexed by the set S
$\mathbf{x}_{S^c}$	subvector of $\mathbf{x}$ consisting of elements indexed by the complement of the set S
$\mathbf{x}^\top$	transpose of vector $\mathbf{x}$
$\mathbf{x}^\mathsf{H}$	Hermitian transpose of vector $\mathbf{x}$, i.e., $\mathbf{x}^\mathsf{H} = (\mathbf{x}^*)^\top$
$\mathbf{X}$	generic matrix
$\mathbf{X}_{-j}$	matrix $\mathbf{X}$ with the j^{th} column removed
$\mathbf{X}^\top$	transpose of matrix $\mathbf{X}$
$\mathbf{X}^\mathsf{H}$	Hermitian transpose of matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	inverse of matrix $\mathbf{X}$
$\mathbf{X}_S$	submatrix of $\mathbf{X}$ consisting of columns indexed by the set S
$\mathbf{X}_{S^c}$	submatrix of $\mathbf{X}$ consisting of columns indexed by the complement of the set S
$(\mathbf{X})_{ij}$	the element of $\mathbf{X}$ corresponding to the i^{th} row and j^{th} column
$\mathbf{I}$	the identity matrix
$\operatorname{diag}(a_1, \ldots, a_n)$	a diagonal matrix whose diagonal entries are $a_1, \ldots, a_n$
$\lvert S \rvert$	cardinality of the set S
$\operatorname{Re}(\cdot)$	real part operator, i.e., $\operatorname{Re}(x) = (x + x^*)/2$
$\operatorname{Im}(\cdot)$	imaginary part operator, i.e., $\operatorname{Im}(x) = (x - x^*)/(2\jmath)$
$\lvert \cdot \rvert$	modulus of its argument, i.e., $\lvert x \rvert = \sqrt{\operatorname{Re}(x)^2 + \operatorname{Im}(x)^2}$
$\lVert\cdot\rVert_1$	ℓ_1-norm, i.e., $\lVert\mathbf{x}\rVert_1 = \sum_i \lvert x_i \rvert$
$\lVert\cdot\rVert_2$	ℓ_2-norm (Euclidean norm), i.e., $\lVert\mathbf{x}\rVert_2 = \sqrt{\mathbf{x}^\mathsf{H}\mathbf{x}}$
$\lVert\cdot\rVert_\infty$	the infinity norm, i.e., $\lVert\mathbf{x}\rVert_\infty = \max\{\lvert x_i \rvert, \ldots, \lvert x_n \rvert\}$
$1_A(\cdot)$	indicator function, i.e., 1 if its argument belongs to the set A, otherwise 0
$\operatorname{Trace}(\cdot)$	matrix trace
$\langle \cdot, \cdot \rangle$	Hermitian inner product, i.e., $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\mathsf{H}\mathbf{y}$
$\mathbb{E}[\cdot]$	the expectation operator
$\operatorname{Var}[\cdot]$	the variance operator
$\operatorname{Cov}[\cdot]$	the covariance operator
$\operatorname{sign}(\cdot)$	the sign operator, i.e., $\operatorname{sign}(x) = x/\lvert x \rvert$ for $x \neq 0$ and $\operatorname{sign}(x) = 0$ for $x = 0$
$(\cdot)_+$	the subplus operator, i.e., $(x)_+ = \max\{x, 0\}$ for $x \in \mathbb{R}$
$\mathcal{S}(\cdot, \lambda)$	the soft-thresholding operator, i.e., $\mathcal{S}(x, \lambda) = \operatorname{sign}(x)(\lvert x \rvert - \lambda)_+$
∇	gradient
∂	subdifferential
δ_{ij}	the Kronecker delta function, i.e., $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$
$\triangleq$	defined as

Abbreviations

BLUE	best linear unbiased estimator
CBF	conventional beamformer
CPR	correct peak rate
DOA	direction of arrival
EN	elastic net
FNR	false negative rate
FPR	false positive rate
i.i.d.	independent and identically distributed
lasso	least absolute shrinkage and selection operator
LSE	least squares estimate/estimator
MSE	mean squared error
MUSIC	multiple signal classification
MVDR	minimum variance distortionless response
MVLUE	minimum variance linear unbiased estimator
RE	restricted eigenvalues
RSS	residual sum of squares
SNR	signal-to-noise ratio
TDOA	time difference of arrival
ULA	uniform linear array

1 Introduction

We are concerned with the linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\mathbf{y}$ is an n-dimensional vector of response variables (measurements) and $\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_p)$ is a fixed $n \times p$ design matrix of features, $\boldsymbol{\beta}$ is a p-dimensional vector of unknown regression coefficients, and $\boldsymbol{\varepsilon}$ is an unobserved error vector consisting of i.i.d. random variables from a symmetric distribution with $\mathbb{E}[\varepsilon_i] = 0$ and $\operatorname{Var}[\varepsilon_i] = \sigma^2$, for all $i \in \{1, \ldots, n\}$. Furthermore, we assume without loss of generality, that the columns of the design matrix are normalized as $\|\mathbf{x}_j\|_2 = 1$, for $j \in \{1, \ldots, p\}$.

In this context, the measured data $\mathbf{y}$ can be any signal, for example, a recorded audio signal or the output from some transducers, as in sensor array processing. The columns of the design matrix $\mathbf{X}$ constitute a basis of possible features that are believed to construct the true signal. In the case that the true model generating the data is in fact linear, then the term $\mathbf{X}\boldsymbol{\beta}$ represents the true signal and the term $\boldsymbol{\varepsilon}$ represents the additive noise, which corrupts the observed measurement. Given this model, we may have different estimation objectives, which are described next.

If we need to predict the outcome given new data, then the problem is a *prediction problem*, and we want to find an estimator of the vector $\mathbf{X}\boldsymbol{\beta}$ which minimizes the *expected prediction error*

$$\mathbb{E}\left[\left\|\mathbf{X}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\right\|_2^2\right]. \tag{1.2}$$

If instead, we are interested in recovering the true coefficient vector $\boldsymbol{\beta}$, then we have a *parameter estimation problem* or an *inverse problem*, and we want to find an estimator which minimizes the *mean squared error* (MSE)

$$\mathbb{E}\left[\left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_2^2\right]. \tag{1.3}$$

If we are interested in finding only the correct set of indices corresponding to the non-zero or significantly large elements in the true coefficient vector, the problem is the *support recovery* or *variable selection* problem.

In real world problems, the linear model (1.1) is most certainly a simplification of the underlying true model. However, it may still be a good approximation. This is especially true in high-dimensional problems, where overfitting is a major concern. Simple methods, such as penalized linear regression has proven to be very useful in these cases [1].

1.1 Least squares estimation

Suppose we are given an estimate $\hat{\boldsymbol{\beta}}$ of the unknown coefficient vector $\boldsymbol{\beta}$. A way of measuring the goodness of the estimate is to evaluate a loss function which outputs a positive real quantity that reflects the quality of the estimate. In linear regression, by far the most popular loss function is the residual sum of squares (RSS), defined as

$$\mathrm{RSS}(\hat{\boldsymbol{\beta}}) \triangleq \sum_{i=1}^{n} (y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}})^2 = \left\| \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \right\|_2^2 = \|\mathbf{r}\|_2^2 ,$$

where $\mathbf{r} \triangleq \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ is the *residual vector* obtained by computing the difference of the observed data $\mathbf{y}$ and the linear model $\mathbf{X}\hat{\boldsymbol{\beta}}$ which is assumed to generate the data. The RSS is the squared Euclidean distance between the observed data and the assumed model.

The particular estimator, which minimizes the RSS,

$$\hat{\boldsymbol{\beta}}_{\text{ls}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \ \mathrm{RSS}(\boldsymbol{\beta}), \tag{1.4}$$

is called the *least squares estimator* (LSE). In a two dimensional problem, finding the least squares estimate corresponds to fitting a line to the data such that the squared Euclidean distances from the data points to the line are minimized. Analogously, in a three dimensional problem, finding the least squares estimate corresponds to fitting a plane to the data, which is illustrated in Figure 1.

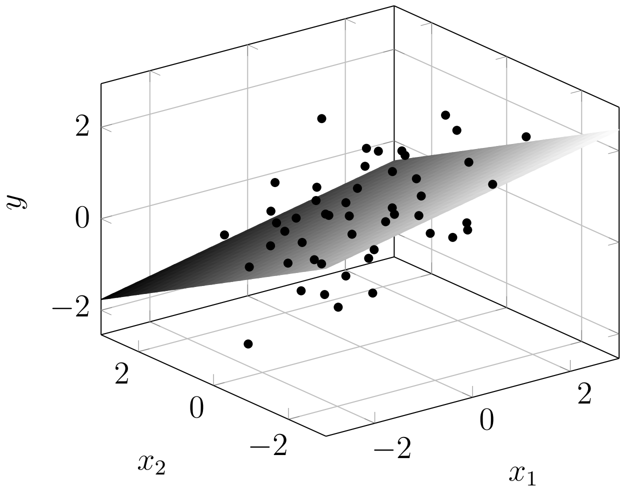

Figure 1: A plane fitted to the data samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ with the method of least squares.

The solution to the least squares problem can be derived straightforwardly. Since the RSS is a quadratic form, the minimum can be found by solving the zero gradient equation

$$\nabla_{\boldsymbol{\beta}} \mathrm{RSS}(\boldsymbol{\beta}) = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{ls}}) = 0,$$

which leads to the *normal equation*

$$\mathbf{X}^\top \mathbf{X} \hat{\boldsymbol{\beta}}_{\text{ls}} = \mathbf{X}^\top \mathbf{y}.$$

If $\mathbf{X}$ has full rank, then $\mathbf{X}^\top \mathbf{X}$ is invertible and the unique solution to the least squares problem is

$$\hat{\boldsymbol{\beta}}_{\text{ls}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{1.5}$$

However, if there exist linearly dependent columns in $\mathbf{X}$, it is not full rank, and the matrix $\mathbf{X}^\top\mathbf{X}$ will not be invertible. In that case, the least squares solution will be an infinite set of points. Note that this is always the case when $\mathbf{X}$ has more columns than rows ($p > n$), which occurs in high-dimensional problems.

The *fitted values* are defined as

$$\hat{\mathbf{y}} \triangleq \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{ls}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{P}\mathbf{y},$$

where $\mathbf{P} \triangleq \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is an idempotent and symmetric projection matrix, i.e., $\mathbf{P}^2 = \mathbf{P} = \mathbf{P}^\top$. The fitted values are projections of the observations $\mathbf{y}$ onto the subspace spanned by the columns of $\mathbf{X}$, as is illustrated in Figure 2.

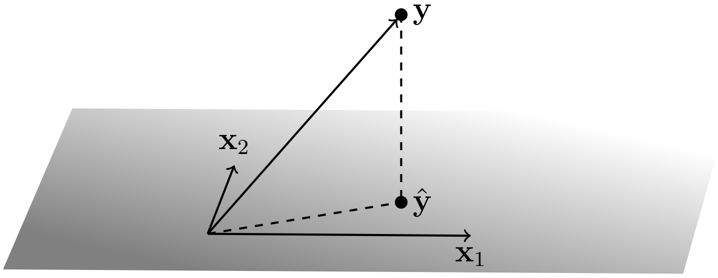

Figure 2: The observations $\mathbf{y}$ projected onto the column space of $\mathbf{X}$.

1.2 Bias and variance

Any estimator can be characterized by its *bias* and *variance*. In general, *unbiasedness* is a desirable property of an estimator since then the estimator will, on average, give the true value of the estimated parameter, i.e., $\mathbb{E}\left[\hat{\boldsymbol{\beta}}\right] = \boldsymbol{\beta}$. A small variance is also a desirable property of an estimator since the variance quantifies how much the value of the estimate varies among different realizations. The MSE (1.3), which measures the squared Euclidean distance between the unknown true vector and its estimate, can be decomposed into a squared bias term and a variance term as follows:

$$\begin{aligned}\mathbb{E}\left[\left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_2^2\right] &= \mathbb{E}\left[\sum_{i=1}^{p} (\beta_i - \hat{\beta}_i)^2\right] \\ &= \sum_{i=1}^{p} \left(\left(\mathbb{E}\left[\hat{\beta}_i\right] - \beta_i\right)^2 + \mathbb{E}\left[\hat{\beta}_i^2\right] - \mathbb{E}\left[\hat{\beta}_i\right]^2\right) \\ &= \sum_{i=1}^{p} \left(\left(\text{Bias}\left[\hat{\beta}_i\right]\right)^2 + \text{Var}\left[\hat{\beta}_i\right]\right).\end{aligned}$$

The best possible estimator is naturally the one which has the optimal balance between the two terms such that the MSE is minimized. If we narrow down and consider only linear estimators of the form $\hat{\boldsymbol{\beta}} = \mathbf{K}\mathbf{y}$, where the matrix $\mathbf{K} \in \mathbb{R}^{n\times p}$

transforms the data to the estimate, then the following also holds:

$$\begin{aligned}\mathbb{E}\left[\left\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}\right\|_2^2\right] &= \mathbb{E}\left[\|\boldsymbol{\beta}-\mathbf{K}\mathbf{y}\|_2^2\right] \\ &= \boldsymbol{\beta}^\top\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top\mathbf{K}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\mathbf{X}^\top\mathbf{K}^\top\mathbf{K}\mathbf{X}\boldsymbol{\beta} + \mathbb{E}\left[\boldsymbol{\varepsilon}^\top\mathbf{K}^\top\mathbf{K}\boldsymbol{\varepsilon}\right] \\ &= \underbrace{\|\boldsymbol{\beta}-\mathbf{K}\mathbf{X}\boldsymbol{\beta}\|_2^2}_{\text{squared bias term}} + \underbrace{\operatorname{Trace}\left(\mathbf{K}\boldsymbol{\Sigma}\mathbf{K}^\top\right)}_{\text{variance term}}, \end{aligned} \tag{1.6}$$

where $\boldsymbol{\Sigma}$ is the covariance of the (zero mean) error terms [2]. In the special case when $\mathbf{K}\mathbf{X} = \mathbf{I}$, the squared bias term in (1.6) equals zero and the estimator is unbiased. This is the case with the LSE, where we have $\mathbf{K}_{\text{ls}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$, that is,

$$\begin{aligned}\mathbb{E}\left[\hat{\boldsymbol{\beta}}_{\text{ls}}\right] &= \mathbb{E}[\mathbf{K}_{\text{ls}}\mathbf{y}] \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\mathbf{y}] \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}.\end{aligned}$$

The famous *Gauss-Markov theorem* states that, if the errors $(\boldsymbol{\varepsilon})_i = \varepsilon_i$, where $i = 1, 2, \ldots, n$, are i.i.d. random variables with zero mean and equal variance, then the LSE is the *best linear unbiased estimator* (BLUE) of $\boldsymbol{\beta}$. That is, no other unbiased estimator can attain a lower mean squared error (MSE). To show this, let us assume that the covariance matrix of the errors is positive definite, i.e., $\boldsymbol{\Sigma} \succ 0$. Then, the *minimum variance linear unbiased estimator* (MVLUE) can be found by solving the constrained optimization program

$$\begin{array}{ll}\text{minimize} & \operatorname{Trace}\left(\mathbf{K}\boldsymbol{\Sigma}\mathbf{K}^\top\right) \\ \text{subject to} & \mathbf{K}\mathbf{X} = \mathbf{I}.\end{array}$$

The solution can found by using the method of Lagrangian multipliers. The Lagrangian function with the variable $\mathbf{K}$ and the Lagrange multiplier $\mathbf{Z}$ is

$$\mathcal{L}(\mathbf{K},\mathbf{Z}) = \operatorname{Trace}\left(\mathbf{K}\boldsymbol{\Sigma}\mathbf{K}^\top\right) + \operatorname{Trace}\left(\mathbf{Z}(\mathbf{K}\mathbf{X}-\mathbf{I})\right).$$

The critical point of the Lagrangian can be found by differentiation with respect to $\mathbf{K}$:

$$\frac{\partial}{\partial \mathbf{K}}\mathcal{L}(\mathbf{K},\mathbf{Z}) = 2\boldsymbol{\Sigma}\mathbf{K}^\top + \mathbf{X}\mathbf{Z} = \mathbf{0},$$

from which we get

$$\mathbf{K} = -(1/2)\mathbf{Z}\mathbf{X}\boldsymbol{\Sigma}^{-1}. \tag{1.7}$$

By substituting the expression of $\mathbf{K}$ into the constraint $\mathbf{K}\mathbf{X} = \mathbf{I}$, we can solve for the Lagrange multiplier $\mathbf{Z}$:

$$\begin{aligned}\mathbf{K}\mathbf{X} = -(1/2)\mathbf{Z}\mathbf{X}\boldsymbol{\Sigma}^{-1}\mathbf{X} = \mathbf{I} \\ \Rightarrow \mathbf{Z} = -2(\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X})^{-1}.\end{aligned}$$

Substituting this into (1.7) gives $\mathbf{K} = \left(\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}$. Thus, the minimum variance linear unbiased estimator is

$$\hat{\boldsymbol{\beta}}_{\text{MVLUE}} = \mathbf{K}\mathbf{y} = \left(\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-1}\mathbf{X}^\top\boldsymbol{\Sigma}^{-1}\mathbf{y}.$$

It is easy to see that if $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, then we recover the LSE. Therefore, when the errors are i.i.d. with an equal variance, the LSE is the *best linear unbiased estimator* (BLUE).

The variance of the LSE can be obtained straightforwardly as

$$\begin{aligned}
\operatorname{Var}\left[\hat{\boldsymbol{\beta}}_{\text{ls}}\right] &= \operatorname{Var}\left[\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{y}\right] \\
&= \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\operatorname{Var}\left[\mathbf{y}\right]\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \\
&= \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\operatorname{Var}\left[\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\right]\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \\
&= \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top(\sigma^2\mathbf{I})\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1} \\
&= \sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}.
\end{aligned}$$

It is important to note that the variance depends on the inverse of the Gram matrix $\mathbf{X}^\top\mathbf{X}$. If the Gram matrix is badly conditioned, the variance of the estimator becomes too large which makes the estimate very inaccurate.

The LSE is a great method for approximating overdetermined inverse problems, that is, the case when $p < n$. However, as we will see in Section 2, it is possible to improve on the LSE in terms of MSE if we relax the condition of unbiasedness. By constraining or penalizing the norm of the coefficient vector while minimizing the RSS, it is possible to limit the variance of the estimator and acquire low error solutions, even in underdetermined inverse problems in which case $p > n$, and hence the LSE doesn't have a unique solution. However, in penalized linear regression, the level of penalization has to be somehow determined, which is complicated by the fact that in conventional penalized regression methods such as in ridge regression and the lasso, the optimal penalty level is well-known to depend on the unknown noise scale.

1.3 Contributions of the thesis

In *scaled sparse linear regression*, such as in scaled lasso and square-root lasso, the penalty parameter is independent of the noise scale, which enables the usage of predetermined universal penalty levels. The square-root lasso was extended to the group square-root lasso in [3]. The main contribution of this thesis, published in [4], is to extend the scaled lasso to the elastic net framework. Two potentially interesting elastic net extensions to the scaled lasso are proposed: *the scaled elastic net* and *the square-root elastic net.* Convergent algorithms are derived for their computation. The proposed estimators are able to outperform the scaled lasso, when there are high correlations among the variables, which is demonstrated with numerical examples and a single snapshot direction-of-arrival (DOA) estimation simulation study.

1.4 Structure of the thesis

The structure of this thesis is as follows. In Section 2, we discuss penalized linear regression. In Section 3, we discuss scaled sparse linear regression. In Section 4, we introduce the main contributions of this thesis, which are the scaled and square-root elastic net estimators [4]. In Section 5, we illustrate via numerical simulations that the proposed estimators outperform the scaled lasso in the case of highly correlated variables. Next, in Section 6, we briefly review $\mathbb{C}\mathbb{R}$-calculus and the complex gradient operator in order to extend the estimators for complex-valued data and linear model. In Section 7, we review the fundamentals of direction-of-arrival (DOA) estimation. In Section 8, we apply the proposed estimators to a single snapshot direction-of-arrival (DOA) estimation problem via numerical simulations. Finally, Section 9 concludes the thesis.

2 Penalized linear regression

As was shown in Section 1.2, the least squares estimator is the best linear unbiased estimator since it has the lowest variance, and thus the lowest MSE, of all unbiased linear estimators. However, by accepting some bias in the estimate, it is possible to achieve a lower variance than in the LSE, which may result in lower MSE. In practice, the trade-off of reducing the variance by introducing some bias is realized by imposing constraints on the minimization of the RSS objective function. For example, in a norm constrained minimization problem, we want to find the estimate

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p, \|\boldsymbol{\beta}\| \leq \gamma}{\arg\min}\ \mathrm{RSS}(\boldsymbol{\beta}),$$

where $\gamma \geq 0$ is a limit on the size of the norm of the coefficients. Alternatively, the same can be achieved by augmenting the RSS objective function (1.4) with a penalty term $\mathcal{P}(\boldsymbol{\beta})$ acting on the coefficients. The penalized optimization problem is of the form

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\ \mathrm{RSS}(\boldsymbol{\beta}) + \lambda \mathcal{P}(\boldsymbol{\beta}),$$

where $\lambda \geq 0$ is a tuning parameter, which determines the weighting between the two terms. The constraints or the penalty function can be chosen by using some *a priori* knowledge regarding the solution. For example, if we know that the solution should be sparse, then we should penalize the number of non-zero coefficients in the solution. The two most popular penalty functions are known as ridge regression [5] with $\mathcal{P}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2$, which penalizes vectors $\boldsymbol{\beta} \in \mathbb{R}^p$ with large ℓ_2-norm quadratically; and lasso [6] with $\mathcal{P}(\boldsymbol{\beta}) = 2\,\|\boldsymbol{\beta}\|_1$, which favours sparse solutions. The multiplicative constant 2 in the lasso penalty is included solely in order to simplify the mathematical expressions in further developments.

The ridge regression and lasso solutions can be visualized by plotting the RSS equicontours around the least squares solution, which for $n \geq p$ and $\mathbf{X}$ of full rank are ellipsoids, and then finding the point where the equicontour touches the edge of the constraint region. This is illustrated in Figure 3, where the optimization landscape is pictured in the two dimensional ($p = 2$) case. The ridge regression constraint is an Euclidean ℓ_2-ball centered at the origin, whereas the lasso constraint is a ℓ_1-ball. Due to the sharp corners of the ℓ_1-constraint in lasso, the minimum RSS equicontour is likely to touch the constraint region at an axis point, thus making the other coordinate zero. In higher dimensions, this phenomenon becomes more probable making the lasso a powerful sparsity inducing method.

2.1 Ridge regression

In the case when the basis vectors $\mathbf{x}_j$ are highly correlated, i.e., the columns of $\mathbf{X}$ are almost linearly dependent, the LSE is very susceptible to noise. This condition is usually referred to as *multicollinearity*, and it causes the Gram matrix $\mathbf{X}^\top\mathbf{X}$ to be badly conditioned. Thus the required inversion of the Gram matrix in the LSE (1.5) can make some of the regression coefficients become considerably large. In effect, this leads to high variability in the performance of the estimator.

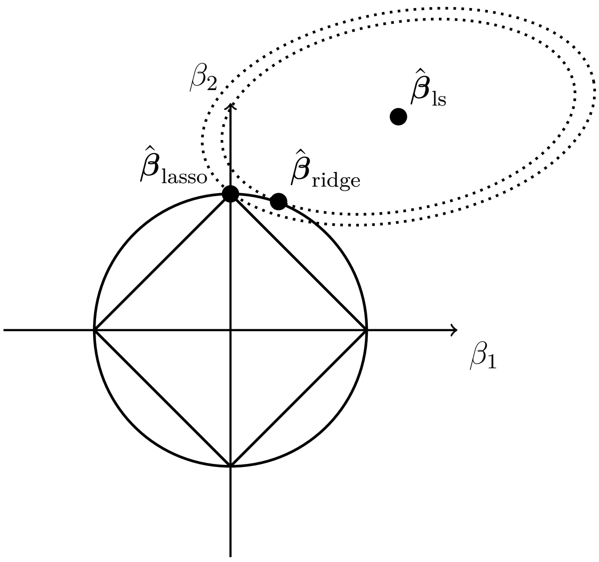
β_2
$\hat{\boldsymbol{\beta}}_{\text{ls}}$
$\hat{\boldsymbol{\beta}}_{\text{lasso}}$
$\hat{\boldsymbol{\beta}}_{\text{ridge}}$
β_1

Figure 3: A two dimensional visualization of the solutions to the least squares problem, ridge regression, and lasso.

Ridge regression [5] is a popular method which overcomes the aforementioned problems of inverting the ill-conditioned Gram matrix. Essentially, in ridge regression, the ℓ_2-norm of the coefficients are constrained to be within some limit when the RSS objective is minimized. The constrained form of the ridge regression optimization program is defined as

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \;\; \text{RSS}(\boldsymbol{\beta}) \;\; \text{subject to} \;\; \sum_{j=1}^{p} \beta_j^2 \leq \gamma, \tag{2.1}$$

where $\gamma \geq 0$ is the limit on how large the coefficient vector is allowed to grow (in Euclidean distance). The downside of constraining the coefficients is that it introduces bias to the estimate. However, the reduction in variance will usually outweight the effect of the bias, and thus reduce the overall MSE. Most commonly, the ridge regression estimate is formulated in the penalized, or Lagrangian, form

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}} \;\; \text{RSS}(\boldsymbol{\beta}) \; + \lambda \left\|\boldsymbol{\beta}\right\|_2^2, \tag{2.2}$$

where $\lambda \geq 0$ is a tuning parameter which controls the trade-off between the bias and the variance of the estimator. It can be shown that there is a one-to-one correspondence between λ in (2.2) and γ in (2.1); see, e.g., [1].

Since the penalized ridge regression optimization program (2.2) is convex, the solution can be derived simply by computing the gradient of the objective function with respect to $\boldsymbol{\beta}$ and finding the stationary point. The gradient of (2.2) is

$$\begin{aligned}\nabla_{\boldsymbol{\beta}} \left(\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2^2 + \lambda \left\|\boldsymbol{\beta}\right\|_2^2 \right) &= -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + 2\lambda\hat{\boldsymbol{\beta}} \\ &= -2\mathbf{X}^\top\mathbf{y} + 2(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})\hat{\boldsymbol{\beta}}.\end{aligned}$$

By equating this to zero and solving for $\hat{\boldsymbol{\beta}}$, we obtain the ridge regression estimate

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}. \tag{2.3}$$

As can be observed from (2.3), a constant value λ is added to the diagonal of the Gram matrix $\mathbf{X}^\top\mathbf{X}$ prior to its inversion. Therefore, even in the case that $\mathbf{X}$ is not full rank, and hence the Gram matrix is not invertible, the ridge estimator will exist. Furthermore, since a constant value is added to the diagonal of the Gram matrix, the ridge regression estimate is dependent of the scaling of the columns of $\mathbf{X}$. It is therefore important to normalize the columns prior to computing the ridge estimate.

2.1.1 Choosing the tuning parameter

In the special case when $\mathbf{X}$ is orthonormal, we have an analytical solution to the optimal tuning parameter in terms of the MSE. The ridge estimate can be written as

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{K}\mathbf{y},$$

where $\mathbf{K} = \left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top$. If $\mathbf{X}$ is orthonormal, then $\mathbf{K}$ simplifies to $\mathbf{K} = (1+\lambda)^{-1}\mathbf{X}^\top$ and the ridge estimate becomes

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \frac{\mathbf{X}^\top\mathbf{y}}{1+\lambda} = \frac{\hat{\boldsymbol{\beta}}_{\text{ls}}}{1+\lambda}. \tag{2.4}$$

Using (1.6) and some simple algebra, the expression for the MSE in the orthonormal case is easily obtained as

$$\begin{aligned} \text{MSE}(\hat{\boldsymbol{\beta}}) &= \left\|\boldsymbol{\beta} - \frac{1}{1+\lambda}\boldsymbol{\beta}\right\|_2^2 + \sigma^2 \operatorname{Trace}\left(\frac{1}{(1+\lambda)^2}\mathbf{I}\right) \\ &= \frac{1}{(\lambda+1)^2}\left(\lambda^2 \|\boldsymbol{\beta}\|_2^2 + p\sigma^2\right). \end{aligned} \tag{2.5}$$

The optimal tuning parameter can be found simply by solving the zero derivative equation of (2.5) with respect to λ, yielding

$$\lambda_{\text{opt}} = \frac{p\sigma^2}{\|\boldsymbol{\beta}\|_2^2}. \tag{2.6}$$

In the general case when $\mathbf{X}$ is not orthogonal, there is no closed form solution for the optimal tuning parameter. However, if we assume $n \geq p$, an optimal tuning parameter can be derived for a general form of ridge regression [5]. Let $\mathbf{X}^\top\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{U}^\top$ be an eigenvalue decomposition of the Gram matrix, where $\mathbf{U}$ is a unitary matrix of eigenvectors and $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_p)$ is a diagonal matrix of eigenvalues. Then, by making the change of variables $\boldsymbol{\alpha} = \mathbf{U}^\top\boldsymbol{\beta}$ and $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$, the Gram matrix of $\tilde{\mathbf{X}}$ becomes diagonal, i.e., $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}} = \mathbf{U}^\top\mathbf{X}^\top\mathbf{X}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U}\mathbf{D}\mathbf{U}^\top\mathbf{U} = \mathbf{D}$. The general form of ridge regression can then be defined as

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^p}{\text{minimize}} \; \left\|\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\alpha}\right\|_2^2 + \boldsymbol{\alpha}^\top\mathbf{T}\boldsymbol{\alpha}, \tag{2.7}$$

where $\mathbf{T} = \operatorname{diag}(t_1, t_2, \ldots, t_p)$ is a diagonal matrix. The solution to (2.7) is easily obtained as

$$\hat{\boldsymbol{\alpha}} = \left(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} + \mathbf{T}\right)^{-1} \tilde{\mathbf{X}}^\top \mathbf{y} = (\mathbf{D} + \mathbf{T})^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}.$$

The solution only contains a simple inversion of a diagonal matrix. As before, the MSE of the estimator can be computed using (1.6), with $\mathbf{K} = (\mathbf{D} + \mathbf{T})^{-1}\tilde{\mathbf{X}}^\top$, as

$$\begin{aligned}
\operatorname{MSE}(\hat{\boldsymbol{\alpha}}) &= \left\|\boldsymbol{\alpha} - \mathbf{K}\tilde{\mathbf{X}}\boldsymbol{\alpha}\right\|_2^2 + \sigma^2 \operatorname{Trace}\left(\mathbf{K}\mathbf{K}^\top\right) \\
&= \left\|\boldsymbol{\alpha} - (\mathbf{D} + \mathbf{T})^{-1}\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}\boldsymbol{\alpha}\right\|_2^2 + \sigma^2 \operatorname{Trace}\left((\mathbf{D} + \mathbf{T})^{-1}\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}(\mathbf{D} + \mathbf{T})^{-T}\right) \\
&= \left\|\left(\mathbf{I} - (\mathbf{D} + \mathbf{T})^{-1}\mathbf{D}\right)\boldsymbol{\alpha}\right\|_2^2 + \sigma^2 \operatorname{Trace}\left((\mathbf{D} + \mathbf{T})^{-2}\mathbf{D}\right) \\
&= \sum_{i=1}^p \alpha_i^2 \left(1 - \frac{d_i}{(d_i + t_i)}\right)^2 + \sigma^2 \sum_{i=1}^p \frac{d_i}{(d_i + t_i)^2} \\
&= \sum_{i=1}^p \alpha_i^2 \frac{t_i^2}{(d_i + t_i)^2} + \sigma^2 \sum_{i=1}^p \frac{d_i}{(d_i + t_i)^2}. \qquad (2.8)
\end{aligned}$$

The optimal values of t_i, where $i \in \{1, \ldots, p\}$, can be found by setting the derivative of (2.8) equal to zero. Thus, the optimal penalty parameter t_j regarding the coefficient α_j, where $j \in \{1, \ldots, p\}$, is solved from

$$0 = \frac{\partial}{\partial t_j}\operatorname{MSE}(\hat{\boldsymbol{\alpha}}) = 2\alpha_j^2 \frac{t_j d_j}{(d_j + t_j)^3} - 2\sigma^2 \frac{d_j}{(d_j + t_j)^3},$$

which yields

$$t_j = \frac{\sigma^2}{\alpha_j^2}. \qquad (2.9)$$

Unfortunately, since the terms σ^2 and α_j^2 are unknown, the optimal tuning parameter can not be implemented as such.

There have been, however, several proposals of theoretically justified tuning parameters in the literature; see, e.g., [7]. The original authors of ridge regression suggested to use the harmonic mean of the optimal tuning parameter (2.9) [8]:

$$\hat{\lambda} = \frac{p\sigma^2}{\sum_{j=1}^p \alpha_j^2} = \frac{p\sigma^2}{\boldsymbol{\alpha}^\top \boldsymbol{\alpha}} = \frac{p\sigma^2}{\boldsymbol{\beta}^\top \boldsymbol{\beta}} = \frac{p\sigma^2}{\|\boldsymbol{\beta}\|_2^2},$$

where the variance σ^2 is replaced by the sample variance $s^2 = \left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{ls}}\right\|_2^2 / (n - p)$ and $\boldsymbol{\beta}$ is replaced by the least squares solution $\hat{\boldsymbol{\beta}}_{\text{ls}}$. An apparent drawback of this implementation is that it doesn't allow for the case $p > n$. Additionally, in the case of multicollinearity, i.e., when ridge regression would be most useful, the least squares estimate $\hat{\boldsymbol{\beta}}_{\text{ls}}$ and s^2 will be way off the mark. Therefore, in real applications, the tuning parameter is usually chosen by cross-validation rather than on a theoretical basis.

2.2 Lasso

The lasso [6] (least absolute shrinkage and selection operator) is a regression method for the linear model in which the RSS objective function is minimized subject to a constraint on the ℓ_1-norm of the coefficients. The lasso estimate is defined as the minimizer of the criterion

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\text{minimize}}\ \text{RSS}(\boldsymbol{\beta})\ \text{ subject to }\ \sum_{j=1}^{p}|\beta_j| \leq \gamma. \tag{2.10}$$

By choosing the constant $\gamma \geq 0$ appropriately, it is possible to induce sparse solutions of the coefficient vector in which there are only a small number of non-zero elements. The lasso is therefore a regression method which incorporates a variable selection property.

The lasso is usually formulated in the penalized, or Lagrangian, form:

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\ \text{RSS}(\boldsymbol{\beta}) + 2\lambda\left\|\boldsymbol{\beta}\right\|_1, \tag{2.11}$$

where $\lambda \geq 0$ is a parameter that controls the trade-off between the RSS error and the level of sparsity in the solution. These two different formulations, (2.10) and (2.11), are equivalent, and it can be shown that for every γ in (2.10) there exist a unique λ in (2.11) which will give the same solution.

The lasso optimization problem (2.11) is convex. However, since the ℓ_1-norm penalty is not differentiable at the origin, we will make use of subdifferentials which generalize the concept of the differential for convex functions that are not differentiable everywhere. See Appendix A for the definition of subdifferentials. The minimizer of (2.11) satisfies the zero subgradient equation

$$\partial_{\boldsymbol{\beta}}\left(\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2^2 + 2\lambda\left\|\boldsymbol{\beta}\right\|_1\right) \in \mathbf{0},$$

which yields

$$\mathbf{X}^\top\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\right) = \lambda\hat{\mathbf{t}}, \tag{2.12}$$

where $\hat{\mathbf{t}} = \left(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_p\right)^\top$ is a vector whose j^{th} element belongs to the subdifferential of the modulus of $\hat{\beta}_j$, that is,

$$\hat{t}_j = \partial|\hat{\beta}_j| = \begin{cases} \text{sign}\left(\hat{\beta}_j\right) & \hat{\beta}_j \neq 0 \\ \{z\in\mathbb{R} : |z| \leq 1\} & \hat{\beta}_j = 0. \end{cases}$$

The equation (2.12) has an analytical solution in the special case when the columns of the design matrix are orthogonal, or, if there is only a single predictor, i.e., $\mathbf{X} \in \mathbb{R}^{n\times 1}$. In the general case, when the columns of the design matrix are not orthogonal, the lasso estimate can be found, e.g., by applying an iterative *cyclic coordinate-wise descent* [9] algorithm which is the topic of Section 2.2.2. However, first we study the single predictor case.

2.2.1 Single predictor case

When there is only a single predictor, the lasso estimate has an analytical solution. The lasso estimate with a single predictor is

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta \in \mathbb{R}} \ \|\mathbf{y} - \mathbf{x}\beta\|_2^2 + 2\lambda|\beta|. \tag{2.13}$$

The minimizer of (2.13) satisfies $\mathbf{x}^\top \left(\mathbf{y} - \mathbf{x}\hat{\beta}\right) = \lambda\hat{t}$, from which solving for $\hat{\beta}$ yields

$$\hat{\beta} = \frac{\mathbf{x}^\top\mathbf{y} - \lambda\hat{t}}{\mathbf{x}^\top\mathbf{x}} = \mathbf{x}^\top\mathbf{y} - \lambda\hat{t}.$$

If we assume $\hat{\beta} \neq 0$, then the subdifferential is $\hat{t} = \operatorname{sign}\left(\hat{\beta}\right) = \hat{\beta}/|\hat{\beta}|$, and we have

$$\hat{\beta} = \mathbf{x}^\top\mathbf{y} - \lambda\frac{\hat{\beta}}{|\hat{\beta}|}.$$

By collecting the terms containing $\hat{\beta}$ to the left-hand side of the equation yields

$$\hat{\beta}\left(1 + \frac{\lambda}{|\hat{\beta}|}\right) = \mathbf{x}^\top\mathbf{y}. \tag{2.14}$$

Taking the absolute value of both sides of the equation and solving for $|\hat{\beta}|$ yields

$$|\hat{\beta}| = |\mathbf{x}^\top\mathbf{y}| - \lambda. \tag{2.15}$$

Since we assumed $\hat{\beta} \neq 0$, we must have the condition $|\mathbf{x}^\top\mathbf{y}| > \lambda$. By substituting (2.15) into (2.14), and some simple algebra, we obtain the solution:

$$\hat{\beta}_{\text{lasso}} = \begin{cases} \operatorname{sign}\left(\mathbf{x}^\top\mathbf{y}\right)\left(|\mathbf{x}^\top\mathbf{y}| - \lambda\right), & |\mathbf{x}^\top\mathbf{y}| > \lambda \\ 0, & |\mathbf{x}^\top\mathbf{y}| \leq \lambda, \end{cases}$$

which can be conveniently expressed as

$$\hat{\beta}_{\text{lasso}} = \mathcal{S}\left(\mathbf{x}^\top\mathbf{y}, \lambda\right), \tag{2.16}$$

where $\mathcal{S}(\cdot, \lambda) \triangleq \operatorname{sign}(\cdot)(|\cdot| - \lambda)_+$ is the soft-thresholding operator and $(\cdot)_+ \triangleq \max\{\cdot, 0\}$.

2.2.2 Non-orthogonal case

For the general case where the columns of $\mathbf{X}$ are not orthogonal, it is possible to derive an iterative coordinate-wise algorithm in which the criterion function is minimized with respect to one coordinate at a time. The derivation presented here is similar to that of [10].

Let the vector $\boldsymbol{\beta}_{-j}$ denote the vector $\boldsymbol{\beta}$ with its j^{th} element removed. Likewise, let $\mathbf{X}_{-j}$ denote the matrix $\mathbf{X}$ with its j^{th} column removed. Using this notation, we can rewrite the lasso (2.11) as

$$\left\|\mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j} - \mathbf{x}_j\beta_j\right\|_2^2 + 2\lambda\left(\left\|\boldsymbol{\beta}_{-j}\right\|_1 + |\beta_j|\right). \tag{2.17}$$

Let us next define the partial residual as $\mathbf{r}^{(j)} \triangleq \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j}$. By treating all other variables fixed except for β_j, the minimizer of (2.17) with respect to β_j satisfies

$$\mathbf{x}_j^\top \mathbf{r}^{(j)} = \lambda\hat{t}_j,$$

where $\hat{t}_j$ belongs to the subdifferential of $|\hat{\beta}_j|$. Thus, the problem reduces to a single predictor problem similar to that of Section 2.2.1. The solution for $\hat{\beta}_j$ is therefore simply

$$\hat{\beta}_j = \mathcal{S}\left(\mathbf{x}_j^\top \mathbf{r}^{(j)}, \lambda\right), \tag{2.18}$$

which can (because of the normalization $\mathbf{x}_j^\top \mathbf{x}_j = 1$) be expressed in terms of the full residual, $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$, as

$$\hat{\beta}_j = \mathcal{S}\left(\hat{\beta}_j + \mathbf{x}_j^\top \mathbf{r}, \lambda\right). \tag{2.19}$$

By updating the coefficients $\hat{\beta}_j$ sequentally via solving (2.18) for each j in a cyclical manner, we eventually arrive at the solution of the lasso (2.11). This iterative minimization procedure is known as *cyclic coordinate-wise descent* (CCD).

2.2.3 A basic error bound and tuning parameter selection

In this section, we derive a basic consistency result for the lasso. However, in order to carry out the derivations, we first need to introduce the concept of *restricted eigenvalues* as well as some notation.

Assuming a sparse coefficient vector $\boldsymbol{\beta}$ with many exactly zero elements, let $S \in \{1, \ldots, p\}$ denote a support set consisting of the indices of the non-zero elements of $\boldsymbol{\beta}$, and let S^c denote the complement of the set S. The notation $\boldsymbol{\beta}_S \in \mathbb{R}^{|S|}$ then refers to a subvector, which consists only of the particular elements of $\boldsymbol{\beta}$ indexed by the set S.

We will proceed by defining the concepts of *strong convexity* and *the restricted eigenvalue condition*, whereafter we derive a bound on the ℓ_2-error of the lasso. The primary source used for this section is [10]. A more comprehensive reference on lasso theory can be found from [11].

Restricted eigenvalues is a condition related to strong convexity. For example, in order that the LSE is unique, the RSS criterion must be strictly convex. In other words, it is required that the Hessian of the RSS criterion, $\nabla_{\beta}^2 \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = 2\mathbf{X}^\top\mathbf{X}$, is positive definite, i.e., the matrix $\mathbf{X}^\top\mathbf{X}$ has only positive eigenvalues. If we denote the smallest eigenvalue of $\mathbf{X}^\top\mathbf{X}$ with γ, we can say that the RSS criterion is strongly convex with the parameter γ. Thus, we have the following definition:

Definition 1. *A function $f : \mathbb{R}^p \to \mathbb{R}$ with parameter $\gamma > 0$ is said to be strongly convex at $\boldsymbol{\beta} \in \mathbb{R}^p$ if*

$$f(\boldsymbol{\beta}') - f(\boldsymbol{\beta}) \geq \nabla f(\boldsymbol{\beta})^\top (\boldsymbol{\beta}' - \boldsymbol{\beta}) + \frac{\gamma}{2} \|\boldsymbol{\beta}' - \boldsymbol{\beta}\| \tag{2.20}$$

holds for all $\boldsymbol{\beta}' \in \mathbb{R}^p$.

In high-dimensional linear regression in which the number of variables is larger than the number of samples ($p > n$), the rank of $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{p \times p}$ is at most n, and the nullspace, $\{\boldsymbol{\nu} \in \mathbb{R}^p, \boldsymbol{\nu} \neq \mathbf{0} : \mathbf{X}^\top \mathbf{X} \boldsymbol{\nu} = \mathbf{0}\}$, is of dimension $p - n$. In that case $\mathbf{X}^\top \mathbf{X}$ is obviously not strongly convex. Therefore, we need a form of restricted strong convexity. For the linear model, this can be described by the *restricted eigenvalues* (RE) condition with respect to the set $\mathcal{C}$, which requires

$$\frac{\|\mathbf{X}\boldsymbol{\nu}\|_2^2}{\|\boldsymbol{\nu}\|_2^2} \geq \gamma \tag{2.21}$$

to hold for all non-zero vectors $\boldsymbol{\nu} \in \mathcal{C}$ and $\gamma > 0$.

Let us assume that the true coefficient vector $\boldsymbol{\beta}$ is sparse with the support set S. Furthermore, let us define the lasso error vector $\hat{\boldsymbol{\nu}} \triangleq \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$. As will be shown in Theorem 1, the lasso error is restricted to a cone set of the form

$$\mathcal{C}(S; \zeta) \triangleq \{\hat{\boldsymbol{\nu}} \in \mathbb{R}^p : \|\hat{\boldsymbol{\nu}}_{S^c}\|_1 \leq \zeta \|\hat{\boldsymbol{\nu}}_S\|_1\}, \tag{2.22}$$

where $\zeta \geq 1$ is a constant whose value depends on the chosen penalty parameter λ.

Theorem 1. *If the design matrix $\mathbf{X}$ satisfies the restricted eigenvalue condition with $\gamma > 0$ over $\mathcal{C}(S; 3)$, and we choose the tuning parameter $\lambda \geq 2 \left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty$, we obtain the following lower bound:*

$$\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|_2 \leq \frac{3}{\gamma}\sqrt{|S|}\lambda,$$

where $|S|$ is the cardinality of S, i.e., the number of non-zero elements in the true coefficient vector $\boldsymbol{\beta}$.

Proof. Let $\hat{\boldsymbol{\nu}} \triangleq \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$. We define the function

$$f(\boldsymbol{\nu}) \triangleq \|\mathbf{y} - \mathbf{X}(\boldsymbol{\beta} + \boldsymbol{\nu})\|_2^2 + 2\lambda \|\boldsymbol{\beta} + \boldsymbol{\nu}\|_1.$$

Since $\hat{\boldsymbol{\beta}}$ is the minimizer of the lasso objective, we have $f(\hat{\boldsymbol{\nu}}) \leq f(\mathbf{0})$, that is,

$$\|\mathbf{y} - \mathbf{X}(\boldsymbol{\beta} + \hat{\boldsymbol{\nu}})\|_2^2 + 2\lambda \|\boldsymbol{\beta} + \hat{\boldsymbol{\nu}}\|_1 \leq \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + 2\lambda \|\boldsymbol{\beta}\|_1.$$

Rearranging and some algebra yields,

$$\frac{\|\mathbf{X}\hat{\boldsymbol{\nu}}\|_2^2}{2} \leq \boldsymbol{\varepsilon}^\top \mathbf{X}\hat{\boldsymbol{\nu}} + \lambda \left(\|\boldsymbol{\beta}\|_1 - \|\boldsymbol{\beta} + \hat{\boldsymbol{\nu}}\|_1\right).$$

Using the Hölder inequality, we have $\boldsymbol{\varepsilon}^\top \mathbf{X}\hat{\boldsymbol{\nu}} \leq \left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty \|\hat{\boldsymbol{\nu}}\|_1$. Furthermore, since $\|\boldsymbol{\beta}\|_1 = \|\boldsymbol{\beta}_S\|_1$, we also have the inequality $\|\boldsymbol{\beta} + \hat{\boldsymbol{\nu}}\|_1 = \|\boldsymbol{\beta}_S + \hat{\boldsymbol{\nu}}_S\|_1 + \|\hat{\boldsymbol{\nu}}_{S^c}\|_1 \geq \|\boldsymbol{\beta}_S\|_1 - \|\hat{\boldsymbol{\nu}}_S\|_1 + \|\hat{\boldsymbol{\nu}}_{S^c}\|_1$. Putting these together yields

$$\frac{\|\mathbf{X}\hat{\boldsymbol{\nu}}\|_2^2}{2} \leq \left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty \|\hat{\boldsymbol{\nu}}\|_1 + \lambda \left(\|\hat{\boldsymbol{\nu}}_S\|_1 - \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right). \tag{2.23}$$

Since by assumption, we have $\lambda \geq 2\left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty$, we can write

$$\begin{aligned}
\frac{\|\mathbf{X}\hat{\boldsymbol{\nu}}\|_2^2}{2} &\leq \frac{\lambda}{2}\|\hat{\boldsymbol{\nu}}\|_1 + \lambda \left(\|\hat{\boldsymbol{\nu}}_S\|_1 - \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right) \\
&= \frac{\lambda}{2}\left(\|\hat{\boldsymbol{\nu}}_S\|_1 + \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right) + \lambda \left(\|\hat{\boldsymbol{\nu}}_S\|_1 - \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right) \\
&= \frac{3}{2}\lambda \|\hat{\boldsymbol{\nu}}_S\|_1 - \frac{1}{2}\lambda \|\hat{\boldsymbol{\nu}}_{S^c}\|_1 \\
&\leq \frac{3}{2}\lambda \|\hat{\boldsymbol{\nu}}_S\|_1 .
\end{aligned}$$

Using the Cauchy-Schwartz inequality, we have

$$\|\hat{\boldsymbol{\nu}}_S\|_1 = \sum_{i\in S} |1 \cdot \hat{\nu}_i| \leq \sqrt{\sum_{i\in S} 1^2}\sqrt{\sum_{i\in S} |\hat{\nu}_i|^2} \leq \sqrt{|S|}\sqrt{\sum_{i\in p} |\hat{\nu}_i|^2} = \sqrt{|S|}\,\|\hat{\boldsymbol{\nu}}\|_2 .$$

Thus, if the restricted eigenvalue condition (2.21) holds, we have

$$\frac{\gamma}{2}\|\hat{\boldsymbol{\nu}}\|_2^2 \leq \frac{\|\mathbf{X}\hat{\boldsymbol{\nu}}\|_2^2}{2} \leq \frac{3}{2}\sqrt{|S|}\lambda \|\hat{\boldsymbol{\nu}}\|_2 .$$

By rearranging the terms, we get the ℓ_2-error bound:

$$\left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_2 \leq \frac{3}{\gamma}\sqrt{|S|}\lambda. \tag{2.24}$$

In order to show that the lasso error $\hat{\boldsymbol{\nu}} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$ belongs to the cone $\mathcal{C}(S;3)$, we first observe that (2.23) implies

$$0 \leq \frac{\lambda}{2}\left(\|\hat{\boldsymbol{\nu}}_S\|_1 + \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right) + \lambda \left(\|\hat{\boldsymbol{\nu}}_S\|_1 - \|\hat{\boldsymbol{\nu}}_{S^c}\|_1\right).$$

By rearranging the terms, we obtain

$$\|\hat{\boldsymbol{\nu}}_{S^c}\|_1 \leq 3\|\hat{\boldsymbol{\nu}}_S\|, \tag{2.25}$$

which completes the proof. □

In Theorem 1, the tuning parameter was chosen to be twice the maximum correlation between the noise and the predictors, i.e., $\lambda = 2\left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty$, which is obviously an unknown quantity. However, if we assume that the noise vector $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ has independent and identically distributed (i.i.d.) Gaussian entries with zero mean and variance σ^2, then the affine transformation $\mathbf{x}_j^\top \boldsymbol{\varepsilon}$ is distributed as $\mathcal{N}\left(0, \sigma^2 \|\mathbf{x}_j\|_2^2\right)$, which reduces to $\mathbf{x}_j^\top \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2)$ because of the normalization $\|\mathbf{x}_j\| = 1$. Using a well-known *concentration inequality* for Gaussian random variables, we have

$$\mathbb{P}\Big[|\mathbf{x}_j^\top \boldsymbol{\varepsilon}| \geq t\Big] \leq 2e^{-\frac{t^2}{2\sigma^2}},$$

for $t \geq 0$. Taking the union bound over all predictors yields

$$\begin{aligned}
\mathbb{P}\Big[\left\|\mathbf{X}^\top \boldsymbol{\varepsilon}\right\|_\infty \geq t\Big] &\leq 2pe^{-\frac{t^2}{2\sigma^2}} \\
&= 2e^{-\frac{t^2}{2\sigma^2}+\log(p)} \\
&= 2e^{-\frac{1}{2}\left(\frac{t^2}{\sigma^2\log(p)}-2\right)\log(p)} \\
&= 2e^{-\frac{1}{2}(\tau-2)\log(p)},
\end{aligned}$$

where $t = \sigma\sqrt{\tau \log(p)}$. By choosing the tuning parameter as

$$\lambda = 2\sigma\sqrt{\tau \log(p)}, \tag{2.26}$$

the error bound

$$\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|_2 \leq \frac{6\sigma}{\gamma}\sqrt{|S|\tau\log(p)} \tag{2.27}$$

will hold with probability at least $1 - 2e^{-\frac{1}{2}(\tau-2)\log(p)}$. Depending on the number of predictors p, a value of $\tau > 2$ yields reasonable bounds.

2.3 Elastic net

One deficiency of the lasso is that it performs poorly under the conditions of multicollinearity, which is the case where ridge regression proves to be most useful. The elastic net [12] (EN) is a popular regularization and variable selection method which merges the useful properties of ridge regression and lasso, namely it is able to handle multicollinearity and it possesses the variable selection property. The elastic net estimator is defined as

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \ \mathrm{RSS}(\boldsymbol{\beta}) + \lambda\left((1-\alpha)\|\boldsymbol{\beta}\|_2^2 + 2\alpha\|\boldsymbol{\beta}\|_1\right), \tag{2.28}$$

where $\alpha \in [0,1]$ is an elastic net tuning parameter that controls the mixing between the ℓ_1-norm and ℓ_2-norm terms in the penalty. The elastic net constraint region induced by the penalty function is shown in Figure 4 along with lasso, ridge, and ℓ_q-ball constraint regions. The elastic net constraint region is curved making it more robust to multicollinearity. It also possesses sharp corners at the axes due to the ℓ_1-norm term in the penalty. This gives it the variable selection property. It should be noted that, the ℓ_q-ball penalty with $q > 1$, although similar looking as the elastic net penalty, does not have sharp corners, and thus does not yield sparse solutions

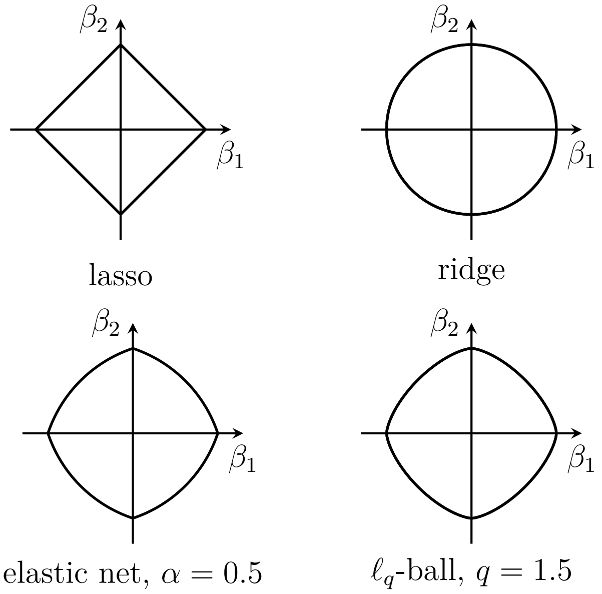

Figure 4: The constraint regions of lasso, ridge, elastic net, and ℓ_q-norm penalties.

with exactly zero coefficients. For an analysis of ℓ_2-error bounds for the elastic net, see [13] and [14].

As with the lasso, the elastic net can be solved via a coordinate-wise descent algorithm, which we will derive next. We rewrite the criterion (2.28) as

$$\left\|\mathbf{r}^{(j)} - \mathbf{x}_j\beta_j\right\|_2^2 + \lambda(1-\alpha)\left(\left\|\boldsymbol{\beta}_{-j}\right\|_2^2 + \beta_j^2\right) + 2\lambda\alpha\left(\left\|\boldsymbol{\beta}_{-j}\right\|_1 + |\beta_j|\right), \tag{2.29}$$

where $\mathbf{r}^{(j)} = \mathbf{y} - \mathbf{X}_{-j}\boldsymbol{\beta}_{-j}$ is the partial residual. The zero subgradient equation of (2.29) with respect to β_j satisfies

$$-\mathbf{x}_j^\top\left(\mathbf{r}^{(j)} - \mathbf{x}_j\hat{\beta}_j\right) + \lambda(1-\alpha)\hat{\beta}_j + \lambda\alpha\hat{t}_j = 0, \tag{2.30}$$

where $\hat{t}_j$ belongs to the subdifferential of $|\hat{\beta}_j|$. If we assume that $\hat{\beta}_j \neq 0$, then we have

$$\left(1 + \lambda(1-\alpha) + \frac{\lambda\alpha}{|\hat{\beta}_j|}\right)\hat{\beta}_j = \mathbf{x}_j^\top\mathbf{r}^{(j)}. \tag{2.31}$$

By taking the modulus of both sides of the equation and solving for $|\hat{\beta}_j|$ yields

$$|\hat{\beta}_j| = \frac{|\mathbf{x}_j^\top\mathbf{r}^{(j)}| - \lambda\alpha}{1 + \lambda(1-\alpha)}. \tag{2.32}$$

This shows that when $\hat{\beta}_j \neq 0$, we must have $|\mathbf{x}_j^\top\mathbf{r}^{(j)}| > \lambda\alpha$; and when $|\mathbf{x}_j^\top\mathbf{r}^{(j)}| \leq \lambda\alpha$, we must have $\hat{\beta}_j = 0$. Substituting (2.32) into (2.31) and some algebra yields the solution

$$\hat{\beta}_j = \frac{\mathbf{x}_j^\top\mathbf{r}^{(j)}}{|\mathbf{x}_j^\top\mathbf{r}^{(j)}|}\frac{\left(|\mathbf{x}_j^\top\mathbf{r}^{(j)}| - \lambda\alpha\right)_+}{1 + \lambda(1-\alpha)},$$

which can be expressed with the full residual $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ and the soft-thresholding operator in the form

$$\hat{\beta}_j = \frac{\mathcal{S}\left(\hat{\beta}_j + \mathbf{x}_j^\top \mathbf{r}, \lambda\alpha\right)}{1 + \lambda(1-\alpha)}. \tag{2.33}$$

By updating each coefficient cyclically, we eventually reach a stationary point which is the solution to the elastic net (2.28).

2.3.1 The double shrinkage effect

The elastic net estimator might suffer from a double shrinkage effect due to using both ℓ_1-norm and ℓ_2-norm penalties. To see this, consider the case that the design matrix is orthonormal and $n = p$, i.e., $\mathbf{x}_i^\top \mathbf{x}_j = \delta_{ij}$. The argument of the soft-thresholding operator then simplifies to $\mathbf{x}_j^\top \mathbf{y}$. Thus, we have the non-iterative explicit solution

$$\hat{\boldsymbol{\beta}}_{\text{EN}} = \frac{\mathcal{S}\left(\mathbf{X}^\top \mathbf{y}, \lambda\alpha\right)}{1 + \lambda(1-\alpha)}.$$

It can be observed that the numerator is identical to the orthonormal lasso solution (2.16), which is then scaled as in orthonormal ridge regression (2.4). The coefficients are thus first shrinked via lasso, whereafter they are further shrinked via ridge regression. This is known as the *double shrinkage effect* and it introduces excess bias which might reduce the predictive power of the estimator. Hence, the original authors of elastic net [12] proposed a bias correction by defining the corrected elastic net estimator as

$$\hat{\boldsymbol{\beta}}^*_{\text{EN}} = (1 + \lambda(1-\alpha))\hat{\boldsymbol{\beta}}_{\text{EN}}.$$

The uncorrected elastic net estimate (2.28) is often referred to as the *naive* elastic net.

2.4 Comparison of shrinkage functions

It is informative to compare the different shrinkage methods: ridge regression, lasso, and elastic net by considering their shrinkage functions: (2.4), (2.19), and (2.33). The shrinkage function is the solution of the optimization problem in the orthonormal case as well as the solution in the general case with respect to one coordinate β_j, when holding the other coordinates β_k, $k \neq j$, fixed. The shrinkage function thus determines the updated value of the coefficient in a cyclic coordinate-wise algorithm. The shrinkage functions of lasso, ridge regression, and elastic net are defined as

$$\begin{aligned} f_{\text{lasso}}(\beta_j; \lambda) &= \mathcal{S}\left(\beta_j, \lambda\right), \\ f_{\text{ridge}}(\beta_j; \lambda) &= \frac{\beta_j}{1+\lambda}, \text{ and} \\ f_{\text{EN}}(\beta_j; \lambda, \alpha) &= \frac{\mathcal{S}\left(\beta_j, \lambda\right)}{1 + \lambda(1-\alpha)}, \end{aligned}$$

and they are depicted in Figure 5. The least squares estimator, which doesn't perform shrinkage, can be thought of as having the shrinkage function $f_{\text{ls}}(\beta_j) = \beta_j$, which is plotted as a dotted line in the figures.

The shrinkage functions are informative in understanding the relationship between the solutions of the different penalized estimators with respect to the least squares estimator. For example, by comparing these shrinkage functions, one can understand why the lasso and elastic net produce sparse solutions, whereas ridge regression does not.

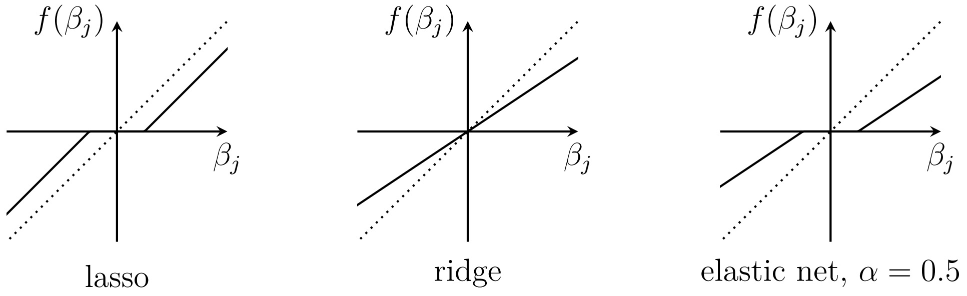

Figure 5: The shrinkage functions of the lasso, ridge regression, and elastic net.

3 Scaled sparse linear regression

In ridge regression and lasso, the optimal tuning parameters depend on the unknown error scale, as was shown in Sections 2.1.1 and 2.2.3. This makes the determination of the tuning parameter difficult, and thus methods such as cross-validation are commonly utilized in choosing an appropriate penalty level. There exist, however, a class of estimators which have the property of being independent of the error scale. This is the topic of this section.

In *scaled linear regression*, the unknown regression coefficients along with the noise scale are estimated jointly. This has the benefit of making the optimal tuning parameter independent of the noise scale enabling the usage of universal tuning parameters. The *scaled lasso* [15], covered in Section 3.2, is a version of the lasso, which has this property. The *square-root lasso*, proposed independently in [16], although having a slightly different formulation, is equivalent to the scaled lasso.

The idea of the scaled lasso arose out of the discussion of [17] in which the authors considered ℓ_1-penalized regression for mixture models, wherein good estimates of the noise scales of the different mixtures are essential. Relating to that, we will first cover the ℓ_1-penalized maximum likelihood estimator, proposed in [17], which served as the motivation for the scaled lasso. Thereafter, we will cover the scaled lasso and the square-root lasso, which are central to this thesis.

3.1 ℓ_1-penalized maximum likelihood estimator

In addition to the ℓ_1-penalized mixture models, which was the main topic of the paper [17], the authors also considered a ℓ_1-penalized non-mixture linear regression model. Assuming the errors are i.i.d. with a zero mean Gaussian distribution, the negative log-likelihood function takes the form

$$l(\boldsymbol{\beta}, \sigma) = \frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma^2} + n \log \sigma + C, \tag{3.1}$$

where C is a constant not depending on the noise scale or regression coefficients. The authors reasoned that in order to incorporate the ℓ_1-penalty to the log-likelihood, one can not simply add the penalty to the negative of the log-likelihood function since the obtained optimization problem,

$$\left(\hat{\boldsymbol{\beta}}, \hat{\sigma}\right) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p, \sigma > 0}{\arg\min} \frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma^2} + n \log \sigma + \lambda \|\boldsymbol{\beta}\|_1, \tag{3.2}$$

is both non-convex and non-equivariant. The latter means that, if the data is scaled, then the estimators are not scaled in the same proportion. Non-equivariance can, however, easily be fixed by also scaling the penalty parameter, which results in the *penalized maximum likelihood estimators* of regression and scale [17]

$$\left(\hat{\boldsymbol{\beta}}_{\text{pmle}}, \hat{\sigma}_{\text{pmle}}\right) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p, \sigma > 0}{\arg\min} \frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma^2} + n \log \sigma + \frac{\lambda}{\sigma} \|\boldsymbol{\beta}\|_1. \tag{3.3}$$

Furthermore, the estimator can be made convex by applying a simple re-parametrization ($\phi_j = \beta_j/\sigma$, $\rho = \sigma^{-1}$) resulting in

$$\left(\hat{\boldsymbol{\phi}}, \hat{\rho}\right) = \underset{\boldsymbol{\phi}\in\mathbb{R}^p, \rho>0}{\arg\min}\ \frac{1}{2}\left\|\rho\mathbf{y} - \mathbf{X}\boldsymbol{\phi}\right\|_2^2 - n\log(\rho) + \lambda\left\|\boldsymbol{\phi}\right\|_1. \tag{3.4}$$

Since (3.4) is minimized by $(\hat{\boldsymbol{\phi}}, \hat{\rho})$, therefore $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\phi}}/\hat{\rho}$ and $\hat{\sigma} = 1/\hat{\rho}$ must be the joint unique local minimizer of (3.3) as well. This was further studied in [18].

3.2 Scaled and square-root lasso

An alternative approach to the estimation of regression and scale was proposed in a discussion paper [19] of [17], where it was suggested to use the ideas of robust regression by Huber [20], namely the concomitant loss function

$$\frac{\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2^2}{2\sigma} + \frac{n\sigma}{2}. \tag{3.5}$$

This lead to the scaled lasso (3.6), which was then studied extensively in [15]. At the time, the idea of penalizing Huber's concomitant loss function with an ℓ_1-norm penalty wasn't exactly entirely new since it was first studied independently in a more general framework in [21].

The scaled lasso estimates of regression coefficients and noise scale are defined as

$$\left(\hat{\boldsymbol{\beta}}_{\text{sl}}, \hat{\sigma}_{\text{sl}}\right) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p, \sigma>0}{\arg\min}\ \frac{\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2^2}{2\sigma} + \frac{n\sigma}{2} + \lambda\left\|\boldsymbol{\beta}\right\|_1, \tag{3.6}$$

which is a jointly convex optimization problem in $(\boldsymbol{\beta}, \sigma)$. The objective is strictly convex in σ and therefore the uniqueness of the solution follows from (3.9) below, via the uniqueness of the lasso. Since the feasible region is not compact, in the pathological case in which $\mathbf{y}$ lies in the subspace spanned by $\mathbf{X}$, i.e., $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$, the estimate of the noise scale $\hat{\sigma}$ will equal zero, and hence be outside the feasible region.

Interestingly, the *square-root lasso* [16], defined as

$$\hat{\boldsymbol{\beta}}_{\text{sr-lasso}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2 + \frac{\lambda}{\sqrt{n}}\left\|\boldsymbol{\beta}\right\|_1, \tag{3.7}$$

is equivalent to the scaled lasso (3.6). In order to show the equivalence, we have to consider the minimizers of the scaled lasso criterion (3.6). The conditional minimizer for a fixed $\boldsymbol{\beta}$ with respect to σ is

$$\hat{\sigma}(\boldsymbol{\beta}) = \frac{\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2}{\sqrt{n}}. \tag{3.8}$$

Respectively, for a fixed σ, the minimizer with respect to the coefficient vector $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}(\sigma) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\ \frac{\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\right\|_2^2}{\sigma} + 2\lambda\left\|\boldsymbol{\beta}\right\|_1, \tag{3.9}$$

which is equivalent to the lasso solution with the tuning parameter scaled by the noise scale. By inserting (3.8) into (3.6), we get

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2^2}{2\frac{\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2}{\sqrt{n}}} + \frac{n}{2}\frac{\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2}{\sqrt{n}} + \lambda\left\|\boldsymbol{\beta}\right\|_1 \\ &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{\sqrt{n}}{2}\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2 + \frac{\sqrt{n}}{2}\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2 + \lambda\left\|\boldsymbol{\beta}\right\|_1 \\ &= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sqrt{n}\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2 + \lambda\left\|\boldsymbol{\beta}\right\|_1,\end{aligned}$$

which is equivalent to (3.7).

The scaled lasso can be solved by a *cyclic coordinate-wise descent* algorithm in which the estimate of the noise scale $\hat{\sigma}$ and the unknown regression coefficients are estimated cyclically until convergence. The procedure is given in Algorithm 1.

Algorithm 1: Scaled lasso

```
Input  : $\mathbf{X}$, $\mathbf{y}$, $\lambda$, $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{0}$
while not converged do
    $\hat{\sigma} \leftarrow \left\|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\right\|_2 / \sqrt{n}$;
    $\hat{\boldsymbol{\beta}} \leftarrow \arg\min_{\boldsymbol{\beta}} \left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2^2 + 2\lambda\hat{\sigma}\left\|\boldsymbol{\beta}\right\|_1$;
Output : $(\hat{\boldsymbol{\beta}}_{\mathrm{sl}}, \hat{\sigma}_{\mathrm{sl}})$
```

4 Scaled and square-root elastic net

As discussed in Section 3, the main benefit of the scaled lasso is that the penalty level is scale-free. This means that the tuning parameter can be predetermined from pure analytical considerations, independently of the actual data. Furthermore, the scaled lasso is also an accurate method in the estimation of the error variance in high-dimensional settings [22].

Nevertheless, as the regular lasso, the scaled lasso also performs poorly when there exist strong correlations between the predictors. Furthermore, in the case there are more predictors than samples ($p > n$), the scaled lasso picks at most n variables.

As discussed in Section 2.3, the elastic net (EN) [12] overcomes the mentioned deficiencies of the lasso by utilizing a penalty function that is a mixture of the ℓ_1 and ℓ_2-norm penalties, defined as

$$\mathcal{P}_{\text{EN}}(\boldsymbol{\beta};\alpha) = \frac{1}{2}(1-\alpha)\left\|\boldsymbol{\beta}\right\|_2^2 + \alpha\left\|\boldsymbol{\beta}\right\|_1, \tag{4.1}$$

where $\alpha \in [0,1]$ is an elastic net tuning parameter. The superiority of the elastic net to the lasso in situations of high correlations between the predictors motivates the idea of extending the scaled lasso to the elastic net framework.

In this section, we propose two different scaled elastic net formulations to remedy the aforementioned shortcomings of the scaled lasso. We will also derive convergent algorithms for their computation. As with scaled lasso, both of the methods are based on penalizing Huber's concomitant loss function. The first formulation uses a conventional elastic net penalty, whereas the second formulation differs from the former in that the ℓ_2-norm term is not squared. The former approach is referred to as the *scaled elastic net estimator* and the latter as the *square-root elastic net estimator*. Later, in Section 5, we illustate via numerical examples and simulations that the proposed methods outperform the scaled lasso, especially in the presence of multicollinearity in the feature space.

Thus we solve the optimization program

$$\underset{\boldsymbol{\beta}\in\mathbb{R}^p,\sigma>0}{\text{minimize}} \frac{\left\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right\|_2^2}{2\sigma} + \frac{n\sigma}{2} + \lambda\mathcal{P}(\boldsymbol{\beta};\alpha), \tag{4.2}$$

where $\mathcal{P}(\boldsymbol{\beta};\alpha)$ is either the conventional elastic net penalty (2.28), or

$$\mathcal{P}_{\sqrt{\text{EN}}}(\boldsymbol{\beta};\alpha) = (1-\alpha)\left\|\boldsymbol{\beta}\right\|_2 + \alpha\left\|\boldsymbol{\beta}\right\|_1, \tag{4.3}$$

referred to as the *square-root elastic net penalty*, as it utilizes a non-squared ℓ_2-norm, as does the square-root lasso in (3.7). When $\alpha = 1$, both estimators reduce to the conventional scaled lasso, but for intermediate values they differ; and when $\alpha = 0$, they yield different scaled ridge regression estimators. Both approaches are potentially interesting elastic net penalties to be used in scaled sparse regression.

4.1 Scaled elastic net

Let us first note that the minimizer of both estimators in (4.2) with respect to the noise scale σ is the same and can be solved from the zero derivative equation,

$$-\frac{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2}{2\hat{\sigma}^2}+\frac{n}{2}=0,$$

yielding

$$\hat{\sigma}(\boldsymbol{\beta})=\frac{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2}{\sqrt{n}}. \tag{4.4}$$

The minimizer with respect to $\boldsymbol{\beta}$ will, however, differ between the two variants.

The scaled elastic net estimators of regression and scale, $(\hat{\boldsymbol{\beta}},\hat{\sigma})$, are defined as the minimizers of the criterion

$$\frac{\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma}+\frac{n\sigma}{2}+\lambda\left(\frac{(1-\alpha)}{2}\|\boldsymbol{\beta}\|_2^2+\alpha\|\boldsymbol{\beta}\|_1\right) \tag{4.5}$$

over $(\boldsymbol{\beta},\sigma)\in\mathbb{R}^p\times(0,\infty)$. The criterion function in (4.5) is *separable* as it can be written in the form $f(\boldsymbol{\beta},\sigma)=g(\boldsymbol{\beta},\sigma)+\sum_{j=1}^p h_j(\beta_j)$, where $g:\mathbb{R}^p\times(0,\infty)\to\mathbb{R}$ is convex and differentiable and $h_j:\mathbb{R}\to\mathbb{R}$ are convex [10]. A coordinate-wise descent algorithm is therefore guaranteed to converge [23]. Hence we will derive a cyclic coordinate-wise descent (CCD) algorithm for the problem in which the function is minimized cyclically with respect to one coordinate at a time.

We will proceed in a similar fashion to the previous derivations of the lasso and the elastic net in Sections 2.2 and 2.3, respectively. Thus, let $\mathbf{X}_{-j}$ and $\boldsymbol{\beta}_{-j}$ denote the matrix $\mathbf{X}$ and vector $\boldsymbol{\beta}$ with its j^{th} column and element excluded. We can then rewrite the objective function in (4.5) as

$$\begin{aligned}&\frac{\left\|\mathbf{r}^{(j)}-\mathbf{x}_j\beta_j\right\|_2^2}{2\sigma}+\frac{n\sigma}{2}+\lambda\left(\frac{(1-\alpha)}{2}\beta_j^2+\alpha|\beta_j|\right)\\&+\lambda\left(\frac{(1-\alpha)}{2}\left\|\boldsymbol{\beta}_{-j}\right\|_2^2+\alpha\left\|\boldsymbol{\beta}_{-j}\right\|_1\right)\end{aligned} \tag{4.6}$$

where $\mathbf{r}^{(j)}=\mathbf{y}-\mathbf{X}_{-j}\boldsymbol{\beta}_{-j}$ denotes the partial residual vector. The minimizer of (4.6) with respect to β_j, when holding other coefficients β_k, $k\neq j$, and σ fixed, needs to verify the subgradient equation

$$-\mathbf{x}_j^\top(\mathbf{r}^{(j)}-\mathbf{x}_j\hat{\beta}_j)+\lambda(1-\alpha)\sigma\hat{\beta}_j+\lambda\alpha\sigma\hat{t}_j=0, \tag{4.7}$$

where $\hat{t}_j$ belongs to the subdifferential of $|\beta_j|$ evaluated at $\hat{\beta}_j$, i.e., equal to $\hat{\beta}_j/|\hat{\beta}_j|$ if $\hat{\beta}_j\neq 0$ and some number in $[-1,1]$ otherwise. We notice that (4.7) is essentially of the same form as the zero subgradient equation of the elastic net (2.30). Thus, the solution for β_j is simply

$$\hat{\beta}_j=\frac{\mathcal{S}\left(\hat{\beta}_j+\mathbf{x}_j^\top\mathbf{r},\lambda\alpha\sigma\right)}{1+\lambda(1-\alpha)\sigma}. \tag{4.8}$$

The cyclic coordinate-wise minimization proceeds as follows. We first update the scale via (4.4) by using $\hat{\boldsymbol{\beta}}$ of the current full iterate in place of $\boldsymbol{\beta}$. Thereafter, we update all $\hat{\beta}_j$, $j \in \{1, \ldots, p\}$, according to (4.8) holding β_k, $k \neq j$, and σ fixed at their current values $\hat{\beta}_k$ and $\hat{\sigma}$. These updates are cycled until convergence is reached, as described in Algorithm 2.

It is instructive to consider the orthonormal design matrix case, i.e., $\mathbf{X}^\top \mathbf{X} = \mathbf{I}$ and $n = p$. With a little algebra, it is easy to show that $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$ then solves

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \frac{\mathcal{S}\left(\mathbf{X}^\top \mathbf{y}, \lambda\alpha\hat{\sigma}\right)}{1 + \lambda(1-\alpha)\hat{\sigma}} \quad \text{and} \\ \hat{\sigma} &= \frac{\left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|_2}{\sqrt{n}}. \end{aligned}$$

As the conventional elastic net, one can argue that also the scaled elastic net can suffer from the double shrinkage effect due to both lasso and ridge regression type of shrinkage, as discussed in Section 2.3.1. To remedy for the double shrinkage effect, we define the corrected scaled elastic net estimates of regression and scale, $(\hat{\boldsymbol{\beta}}^*, \hat{\sigma}^*)$, as

$$\begin{aligned} \hat{\boldsymbol{\beta}}^* &= (1 + \lambda(1-\alpha)\hat{\sigma})\hat{\boldsymbol{\beta}} \quad \text{and} \\ \hat{\sigma}^* &= \hat{\sigma}(\hat{\boldsymbol{\beta}}^*). \end{aligned}$$

4.2 Square-root elastic net

The square-root elastic net estimators, $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$, are defined as the minimizers of the criterion

$$\frac{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2}{2\sigma} + \frac{n\sigma}{2} + \lambda\left((1-\alpha)\|\boldsymbol{\beta}\|_2 + \alpha\|\boldsymbol{\beta}\|_1\right) \tag{4.9}$$

over $(\boldsymbol{\beta}, \sigma) \in \mathbb{R}^p \times (0, \infty)$. The zero subgradient equation of (4.9) with respect to $\boldsymbol{\beta}$ is

$$-\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) + \lambda(1-\alpha)\sigma\hat{\mathbf{s}} + \lambda\alpha\sigma\hat{\mathbf{t}} = \mathbf{0}, \tag{4.10}$$

where $\hat{\mathbf{t}}$ is a p-vector whose j^{th} element belongs to the subdifferential of $|\beta_j|$ evaluated at $\hat{\beta}_j$, and $\hat{\mathbf{s}}$ belongs to the subdifferential of $\|\boldsymbol{\beta}\|_2$ evaluated at $\hat{\boldsymbol{\beta}}$, so $\hat{\mathbf{s}} = \hat{\boldsymbol{\beta}} / \left\|\hat{\boldsymbol{\beta}}\right\|_2$ if $\hat{\boldsymbol{\beta}} \neq \mathbf{0}$ and $\hat{\mathbf{s}} \in \{\mathbf{s} \in \mathbb{R}^p : \|\mathbf{s}\|_2 \leq 1\}$ if $\hat{\boldsymbol{\beta}} = \mathbf{0}$.

The subgradient equations are satisfied with $\hat{\boldsymbol{\beta}} = \mathbf{0}$ if and only if

$$-\mathbf{X}^\top \mathbf{y} + \lambda(1-\alpha)\sigma\hat{\mathbf{s}} + \lambda\alpha\sigma\hat{\mathbf{t}} = \mathbf{0} \tag{4.11}$$

has a solution with $\|\hat{\mathbf{s}}\|_2 \leq 1$ and $\hat{t}_j \in [-1, 1]$, for $j \in 1, \ldots, p$ [10]. To check for this condition, we first rearrange (4.11) and then take the ℓ_2-norm of both sides, yielding

$$\left\|\mathbf{X}^\top \mathbf{y} - \lambda\alpha\sigma\hat{\mathbf{t}}\right\|_2 = \lambda(1-\alpha)\sigma\|\hat{\mathbf{s}}\|_2 \leq \lambda(1-\alpha)\sigma.$$

Algorithm 2: Scaled elastic net and Square-root elastic net

```
Input  : $\mathbf{X}, \mathbf{y}, \lambda, \alpha, \hat{\boldsymbol{\beta}} \leftarrow \mathbf{0}$
while not converged do
    $\hat{\sigma} \leftarrow \left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|_2 / \sqrt{n}$;
    $\lambda_1 \leftarrow \lambda\alpha\hat{\sigma}$;
    $\lambda_2 \leftarrow \lambda(1-\alpha)\hat{\sigma}$;
    for $j = 1$ to $p$ do
        $\mathbf{r} \leftarrow \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$;
        if Scaled EN then
            $\hat{\beta}_j \leftarrow \frac{\mathcal{S}\left(\hat{\beta}_j + \mathbf{x}_j^\top \mathbf{r}\,,\ \lambda_1\right)}{1 + \lambda_2}$
        else if Square-root EN then
            if condition (4.12) is met then
                $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{0}$;
            else
                $\hat{\beta}_j \leftarrow \frac{\mathcal{S}\left(\hat{\beta}_j + \mathbf{x}_j^\top \mathbf{r}\,,\ \lambda_1\right)}{1 + \lambda_2 / \left\|\hat{\boldsymbol{\beta}}\right\|_2}$
Output: $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$
```

Since the subdifferential $\hat{\mathbf{t}}$ is a set of subgradients which satisfy (4.11), we can choose the particular subgradient that minimizes the left-hand side, resulting in

$$\left\|\mathcal{S}\left(\mathbf{X}^\top\mathbf{y}, \lambda\alpha\sigma\right)\right\|_2 \leq \lambda(1-\alpha)\sigma,$$

which holds if and only if $\hat{\boldsymbol{\beta}} = \mathbf{0}$. Moreover, if $\hat{\boldsymbol{\beta}} = \mathbf{0}$, then $\hat{\sigma} = \|\mathbf{y}\|_2 / \sqrt{n}$. Putting this together, we have the condition that $\hat{\boldsymbol{\beta}} = \mathbf{0}$ if and only if

$$\left\|\mathcal{S}\left(\mathbf{X}^\top\mathbf{y}, \lambda\alpha\,\|\mathbf{y}\|_2/\sqrt{n}\right)\right\|_2 \leq \lambda(1-\alpha)\,\|\mathbf{y}\|_2/\sqrt{n}. \tag{4.12}$$

Next, we derive the CCD algorithm for solving the square-root elastic net. Let us consider the j^{th} element of (4.10) and assume that it is non-zero, i.e., $\hat{\beta}_j \neq 0$. Then, by rearranging, we obtain

$$\left(1 + \frac{\lambda(1-\alpha)\sigma}{\left\|\hat{\boldsymbol{\beta}}\right\|_2} + \frac{\lambda\alpha\sigma}{|\hat{\beta}_j|}\right)\hat{\beta}_j = \mathbf{x}_j^\top \mathbf{r}^{(j)}. \tag{4.13}$$

Taking the modulus of both sides and solving for $|\hat{\beta}_j|$ yields

$$|\hat{\beta}_j| = \left(1 + \frac{\lambda(1-\alpha)\sigma}{\left\|\hat{\boldsymbol{\beta}}\right\|_2}\right)^{-1}\left(|\mathbf{x}_j^\top \mathbf{r}^{(j)}| - \lambda\alpha\sigma\right)_+.$$

Substituting this back into (4.13) and solving for $\hat{\beta}_j$ gives

$$\hat{\beta}_j = \frac{\mathbf{x}_j^\top \mathbf{r}^{(j)}}{|\mathbf{x}_j^\top \mathbf{r}^{(j)}|} \frac{\left(|\mathbf{x}_j^\top \mathbf{r}^{(j)}| - \lambda\alpha\sigma\right)_+}{1 + \lambda(1-\alpha)\sigma / \left\|\hat{\boldsymbol{\beta}}\right\|_2} = \frac{\mathcal{S}\left(\hat{\beta}_j + \mathbf{x}_j^\top \mathbf{r}, \lambda\alpha\sigma\right)}{1 + \lambda(1-\alpha)\sigma / \left\|\hat{\boldsymbol{\beta}}\right\|_2}.$$

As a consequence of non-separability, the formula depends on the norm of the optimal coefficients. In order to use the formula, we simply use the norm of the previous full iterate. The procedure is given in Algorithm 2. For this problem, one could also have utilized a generalized gradient descent scheme as in [24].

It is again instructive to consider the orthonormal design matrix case, i.e., $\mathbf{X}^\top \mathbf{X} = \mathbf{I}$ and $n = p$. With a little algebra, it is easy to show that $(\hat{\boldsymbol{\beta}}, \hat{\sigma})$ then solves

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \left(1 - \frac{\lambda(1-\alpha)\hat{\sigma}}{\left\|\mathcal{S}\left(\mathbf{X}^\top \mathbf{y}, \lambda\alpha\hat{\sigma}\right)\right\|_2}\right)_+ \mathcal{S}\left(\mathbf{X}^\top \mathbf{y}, \lambda\alpha\hat{\sigma}\right) \quad \text{and} \\ \hat{\sigma} &= \frac{\left\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right\|_2}{\sqrt{n}}. \end{aligned}$$

To remedy for the double shrinkage effect, we define the corrected square-root elastic net estimates of regression and scale, $(\hat{\boldsymbol{\beta}}^*, \hat{\sigma}^*)$, as

$$\begin{aligned} \hat{\boldsymbol{\beta}}^* &= \left(1 - \frac{\lambda(1-\alpha)\hat{\sigma}}{\left\|\mathcal{S}\left(\mathbf{X}^\top \mathbf{y}, \lambda\alpha\hat{\sigma}\right)\right\|_2}\right)_+^{-1} \hat{\boldsymbol{\beta}} \quad \text{and} \\ \hat{\sigma}^* &= \hat{\sigma}(\hat{\boldsymbol{\beta}}^*). \end{aligned}$$

5 Numerical results

In this section, we present simulation results which show that the scaled elastic net and the square-root elastic net estimators outperform the scaled lasso in the case of multicollinearity.

As shown in Section 2.2.3, the optimal tuning parameter is proportional to $\sqrt{\log(p)}$. When the sample size is finite, the performance will depend on the chosen proportionality constant. In the original paper [15], three different values for λ are considered; namely $\sqrt{2^{j-1}\log(p)}$, where $j = 1, 2$, and 3. Herein, we consider those same values.

5.1 Example 1: Grouping effect of collinear variables

The first set-up illustrates the superiority of the elastic net penalties to lasso in situations of high correlations in the feature space as well as the grouping effect. The set-up is as in [10], where the linear model consists of two groups of three highly correlated variables. The data is generated as

$$\mathbf{y} = 3\mathbf{z}_1 - 1.5\mathbf{z}_2 + 2\boldsymbol{\varepsilon},$$

where $\mathbf{z}_1$, $\mathbf{z}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$, and the design matrix is generated as: $\mathbf{x}_j = \mathbf{z}_1\, 1_{\{1,2,3\}}(j) + \mathbf{z}_2\, 1_{\{4,5,6\}}(j) + (1/5)\boldsymbol{\varepsilon}_j$, where $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n\times n})$, for $j = 1, 2, \ldots, 6$. As the tuning parameter is varied, the estimated regression coefficients trace a path in $\mathbb{R}^p$, referred to as the solution path, shown in Figure 6. Even with a very mild EN parameter value of $\alpha = 0.95$, the scaled and square-root elastic net are able to identify the two groups of correlated variables and connect them by setting them to zero at the same value of λ. By contrast, the scaled lasso fails to do so.

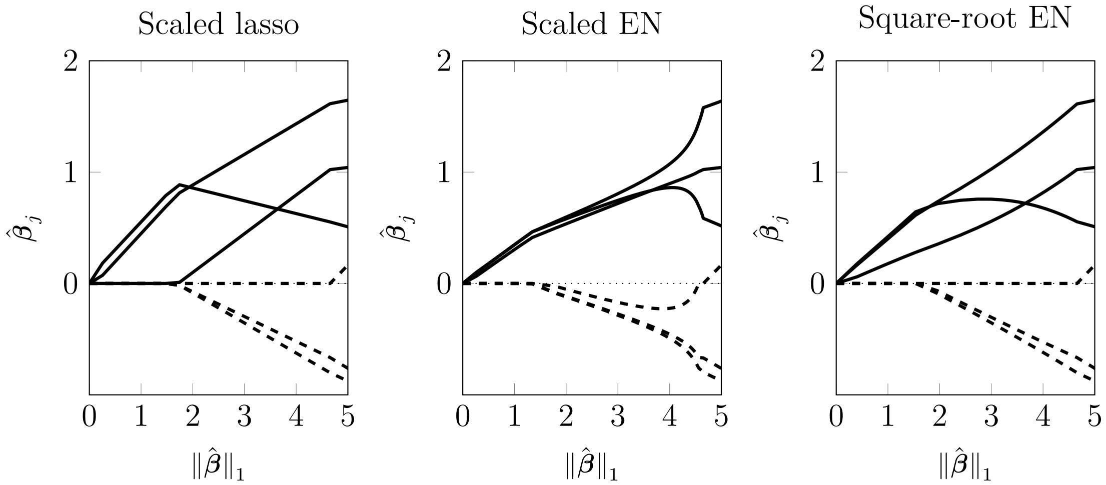

Figure 6: The solution path of scaled lasso (*left*), scaled elastic net (*middle*), and square-root elastic net (*right*), with the EN parameter value of $\alpha = 0.95$.

5.2 Example 2: Performance vs. SNR

In the second set-up, we consider a linear model with the dimensions $(n, p) = (50, 10)$. The row vectors of $\mathbf{X}$ are normally distributed with a mean vector $\mathbf{0}_{p\times 1}$ and a covariance matrix $\mathbf{\Sigma}$ such that $(\mathbf{\Sigma})_{ij} = 0.9^{|i-j|}$, for $i, j = 1, \ldots, p$. In Figure 7, on the left, we have $\boldsymbol{\beta} = (1, 1, 1, 1, 1, 0, 0, 0, 0, 0)^\top = \boldsymbol{\beta}^{(1)}$, and on the right, we have $\boldsymbol{\beta} = (1, 2, 3, 4, 5, 0, 0, 0, 0, 0)^\top = \boldsymbol{\beta}^{(2)}$. For all simulated methods, the tuning parameter value is set to $\lambda = \sqrt{2\log(p)}$. The EN tuning parameter is set to $\alpha = 0.9$ for both the scaled elastic net and the square-root elastic net. Figure 7 depicts the (empirical) mean squared error (MSE) versus the signal-to-noise ratio (SNR). As can be seen, both the scaled elastic net and the square-root elastic net outperform the scaled lasso. The MSE is defined as

$$\mathrm{MSE}(\hat{\boldsymbol{\beta}}) \triangleq \mathrm{Ave}\left\{\frac{1}{p}\left\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|_2^2\right\}, \tag{5.1}$$

where the average is over 200 Monte-Carlo trials. The signal-to-noise ratio (SNR), in decibels, is defined as

$$\mathrm{SNR} \triangleq 10\log_{10}\frac{\sigma_{\boldsymbol{\beta}}^2}{\sigma^2}, \tag{5.2}$$

where $\sigma_{\boldsymbol{\beta}}^2 = \sum_j |\beta_j|^2 / \|\boldsymbol{\beta}\|_0$.

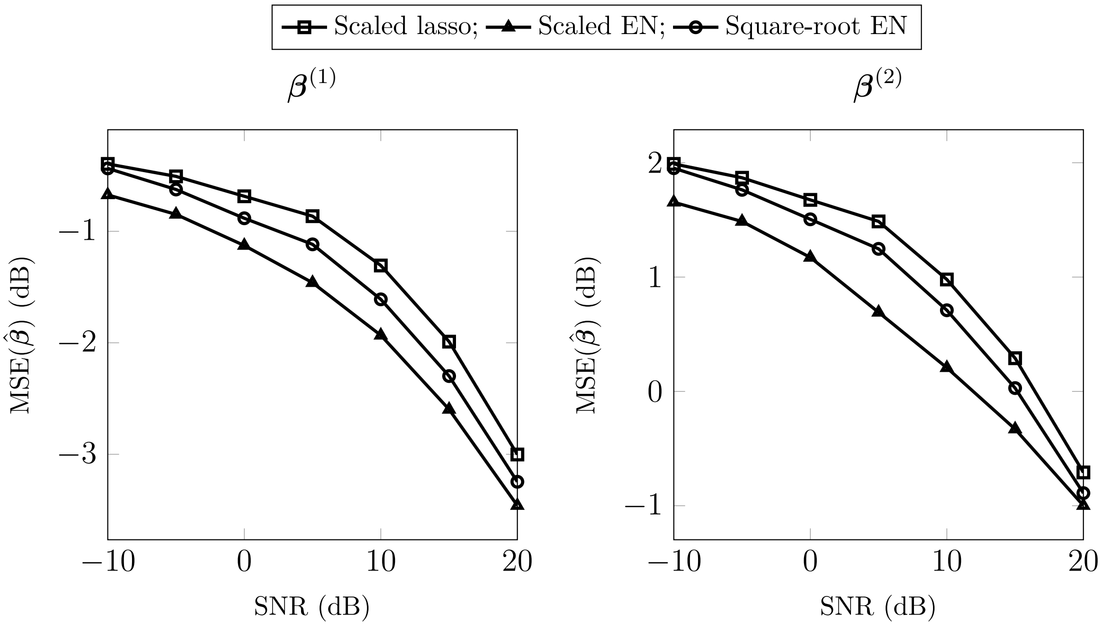

Figure 7: MSE vs. SNR for the scaled lasso, scaled elastic net, and square-root elastic net for $\boldsymbol{\beta}^{(1)}$ (*left*) and $\boldsymbol{\beta}^{(2)}$ (*right*).

5.3 Example 3: A high-dimensional setting

Next, we consider a high-dimensional problem where $(n, p) = (30, 150)$. The design matrix is generated as in Example 2: the row vectors of $\mathbf{X}$ are normally distributed with a mean vector $\mathbf{0}_{p\times 1}$ and a covariance matrix $\mathbf{\Sigma}$ such that $(\mathbf{\Sigma})_{ij} = 0.9^{|i-j|}$, for $i, j = 1, \ldots, p$. The SNR is set to 0 dB. The true 150×1 coefficient vector is $\boldsymbol{\beta} = (\mathbf{1}_{1\times 20}, \mathbf{0}_{1\times 130})^\top$. The elastic net tuning parameter is set to $\alpha = 0.9$. Three different values for λ are considered, namely $\lambda_j = \sqrt{2^{j-1}\log(p)}$, for $j \in \{1, 2, 3\}$. Table 1 tabulates the MSE, the ratio of the estimated and true error scale, $\hat{\sigma}/\sigma$, the mean false positive rate (FPR) and the mean false negative rate (FNR) defined as

$$\text{FPR} = \frac{\text{false positives}}{\text{false positives} + \text{true negatives}} \tag{5.3}$$

and

$$\text{FNR} = \frac{\text{false negatives}}{\text{false negatives} + \text{true positives}}, \tag{5.4}$$

respectively. The reported results are averages over 100 Monte-Carlo trials. The standard deviation (×10) is given in the parenthesis.

Based on the results in Table 1, $\lambda_2 = \sqrt{2\log(p)}$ appears to be the best compromise giving the best estimates of the error scale and mean squared error. With the chosen elastic net tuning parameter $\alpha = 0.9$, the scaled elastic net estimator performs the best.

Table 1: Simulation results of Example 3. The tuning parameters are defined as $\lambda_j = \sqrt{2^{j-1}\log(p)}$, for $j \in \{1, 2, 3\}$. The standard deviation (×10) is given in the parenthesis.

		MSE($\hat{\boldsymbol{\beta}}$)	$\hat{\sigma}/\sigma$	FPR	FNR
Scaled lasso	λ_1	0.23 (0.7)	0.85 (1.7)	0.01 (0.1)	0.59 (0.8)
	λ_2	0.21 (0.6)	1.23 (2.1)	0.00 (0.0)	0.67 (0.8)
	λ_3	0.13 (0.2)	2.58 (2.9)	0.00 (0.0)	0.93 (0.6)
Scaled EN	λ_1	0.08 (0.2)	0.87 (1.7)	0.03 (0.2)	0.26 (1.0)
	λ_2	0.07 (0.1)	1.25 (2.0)	0.01 (0.1)	0.28 (1.2)
	λ_3	0.10 (0.2)	2.38 (2.4)	0.00 (0.0)	0.63 (1.8)
Square-root EN	λ_1	0.18 (0.5)	0.79 (1.7)	0.02 (0.2)	0.49 (0.9)
	λ_2	0.14 (0.4)	1.11 (1.9)	0.00 (0.1)	0.51 (1.0)
	λ_3	0.11 (0.1)	2.21 (3.3)	0.00 (0.0)	0.71 (1.6)

5.4 Example 4: Proportionality constant vs. MSE

In this example, we study how the MSE performance changes as the value of the tuning parameter is varied. We consider the cases of correlated and uncorrelated design matrices. The simulation setting is as follows. The dimensions are $(n, p) = (30, 150)$, the true coefficient vector is $\boldsymbol{\beta} = (\mathbf{1}_{1\times 20}, \mathbf{0}_{1\times 130})^\top$. The row vectors of $\mathbf{X}$ are normally distributed with the mean vector $\mathbf{0}_{p\times 1}$ and the covariance matrix $\mathbf{\Sigma}$ such that for the correlated case, we have $(\mathbf{\Sigma})_{ij} = 0.9^{|i-j|}$, and for the uncorrelated case, we have

$(\boldsymbol{\Sigma})_{ij} = \delta_{ij}$, for $i, j = 1, \ldots, p$. The EN tuning parameter is fixed to $\alpha = 0.9$. The SNR is set to 10 dB. In the simulations, the tuning parameter is defined as

$$\lambda = \sqrt{\tau \log(p)}, \tag{5.5}$$

where τ is the proportionality constant which is varied over the set $\tau \in [0.5, 6]$. The results are averaged over 200 Monte-Carlo trials and are shown in Figure 8. From the results, it can be observed that for scaled and square-root elastic net, in the uncorrelated case, the optimal value is $\tau \approx 1$, and in the correlated case, the optimal value is $\tau \approx 2$. In the correlated case, the scaled lasso performs poorly with any value of τ.

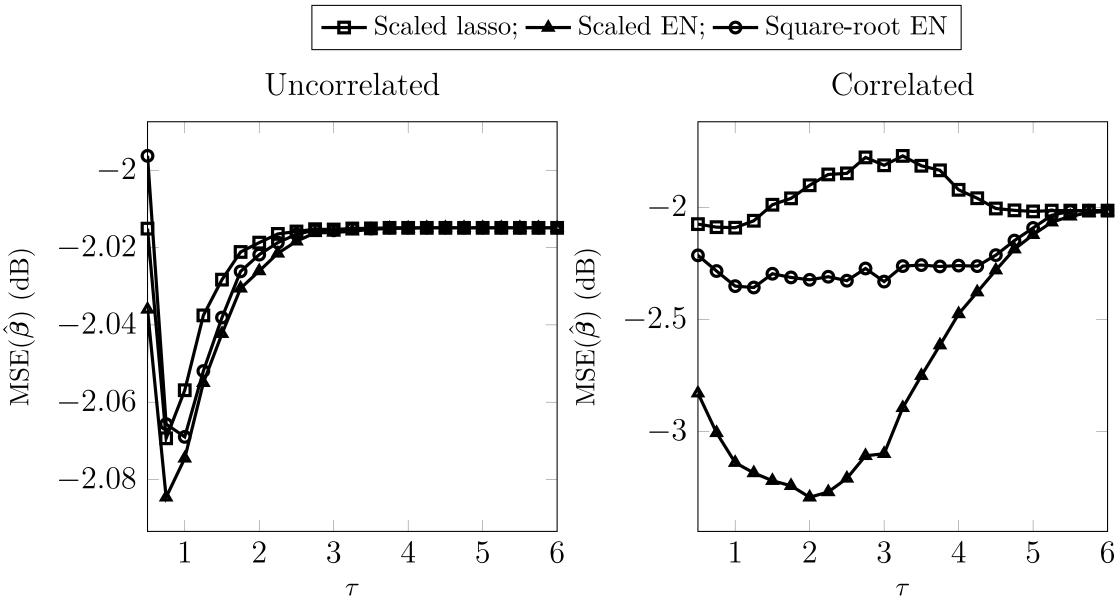

Figure 8: The effect of the proportionality constant τ to the MSE. Uncorrelated design matrix (*left*). Correlated design matrix (*right*).

6 Extension to complex-valued data

The complex field is a very natural domain for representing signals in many engineering applications such as communications, radio science, biomedical imaging, sensor array and radar processing, to name only a few. The reason for this lies in the fact that the complex representation provides a mathematically simple and convenient representation for periodic signals, which are inherent in nature. In this section, we will review some complex differential calculus insofar as what is needed in order to extend the previously proposed estimators to the complex field. The main sources and recommended reading for this section are [25]–[31].

In order to provide a motivating example, consider conventional gradient-based optimization methods, such as gradient descent. It requires the evaluation of the first derivative of the objective function, or loss function. Obviously, the negative gradient gives the direction of the greatest rate of descent of the function. In the complex-valued setting, however, differentiation is not as straightforward. That is because in complex analysis, a non-constant real-valued objective function is not holomorphic (complex analytic), that is, it is not complex differentiable. More formally, consider a complex function $f(z)$ of a complex variable $z = x + \jmath y$:

$$f(z) = f(x, y) = u(x, y) + \jmath v(x, y). \tag{6.1}$$

In order for $f(z)$ to be holomorphic (complex analytic), it must satisfy the well-known Cauchy-Riemann equations:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \qquad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}. \tag{6.2}$$

For a real-valued loss function over a complex-valued parameter, $f : \mathbb{C}^p \to \mathbb{R}$, we have $v = 0$ in (6.1), and hence, unless it is a constant function, it does not satisfy the Cauchy-Riemann criteria (6.2). As a consequence, the loss function is neither holomorphic nor differentiable and a local optimum can not be found by solving the zero gradient equation.

This problem can be circumvented by treating the real and imaginary parts independently and computing the real gradients with respect to the real and imaginary parts separately in real space. This is possible since the complex space $\mathbb{C}^p$ can equivalently be described in the real space $\mathbb{R}^{2p}$. That is, the cost function $f : \mathbb{C}^p \to \mathbb{R}$ can be reformulated as the mapping $f : \mathbb{R}^{2p} \to \mathbb{R}$, where the mere existence of the real partial derivatives of u and v is a necessary and sufficient condition for finding a stationary point of the cost function.

The described approach is, however, somewhat cumbersome. Fortunately, there exists a more elegant method which avoids switching back and forth between the complex and real domain. By relaxing the requirements of the Cauchy-Riemann equations (6.2), we can define a complex gradient operator that only requires the existence of the partial derivatives with respect to the real and imaginary parts. This relaxation is developed in the $\mathbb{CR}$-calculus, also known as *Wirtinger* calculus.

6.1 Complex differentiation

The elegance of $\mathbb{CR}$-calculus comes from the fact that it enables treating the complex number z and its conjugate z^* as independent variables. If we write the complex function (6.1) as a function in the variables z and z^*, i.e., $f(x,y) = u(x,y) + \jmath v(x,y) = f(z) = f(z, z^*)$, then by using $\mathbb{CR}$-calculus, we can take the partial derivative of $f(z, z^*)$ with respect to z or z^* independently while treating the other one as a constant.

We can derive the expressions for the partial derivatives $\partial/\partial z$ and $\partial/\partial z^*$ in terms of $\partial/\partial x$ and $\partial/\partial y$. By using the chain-rule for the total derivative, we find that

$$\begin{aligned}\frac{\partial f}{\partial x} &= \frac{\partial f}{\partial z}\frac{\partial z}{\partial x} + \frac{\partial f}{\partial z^*}\frac{\partial z^*}{\partial x} \\ &= \frac{\partial f}{\partial z}\frac{\partial (x + \jmath y)}{\partial x} + \frac{\partial f}{\partial z^*}\frac{\partial (x - \jmath y)}{\partial x} \\ &= \frac{\partial f}{\partial z} + \frac{\partial f}{\partial z^*} \end{aligned} \tag{6.3}$$

and

$$\begin{aligned}\frac{\partial f}{\partial y} &= \frac{\partial f}{\partial z}\frac{\partial z}{\partial y} + \frac{\partial f}{\partial z^*}\frac{\partial z^*}{\partial y} \\ &= \frac{\partial f}{\partial z}\frac{\partial (x + \jmath y)}{\partial y} + \frac{\partial f}{\partial z^*}\frac{\partial (x - \jmath y)}{\partial y} \\ &= \jmath\left(\frac{\partial f}{\partial z} - \frac{\partial f}{\partial z^*}\right). \end{aligned} \tag{6.4}$$

By combining (6.3) and (6.4), we can solve for $\partial/\partial z$ and $\partial/\partial z^*$, yielding

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - \jmath\frac{\partial f}{\partial y}\right) \quad \text{and} \quad \frac{\partial f}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \jmath\frac{\partial f}{\partial y}\right). \tag{6.5}$$

Thus, we can define the expressions for the differential operators with respect to z and z^* as

$$\frac{\partial}{\partial z} \triangleq \frac{1}{2}\left(\frac{\partial}{\partial x} - \jmath\frac{\partial}{\partial y}\right) \quad \text{and} \quad \frac{\partial}{\partial z^*} \triangleq \frac{1}{2}\left(\frac{\partial}{\partial x} + \jmath\frac{\partial}{\partial y}\right). \tag{6.6}$$

This definition is compatible with the Cauchy-Riemann conditions (6.2). To show this, consider a function $f(x,y) = u(x,y) + \jmath v(x,y)$, which is holomorphic. Since a holomorphic complex function does not depend on z^*, it is true that $\frac{\partial f}{\partial z^*} = 0$. That is,

$$0 = \frac{1}{2}\left(\frac{\partial}{\partial x} + \jmath\frac{\partial}{\partial y}\right)(u + \jmath v) = \frac{1}{2}\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) + \frac{\jmath}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right).$$

Since both the left-hand side and right-hand side of the equation is equal to zero, both its real and imaginary parts must equal zero, which yields the Cauchy-Riemann conditions (6.2).

For a complex-valued vector $\mathbf{z} = (z_1, \ldots, z_n)^\top$, we define the complex gradient with respect to $\mathbf{z}$ as

$$\nabla_{\mathbf{z}} \triangleq \left(\frac{\partial}{\partial z_1}, \ldots, \frac{\partial}{\partial z_n} \right)^\top,$$

and similarly, we define the complex gradient with respect ot $\mathbf{z}^*$ as

$$\nabla_{\mathbf{z}^*} \triangleq \left(\frac{\partial}{\partial z_1^*}, \ldots, \frac{\partial}{\partial z_n^*} \right)^\top.$$

Theorem 2. *For a real-valued function with a complex-valued argument, $f : \mathbb{C}^n \to \mathbb{R}$, let $f(z) = g(z, z^*)$, where $g : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{R}$, be analytic with respect to z and z^* individually by treating the other one constant. Then, a necessary and sufficient condition for a stationary point of f is $\nabla_{\mathbf{z}} f = \mathbf{0}$ or, alternatively, $\nabla_{\mathbf{z}^*} f = \mathbf{0}$ [25].*

Proof. Since $g(z(x,y), z^*(x,y)) = u(x,y) + \jmath v(x,y)$ is a real-valued function, we must have $v(x,y) = 0$. Then, regarding the $\jmath^{\text{th}}$ component of the gradient, we have

$$(\nabla_{\mathbf{z}} f)_j = \frac{\partial f}{\partial z_j} = \frac{1}{2} \left(\frac{\partial u}{\partial x_j} - \jmath \frac{\partial u}{\partial y_j} \right).$$

We see that, if the derivative with respect to x_j and y_j vanishes, it also vanishes for z_j, and vice versa. So, we have

$$\frac{\partial u}{\partial x_j} = \frac{\partial u}{\partial y_j} = 0 \Leftrightarrow \frac{\partial u}{\partial z_j} = 0,$$

and likewise,

$$\frac{\partial u}{\partial x_j} = \frac{\partial u}{\partial y_j} = 0 \Leftrightarrow \frac{\partial u}{\partial z_j^*} = 0.$$

□

Theorem 3. *For a real-valued function with a complex-valued argument, $f : \mathbb{C}^n \to \mathbb{R}$, let $f(z) = g(z, z^*)$, where $g : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{R}$, be analytic with respect to z and z^* individually by treating the other one constant. Then, the complex gradient of a real-valued function, $f : \mathbb{C}^n \to \mathbb{R}$, with respect to $\mathbf{z}^*$ determines the direction of the maximum rate of change with respect to $\mathbf{z}$ [25].*

Proof. The differential of g is

$$\mathrm{d}g = \sum_{j=1}^{n} \left(\frac{\partial g}{\partial z_j} \mathrm{d}z_j + \frac{\partial g}{\partial z_j^*} \mathrm{d}z_j^* \right).$$

Since $\frac{\partial}{\partial z^*} = \left(\frac{\partial}{\partial z} \right)^*$, we have

$$\begin{aligned} \mathrm{d}g &= \sum_{j=1}^{n} \left(\frac{\partial g}{\partial z_j} \mathrm{d}z_j + \left(\frac{\partial g^*}{\partial z_j} \mathrm{d}z_j \right)^* \right) \\ &= 2 \operatorname{Re} \left(\sum_{j=1}^{n} \frac{\partial g}{\partial z_j} \mathrm{d}z_j \right) \\ &= 2 \operatorname{Re} \left((\nabla_{\mathbf{z}} g)^\top \mathrm{d}\mathbf{z} \right) \\ &= 2 \operatorname{Re} \left((\nabla_{\mathbf{z}^*} g)^{\mathsf{H}} \mathrm{d}\mathbf{z} \right). \end{aligned} \tag{6.7}$$

In a complex Hilbert space, the scalar product $(\nabla_{\mathbf{z}^*} g)^{\mathsf{H}} \mathrm{d}\mathbf{z}$ satisfies the properties of the inner product. Thus, (6.7) will attain its maximum when $\nabla_{\mathbf{z}^*} g$ and $\mathrm{d}\mathbf{z}$ are vectors pointing in the same direction. This shows that $\nabla_{\mathbf{z}^*} g$ determines the direction of maximum rate of change with respect to $\mathbf{z}$. □

6.2 Some examples of complex differentials

It is straightforward to extend the real-valued estimators presented in the earlier sections to the complex-valued setting. By applying the rules of $\mathbb{CR}$-calculus and subgradients (see Appendix A), we have the following results

a. $\nabla_{\boldsymbol{\beta}^*} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 = \nabla_{\boldsymbol{\beta}^*} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^{\mathsf{H}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = -\mathbf{X}^{\mathsf{H}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$

b. $\nabla_{\boldsymbol{\beta}^*} \|\boldsymbol{\beta}\|_2^2 = \nabla_{\boldsymbol{\beta}^*} \boldsymbol{\beta}^{\mathsf{H}} \boldsymbol{\beta} = \boldsymbol{\beta}$

c. $\partial_{\boldsymbol{\beta}^*} \|\boldsymbol{\beta}\|_2 = \partial_{\boldsymbol{\beta}^*} \left(\boldsymbol{\beta}^{\mathsf{H}} \boldsymbol{\beta}\right)^{\frac{1}{2}} = \frac{1}{2}\mathbf{s} = \begin{cases} \frac{1}{2} \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_2}, & \text{for } \boldsymbol{\beta} \neq \mathbf{0} \\ \{\frac{1}{2}\mathbf{s} : \|\mathbf{s}\|_2 \leq 1\}, & \text{for } \boldsymbol{\beta} = \mathbf{0} \end{cases}$

d. $\left(\partial_{\boldsymbol{\beta}^*} \|\boldsymbol{\beta}\|_1\right)_j = \partial_{\beta_j^*} \left(\beta_j^* \beta_j\right)^{\frac{1}{2}} = \frac{1}{2} t_j = \begin{cases} \frac{1}{2} \operatorname{sign}(\beta_j), & \text{for } \beta_j \neq 0 \\ \{\frac{1}{2} t_j : |t_j| \leq 1\}, & \text{for } \beta_j = 0. \end{cases}$

Using these results, it is easy to verify that the subgradient equations for the lasso, elastic net, and the square-root elastic net stay the same in both real and complex-valued settings.

7 Direction-of-arrival estimation

Direction-of-arrival (DOA) estimation is an active and important field of research with a vast number of distinct applications, typically in radar, sonar-, seismic-, acoustic-, communications, and biomedical systems [32].

In DOA estimation, the objective is to estimate the directions of a possibly unknown number of radiating sources. In the problem setting, an array of passive sensors or transducers capture the signals emitted by stationary or moving radiating sources. Depending on the application, the captured signal can be by nature, e.g., acoustic, electromagnetic, or seismic. From the measured array data, it is then possible to estimate the source directions by means of estimating the angles of incidence of the incoming signals with respect to the sensor array axis.

In our developments, we consider the signal capturing device to be a uniform linear array (ULA) in which the sensors are aligned in line with a uniform interspacing as shown in Figure 9. The basic working principle of the ULA is very simple. Consider a planewave incoming to the ULA. If the angle of incidence is oblique, then the wavefront will reach the sensors at slightly different time instants, referred to as time difference of arrival (TDOA). This constitutes a phase difference in the output of the sensors. Since the sensor locations are known, by estimating the phase difference it is possible to approximately resolve the direction of the source.

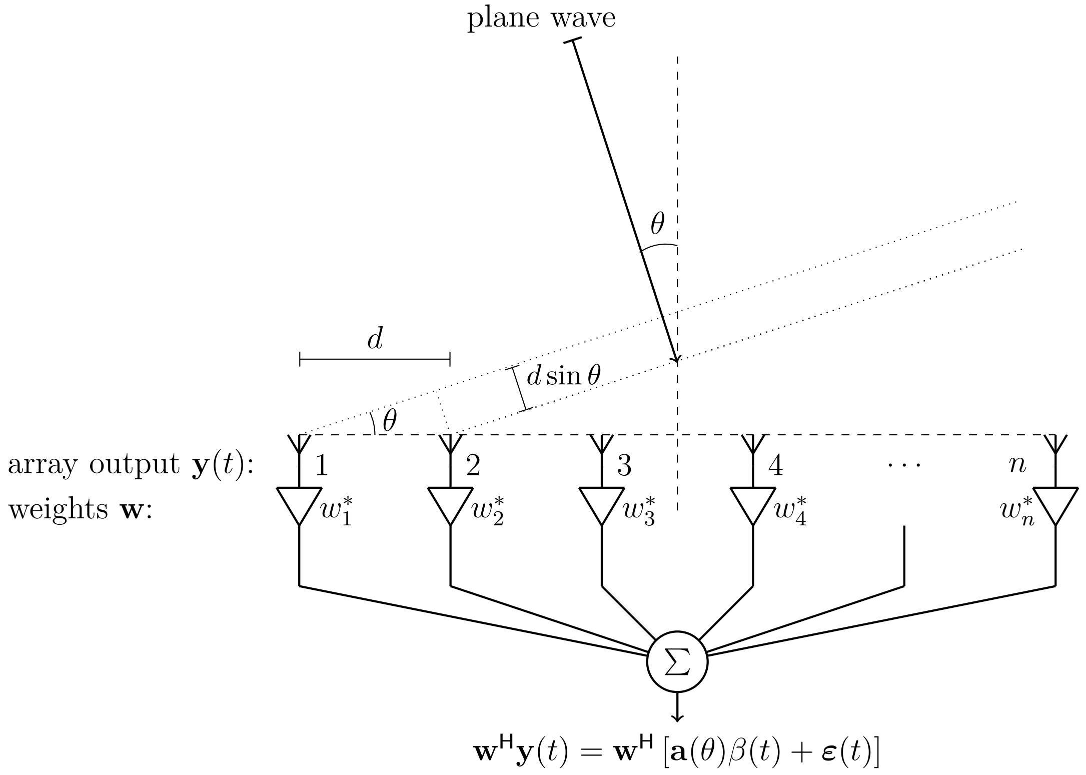
plane wave
θ
d
$d \sin\theta$
θ
array output $\mathbf{y}(t)$:
weights $\mathbf{w}$:
1
2
3
4
$\cdots$
n
w_1^*
w_2^*
w_3^*
w_4^*
w_n^*
Σ
$\mathbf{w}^{\mathsf{H}}\mathbf{y}(t) = \mathbf{w}^{\mathsf{H}}\left[\mathbf{a}(\theta)\beta(t) + \boldsymbol{\varepsilon}(t)\right]$

Figure 9: A uniform linear array (ULA).

Since the data is spatially sampled, the estimation of the DOA can be considered a spatio-spectral estimation problem, and consequently, most of the well-known methods in spectral analysis and frequency estimation can be applied to DOA estimation. The distinction is that, the spectrum now represents the angles of incidence of the plane waves [32].

Traditional methods for estimating the DOA include: conventional beamforming (CBF), the minimum variance distortionless response (MVDR), also known as Capon's method [33], and subspace methods such as multiple signal classification (MUSIC) [34]. Some methods, such as the MVDR, require the inversion of the covariance matrix of the measured data, and thus require at least as many *snapshots* as there are sensors. Here, *snapshot* refers to a single measurement from the sensors at a particular time instant.

In this section, we will first formulate the signal model of the direction-of-arrival problem in the context of the ULA, after which we will overview some traditional DOA methods. Then, we will show how sparse linear regression can be applied to the problem. In Section 8, we provide a simulation study where the scaled sparse linear regression estimators are applied to the problem.

For the reader interested in DOA estimation, the books [32] and [35] provide an introduction to the subject, and for a more comprehensive treatment, the book [36] can be recommended.

It should be mentioned that in the recent years, there has been some interesting advancements in line spectral estimation using semidefinite programming; see, e.g., [37]–[39]. These grid-free methods can also be applied to DOA estimation; see, e.g., [40].

7.1 Signal model

In order to simplify the problem, we make the following assumptions. It is assumed that the medium of propagation is non-dispersive and homogeneous. The sources are assumed to be isotropically radiating point sources located in the far field such that the incoming signals can be regarded as plane waves. We also restrict our analysis to a two-dimensional plane such that the direction of a source k can be defined solely with one parameter, the angle θ_k, which is referred to as the *steering angle*. Furthermore, we assume the incoming signal to be centered at a carrier frequency f_c and occupying a narrowband of the frequency spectrum. Lastly, the sensor elements are assumed to be identical with a flat frequency response.

When the ULA receives a signal, the wavefront of the plane wave reaches the sensor elements sequentially with a time delay τ referred to as time difference of arrival (TDOA). This constitutes a phase difference in the sensor outputs, which will in turn depend on the angle of incidence of the plane wave and the interspacing d of the sensors. Assuming that the plane wave strikes the outermost sensor $i = 1$ at time $\tau_1 = 0$ with an angle $\theta \in [-90°, 90°]$, the time instants for the plane wave hitting the sensors will be

$$\tau_i = (i-1)\frac{d \sin\theta}{c}, \quad i = 1, \dots, n, \tag{7.1}$$

where c is the propagation velocity of the plane wave [32, p. 271]. The phase difference due to propagation delay in the output of sensor i is $\phi_i = 2\pi f_c \tau_i$. The spatial phase differences between the sensors are collected into the *steering vector*

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{n}}(e^{-\jmath\phi_1}, \dots, e^{-\jmath\phi_n})^\top, \tag{7.2}$$

$$= \frac{1}{\sqrt{n}}(e^{-\jmath 2\pi f_c \tau_1}, \dots, e^{-\jmath 2\pi f_c \tau_n})^\top, \tag{7.3}$$

$$= \frac{1}{\sqrt{n}}(1, e^{-\jmath 2\pi f_c \frac{d \sin\theta}{c}}, \dots, e^{-\jmath 2\pi f_c (n-1)\frac{d \sin\theta}{c}})^\top, \tag{7.4}$$

where f_c is the carrier frequency and n is the number of sensors. Since the elements of the steering vector (7.4) are integer powers of the same number, $\mathbf{a}(\theta)$ is a Vandermonde vector. In order to avoid spatial aliasing, the interspacing of the sensors projected to the direction of the incoming plane wave must be smaller than half the wavelength of the incoming plane wave, i.e.,

$$d|\sin\theta| \leq \frac{\lambda}{2} \Leftrightarrow d \leq \frac{\lambda}{2}.$$

If we choose $d = \lambda/2$ the steering vector simplifies to

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{n}}(1, e^{-\jmath\pi\sin\theta}, \dots, e^{-\jmath\pi(n-1)\sin\theta})^\top.$$

Let $\mathbf{y}(t) \in \mathbb{C}^n$ denote a measurement from the n sensors at a discrete time instant t. In case of a single stationary source with DOA θ_1, the measurement is of the form

$$\mathbf{y}(t) = \mathbf{a}(\theta_1)\beta(t) + \boldsymbol{\varepsilon}(t), \tag{7.5}$$

where $\beta(t) \in \mathbb{C}$ is the *signal of interest* (SOI) and $\boldsymbol{\varepsilon}(t) \in \mathbb{C}^n$ is a complex-valued noise vector. In the case of k sources, the measurement $\mathbf{y}$ has the form

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\boldsymbol{\beta}(t) + \boldsymbol{\varepsilon}(t), \tag{7.6}$$

where $\mathbf{A}(\boldsymbol{\theta}) = (\mathbf{a}(\theta_1) \cdots \mathbf{a}(\theta_k)) \in \mathbb{C}^{n \times k}$ is referred to as the *steering matrix* and $\boldsymbol{\beta}(t) = (\beta_1(t), \dots, \beta_k(t))^\top \in \mathbb{C}^k$ is the signal vector consisting of the complex-valued source amplitudes.

When we have multiple snapshots, i.e., the signal is sampled at L time instants, $t \in \{t_1, t_2, \dots, t_L\}$, then we can write the measurements in a *multiple measurement vector* (MMV) form:

$$\mathbf{Y} = \mathbf{AB} + \mathbf{E}, \tag{7.7}$$

where the L snapshots are stacked as columns into the matrix $\mathbf{Y} = (\mathbf{y}(t_1) \cdots \mathbf{y}(t_L)) \in \mathbb{C}^{n \times L}$, $\mathbf{B} = (\boldsymbol{\beta}(t_1) \cdots \boldsymbol{\beta}(t_L)) \in \mathbb{C}^{k \times L}$ is a matrix consisting of L snapshots of k source signals, and $\mathbf{E} = (\boldsymbol{\varepsilon}(t_1) \cdots \boldsymbol{\varepsilon}(t_L)) \in \mathbb{C}^{n \times L}$ is a matrix of complex noise vectors.

7.2 Conventional beamformer (CBF)

In conventional beamforming (CBF), also known as the Bartlett beamformer, or the delay-and-sum beamformer, we are interested in finding the filter coefficients $\mathbf{w} \in \mathbb{C}^n$, so that when applied to the array output $\mathbf{y}$, the phase differences due to propagation delay are cancelled for a plane wave incident from a specific *look direction* θ. In order to provide a more formal definition, we introduce the concept of *array output power* $P(\theta)$ which is a function of the steering angle, and it is defined as

$$\begin{aligned} P(\theta) &\triangleq \mathbb{E}\left[\left|\mathbf{w}(\theta)^{\mathsf{H}}\mathbf{y}\right|^2\right] \\ &= \mathbf{w}^{\mathsf{H}}(\theta)\boldsymbol{\Sigma}\mathbf{w}(\theta), \end{aligned} \tag{7.8}$$

where $\mathbf{w}$ is the filter coefficient vector and $\boldsymbol{\Sigma} = \mathbb{E}\left[\mathbf{y}\mathbf{y}^{\mathsf{H}}\right]$ is the covariance matrix. As the true covariance matrix is usually unknown, the sample covariance matrix $\mathbf{S} = \frac{1}{L}\sum_{i=1}^{L}\mathbf{y}_i\mathbf{y}_i^{\mathsf{H}}$ is typically used in place of $\boldsymbol{\Sigma}$.

In CBF, we minimize the array output power for a spatially white signal, i.e., $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, while keeping the array output power fixed to unity for a specific look direction θ. Thus, we have the optimization problem

$$\begin{aligned} \underset{\mathbf{w}\in\mathbb{C}^n}{\text{minimize}} \quad & \mathbf{w}^{\mathsf{H}}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) = 1. \end{aligned} \tag{7.9}$$

The optimal filter coefficients can be acquired, e.g., with the method of Lagrange multipliers, yielding $\mathbf{w} = \mathbf{a}(\theta)$ [41]. For the derivation, see Appendix B. By inserting the solution into 7.8, we obtain the estimates for the direction of arrivals as the largest peaks of the function

$$P_{\text{CBF}}(\theta) = \mathbf{a}(\theta)^{\mathsf{H}}\mathbf{S}\mathbf{a}(\theta), \tag{7.10}$$

where $\mathbf{S}$ is the sample covariance matrix of the sensor output signal.

The resolution of the conventional beamformer depends on the number of array elements via the formula $\phi_{\text{res}} = \frac{2\pi}{n}$, e.g., for a $n = 10$ element ULA, the smallest angle between DOAs that will still be resolved is $2\pi/10 \approx 0.63$ rad corresponding to $12°$ [41].

7.3 Minimum variance distortionless response (MVDR)

The *minimum variance distortionless response* (MVDR), which is known as the Capon method [33], [42], is also based on minimizing the average power of the filter output while keeping the signal at unity in the specified look direction. However, the MVDR differs from the CBF method in that it uses the actual data in the optimization of the filter coefficients. The MVDR beamformer can be obtained by solving the quadratic problem

$$\begin{aligned} \underset{\mathbf{w}\in\mathbb{C}^n}{\text{minimize}} \quad & \mathbf{w}^{\mathsf{H}}\mathbf{S}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) = 1. \end{aligned} \tag{7.11}$$

The solution to this optimization problem is

$$\mathbf{w} = \frac{\mathbf{S}^{-1}\mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{S}^{-1}\mathbf{a}(\theta)}. \tag{7.12}$$

For the derivations, see Appendix B. By inserting 7.12 into 7.8, the DOA estimates are obtained as the largest peaks of the function

$$P_{\mathrm{MVDR}}(\theta)=\frac{1}{\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{S}^{-1}\mathbf{a}(\theta)}. \tag{7.13}$$

Note that the sample covariance matrix $\mathbf{S}$ has to be full rank to be invertible, i.e., we require at least n snapshots.

7.4 Multiple signal classification (MUSIC)

Multiple signal classification (MUSIC) [34] is a method based on separating the signal subspace from the noise subspace via an eigendecomposition of the sample covariance matrix. The covariance matrix can be written as follows:

$$\begin{aligned}\boldsymbol{\Sigma}&=\mathbb{E}\Big[\mathbf{y}\mathbf{y}^{\mathsf{H}}\Big]\\&=\mathbb{E}\Big[(\mathbf{A}\boldsymbol{\beta}+\boldsymbol{\varepsilon})(\mathbf{A}\boldsymbol{\beta}+\boldsymbol{\varepsilon})^{\mathsf{H}}\Big]\\&=\mathbb{E}\Big[\mathbf{A}\boldsymbol{\beta}\boldsymbol{\beta}^{\mathsf{H}}\mathbf{A}^{\mathsf{H}}+\mathbf{A}\boldsymbol{\beta}\boldsymbol{\varepsilon}^{\mathsf{H}}+\boldsymbol{\varepsilon}\boldsymbol{\beta}^{\mathsf{H}}\mathbf{A}^{\mathsf{H}}+\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathsf{H}}\Big]\\&=\mathbf{A}\mathbb{E}\Big[\boldsymbol{\beta}\boldsymbol{\beta}^{\mathsf{H}}\Big]\mathbf{A}^{\mathsf{H}}+\mathbb{E}\Big[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathsf{H}}\Big]\\&=\mathbf{A}\boldsymbol{\Sigma}_s\mathbf{A}^{\mathsf{H}}+\sigma_{\varepsilon}^2\mathbf{I},\end{aligned} \tag{7.14}$$

where $\mathbf{A}\in\mathbb{C}^{n\times k}$ is the steering matrix with its columns corresponding to the steering vectors of the k sources, $\boldsymbol{\Sigma}_s\in\mathbb{C}^{k\times k}$ is the signal covariance matrix consisting of the squared amplitudes of the sources on its main diagonal, and $\sigma_{\varepsilon}^2\mathbf{I}$ is the covariance matrix of the noise. This structure of the covariance matrix is exploited in the MUSIC method as follows. The eigendecomposition of the sample covariance matrix is $\mathbf{S}=\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\mathsf{H}}$, where $\mathbf{U}\in\mathbb{C}^{n\times n}$ is a matrix of eigenvectors and $\boldsymbol{\Lambda}=\operatorname{diag}(\lambda_1,\lambda_2,\ldots,\lambda_n)$ is a diagonal matrix consisting of eigenvalues. The k largest eigenvalues of $\boldsymbol{\Lambda}$ correspond to the k source signals and the rest of the eigenvalues correspond to the noise. Thus, the eigendecomposition of the sample covariance can be decomposed as

$$\mathbf{S}=\mathbf{U}_s\boldsymbol{\Lambda}_s\mathbf{U}_s^{\mathsf{H}}+\mathbf{U}_{\varepsilon}\boldsymbol{\Lambda}_{\varepsilon}\mathbf{U}_{\varepsilon}^{\mathsf{H}}, \tag{7.15}$$

where $\boldsymbol{\Lambda}_s$ and $\mathbf{U}_s$ are the matrices of eigenvalues and eigenvectors corresponding to the k largest eigenvalues; and $\boldsymbol{\Lambda}_{\varepsilon}$ and $\mathbf{U}_{\varepsilon}$ are the matrices of eigenvalues and eigenvectors corresponding to the noise, respectively. The insight of MUSIC is that the eigenvectors $\mathbf{U}_s$ lie in the same subspace as the steering vectors corresponding to the actual DOA directions, and they are orthogonal to the noise eigenvectors, i.e., $\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{U}_{\varepsilon}=\mathbf{0}$. Hence the DOA estimates can be found as the largest peaks of the function

$$P_{\mathrm{MUSIC}}(\theta)=\frac{1}{\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{U}_{\varepsilon}\mathbf{U}_{\varepsilon}^{\mathsf{H}}\mathbf{a}(\theta)}, \tag{7.16}$$

which is sometimes called as the *pseudospectrum*.

7.5 Direction-of-arrival estimation using ℓ_1-penalization

It is possible to utilize sparse regression methods in direction-of-arrival estimation. Recalling that the signal model for the array measurement is of the form

$$\mathbf{y} = \mathbf{A}_{\text{true}}(\theta)\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{7.17}$$

where $\mathbf{A}_{\text{true}}(\theta) \in \mathbb{C}^{n\times k}$ is the unknown steering matrix consisting of k steering vectors corresponding to the true sources. If we assume that there is only a small number of actual sources, then we can formulate the problem as an underdetermined linear system by constructing a steering matrix $\mathbf{A}(\theta) \in \mathbb{C}^{n\times p}$ with a dense grid of hypothetical directions, i.e., $p \gg k$. Assuming the true source directions fall exactly on the grid, the true coefficient vector $\boldsymbol{\beta} \in \mathbb{C}^{p}$ will be sparse with only k non-zero coefficients corresponding to the sources. Hence, the problem is sparse and we can use sparse regression methods for estimating the DOAs. The method can easily be extended to the multiple snapshot case as well; see, e.g., [43]. The idea of incorporating sparse regression methods in DOA estimation has become to be known as *compressive beamforming.* It has proven to be a successful approach outperforming many conventional methods by producing high-resolution estimates of the source directions. It also has the benefit that it can be applied even with a single snapshot; see, e.g., [43]–[46] and references therein.

8 A simulation study: single snapshot compressive beamforming

In this section, we examine the performance of the scaled lasso, the scaled elastic net, and the square-root elastic net when applied to single snapshot compressive beamforming. We study how the performance changes as a function of the EN tuning parameter α, when it is decreased from unity, corresponding to the scaled lasso, towards zero.

The steering matrix $\mathbf{A}(\boldsymbol{\theta})$ is a coherent basis since it is a oversampled DFT basis. When using a half wavelength interelement spacing, the correlation between the columns i and j of the steering matrix takes the form

$$|\mathbf{a}(\theta_i)^{\mathsf{H}}\mathbf{a}(\theta_j)| = \frac{1}{n}\left|\sum_{q=0}^{n-1} e^{\jmath\pi q(\sin\theta_i - \sin\theta_j)}\right|. \tag{8.1}$$

Since the correlation between the steering vectors depend on the difference of the sine of the steering angles in (8.1), the mutual correlation between the steering vectors is not uniform. Instead, the correlation of a given steering vector with the rest of the basis depends on the steering angle θ, and the correlation generally increases as the steering angle deviates further away from 0°. Thus, we would expect the proposed elastic net formulations to improve upon the scaled lasso when the angle of incidence of the signal becomes more oblique with respect to the ULA axis.

To test this hypothesis, we simulate two cases. In the first set-up, we have two source signals with the directions-of-arrivals $\theta_1 = 0°$ and $\theta_2 = 8°$. In the second set-up, the signals have the direction-of-arrivals $\theta_1 = 50°$ and $\theta_2 = 58°$. Thus, the angles of incidence are more oblique in the second set-up, which makes their corresponding steering vectors more correlated with the basis, as shown in Table 2. In both cases, the signals have the amplitudes $|\beta_1| = 1$ and $|\beta_2| = 1$, and a random phase generated from the uniform distribution, $\text{unif}(0, 2\pi)$. The additive noise is complex Gaussian with a variance chosen such that the signal-to-noise ratio (SNR) is 20 dB. The steering matrix is formed by a discrete grid from $-90°$ to $90°$ in steps of 2°. Both set-ups are simulated for the penalty parameters $\lambda_1 = \sqrt{\log(p)}$ and $\lambda_2 = \sqrt{2\log(p)}$, and the EN tuning parameter ranging from $\alpha = 0.8$ to $\alpha = 1$ in steps of 0.01. The results are reported in terms of the mean squared error (MSE) as defined in (5.1) and correct peak rate (CPR), which is defined as the rate in which the peaks of the solution vector corresponds to the true DOAs. More explicitly, a peak is defined as an element $\hat{\beta}_j$ satisfying $|\hat{\beta}_{j-1}| < |\hat{\beta}_j|$ and $|\hat{\beta}_{j+1}| < |\hat{\beta}_j|$. The results are averages over 500 Monte-Carlo trials and are shown in Figure 10 for the first set-up, and in Figure 11 for the second set-up.

From the results of the first set-up, it can be deduced that, since the sources were well separated and both of the sources had a small angle of incidence with respect to the ULA axis, their DOAs were relatively easy to estimate. This can be seen from the high CPR in Figure 10. With the penalty value λ_1, the scaled lasso ($\alpha = 1$) performs well and there are no obvious benefits of introducing the elastic net penalty in terms of the CPR. The penalty level λ_2 is obviously too high for the scaled lasso.

Table 2: The correlations between the steering vectors corresponding to the source directions as well as the correlations between one of the sources and its adjacent steering vector in the basis.

	θ_1	θ_2	$\lvert\mathbf{a}(\theta_1)^{\mathsf{H}}\mathbf{a}(\theta_2)\rvert$
DOAs in set-up 1	0°	8°	0.22
Correlation with basis in set-up 1	8°	10°	0.82
DOAs in set-up 2	50°	58°	0.21
Correlation with basis in set-up 2	58°	60°	0.95

However, for the elastic net formulations, when α is decreased, the performance in terms of CPR improves. This may probably be caused by the fact that decreasing α also decreases the weight of the (too high) ℓ_1-norm penalty.

From the results of the second set-up, which are shown in Figure 11, we can notice the benefits of the elastic net formulations. With penalty parameter λ_1, the elastic net formulations outperform the scaled lasso both in terms of CPR and MSE when the EN tuning parameter is in the range $\alpha \in (0.9, 1)$. When the penalty level is λ_2, which is again too high for the scaled lasso, the performance in terms of CPR increases as the EN tuning parameter is lowered, as in the first set-up.

It is important to notice that even though the MSE was smaller with the penalty parameter λ_2 in both of the cases, the smaller penalty level λ_1 yielded better results in terms of the CPR.

It can be concluded that when we have an oblique angle of incidence, the steering vectors are more correlated with the basis and the scaled and square-root elastic net estimators outperform the scaled lasso. On the other hand, when the angle of incidence is small, the benefit of using the scaled and square-root elastic net estimators is not as pronounced.

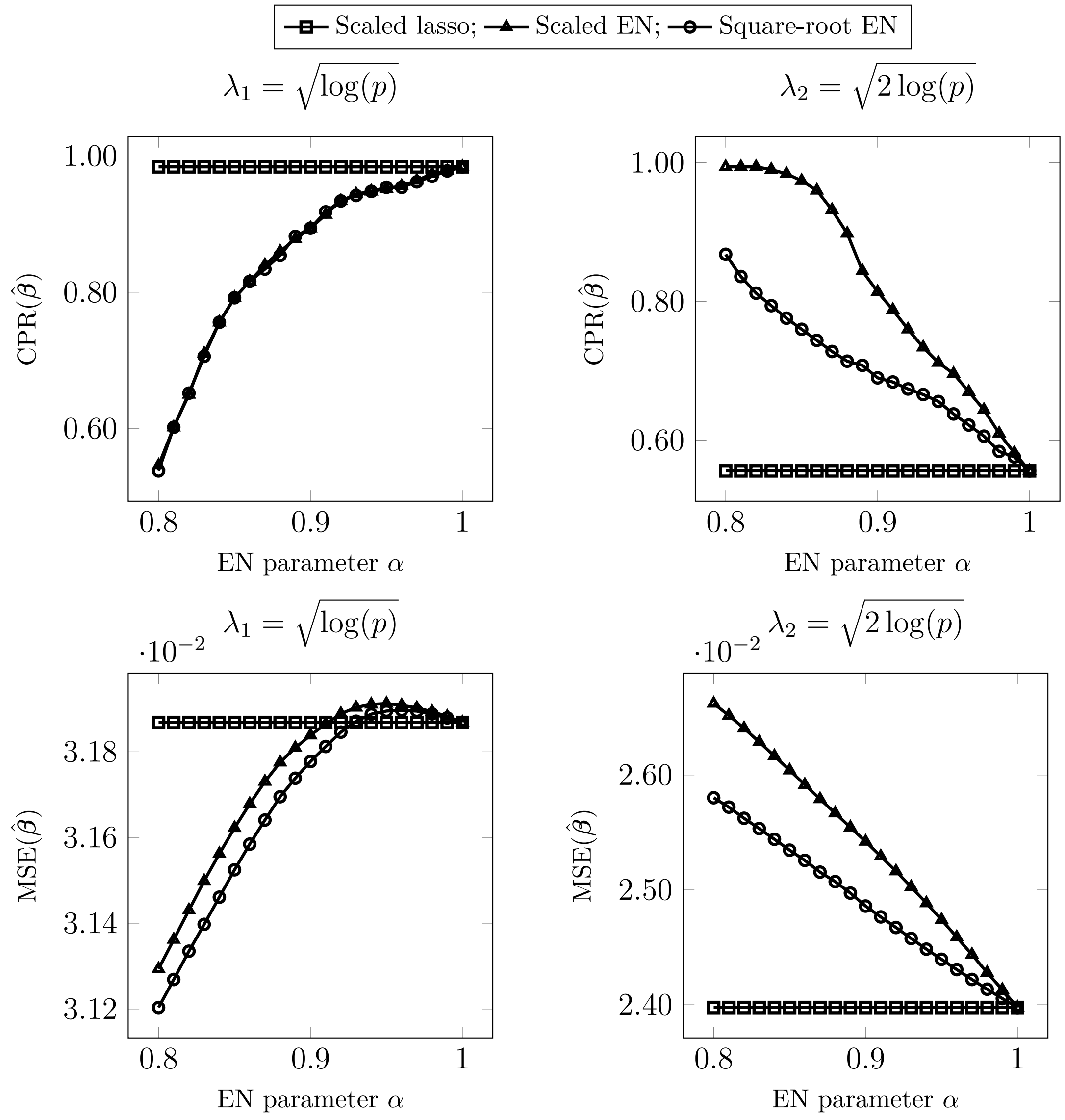

Figure 10: Correct peak rate (CPR) and mean squared error (MSE) as a function of the EN tuning parameter for two sources at 0° and 8°, and the penalty parameters $\lambda_1 = \sqrt{\log(p)}$ (*left*) and $\lambda_2 = \sqrt{2\log(p)}$ (*right*).

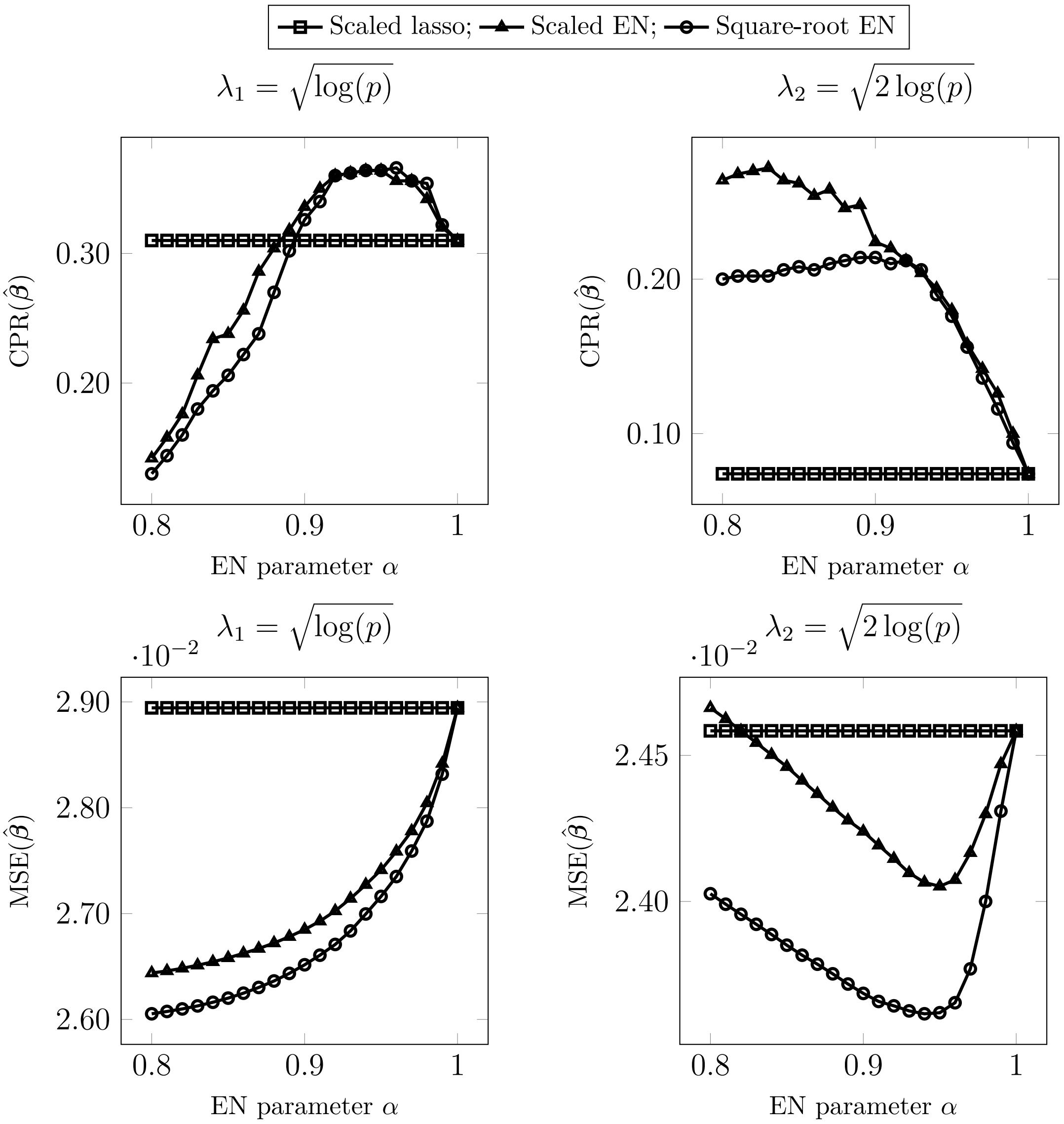

Figure 11: Correct peak rate (CPR) and mean squared error (MSE) as a function of the EN tuning parameter for two sources at 50° and 58°, and the penalty parameters $\lambda_1 = \sqrt{\log(p)}$ (*left*) and $\lambda_2 = \sqrt{2\log(p)}$ (*right*).

9 Conclusions

In scaled sparse linear regression, the regression coefficients and the noise scale are estimated jointly. The main benefit is that the penalty parameter controlling the trade-off between the sparsity level and the fit is no longer dependent of the noise level. This enables the usage of analytically derived optimal penalty levels.

In this thesis and in the publication [4], two elastic net extensions of the scaled lasso estimator were proposed: the scaled elastic net and the square-root elastic net. It was demonstrated via numerical examples and simulations that the proposed estimators outperformed the scaled lasso in terms of MSE, especially in the presence of high correlations in the feature space. The proposed estimators also encouraged the grouping effect of collinear variables. Furthermore, they don't have the particular shortcoming of the scaled lasso of being able to only choose at most n predictors when $p > n$.

The proposed estimators have an additional EN tuning parameter $\alpha \in [0, 1]$, which controls the mixing between the ℓ_1-norm and ℓ_2-norm terms in the penalty. In our experience, the EN tuning parameter should be chosen close to unity such that the estimators induce sparsity, but are still able to encourage the grouping effect, and in effect reduce the MSE. This view was also supported by the numerical examples and simulations conducted in this thesis. However, choosing an optimal EN tuning parameter is an important topic of its own, which will be addressed in the future research.

In the application of single snapshot direction-of-arrival estimation, the proposed estimators performed better than scaled lasso when the DOAs had an oblique angle of incidence with respect to the ULA axis; a situation in which the significant variables are highly correlated with the rest of the basis vectors. In direction-of-arrival estimation, the most important criteria was the correct peak rate (CPR), which is used to determine the DOA angles. However, a good CPR did not guarantee a low MSE and vice versa. In fact, a lower penalty level gave a better CPR, although a higher penalty level gave a lower MSE.

As was shown in this thesis, the optimal penalty level does not only depend on the noise scale, but also of the true sparsity level and the norm of the true coefficients. Therefore, for a given problem, in order to guarantee the best performance, the penalty level as well as the EN tuning parameter must be chosen among a set of candidate values, e.g., as done in Section 5.3.

It is fair to conclude that, scaled linear regression improve on traditional penalized regression methods by simplifying the selection of the correct penalty level with little to none computational overhead.

References

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*, 2nd ed. Springer, 2009.

[2] D. G. Luenberger, *Optimization by vector space methods.* Wiley, 1969.

[3] F. Bunea, J. Lederer, and Y. She, “The group square-root lasso: Theoretical properties and fast algorithms”, *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1313–1325, 2014.

[4] E. Raninen and E. Ollila, “Scaled and square-root elastic net”, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’17)*, New Orleans, Louisiana, U.S.A., 2017, pp. 4336–4340.

[5] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems”, *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[6] R. Tibshirani, “Regression shrinkage and selection via the lasso”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[7] B. M. G. Kibria, “Performance of Some New Ridge Regression Estimators”, *Communications in Statistics - Simulation and Computation*, vol. 32, no. 2, pp. 419–435, 2003.

[8] A. E. Hoerl, R. W. Kennard, and K. F. Baldwin, “Ridge regression: Some simulations”, *Communications in Statistics - Theory and Methods*, vol. 4, no. 2, pp. 105–123, 1975.

[9] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, “Pathwise coordinate optimization”, *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[10] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: The lasso and generalizations.* CRC Press, 2015.

[11] P. Bühlmann and S. van de Geer, *Statistics for high-dimensional data: Methods, theory and applications.* Springer, 2011.

[12] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net”, *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[13] H. Liu and J. Jia, “On ℓ_2 error bounds of the elastic net when $p \gg n$”, *Statistica Sinica*, vol. 20, pp. 595–611, 2012.

[14] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, “A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers”, *Statistical Science*, vol. 27, no. 4, pp. 538–557, 2012.

[15] T. Sun and C.-H. Zhang, “Scaled sparse linear regression”, *Biometrika*, vol. 99, no. 4, pp. 879–898, 2012.

[16] A. Belloni, V. Chernozhukov, and L. Wang, “Square-root lasso: Pivotal recovery of sparse signals via conic programming”, *Biometrika*, vol. 98, no. 4, pp. 791–806, 2011.

[17] N. Städler, P. Bühlmann, and S. van de Geer, “ℓ_1-penalization for mixture regression models”, *Test*, vol. 19, no. 2, pp. 209–256, 2010.

[18] T. Sun and C.-H. Zhang, “Comments on: ℓ_1-penalization for mixture regression models”, *Test*, vol. 19, no. 2, pp. 270–275, 2010.

[19] A. Antoniadis, “Comments on: $\ell_1$1-penalization for mixture regression models”, *Test*, vol. 19, no. 2, pp. 257–258, 2010.

[20] P. Huber, *Robust statistics.* Wiley, 1981.

[21] A. B. Owen, “A robust hybrid of lasso and ridge regression”, *Contemporary Mathematics*, vol. 443, pp. 59–72, 2007.

[22] S. Reid, R. Tibshirani, and J. Friedman, “A study of error variance estimation in lasso regression”, *Preprint*, 2013. arXiv: 1311.5274v2.

[23] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization”, *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[24] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, “A sparse-group lasso”, *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[25] D. Brandwood, “A complex gradient operator and its application in adaptive array theory”, *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 130, no. 1, pp. 11–16, Feb. 1983.

[26] A. van den Bos, “Complex gradient and hessian”, *IEE Proceedings - Vision, Image, and Signal Processing*, vol. 141, no. 6, pp. 380–383, 1994.

[27] K. Kreutz-Delgado, *The complex gradient operator and the cr-calculus*, Electrical and Computer Engineering, University of California, San Diego., 2009. [Online]. Available: http://arxiv.org/abs/0906.4835.

[28] T. Adali and H. Li, “Complex-valued adaptive signal processing”, in *Adaptive Signal Processing: Next Generation Solutions.* Wiley, 2010, pp. 1–85.

[29] J. Eriksson, E. Ollila, and V. Koivunen, “Essential statistics and tools for complex random variables”, *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5400–5408, 2010.

[30] E. Ollila, “Contributions to independent component analysis, sensor array and complex-valued signal processing”, PhD thesis, Helsinki University of Technology, 2010.

[31] S. Haykin, *Adaptive filter theory*, 5th ed. Pearson, 2013.

[32] P. Stoica and R. Moses, *Spectral analysis of signals.* Prentice-Hall, 2005.

[33] J. Capon, “High-resolution frequency-wavenumber spectrum analysis”, *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[34] R. Schmidt, “Multiple emitter location and signal parameter estimation”, *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[35] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and adaptive signal processing: Spectral estimation, signal modeling, adaptive filtering and array processing.* Artech House, 2005.

[36] H. L. V. Trees, *Detection, estimation, and modulation theory, optimum array processing (part iv).* Wiley, 2002.

[37] E. J. Candès and C. Fernandez-Granda, “Towards a mathematical theory of super-resolution”, *Communications on Pure and Applied Mathematics*, vol. 67, no. 6, pp. 906–956, 2014.

[38] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht, “Compressed sensing off the grid”, *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.

[39] B. N. Bhaskar, G. Tang, and B. Recht, “Atomic norm denoising with applications to line spectral estimation”, *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5987–5999, 2013.

[40] A. Xenaki and P. Gerstoft, “Grid-free compressive beamforming”, *The Journal of the Acoustical Society Of America*, vol. 137, pp. 1923–1935, 2015.

[41] H. Krim and M. Viberg, “Two decades of array signal processing research: The parametric approach”, *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.

[42] R. T. Lacoss, “Data adaptive spectral analysis methods”, *Geophysics*, vol. 36, no. 4, pp. 661–675, 1971.

[43] D. Malioutov, M. Çetin, and A. S. Willsky, “A sparse signal reconstruction perspective for source localization with sensor arrays”, *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[44] S. Fortunati, R. Grasso, F. Gini, M. S. Greco, and K. LePage, “Single-snapshot doa estimation by using compressed sensing”, *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, p. 120, 2014.

[45] A. Xenaki, P. Gerstoft, and K. Mosegaard, “Compressive beamforming”, *The Journal of the Acoustical Society Of America*, vol. 136, no. 1, pp. 260–271, 2014.

[46] P. Gerstoft, A. Xenaki, and C. F. Mecklenbräuker, “Multiple and single snapshot compressive beamforming”, *The Journal of the Acoustical Society Of America*, vol. 138, no. 4, pp. 2003–2014, 2015.

[47] R. T. Rockafellar, *Convex analysis.* Princeton University Press, 1970.

A The subdifferential

The subdifferential generalizes the derivative of convex functions to include also *not everywhere* differentiable functions. For example, the ℓ_1-norm is not differentiable when its argument equals zero. However, since the norm is convex, we can make use of subdifferentiation [47, p.215]. The *subgradient* is defined in the next definition.

Definition 2. *A vector* $\mathbf{g} \in \mathbb{R}^p$ *is called a subgradient of the convex function* $f : \mathbb{R}^p \to \mathbb{R}$ *at the point* $\boldsymbol{\beta}$ *if*

$$f(\boldsymbol{\beta}') \geq f(\boldsymbol{\beta}) + \langle \mathbf{g}, \boldsymbol{\beta}' - \boldsymbol{\beta} \rangle$$

holds for all $\boldsymbol{\beta}' \in \mathbb{R}^p$*.*

The subgradient is not unique and the set of all subgradients at a certain point is called the *subdifferential*, which is defined in the next definition.

Definition 3. *The subdifferential of a convex function* $f : \mathbb{R}^p \to \mathbb{R}$ *at the point* $\boldsymbol{\beta}$ *is the set of all subgradients,*

$$\partial f(\boldsymbol{\beta}) = \left\{ \mathbf{g} \in \mathbb{R}^p : \forall \boldsymbol{\beta}' \in \mathbb{R}^p, f(\boldsymbol{\beta}') \geq f(\boldsymbol{\beta}) + \langle \mathbf{g}, \boldsymbol{\beta}' - \boldsymbol{\beta} \rangle \right\}.$$

Remark 1. *At points where the function is differentiable, the subdifferential reduces to the gradient, i.e.,* $\partial f(\boldsymbol{\beta}) = \{\nabla f(\boldsymbol{\beta})\}$*.*

A geometric visualization of the subdifferential of the ℓ_1-norm is shown in Figure 12, where two different subgradients define different slopes to lines touching the origin; the point, where the ℓ_1-norm is not differentiable.

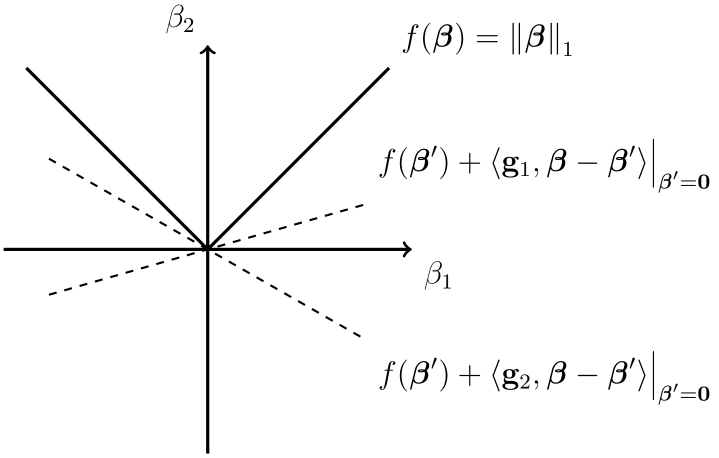
β_2
$f(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$
$f(\boldsymbol{\beta}') + \langle \mathbf{g}_1, \boldsymbol{\beta} - \boldsymbol{\beta}' \rangle \big|_{\boldsymbol{\beta}'=\mathbf{0}}$
β_1
$f(\boldsymbol{\beta}') + \langle \mathbf{g}_2, \boldsymbol{\beta} - \boldsymbol{\beta}' \rangle \big|_{\boldsymbol{\beta}'=\mathbf{0}}$

Figure 12: An illustration of the subdifferential of the ℓ_1-norm. The vectors $\mathbf{g}_1$ and $\mathbf{g}_2$ belong to the subdifferential of the ℓ_1-norm.

The subdifferential can also be extended to convex functions with complex-valued arguments by the following definitions.

Definition 4. *A vector* $\mathbf{g} \in \mathbb{C}^p$ *is called a subgradient of a convex function* $f : \mathbb{C}^p \to \mathbb{R}$ *at the point* $\boldsymbol{\beta}$ *if*

$$f(\boldsymbol{\beta}') \geq f(\boldsymbol{\beta}) + 2\,\mathrm{Re}\,\langle \mathbf{g}, \boldsymbol{\beta}' - \boldsymbol{\beta} \rangle$$

holds for all $\boldsymbol{\beta}' \in \mathbb{C}^p$.

Definition 5. *The subdifferential of the convex function* $f : \mathbb{C}^p \to \mathbb{R}$ *at the point* $\boldsymbol{\beta}$ *is the set of all subgradients,*

$$\partial f(\boldsymbol{\beta}) = \left\{ \mathbf{g} \in \mathbb{C}^p : \forall \boldsymbol{\beta}' \in \mathbb{C}^p, f(\boldsymbol{\beta}') \geq f(\boldsymbol{\beta}) + 2\,\mathrm{Re}\,\langle \mathbf{g}, \boldsymbol{\beta}' - \boldsymbol{\beta} \rangle \right\}.$$

B Derivation of beamformer filter coefficients using $\mathbb{CR}$-calculus

The optimization program for the MVDR beamformer is defined as

$$\begin{aligned} \underset{\mathbf{w}\in\mathbb{C}^n}{\text{minimize}} \quad & \mathbf{w}^{\mathsf{H}}\boldsymbol{\Sigma}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) = 1, \end{aligned}$$

where $\mathbf{w} \in \mathbb{C}$ are the filter coefficients to be optimized, $\boldsymbol{\Sigma}$ is the covariance matrix, and $\mathbf{a}(\theta) \in \mathbb{C}$ is a steering vector satisfying the property $\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{a}(\theta) = 1$. The optimization program of the CBF is a special case of the former with the difference of the covariance matrix being the identity matrix. The following derivations presented here are similar to that of [25]. In order to solve the optimization program, we will use the method of Lagrangian multipliers. The Lagrangian takes the form

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \mathbf{w}^*) &= \mathbf{w}^{\mathsf{H}}\boldsymbol{\Sigma}\mathbf{w} + 2\operatorname{Re}\Big(\lambda(\mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) - 1)\Big) \\ &= \mathbf{w}^{\mathsf{H}}\boldsymbol{\Sigma}\mathbf{w} + \lambda(\mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) - 1) + \lambda^*(\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{w} - 1). \end{aligned}$$

For the sake of mathematical convenience, we added the factor 2 to simplify subsequent expressions. Next, we solve for the critical point by differentiating the Lagrangian with respect to $\mathbf{w}^*$. That is,

$$\nabla_{\mathbf{w}^*}\mathcal{L}(\mathbf{w}, \mathbf{w}^*) = \boldsymbol{\Sigma}\mathbf{w} + \lambda\mathbf{a}(\theta) = 0.$$

If the covariance matrix is invertible, the solution for the weights is

$$\mathbf{w} = -\lambda\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta). \tag{B.1}$$

Substituting this into the constraint $\mathbf{w}^{\mathsf{H}}\mathbf{a}(\theta) = 1$ yields

$$\Big(-\lambda\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta)\Big)^{\mathsf{H}}\mathbf{a}(\theta) = 1.$$

Taking the Hermitian transpose of both sides of the equation yields

$$-\mathbf{a}(\theta)^{\mathsf{H}}\lambda\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta) = 1,$$

from which solving for λ gives

$$\lambda = -\frac{1}{\mathbf{a}(\theta)^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta)}.$$

Substituting λ into (B.1) yields the optimal filter coefficients for the MVDR beamformer:

$$\mathbf{w} = \frac{\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathsf{H}}\boldsymbol{\Sigma}^{-1}\mathbf{a}(\theta)}.$$

In CBF, we have $\boldsymbol{\Sigma} = \mathbf{I}$, which reduces the optimal filter coefficients to

$$\mathbf{w} = \frac{\mathbf{a}(\theta)}{\mathbf{a}(\theta)^{\mathsf{H}}\mathbf{a}(\theta)} = \mathbf{a}(\theta).$$