# A Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays

by

Dmitry M. Malioutov

B.S., Electrical and Computer Engineering
Northeastern University, 2001

Submitted to the Department of Electrical Engineering
and Computer Science in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology

July 2003



Signature of Author: _____

Department of Electrical Engineering
and Computer Science
July 29, 2003

Certified by: _____

Müjdat Çetin
Title: Research Scientist,
Laboratory for Information and Decision Systems
Thesis Supervisor

Accepted by: _____

Arthur C. Smith
Chairman
Department Committee on Graduate Students

# A Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays

by Dmitry M. Malioutov

Submitted to the Department of Electrical Engineering and Computer Science
on July 29, 2003
in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

## Abstract

The theme for this thesis is the application of the inverse problem framework with sparsity-enforcing regularization to passive source localization in sensor array processing. The approach involves reformulating the problem in an optimization framework by using an overcomplete basis, and applying sparsifying regularization, thus focusing the signal energy to achieve excellent resolution. We develop numerical methods for enforcing sparsity by using $\ell_1$ and $\ell_p$ regularization. We use the second order cone programming framework for $\ell_1$ regularization, which allows efficient solutions using interior point methods. For the $\ell_p$ counterpart, the numerical solution is based on half-quadratic regularization. We propose several approaches of using multiple time samples of sensor outputs in synergy, and a method for the automatic choice of the regularization parameter. We conduct extensive numerical experiments analyzing the behavior of our approach and comparing it to existing source localization methods. This analysis demonstrates that our approach has important advantages such as superresolution, robustness to noise and limited data, robustness to correlation of the sources and lack of need for accurate initialization. The approach is also extended to allow self-calibration of sensor position errors by using a procedure similar in spirit to block-coordinate descent on an augmented objective function including both the locations of the sources and the positions of the sensors.

The second direction of the work done in the thesis, which is intimately related to our approach for source localization, is theoretical analysis of the noiseless signal representation problem using overcomplete bases. Questions considered in this analysis include the uniqueness of solutions to the noiseless $\ell_0$ problem, and the equivalence of solutions of the $\ell_0$, $\ell_1$ and $\ell_p$ problems. We consider an arbitrary overcomplete basis, and we show that under certain sparsity conditions on the underlying signal, such uniqueness and equivalence holds.

---

Thesis Supervisor: Müjdat Çetin
Title: Research Scientist
Laboratory for Information and Decision Systems

# Acknowledgments

Although my name appears as the sole author of the thesis, in reality a number of people are responsible for the work. First and foremost, I would like to thank my thesis advisor, Mujdat Cetin, and Alan Willsky, the leader of the Stochastic Systems Group. They introduced me to the topics of enforcing sparsity, regularization in inverse problems, and source localization, spent a good deal of time to make sure I understand the basic concepts, steered my research efforts to attack interesting and potentially solvable problems, and contributed many new ideas as the work progressed. I must thank Mujdat, and I encourage the reader to do so as well, for his titanic efforts in making my papers and my thesis readable, and teaching me the process of good scientific writing. Without his help I found myself on several occasions unable to understand my own writing which had been scribbled less than a week before. Working with Alan and Mujdat has been very inspiring: their energy, dedication, creative ideas, excellent organization, and the ability to be actively involved in and have a deep understanding of so many different research topics simultaneously, giving invaluable guidance to so many students on a weekly basis still does not fail to astonish me.

I would also like to thank R. Moses, A. Baggeroer, A. Tsybakov, A. Samarov, M. Zatman, B. Sadler, and V. Stepanov for their discussions of my work and their suggestions for future directions, and improvements thereof. I would like to thank Jos Sturm for his help with SeDuMi, and for sending me patches to make it work for some of the optimization problems encountered in the thesis. I am very grateful to Peter Shor and Robin Blume-Kohout for their explanation of line packing in Euclidean spheres, and for providing me with a proof of the optimality of the regular simplex. Thanks to Brian Sadler for proposing a number of interesting directions for future work, and to Vladimir Stepanov for an interesting conversation on inverse problems.

I have received a great deal of help from my fellow SSG students, which ranges from references to papers, explanation of obscure mathematical ideas, to tips for effective programming. Despite my sincere efforts, some outcomes from the last two years of interaction with this unique group of people did not get reflected in the thesis, but I feel they deserve to be mentioned anyway. I would like to thank my officemates Chen Lei, and Ayres Fan[1] for teaching me the wonders of Mandarin subnormative lexicon, and perfecting my pronunciation; thanks to Jason Johnson for making me realize that I have a potential second career as a billiard hustler, and for introducing me to information geometry, and to structure estimation in graphical models. I have to thank Walter Wall-street Sun for his sincere attempt to try to alleviate my ignorance in financial matters, for almost making me big money (more than a hundred dollars) in FOREX trading,

[1] I found out recently that he also goes by Jī Chaǒ Fàn, and also, to my great surprise, by Eros Fun.

and for stimulating mathematical conversations, especially on the subject of topology. Thanks to Eric Sudderth and Alex Ihler for taking the burden of running the network, and for their help and patience with computing troubles. Thanks to all the other SSG members and former members: Junmo Kim, Dewey Tucker, Andy Tsai (I never ate any of your God-forsaken candy because they tasted awful), Martin Wainwright, Patrick Kreidl, and Ron Dror. Also, despite all the harassment, humiliation, kicks and punches that I received from her, I'd like to thank Taylore Kelly. However, I have to warn her that she is yet to feel the full wrath of introducing me to PhotoShop. Tremble with fear Taylore, your fate is soon to come upon you!

Finally, I would like to thank my friends and my family members for making me forget for brief periods of time about my work, my thesis, about MIT and its problem sets, and focus on fun-filled activities that range from learning how to cook and climb, to fixing a completely wrecked car, and measuring the speed of water in Tena river on a canoe. These moments made the past two years memorable and enjoyable.

# Contents

Chapter 1

# Introduction

In this thesis we consider the problem of sensor array source localization, and present a new approach based on a sparse signal representation perspective. The purpose of this chapter is to introduce the problem addressed in the thesis, motivate the need for a new approach, and describe our main contributions and the organization of the thesis.

## ■ 1.1 Overview of the problem addressed in the thesis

At the core of this thesis is the solution of the sensor array source localization problem by representing it as an inverse problem and imposing sparsifying regularization.

Source localization using sensor arrays is a problem with many important practical applications including wireless communications [1, 2], radar [3, 4], sonar [5], and exploration seismology [6], among many others. The goal is to find the locations of the sources of wavefields which are impinging on an array of sensors. Practical applications require that the estimates of the locations be not only accurate under ideal conditions, but also robust to factors such as measurement noise, limitations in the amount of data, correlation of the sources, and modeling errors. For non-parametric methods, which result in a spatial energy spectrum, it is desired that the spectra have narrow peaks, low sidelobes, and the ability to localize sources even if they are within Rayleigh resolution of each other, i.e. the ability to achieve superresolution. Rayleigh resolution of an array depends on the number of sensors and on the spatial extent of the array, so it is possible to achieve any resolution simply by making larger arrays with more sensors. However, many practical applications have strict limits on the size of the array. One such application is surveillance using sensor networks. For example, suppose a large number of sensors are deployed into unknown terrain to monitor the activity in the area. Sensors are deployed over a large spatial extent, but power consumption limits severely the amount of communications that sensors can have, so source localization cannot be done coherently using all the sensors. Hence local groups of sensors have to provide accurate estimates of the locations of the objects of interest. These small arrays also have to be robust against noise, limited data, and modeling errors.

In many source localization applications the physical dimensions of the sources of energy are small, or the sources are far enough from the array, so that they can be considered point sources. If, in addition, the number of sources is small, then the

spectrum of energy vs. location is sparse. Sparsity is a very valuable property to have. Many advanced source localization techniques for the localization of point-sources achieve superresolution by exploiting sparsity. For example, the key component of the MUSIC method [7] is the assumption of a small-dimensional signal subspace. We follow a different approach for exploiting such structure: we pose source localization as a linear inverse problem and use sparsity enforcing regularization. More specifically, our approach can be viewed as sparse representation of signals in terms of overcomplete bases. In this context, each basis vector corresponds to an array manifold vector for a possible source location among a sampling grid of locations. The representation of the observed sensor data in terms of an overcomplete basis is not unique, and we impose a penalty on lack of sparsity to regain uniqueness and, more importantly, to get sparse spectra.

What penalties enforce sparsity? The ideal penalty to enforce sparsity is the count of nonzero elements of the resulting spectrum (which is sometimes referred to as the $\ell_0$-norm of the spectrum). However, the resulting problem is combinatorial in nature, and requires very heavy computational methods for its solution. We use related $\ell_1$-norm and $\ell_p$-norm penalties instead. The solution of a noiseless signal representation problem using $\ell_0$ penalty has a close connection to solutions using $\ell_1$ and $\ell_p$ penalties. In fact, under some assumptions on the number of sources, we show that in the noiseless case, the solution of these problems is the same for a general overcomplete basis. That means that if the signal of interest has a sparse enough representation in terms of an overcomplete basis, then, instead of using combinatorial optimization associated with $\ell_0$ norms, we can find that sparse representation by imposing an $\ell_1$ or an $\ell_p$ penalty, which leads to more tractable optimization problems. Prior work on this topic considered minimum $\ell_1$-norm representations in terms of a basis consisting of pairs of orthogonal bases [8, 9], and our work extends their results to arbitrary overcomplete bases. The results of equivalence of noiseless representations with minimum $\ell_0$, $\ell_1$, and $\ell_p$ norms for sparse signals serve as a strong motivation for the use of $\ell_1$ and $\ell_p$ penalties to enforce sparsity in the noisy case as well.

The numerical solution of $\ell_1$-norm regularized linear inverse problems is much simpler that the solution of the $\ell_0$ counterpart, since the $\ell_1$-norm penalty leads to convex objective functions. However, the solution is by no means trivial. The objective function is neither linear nor quadratic since we are dealing with $\ell_1$-norms of complex-valued data. We are led to consider second order cone programming (SOC) [10] which can be used to represent the resulting objective function, and also has an efficient procedure for solution through the use of interior point methods. The objective function corresponding to $\ell_p$-norm regularization for $p < 1$ is a closer approximation to the objective function with $\ell_0$ regularization, but, unlike the $\ell_1$ objective function, it is non-convex[1]. We can only expect to converge to local optima using smooth local optimization techniques. Global optimization methods are inherently computation-intensive, thus we do

[1]For $p < 1$ $\ell_p$-norm is not a valid norm, (it does not satisfy the triangle inequality) but we choose to keep the same terminology for convenience.

not use them. For local optimization of the objective function for $\ell_p$-norm regularization we use an iterative half-quadratic method [11].

The representation of the source localization problem in the linear inverse problem form can be immediately used to solve single time sample problems. Unfortunately, there is limited information that can be extracted from a single time sample, and we are faced with the question of how to represent the multiple time sample source localization problem in our framework. In principle, we can represent the data for each of the multiple time samples in the linear inverse problem form and use sparsity enforcing regularization for each problem separately. Much better robustness to noise is achieved if we use multiple time samples in synergy. We look into several possibilities of joint use of multiple time samples. The one that appears the most promising is based on the singular value decomposition of the outputs of the sensors. We also consider additional practical problems, such as removing the effects of the grid, and automatically choosing the regularization parameter which balances the level of sparsity of the resulting spectrum versus the fidelity to the sensor measurements.

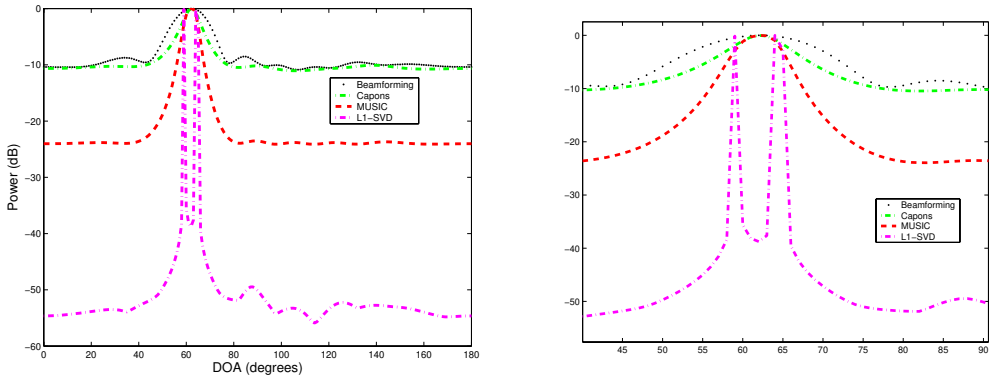



**Figure 1.1.** (a) Comparison of beamforming, Capon's, MUSIC, and $\ell_1$-SVD spectra for SNR=0 dB, and two sources coming from directions $60^\circ$ and $65^\circ$ with respect to the array axis.(b) Detail of (a).

To give a flavor of what we are able to do with our source localization technique, we present a simulation of an 8-sensor uniform linear array with 2 incoming narrowband farfield signals in Figure 1.1. The SNR is very low, 0 dB, and the sources are very close, the angular separation is $5^\circ$, so neither beamforming nor Capon's method nor MUSIC are able to separate these two sources. However, one of the methods that we propose in the thesis, $\ell_1$-SVD has a clear separation between the two peaks in the spectrum. Also the sidelobes are nearly non-existent, which happens due to the fact that we explicitly optimized a measure related to sparsity! These results may look too good to be true, so we have to mention that the $\ell_1$-SVD technique is biased for closely spaced signals when the SNR is low. Nevertheless, small bias seems to be a good compromise for having excellent resolution and robustness to noise.

An additional concern that practical source localization methods have to handle is model errors, such as sensor position uncertainty. To that end we look into using a block-coordinate descent-like procedure on an extended cost function which takes into account both the locations of the sources and the positions of the sensors. The procedure alternates between two steps: the first step is source localization with estimated model parameters, and the second step is calibration of model parameters given the estimates of source locations from the previous step.

## ■ 1.2 Outline and contributions

Before describing the contents of the thesis chapter by chapter, we briefly summarize our main contributions. The first major contribution is the development of a sparse signal reconstruction framework for source localization. In this framework we formulate various optimization problems for single and multiple snapshot sensor data for the narrowband, wideband, farfield, and nearfield scenarios. We adapt and use two paradigms for the numerical solution of the optimization problems. Finally, we carry out an extensive performance analysis of the proposed source localization methods. The second contribution of the thesis is the extension of our source localization framework to perform self-calibration in the case of modeling errors. The third contribution is the development of conditions on the sparsity of the underlying signals that guarantee the uniqueness of solutions to the noiseless $\ell_0$ sparse representation problem, and the equivalence of $\ell_0$, $\ell_1$, and $\ell_p$ problems for a general overcomplete basis.

#### Chapter 2: Introduction to Source Localization using Sensor Arrays

In this chapter we formulate the problem of source localization using an array of sensors. We describe several existing source localization methods. We end the chapter by describing some of the limitations of existing techniques thus motivating the need for our source localization framework.

#### Chapter 3: Introduction to Inverse Problems and Regularization

We start by giving a brief overview of discrete ill-posed inverse problems, and motivate the need for regularization. We summarize the well-established quadratic regularization methods and discuss why they are inappropriate for the purpose of enforcing sparsity. Next we switch to non-quadratic regularization methods, an important subset of which is sparsity-enforcing regularization. Lastly, we describe an important linear inverse problem, sparse representation of signals using overcomplete bases. This problem serves a central role in the thesis: the basis of our work is the transformation of the source localization problem into the problem of sparse signal representation.

#### Chapter 4: $\ell_1$ and $\ell_p$ Regularization

In this chapter we describe numerical optimization of the objective functions corresponding to $\ell_1$ and $\ell_p$ regularization. We start with the noiseless $\ell_1$ signal represen-

tation problem, and continue to several versions of noisy $\ell_1$ problems. The data for source localization is complex-valued, and we are led to consider second order cone (SOC) programming for the numerical optimization of objective functions associated with $\ell_1$-norm penalization of complex quantities. We briefly summarize the SOC framework, and describe how to use it to represent our objective functions. In addition to showing numerical examples of $\ell_1$-regularization, we also solve a small problem analytically using non-smooth optimality conditions. We finish the $\ell_1$ section by describing an interesting observation that we have made concerning sign patterns of exact solutions to the noiseless $\ell_1$ problem.

Next we describe $\ell_p$ regularization using an iterative half-quadratic procedure. Alternatively, it can be viewed as a quasi-Newton method with a positive definite Hessian approximation. The procedure relies on the conjugate gradients method for iterative solution of positive definite linear systems.

Our main contribution in this chapter is the adaptation of the SOC framework for sparse complex signal representation with $\ell_1$ regularization. In addition some theoretical analysis involving the analytic solution of a noisy $\ell_1$ problem, and the observation of the existence of sign patterns of exact solutions, are also original.

#### Chapter 5: Sparse-Regularization Framework for Source Localization

This chapter is the main contribution of the thesis. It describes the application of sparse regularization methodology from Chapter 4 to source localization using sensor arrays.

We start by describing how to represent the nonlinear narrowband source localization problem with one time snapshot as a linear inverse problem. This problem can be viewed as signal representation using an overcomplete basis composed of a grid of samples from the array manifold. Next we present several approaches to use multiple time samples together in an efficient manner, and take a look at how to apply our framework to wideband source localization. Also, we develop an adaptive grid refinement procedure to get rid of the grid effects. An important issue in our framework is the choice of the regularization parameter. We describe a novel method for its automatic choice based on the discrepancy principle. In the course of our research we found that some previous work has been done with a similar flavor of enforcing sparsity for signal processing and even array processing applications, [12–14]. However, most of what we present has not been considered in these papers.

#### Chapter 6: Practical Issues and Performance Analysis

This chapter is devoted to the analysis of the techniques developed in the previous chapter. First, we describe some details of the techniques and their implementation, such as the effects of the grid, comparison of $\ell_1$ and $\ell_p$, initialization, parameter selection, and the number of resolvable sources.

Next we illustrate the benefits of using the sparse regularization framework for source localization. These include superresolution, robustness to SNR, to limited number of samples and to correlated sources, as well as no need for accurate initialization.

Finally, we analyze the bias, and compare the variance of our source localization methodology to the Cramer Rao Bound, as well as to the variances of existing source localization methods, using numerical simulations.

#### Chapter 7: Theoretical Analysis: solving the $\ell_0$ problem by $\ell_p$ and related topics

This chapter is another contribution of our thesis. We address theoretical analysis of uniqueness of solutions to the noiseless $\ell_0$ regularization, and the equivalence of the noiseless $\ell_0$ problems with noiseless $\ell_1$ and $\ell_p$ problems. For the sake of generality, our analysis is separated from the array processing context, and presented in the context of signal representation using an overcomplete basis. This work was motivated by two papers [8] and [9], which consider the question of equivalence of $\ell_0$ and $\ell_1$ optimization for an overcomplete basis composed of two orthogonal bases. We extend their results to the general overcomplete basis case. In addition we prove some novel results: on the uniqueness of solutions of $\ell_0$ problems using the notion of rank-K unambiguity, on the equivalence of $\ell_0$ and $\ell_p$ problems for $p < 1$, and on sensitivity of noisy $\ell_1$ regularization.

#### Chapter 8: Model Errors and Self-Calibration

We motivate self-calibration of sensor arrays, briefly touch upon the observability conditions, and describe two existing methods based on block-coordinate descent. Next we use the same block-coordinate idea to extend our source localization framework to do self-calibration. This extension is also a contribution of the thesis.

#### Chapter 9: Conclusion

This chapter summarizes the main ideas of the thesis and gives suggestions for further research in the area.

Chapter 2

# Introduction to Source Localization using Sensor Arrays

The universal goal of array processing is to gather information from propagating waves. This nontrivial task is approached by sampling the spatiotemporal wavefield using an array of sensors. Some pieces of information that are commonly being sought about the wavefield include: the number and location of the sources of energy (or spatial energy spectrum), the signals generated by these sources, and the time evolution of all of the above. Using an array instead of a single sensor furnishes numerous benefits, comprising an improvement in signal to noise ratio, possibility of electronic steering and jamming suppression (instead of mechanical), and easier reconfiguration, among others. More importantly, source localization with omni-directional sensors is possible only when multiple sensors are available. Sensor array processing lends itself to many applications such as sonar, radar, exploration seismology and radio astronomy. Source localization is a branch of array processing which deals with determining the number and location of multiple sources using an array. In this chapter we formulate the source localization problem mathematically, provide an overview of most notable source localization methods, and describe some of their limitations as a motivation for our work.

## ■ 2.1 Observation model

Before we describe the conventional methods of source localization, it is necessary to present the mathematical model for the problem. For a more thorough covering of the material in this section the reader is referred to [15, 16].

#### Narrowband signal in the farfield of the array

We start with the most basic case, the localization of narrowband sources in the farfield of a uniform linear array. Let the uniform linear array in consideration consist of $M$ omni-directional sensors with equal spacing $d$, residing on the $x$-coordinate axis.

Taking the phase center of the array at the origin, the position of the $m$-th sensor is $p_m = (m - (M+1)/2)d$, $m \in \{1, .., M\}$. For the sources in the farfield of the array,

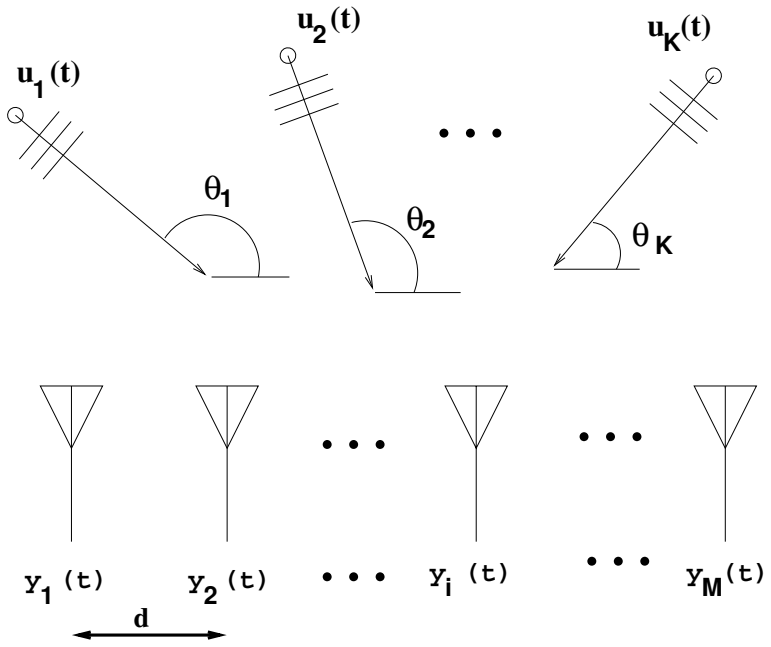



**Figure 2.1.** An illustration of the geometry of source localization: sources $\mathbf{u}_k(t)$, impinging on the array at angles $\theta_k$ producing sensor outputs $\mathbf{y}_m(t)$.

the curvature of the wavefront is insignificant across the aperture of the array, and the plane wavefront approximation works very well. The solution of the wave equation with a single source generating signal $f(t)$ has the form $f(t-\mathbf{p}^T\boldsymbol{\alpha})$, where $\mathbf{p}$ is the position, and $\boldsymbol{\alpha}$ is the so called slowness vector aligned with the direction of propagation of the wave, and whose magnitude is equal to $1/c$, the inverse of the propagation speed. The distance attenuation factor is not considered in the farfield model since it will be almost constant across the array if the sources indeed come from the farfield.

The signal in the narrowband case can be expressed as $u(t)exp(j\omega_0 t)$, where $u(t)$ is the baseband signal. It is modulated to frequency $\omega_0$, which has to be much greater than the bandwidth of $u(t)$ for the narrowband assumption to hold. In order to avoid spatial aliasing, sensor spacing has to be smaller than the half of the wavelength, $d \leq \lambda/2 = 2\pi c/(2\omega_0)$. Unless otherwise stated, we always take $d = \lambda/2$ for the narrowband case. The output of sensor $m$ is $y_m(t) = u(t-\tau_{center})exp(j(\omega_0(t-\tau_{center})-\mathbf{k}^T\mathbf{p}))$, where $\tau_{center}$ is the delay from the source to the phase-center of the array, and the wavenumber is given by $\mathbf{k} = \omega_0\boldsymbol{\alpha}$. Narrowband assumption allows us to ignore the delay between the sensors, $\mathbf{k}^T\mathbf{p}$, in the baseband signal $u(t)$; it is only present in the modulation. The complex envelope of the output of sensor $m$ (i.e. the output after demodulation) can be written as $y_m(t) = u(t-\tau_{center})exp(-j(\omega_0\tau_{center}+\mathbf{k}^T\mathbf{p}))$. By measuring time relative to the phase center, the dependence on $\tau_{center}$ can be dropped. Thus, for a single source the complex envelope of the sensor outputs has the following form: $\mathbf{y}(t) = \mathbf{a}(\theta)u(t)$.

The manifold vector, $\mathbf{a}(\theta) = exp(-j\mathbf{k}^T\mathbf{p})$ contains the phase delay information for the source coming from bearing $\theta$ with respect to the array axis. The parameterization of the manifold vector by $\theta$ can be done since $\mathbf{k}^T\mathbf{p}_m = -(\omega_0/c)(m-(M+1)/2)dcos(\theta)$.

Due to the linearity of the system the superposition principle holds, and the model for $K$ narrowband signals with the same center frequency can be written as $\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t)$. The $M \times K$ matrix $\mathbf{A}(\boldsymbol{\theta})$ is the manifold matrix containing the manifold vectors for different sources as its columns, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), ..., \mathbf{a}(\theta_K)]$. Sensor signal vector $\mathbf{y}(t)$ is a column vector whose $m$-th element is $y_m(t)$, and similarly $\mathbf{u}(t)$ is a column vector containing the signals $u_k(t)$ coming from all $K$ sources. Vector $\boldsymbol{\theta}$ contains source locations for all the $K$ sources: $\boldsymbol{\theta} = [\theta_1, ..., \theta_K]^T$. Taking into account the inevitable presence of noise, and discretizing the waveforms, the final version of the model takes the following form:

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t) + \mathbf{n}(t), \quad t \in \{1, .., T\} \tag{2.1}$$

For simplicity, the noise is assumed to be spatially and temporally stationary and white, uncorrelated with the sources, and circularly symmetric. The covariance matrix takes the following form: $E[\mathbf{n}(t_1)\mathbf{n}^H(t_2)] = \sigma^2\ \mathbf{I}\ \delta(t_1 - t_2)$, where $\delta()$ is the Kronecker delta function, and $\mathbf{I}$ is an identity matrix. The circular symmetry of the noise leads to $E[\mathbf{n}(t_1)\mathbf{n}^T(t_2)] = 0$.

#### Nearfield of the array

The generalization of the model to the case where the sources lie in the nearfield of the model has a number of applications, for example audio speaker separation using a microphone array in enclosed spaces. The plane-wave approximation no longer holds, and the solution to the spherical wave equation at distance $r$ from a single source $f(t)$ is as follows: $f(r,t) = (1/r)f(t - r/c)$. Again, considering the narrowband signal $f(t) = u(t)exp(j\omega_0 t)$, the complex envelope of the array output becomes $y_m(t) = (1/r_m)u(t - r_m/c)exp(-j(\omega_0 r_m/c))$. Here, $r_m$ is the distance from the source to the $m$-th sensor. Let $r_c$ be the distance from the source to the phase center of the array, and $\mathbf{p}_m$ the position of $m$-th sensor. Taking into account the narrowband assumption, and shifting the time origin to correspond to the signal arriving at the phase center, the output can be rewritten as $y_m(t) = (1/r_m)u(t)exp(-j(\omega_0(r_m - r_c)/c)) = a(\mathbf{p}_m)u(t)$. The $m$-th component of the manifold vector, $a(\mathbf{p}_m)$ contains the phase and attenuation factors for the source arriving at sensor $m$. Using superposition, the model for $K$ sources takes the exact same form as (2.1), except the columns of $\mathbf{A}(\boldsymbol{\theta})$ contain the nearfield manifold vectors instead of the farfield ones. Unlike the farfield case, the response at the array depends not only on the bearings of the sources but also on their range, thus source localization furnishes both of these parameters.

#### Wideband signals

Finally, in the wideband case, the signal can no longer be well-approximated by a baseband signal modulated by a carrier. However, using the Fourier transform, a narrowband model can be written for each frequency:

$$\mathbf{y}(\omega) = \mathbf{A}(\boldsymbol{\theta}, \omega)\mathbf{u}(\omega) + \mathbf{n}(\omega), \quad \omega \in \{\omega_1, .., \omega_W\} \tag{2.2}$$

where $\mathbf{y}(\omega)$ and $\mathbf{u}(\omega)$ are Fourier transforms of $\mathbf{y}(t)$ and $\mathbf{u}(t)$ respectively. Note that in the narrowband case there is only one manifold matrix, whereas in the wideband case each frequency component $\omega$ yields a new manifold matrix, $\mathbf{A}(\boldsymbol{\theta}, \omega)$. This happens since phase shift for a given delay depends on the frequency of the signal. To have multiple observations for each frequency, temporal data is usually divided into several blocks, and the Fourier transforms of each block are calculated. Or more generally, short-time Fourier transform, which allows the blocks to overlap, can be used.

#### Second-order statistics

Most modern source localization methods rely on statistical characterization of the sensor outputs. The majority of them considers second-order statistics. The spatial covariance of the sensor outputs is $\mathbf{R} = E[\mathbf{y}(t)\mathbf{y}^H(t)] = \mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}(\boldsymbol{\theta})^H + \sigma^2\mathbf{I}$, where the signal covariance matrix is $\mathbf{P} = E[\mathbf{u}(t)\mathbf{u}^H(t)]$, and as discussed previously noise has a diagonal covariance: $E[\mathbf{n}(t)\mathbf{n}^H(t)] = \sigma^2\mathbf{I}$. Many methods require that $\mathbf{P}$ is nonsingular, however situations in which this is not the case, e.g. due to the presence of multipath or coherent jamming, occur as well. Since the exact expectation is unknown, the standard sample covariance approximation is used: $\hat{\mathbf{R}} = \frac{1}{M}\sum_{t=1}^{M}\mathbf{y}(t)\mathbf{y}^H(t)$. In the rest of this manuscript, we use $\mathbf{R}$ for both the actual and sample expectations, but the meaning of the symbol should be clear from context.

## ■ 2.2 Methods for source localization

### ■ 2.2.1 Classical beamforming

The classical approach to source localization relies on scanning the power from different locations by steering the array. We discuss the farfield case[1]. The array is steered by compensating the delays for the different sensor outputs by appropriately shifting the waveforms. When the weights on all the sensors are unity and no delays are introduced, the array is effectively steered at broadside (perpendicular to the array axis). For waves traveling in that direction, the delays for all the sensors are equal, and the delays with respect to the phase center are zero, requiring no compensation. Thus unity weighting produces constructive interference of the sensor outputs, and achieves the maximum power at broadside among all directions.

Similarly, if the array is steered at an angle $\theta$, the waveforms on the $m$-th sensor are advanced or delayed by $-\tau_m(\theta)$, the negative of the delay relative to the phase

[1] The nearfield case is analogous with the addition of a range parameter instead of just using bearing.

center. The maximum power is achieved by steering at the direction from which the waves are arriving, assuming no aliasing is present. For the narrowband case the delays amount to phase shifts which can be implemented by complex weights $\mathbf{w}$ on the sensors. The array output thus becomes: $z(t) = \mathbf{w}^H \mathbf{y}(t) = \mathbf{w}^H \mathbf{a}(\theta) u(t)$. To steer the array to angle $\theta$, the weights have to be set as $\mathbf{w} = \mathbf{a}(\theta)$. Due to the linearity of the system, the same approach is used to look for a superposition of plane-waves traveling from different directions, with identical carrier-frequencies. The beamforming spectrum can be represented as

$$P_{bf}(\theta) = \sum_{t=1}^{T} \|\mathbf{w}^H(\theta)\mathbf{y}(t)\|_2^2 \tag{2.3}$$

Beamforming is a very simple and robust approach, which is widely used in practice. However, beamforming suffers from the Rayleigh resolution limit [15], which can be mitigated only by increasing the width of the array (the number of sensors): improving SNR or increasing observation time does not change resolution. The method parallels FIR time-series analysis. For example, to decrease the sidelobes levels, windowing can be used; however, no simple extensions are able to improve resolution. In the wideband case, the processing is usually done in frequency domain using short-time Fourier transforms. To work with wideband signals in the time domain, actual delays have to be implemented instead of phase shifts.

### ■ 2.2.2 Optimal beamforming: Capon's method (MVDR)

The classical beamforming method has weights which are independent of the signals and noise. The idea of optimal beamforming is to use the estimated signal and noise parameters to improve the performance. One widely used method is Capon's method, also called Minimum Variance Distortionless Response (MVDR), and Applebaum's array [17]. It attempts to minimize the variance due to noise, while keeping the gain in the direction of steering equal to unity: $\mathbf{w}_{CAP}(\theta) = \arg\min_{\mathbf{w}}(E[\mathbf{w}^H \mathbf{y}\mathbf{y}^H \mathbf{w}])$, subject to $Re[\mathbf{w}^H \mathbf{a}(\theta)] = 1$. The term variance is misleading: if the signals are random and zero-mean, then this is indeed the case, however, when the signals are non-random, $\mathbf{w}^H \mathbf{R} \mathbf{w}$ does not correspond to variance. Also, no attempt is made to separate the signal from the noise, so the aggregate energy is being minimized. The solution of this optimization problem can be shown to have the following form:

$$\mathbf{w}_{CAP}(\theta) = \frac{\mathbf{R}^{-1}\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)} \tag{2.4}$$

The source location estimate is obtained in the same way as for classical beamforming - simply by steering the array at a range of $\theta$'s, and looking for maximum power. The resulting spectrum has an analytic expression:

$$P_{CAP}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{R}^{-1}\mathbf{a}(\theta)} \tag{2.5}$$

The main benefit of this method is a substantial increase in resolution compared with standard beamforming. In fact, as opposed to beamforming, the number of sensors does not impose a limit on resolution. With a non-degenerate array geometry (which avoids spatial aliasing), resolution increases without limit as SNR or the observation time are increased. An additional benefit is the lower amount of ripple in the sidelobes. However, the sidelobe level cannot go below $\sigma^2/M$, the same value as for standard beamforming with unity weights. Some of the other shortcomings include an increase in the amount of computation compared to beamforming, poor performance with small amounts of time-samples (due to the difficulty of estimation of the sensor-data covariance matrix) and inability to handle strongly correlated or coherent sources. Nevertheless, the combination of increased resolution, only moderate increase in computational complexity, and the robustness due to model errors which occur in practice (unlike some of the other conventional super-resolution methods) make this method one of the most widely used in practical applications. A more elaborate discussion of the method with motivations for all of the above assertions can be found in [15].

### ■ 2.2.3 Subspace methods: MUSIC

The MUSIC method [7] is the most prominent member of the family of eigen-expansion based source location estimators. The underlying idea is to separate the eigenspace of the covariance matrix of sensor outputs into the signal and noise components using the knowledge about the covariance of the noise. The sensor output correlation matrix admits the following decomposition:

$$\mathbf{R} = \mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}(\boldsymbol{\theta})^H + \sigma^2\mathbf{I} = \mathbf{U}\Lambda\mathbf{U}^H = \tag{2.6}$$

$$\mathbf{U}_s\Lambda_s\mathbf{U}_s^H + \mathbf{U}_n\Lambda_n\mathbf{U}_n^H = \mathbf{U}_s\Lambda_s\mathbf{U}_s^H + \sigma^2\mathbf{U}_n\mathbf{U}_n^H \tag{2.7}$$

Here, $\mathbf{U}$ and $\Lambda$ form the eigenvalue decomposition of $\mathbf{R}$, and $\mathbf{U}_s$, $\mathbf{U}_n$, $\Lambda_s$, and $\Lambda_n = \sigma^2\mathbf{I}_{M-K}$ are the partitions of the eigenspectrum into signal plus noise and signal subspaces. Provided that $\mathbf{P}$ is nonsingular, $\mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}(\boldsymbol{\theta})^H$ has rank $K$. The number of sources, $K$, has to be strictly less than the number of sensors, $M$, for the method to work. Hence, $\mathbf{R}$ has $K$ eigenvalues which are due to the combined signal plus noise subspace, and $M-K$ eigenvalues due to the noise subspace alone. Assuming that the noise has a flat spectrum of $\sigma^2$, $K$ eigenvalues corresponding to the signal and noise subspace are larger than the remaining $M-K$ noise eigenvalues, which are equal to $\sigma^2$. This information can be used to separate the two eigensubspaces. Due to the orthogonality of eigensubspaces corresponding to different eigenvalues for Hermitian matrices, the noise subspace is orthogonal to the steering vectors corresponding to the direction of propagation, thus $\mathbf{U}_n^H\mathbf{a}(\theta) = 0$ for all directions from which the signals are impinging. MUSIC spectrum is obtained by putting the squared norm of this term into the denominator, which leads to very sharp estimates of the positions of the sources, (in the noiseless case the peaks of the spectrum approach infinity):

$$P_{MUS}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta)} \tag{2.8}$$

In contrast with the previously discussed techniques, MUSIC spectrum has no direct relation to power; it simply exhibits sharp peaks at the estimated source locations. Also, it cannot be used as a beamformer, since the spectrum is not obtained by steering the array. Unlike the the methods previously discussed, MUSIC provides a consistent (in the sense of estimation theory) estimate of the locations of the sources, as SNR and the number of sensors go to infinity. Despite the dramatic improvement in resolution, MUSIC suffers from a high sensitivity to model errors, such as sensor position uncertainty. Also, the resolution capabilities decrease when the signals are correlated. When some of the signals are coherent (perfectly correlated), the method fails to work. The computational complexity is dominated by the computation of the eigenexpansion of the covariance matrix.

There are multiple extensions of MUSIC by using a weight matrix in the denominator, one of which is the Min-Norm algorithm [16]. Root-MUSIC [18] is a variant of MUSIC which instead of computing a spectrum, forms a polynomial using the noise subspace, and the source location estimates are the roots of the polynomial. Root-MUSIC relies on the structure of the steering matrix for a uniform linear array (ULA), and cannot be extended to general arrays. The performance for ULAs is very similar to that of MUSIC, except for a somewhat higher robustness at limited numbers of time samples.

### ■ 2.2.4 Maximum Likelihood techniques

Maximum Likelihood (ML) methods [16,19] belong to the class of parametric methods. In contrast to the methods described above, the spectrum is not computed, but instead parameters of the model are estimated. A variety of methods resides under the ML header. One notable classification is in the assumed form of the signal. When the signals are modeled as deterministic, the method is called Deterministic ML (DML), when the signals are modeled as Gaussian, the method is called Stochastic ML (SML). Noise is usually modeled as stationary Gaussian. For deterministic maximum likelihood, the objective is to find $\boldsymbol{\theta}$, $\mathbf{u}(t)$, and $\sigma^2$, to maximize the likelihood function:

$$L_{DML}(\boldsymbol{\theta}, \mathbf{u}(t), \sigma^2) = \prod_{t=1}^{T}(\pi\sigma^2)^{-M} exp(-\|\mathbf{y}(t) - \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t)\|_2^2/\sigma^2), \tag{2.9}$$

where $\boldsymbol{\theta}$ is the vector of source locations. The log-likelihood is:

$$l_{DML}(\boldsymbol{\theta}, \mathbf{u}(t), \sigma^2) = -2M\log\sigma + \frac{1}{\sigma^2 T}\sum_{t=1}^{T}(-\|\mathbf{y}(t) - \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t)\|_2^2), \tag{2.10}$$

Fortunately, it is not necessary to optimize over all the parameters, $\boldsymbol{\theta}$, $\mathbf{u}(t)$, and $\sigma$ simultaneously, since once $\boldsymbol{\theta}$ is known, we can use $\mathbf{A}(\boldsymbol{\theta})$ to get explicit values for the other parameters:

$$\hat{\sigma}^2 = \frac{1}{M}\text{trace}\{\Pi^{\perp}_{\mathbf{A}(\boldsymbol{\theta})}\mathbf{R}\} \quad \text{and} \quad \hat{\mathbf{u}}(t) = \mathbf{A}(\boldsymbol{\theta})^{\dagger}\mathbf{y}(t) \tag{2.11}$$

where $\mathbf{A}(\boldsymbol{\theta})^{\dagger}$ is the pseudo-inverse of $\mathbf{A}(\boldsymbol{\theta})$, and $\Pi^{\perp}_{\mathbf{A}(\boldsymbol{\theta})}$ is the projection matrix onto the orthogonal complement of the range space of $\mathbf{A}(\boldsymbol{\theta})$. [2]

The remaining unknown, the locations of the sources, can be found by minimizing the following cost function:

$$\hat{\boldsymbol{\theta}}_{DML} = \arg\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \|\Pi^{\perp}_{\mathbf{A}(\boldsymbol{\theta})}\mathbf{y}(t)\|_2^2 = \arg\min_{\boldsymbol{\theta}} \; trace\{\Pi^{\perp}_{\mathbf{A}(\boldsymbol{\theta})}\mathbf{R}\} \tag{2.12}$$

This cost function measures the sum of squares of projections of $\mathbf{y}(t)$ onto the orthogonal complement of the array manifold matrix, i.e. lack of fit of the range space of the manifold matrix to the data $\mathbf{y}(t)$. The optimization involves a $K$-dimensional search, where $K$ is the number of impinging signals. $K$ can be estimated using a variety of methods, such as Akaike information criterion (AIC) or minimum description length (MDL) [16, 20]. The computational complexity is considerably higher than for any of the methods described before. The benefits of ML family of methods is the ability to resolve coherent signals, ability to handle single snapshot scenarios, and better statistical properties [21]. A major problem with the ML-family of methods is the need for a very accurate starting point for the optimization procedure; otherwise the solution may converge to a local extremum.

### ■ 2.2.5 Limitations of current methods

Despite the existence of a multitude of various source localization methods we took the time to develop a new one. Part of the reason for such an undertaking is the desire to improve upon the performance of the existing methods; to that end we summarize some of their limitations.

Beamforming is a very robust and simple source localization technique, but it has limited resolution. In Figure 2.2 we present two plots with beamforming spectra. We simulate a uniform linear array (ULA) with 8 sensors spaced at half-wavelength which is exposed to two farfield narrowband sources. In plot (a) the separation between the two sources is 20°, and beamforming is able to resolve the two sources. However once we move the sources closer together to 13°, within the Rayleigh resolution limit, the two peaks are merged, and the locations of the two sources cannot be determined.

MUSIC and Capon's methods go a long way to improve the resolution capabilities of beamforming. However, when the sources are close, and the SNR is low, they also

[2] $\mathbf{A}(\boldsymbol{\theta})^{\dagger} = (\mathbf{A}(\boldsymbol{\theta})^H\mathbf{A}(\boldsymbol{\theta}))^{-1}\mathbf{A}(\boldsymbol{\theta})^H$, and $\Pi^{\perp}_{\mathbf{A}(\boldsymbol{\theta})} = \mathbf{I} - \mathbf{A}(\boldsymbol{\theta})\mathbf{A}(\boldsymbol{\theta})^{\dagger}$, where $\mathbf{I}$ is an identity matrix.

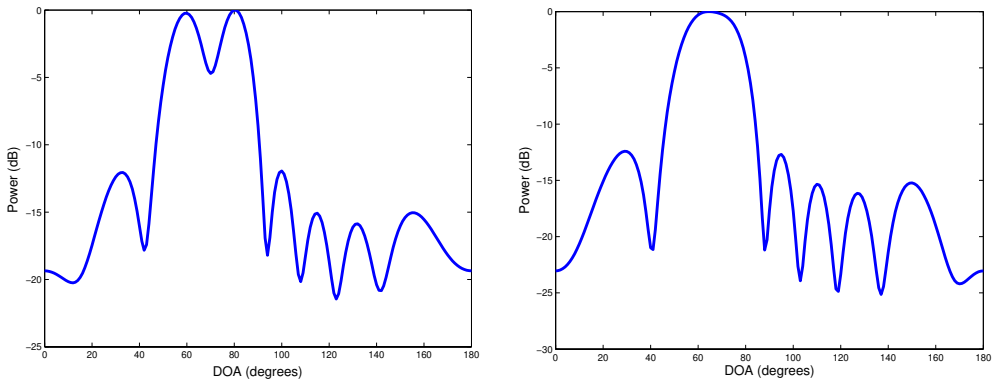



**Figure 2.2.** Resolution limitations of beamforming. (a) Separation between the sources is 20°, peaks are resolved. (b) Separation between the sources is 13°, peaks are merged.

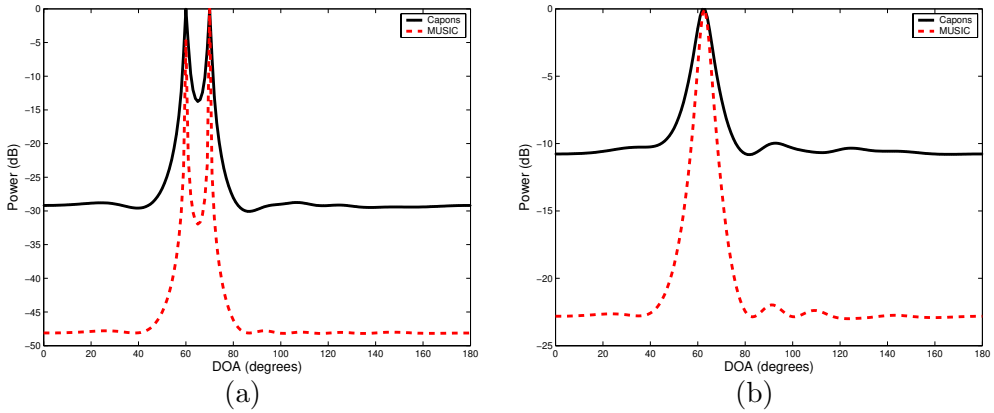



**Figure 2.3.** Limitations of MUSIC and Capon's methods (a) SNR=20 dB, separation between the sources is 10°, peaks are resolved by both MUSIC and Capon's methods. (b) SNR=0 dB, separation between the sources is 5°, peaks are merged for both.

lose resolution and eventually are unable to separate the sources[3]. Figure 2.3 illustrates what happens when we lower the SNR and bring the sources close together. In plot (a) SNR is 20 dB, and separation between the sources is 10°, so both MUSIC and Capon's methods are able to resolve the two sources well. However, plot (b) shows that when SNR is decreased to 0 dB, and source separation is decreased to 5°, neither of the two methods can resolve the two sources. Some additional limitations of these two methods include inferior performance for correlated and coherent sources, and for scenarios with limited number of time samples. We present an in-depth comparison of these methods with our proposed source localization method in Chapter 6.

[3]In fact every source localization technique has a lower limit on the SNR that it can withstand, but the method that we propose in the rest of the thesis has better robustness to low SNR than MUSIC and Capon's methods.

Maximum Likelihood source localization techniques are parametric, so the result is not a spectrum but a set of point estimates of source locations. In general they are more robust than beamforming, MUSIC and Capon's methods, but are computationally more demanding. Apart from computational complexity, the major drawback of ML source localization is the need for accurate initialization to insure convergence to global minima (instead of local ones). The method that we propose in this thesis does not suffer from the need for accurate initialization[4]. A longer discussion of this issue appears in Chapter 6.

[4]But, it does not decrease the computational cost of ML.

Chapter 3

# Introduction to Inverse Problems and Regularization

We start this chapter by describing linear ill-posed inverse problems. Later in the thesis (in Chapter 5) we transform the source localization problem into this form. The solution of ill-posed inverse problems relies on regularization. Quadratic regularization is mentioned first. As we discuss, it is not well-suited for our goals, and we switch next to non-quadratic regularization and in particular sparsifying regularization. Sparsifying regularization is discussed in the context of signal representation using overcomplete bases, a special case of a linear ill-posed inverse problem. Our transformed source localization problem can be viewed as signal representation using overcomplete bases.

## ■ 3.1 Ill-posed inverse problems and regularization

In inverse problems [22–24] the function from the unknown quantity that we wish to find to the observations is known. The goal is to find a meaningful inverse function. Mathematically, we have $\mathbf{y} = \mathbf{T}(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$ is the unknown and $\mathbf{y} \in \mathcal{Y}$ is the vector of observations.[1] Usually, $\mathbf{T}()$ is a well-behaved continuous operator, and the solution of the forward problem (find $\mathbf{y}$ given $\mathbf{x}$) meets no significant obstacles. The inverse mapping from $\mathbf{y}$ to $\mathbf{x}$ in the problems of interest is much less friendly. The difficulties may include lack of solution, non-unique solutions, or a discontinuous dependence of the solution on the observations. The presence of any of these issues makes the problem ill-posed.

As we stated it, the problem is too general, and we make additional assumptions that $\mathcal{X}$ and $\mathcal{Y}$ are finite-dimensional, and $\mathbf{T}$ is a linear operator:

$$\mathbf{y} = \mathbf{T}\mathbf{x}, \quad \mathbf{y} \in \mathbb{C}^M, \quad \mathbf{x} \in \mathbb{C}^N, \quad \mathbf{T} \in \mathbb{C}^{M \times N} \tag{3.1}$$

Lack of solutions means that $\mathbf{y}$ does not lie in the range of $\mathbf{T}$ ($\mathbf{T}$ is not surjective), and lack of uniqueness means that the nullspace of $\mathbf{T}$ is not trivial ($\mathbf{T}$ is not injective). The standard approach to treat these two obstacles is by taking the Moore-Penrose

[1] $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces, i.e. complete metric spaces with an inner product defined.

pseudo-inverse, $\mathbf{T}^\dagger$. Consider the singular value decomposition (SVD):

$$\mathbf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}' = \sum_{i=1}^{\min(M,N)} \mathbf{u}_i \sigma_i \mathbf{v}_i' \tag{3.2}$$

Let $K = \text{rank}(\mathbf{T})$. Then the pseudo-inverse is defined as

$$\mathbf{T}^\dagger = \sum_{i=1}^{K} \mathbf{v}_i \sigma_i^{-1} \mathbf{u}_i' \tag{3.3}$$

By applying the pseudo-inverse we find the minimum-norm least squares solution. If $\mathbf{y} = \mathbf{T}\mathbf{x}$, then the reconstruction is

$$\hat{\mathbf{x}} = \mathbf{T}^\dagger \mathbf{y} = \left( \sum_{j=1}^{K} \mathbf{v}_j \sigma_j^{-1} \mathbf{u}_j' \right) \mathbf{y} = \sum_{j=1}^{K} \mathbf{v}_j \sigma_j^{-1} \mathbf{u}_j' \sum_{i=1}^{\min(M,N)} \mathbf{u}_i \sigma_i \mathbf{v}_i' \mathbf{x} =$$
$$\sum_{j=1}^{K} \sum_{i=1}^{\min(M,N)} \frac{\sigma_i}{\sigma_j} \mathbf{v}_j \mathbf{u}_j' \mathbf{u}_i \mathbf{v}_i' \mathbf{x} = \sum_{i=1}^{K} \mathbf{v}_i \mathbf{v}_i' \mathbf{x} = (\mathbf{I}_N - \sum_{i=K+1}^{N} \mathbf{v}_i \mathbf{v}_i') \mathbf{x}$$

Here $\mathbf{I}_N$ is an $N \times N$ identity matrix. Whenever $K < N$, the reconstruction $\hat{\mathbf{x}}$ is only an approximation to $\mathbf{x}$. The component of $\mathbf{x}$ that lies in the nullspace of $\mathbf{T}$ is set to zero ($\mathbf{T}^\dagger$ chooses the min-norm solution).

Since $\mathbf{T}^\dagger$ is a linear function in a finite dimensional space, then it is necessarily continuous. However, in some applications the condition number of $\mathbf{T}^\dagger$ may be very large, making the pseudo-inverse discontinuous for all practical purposes. Now let us consider what happens when we add noise: $\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{n}$. Even the addition of small amount of noise to the observations may render the solution completely useless: $\mathbf{T}^\dagger \mathbf{y} = \mathbf{T}^\dagger(\mathbf{T}\mathbf{x} + \mathbf{n}) = \hat{\mathbf{x}} + \sum_{i=1}^{K} \mathbf{v}_i \sigma_i^{-1} \mathbf{u}_i' \mathbf{n}$. The power distribution of projections of white noise on all the left singular vectors is uniform, ($E[(\mathbf{u}_i'\mathbf{n})^2]$ is not a function of i). By applying the pseudo-inverse we are multiplying the noise components by inverses of $\sigma_i$, the last few of which are very large since $\mathbf{T}$ is ill-conditioned. The amplification of these noise components dominates the solution, and the signal component of interest becomes hidden under the noise floor. Much of the effort in the field of discrete ill-posed problems is directed at making good approximations to $\mathbf{T}^\dagger$ which are much less sensitive to noise.

Regularization is used to solve ill-posed problems by incorporating *apriori* knowledge about $\mathbf{x}$ to stabilize the problem and to provide reasonable and useful solutions. For example, if it is known that the solution should be a discretization of a continuous function, this knowledge allows us to discard the wildest looking candidates, and to considerably reduce the set of possible solutions. The task is to minimize some measure $J_1(\mathbf{x})$ of proximity of $\mathbf{y}$ to the range space of $\mathbf{T}$, as well as to satisfy as much as possible

the *apriori* information about $\mathbf{x}$, by minimizing some appropriate measure $J_2(\mathbf{x})$. The two objectives typically cannot be both minimized at the same time, so we need a compromise, which can be simply obtained by taking a linear combination of the two:

$$J(\mathbf{x}) = J_1(\mathbf{x}) + \lambda J_2(\mathbf{x}) \tag{3.4}$$

Scalar $\lambda$ is the regularization parameter balancing the tradeoff between the fidelity to the data, $J_1(\mathbf{x})$, and the fidelity to the prior information, $J_2(\mathbf{x})$. There is a whole family of solutions indexed by $\lambda$, with the non-regularized (least squares) solution if $\lambda = 0$, and a solution strongly favoring the *apriori* information when $\lambda$ is large. In general, choosing an appropriate $\lambda$ is problem-dependent, and is a nontrivial task.

With an appropriate choice for $J(\mathbf{x})$, regularization effectively deals with all the three aspects of ill-posedness. Solution exists for any $\mathbf{y} \in \mathcal{Y}$, since we are allowing $\mathbf{y}$ outside the range of $\mathbf{T}$ with the use of $J_1(\mathbf{x})$. Also, proper choice of $J_2(\mathbf{x})$ deals with lack of uniqueness and can dramatically reduce sensitivity to noise (improve the Lipschitz constant of the inverse function), making it "continuous enough" for practical applications.

### ■ 3.1.1 Quadratic regularization methods

One of the most well-known approaches to regularization is due to Tikhonov [25]. Tikhonov's method assumes that the norm of the solution should be small, which limits the amount of amplification due to small eigenvalues. The cost function takes the form

$$J(\mathbf{x}) = \|\mathbf{T}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \tag{3.5}$$

The $\ell_2$-norm of the residual is the data-fidelity term, $J_1(\mathbf{x})$, and the term $\|\mathbf{x}\|_2^2$ serves as $J_2(\mathbf{x})$ in (3.4). The Tikhonov cost function has a closed-form solution:

$$\hat{\mathbf{x}} = \sum_{i=1}^{K} \left( \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \right) \frac{\mathbf{u}_i' \mathbf{y}}{\sigma_i} \mathbf{v}_i \tag{3.6}$$

Other quadratic regularization methods, such as the truncated or damped SVD follow a similar pattern [26]: $\hat{\mathbf{x}} = \sum_{i=1}^{K} w_i \frac{\mathbf{u}_i' \mathbf{y}}{\sigma_i} \mathbf{v}_i$. They can be regarded as weighted pseudo-inverses with weights $w_i$. Tikhonov regularization is a special case with $w_i = \left( \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \right)$. The idea of all of these methods is to leave the large singular values almost unchanged, and to limit the effects of the inverses of small singular values.

To incorporate other forms of prior information, the following generalization of the cost function can be used: $J(\mathbf{x}) = \|\mathbf{T}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|L(\mathbf{x} - \mathbf{x}^*)\|_2^2$, where L is a linear operator most suitable for the prior of interest, and $\mathbf{x}^*$ is an *apriori* estimate. The quadratic family of regularization methods is very suitable for many practical applications, and has the benefit of a closed-form solution, and tractable methods for choosing the regularization parameters. However, since the inverse operator is always a linear function of the data, there are limitations on what can be achieved. In particular, due to the linearity it is impossible to recover the part of $\mathbf{x}$ which belongs to the nullspace of $\mathbf{T}$.

### ■ 3.1.2 Non-quadratic regularization methods

Using a quadratic form for both the data-fidelity and the prior term leads to a linear dependence of the reconstruction on the data. The most important benefit of linearity is the computational tractability of the problem. However, linearity also suffers from a salient drawback of irrecoverability of sharp features. This occurs due to the fact that forward operators $\mathbf{T}$ in most inverse problems of interest have a low-pass frequency response, and a smoothing effect. High-frequency components belong to the nullspace of $\mathbf{T}$, and cannot be recovered by a linear inverse mapping. It has been shown both theoretically and practically that allowing for the more general non-linear form of regularization can lead to a dramatic improvement in this respect, preserving sharp edges and other strong features and leading to super-resolution in the reconstruction [26]. However, computational complexity increases considerably due to this generalization. That is particularly true for the case of non-convex functions.

Two popular non-quadratic cost functions are total variation and entropy [26]. Total variation puts a penalty on the sum of variations of the signal $J_2 = \|D\mathbf{x}\|_1 = \sum |[D\mathbf{x}]_i|$, where $D$ is a discrete approximation to the gradient operator. Total variation is most frequently used in image processing applications, such as image restoration. In comparison to the Tikhonov regularization with $L = D$, the penalty on strong features is less severe, and the reconstruction can contain sharp edges. It works very well in practice with images that can be described as piecewise-smooth.

Maximum entropy regularization uses an entropy-like prior term: $J_2(\mathbf{x}) = \sum |x_i| log(|x_i|)$. Some variations are possible using cross-entropy, and divergence. Cost functions with this form lead to greater energy concentration in the reconstruction (most coefficients are very small, and a few are large). Another term for energy concentration is *sparsity*. This is most suitable for data which exhibit the same behavior, for example in spectrum estimation for signals with several harmonics, or point source localization.

Another regularizing function which has the same sparsifying effect is the $\ell_1$-penalty: $J_2 = \|\mathbf{x}\|_1$. This is similar to the total-variation except we take the $\ell_1$-norm of the values of $\mathbf{x}$ instead of their derivatives. Total variation allows sparse jumps of the gradient of $\mathbf{x}$, whereas the $\ell_1$ penalty favors sparse values of $\mathbf{x}$. To lower the penalty on strong features even further, several non-convex functions have also found use. The $\ell_p$-quasi-norm with $p < 1$, $J_2 = \|\mathbf{x}\|_p^p = \sum_i |x_i|^p$, produces even stronger energy concentration in the reconstruction. Strong energy-concentration property makes $\ell_1$ and $\ell_p$-regularization very suitable for the reconstruction of sparse signals, for example for those arising in array processing. We rely on $\ell_1$ and $\ell_p$ penalization for the rest of the thesis. We discuss the benefits of sparsity in much greater detail in the next section, since it is the heart of our source localization methodology.

## ■ 3.2 Sparsity regularization

The selection of a proper regularizer intimately depends on the property of $\mathbf{x}$ that one wishes to enforce, and that depends on the particular application. In many mathe-

matical inverse problems, priors of choice are different forms of smoothness or energy constraints, and the corresponding regularizers are the $\ell_2$ norms of $\mathbf{x}$ or its derivatives.

Sparsity prior is useful when signals $\mathbf{x}$ that we look for have to be sparse. We define sparsity of a vector $\mathbf{x}$ by the presence of a small number of large elements and zeros elsewhere. An appropriate numerical measure of sparsity is the count of non-zero elements. An important linear inverse problem which has a good use for sparsity priors is the problem of signal representation using overcomplete bases. Our source localization framework, the subject of this manuscript, is built on such overcomplete representation ideas. The base for this discussion is the work of Mallat [27], Donoho [28], Rao [14] and others on function approximation and optimal basis selection.

The problem of choosing an appropriate basis for a family of signals has received a great deal of attention over the past decade, and many new bases were introduced, such as wavelet bases, ridgelets, and curvelets, among many others [29]. Despite the fact that any minimal spanning basis for a finite-dimensional space can represent perfectly any signal in the space, when only a subset of possible signals is of interest, some bases have better representational properties than others. For example, using the Fourier basis for signals consisting of a few harmonics is more natural than using the standard basis. What do we mean by "more natural"? In this context that means that the representation is much sparser with the Fourier basis than with the standard basis. If the signal consists of harmonics with frequencies on the standard discrete Fourier grid, then the number of nonzero elements in the discrete Fourier transform is equal to the number of harmonics. However, using the standard basis, the number of nonzero elements is in general equal to the dimension of the space, which may be much greater than the number of harmonics.

Some applications which benefit greatly from sparsity of representation are signal compression, denoising, and parameter estimation. In compression for information transmission, if the representation of the signal is not sparse then we need to transmit the whole signal. However, if under a change of basis the representation becomes sparse, then substantial savings are possible. Most coefficients of the representation are very small (by definition of sparsity) and if we set them to zero the perceptual quality of the signal will be affected very little. So we are left with transmitting only the large coefficients, which are few in number. This idea found use in commercial compression algorithms. For example, the JPEG2000 standard [2] uses the fact that the representation of natural images using Daubechies maxflat wavelet bases is considerably sparser than the original representation (in terms of the standard basis). Another application where sparsity plays a key role is denoising. If the signal is sparse then separating it from the noise requires considerably less effort than when signal power is evenly distributed along the support of the signal. Therefore, for the purpose of facility of denoising of a class of signals, it is worthwhile to find a basis in which the representation of all signals belonging to this class is as sparse as possible.

The number of sparsely representable signals directly depends on the number of

[2] See http://www.jpeg2000info.com for details.

elements in the signal dictionary. Every minimal basis has the same number of sparsely representable signals. In order to increase the number of signals with sparse representation, an overcomplete basis has to be used. Some overcomplete bases that have been considered include a concatenation of several orthogonal bases, e.g. standard and Fourier which can sparsely represent superpositions of continuous sinusoids and local sharp phenomena. Another possibility is to use a single extended non-orthogonal basis such as a Fourier basis with the number of considered frequencies exceeding the dimension of the space. The overcomplete Fourier basis allows to represent sparsely harmonics with frequencies in between the standard Fourier grid.

By turning to an overcomplete basis we lose a very important property, the uniqueness of representation. To regain uniqueness, we search for the sparsest solution among the many possible solutions. Mathematically, when no noise is present the problem is as follows: given a signal $\mathbf{y} \in \mathbb{C}^M$, and an overcomplete basis $\mathbf{T} \in \mathbb{C}^{M \times N}$, we would like to find $\mathbf{x} \in \mathbb{C}^N$ such that $\mathbf{y} = \mathbf{T}\mathbf{x}$, and $\mathbf{x}$ is sparse. Define $\|\mathbf{x}\|_0^0$ to be the number of non-zero elements of $\mathbf{x}$. We would like to find $\min \ \|\mathbf{x}\|_0^0$ subject to $\mathbf{y} = \mathbf{T}\mathbf{x}$. This is a very hard combinatorial problem. In Chapter 7 we show [3] that under some conditions on $\mathbf{T}$ and $\mathbf{x}$, the optimal value of this problem can be found exactly by solving a related problem: $\min \ \|\mathbf{x}\|_p^p$ subject to $\mathbf{y} = \mathbf{T}\mathbf{x}$, where $0 < p \leq 1$ (we consider separately two cases, $p = 1$, and general $p$, $0 < p \leq 1$).

A natural extension when we allow white Gaussian noise $\mathbf{n}$ is

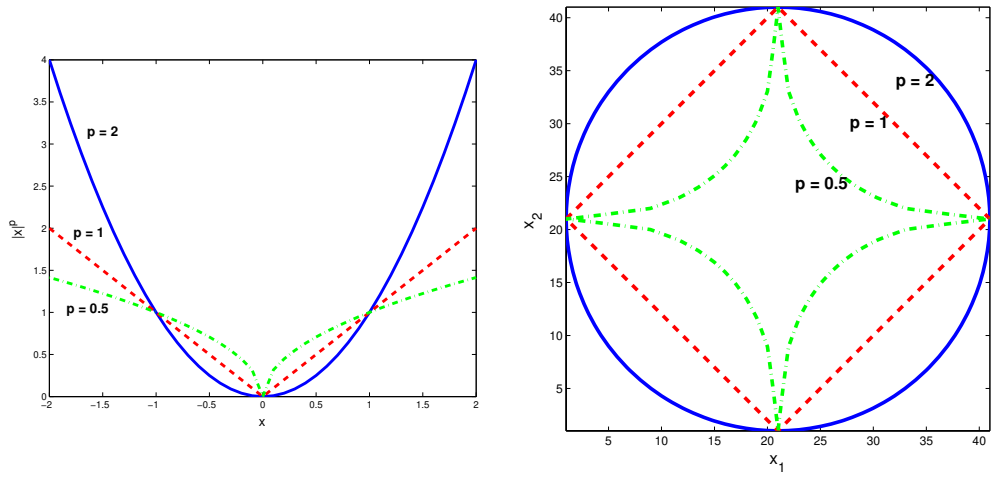
$$\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{n}, \tag{3.7}$$

which can be solved by

$$\min \ \|\mathbf{y} - \mathbf{T}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p. \tag{3.8}$$

If we let $J_1(\mathbf{x}) = \|\mathbf{y} - \mathbf{T}\mathbf{x}\|_2^2$, and $J_2(\mathbf{x}) = \|\mathbf{x}\|_p^p$, then we have nothing but a regularized inverse problem of the form in (3.4). When $p = 1$ this method is called basis pursuit [29] (or LASSO [30] in the statistical literature). The prior term, $J_2(\mathbf{x})$ has an effect of enforcing sparsity. Figure 3.1 gives some insight into why $\ell_p$-regularization with $p \leq 1$ favors sparse $\mathbf{x}$. In plot (b), we show the level sets of $\ell_p$ norms to the $p$-th power ($\|\mathbf{x}\|_p^p$) for $p = 0.5$, $p = 1$, and $p = 2$ of a two-dimensional vector. For a fixed $\ell_2$-norm, i.e. for all vectors that lie on a circle with fixed radius, $\ell_p$ norms with $p \leq 1$ are minimized on the coordinate axes, i.e. preferring that some of the coefficients are set exactly to zero, while others are large. In other words it prefers sparse solutions. This argument can be generalized to vectors in higher dimensions. Plot (a) shows $\ell_p$ norms for the same $p$'s in one dimension. It shows that the penalty on large features (large $x_i$) is less for smaller $p$. Strong features are penalized much less severely in $\ell_p$ penalization with $p \leq 1$ than in $\ell_2$ penalization (Tikhonov regularization). This motivates the smoothing effect of $\ell_2$-penalization, and the feature-preserving behavior of $\ell_p$ for $p \leq 1$.

[3] Our results extend the work of [8] and [9].




(a) (b)

**Figure 3.1.** (a) 1-D Plot of $\|\mathbf{x}\|_p^p$, p = 0.5, 1, 2. (b) 2-D Level sets of $\|\mathbf{x}\|_p^p$, same p.

Another observation from Figure 3.1 is that $\ell_1$ norm is convex, whereas when $p < 1$, $\ell_p$-norm is no longer convex[4]. The computational complexity for the minimization of some non-convex cost functions ($\ell_p$ in particular) can be ameliorated by using the half-quadratic regularization method [31]. The key idea is to introduce a supplementary vector $\mathbf{s}$, and an extended cost function, $Q(\mathbf{x}, \mathbf{s})$ which is quadratic in $\mathbf{x}$ for a fixed $\mathbf{s}$, and $\min_s Q(\mathbf{x}, \mathbf{s}) = J(\mathbf{x})$, for any $\mathbf{x}$. If $Q(\mathbf{x}, \mathbf{s})$ is also easy to minimize in $\mathbf{s}$ (or even better if there is a closed-form solution), then the resulting extended cost function can be optimized with reasonable efficiency by iterative coordinate descent.

The problem of signal representation in overcomplete bases is somewhat different from the more usual inverse problems such as image deconvolution. The focus in the latter one is to make the inverse function continuous. However, for our problem of optimal basis selection, the pseudo-inverse usually already has a small condition number, and no additional regularization is necessary. The trouble with the pseudo-inverse is that it is optimizing an inappropriate measure for the signal representation problem, minimizing the $\ell_2$ norm of $\mathbf{x}$, which does not lead to sparse solutions. We use the other aspect of regularization, finding a unique solution among a large set of possibilities. Regularization is a very flexible framework, and allows us to use an appropriate prior term, an $\ell_p$ norm with $p \leq 1$, which enforces sparsity.

Another reason to use regularization is that we can even relax the overcompleteness assumption. That means that our matrix $\mathbf{T}$ does not have to be overcomplete, and it does not even have to span the space of $\mathbf{y}$. Suppose $\mathbf{y}$ lies in the range of $\mathbf{T}$, but some coefficients of $\mathbf{x}$ obtained by the plain inverse are very small. By using the same approach we have a family of solutions which allow a sparser (but less accurate)

[4]When $p < 1$, the triangle inequality is not satisfied and it would be more precise to use the term "quasi-norm", rather than "norm". However, we will ignore this subtle point, and use the term $\ell_p$-norm for any value of $p$.

representation of $\mathbf{y}$.

Apart from the signal representation in an overcomplete basis, sparsity enforcing regularization has applications in many other fields, such as statistics, data mining and machine learning. An important problem in all three fields is subset selection. Suppose an observed quantity $\mathbf{y}$ depends on many parameters $\mathbf{x} = [x_1, .., x_N]$, but the influence of a small subset of the parameters is much stronger than the influence of the others. In order to build a simple model for $\mathbf{y}$ in terms of $\mathbf{x}$ we must find a small subset of $\{x_i\}$ which predicts $\mathbf{y}$ well. In machine learning a similar problem is called feature selection. Previous approaches to feature selection include stepwise regression [32] (similar to matching pursuit [27]), which is a heuristic method for approximate optimization of the $\ell_0$ norm, i.e. the direct count of variables that we select. If we have a linear model, then the methodology described in this chapter can be used. LASSO [30] ($\ell_1$ penalization) is starting to gain popularity, since it does not have many of the drawbacks of stepwise regression.

In Chapter 4 we describe numerical procedures that can be used to solve our sparsity regularization problems. Regularization using $\ell_1$-norms is described in more detail than $\ell_p$ with $p < 1$ due to the fact that the latter one is non-convex. This makes $\ell_p$ very hard to analyze, and limits the interest of researchers. Quite surprisingly, for the source localization application $\ell_p$ regularization tends to give excellent results despite the fact that the technique for its optimization is only guaranteed to converge to local minima. Arguments to use $\ell_1$ versus $\ell_p$ can be found in Section 6.

Chapter 4

# $\ell_1$ and $\ell_p$ Regularization

The goal of this chapter is to provide intuition and details for the use of $\ell_p$ regularization, and to introduce numerical tools that can be used to optimize the objective functions corresponding to different forms of $\ell_p$-regularization. These numerical tools are used later (in Chapter 5) in the context of source localization. Additionally, we present some observations regarding $\ell_1$ regularization that we made by solving a small problem analytically, and describe a curious property of the noiseless $\ell_1$ problem dealing with sign patterns of exact solutions.

The chapter is divided into two parts: we consider the case when $p = 1$ (e.g. $\ell_1$ regularization) separately from the general $p \leq 1$. The reason for this bifurcation is that for $p < 1$ the cost function is not convex, which makes the problem very challenging. In particular, the theory for $\ell_1$ penalization and the relevant numerical methods are more developed, because few researchers dare to enter the murky realms of nonconvex optimization associated with $p < 1$.

We start with $p = 1$ in Section 4.1. We describe the noiseless problem formulation first, and the noisy problem afterwards. We represent the problems in a second order cone (SOC) programming framework and use an interior point method implementation for their solution. Second order cone programming is motivated in Section 4.1.3. We describe how we recast our objective functions into SOC framework in Section 4.1.4, and illustrate $\ell_1$ techniques on several numerical examples in Section 4.1.5. The analytical solution of a small problem and the topic of sign patterns of exact solutions are presented in Sections 4.1.6 and 4.1.7.

Next, in Section 4.2, we consider general $p \leq 1$. We describe the iterative procedure developed by [11] for $\ell_p$ optimization which is based on the half-quadratic regularization method of [31].

## ■ 4.1 $\ell_1$-regularization

The use of $\ell_1$-norms to achieve sparsity has been known for almost a decade. The Least Absolute Shrinkage and Selection Operator (LASSO) has been introduced in the statistics literature [33], and Basis Pursuit algorithm [29] for choosing a sparse basis has been proposed in the signal representation community at around the same time. Some applications to signal processing, and even to array processing, have been considered

[13,14]. The most important advantages of $\ell_1$ penalization schemes are their convexity, and the strong sparsity of the results (most indices of the result are set exactly to zero). Different versions of $\ell_1$-regularization problems can be reformulated as linear, convex constrained or unconstrained quadratic, or second order cone (SOC) programming, all of which allow efficient and globally convergent algorithms. Another significant benefit of using $\ell_1$ penalization is a number of recent theoretical (e.g. [8]) results showing that under certain sparsity conditions on the underlying unknown signal, the signal can be recovered exactly. This is quite surprising since the direct combinatorial formulation of the problem requires comparing solutions with all possible permutations of non-zero indices, which is very hard. These theoretical results and some extensions that we have made are postponed until Chapter 7.

### ■ 4.1.1 Noiseless case

First we consider the noiseless version of the problem of signal representation in overcomplete bases, and describe its solution using $\ell_1$-penalization. Although the scenario without noise has little practical value in array processing, it has close connection to the noisy formulation, and most theoretical results have been developed for the noiseless case. Thus much insight into the behavior and properties of the noisy counterpart can be acquired by taking the noiseless case into consideration.

We repeat the noiseless signal representation problem from the previous chapter. Suppose we have a signal $\mathbf{y}$, which is a combination of a *few* weighted elements of an *overcomplete* basis[1] $\mathbf{A}$: $\mathbf{y} = \sum_{i=1}^{K} x_i \mathbf{a}_i$, where $\mathbf{A} = [\mathbf{a}_1 ... \mathbf{a}_N]$. We would like to find out the weighting coefficients $x_i$, using the available information, $\mathbf{y}$, and $\mathbf{A}$. $\mathbf{A}$ is overcomplete, and uncovering $x_i$ is an ill-posed inverse problem, since infinitely many solutions exist. The situation is however not hopeless, because only a few $x_i$ are non-zero (although *apriori* we do not know either the number of non-zero coefficients, or their indices). A meaningful way to attempt to find out the solution is to try to solve

$$\min \ \|\mathbf{x}\|_0^0, \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{4.1}$$

Recall that $\|\mathbf{x}\|_0^0$ represents the number of nonzero elements of $\mathbf{x}$. This a hard combinatorial problem, and we replace it by a related problem,

$$\min \ \|\mathbf{x}\|_1, \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{4.2}$$

Equation (4.2) can be reformulated as a linear programming problem when the data is real, and as a second order cone problem when the data is complex. In Section 7 we describe the conditions under which the two problems (4.1) and (4.2) are equivalent.

First, we describe the numerical solution for the case when the data is real. The procedure for reformulating problems involving minimization of $\ell_1$-norms as linear programming problems (provided all other terms are linear functions) is well known to

[1] We used $\mathbf{T}$ in last chapter. For the rest of the thesis we are dealing with the specific inverse problem of signal representation and we use $\mathbf{A}$ instead.

the optimization community. First we introduce two variables $\mathbf{x}^+$ and $\mathbf{x}^-$, defined as $x_i^+ = max\{x_i, 0\}$, $x_i^- = max\{-x_i, 0\}$, from which $\mathbf{x}$ can be simply recovered as $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$. The variables $\mathbf{x}^+$ and $\mathbf{x}^-$ are limited to the positive orthant, and they must satisfy $x_i^+ x_i^- = 0, \forall i$. The last condition gets automatically satisfied when we consider minimization problems. Rewriting the problem (4.2), we have

$$\min \mathbf{1}' \begin{pmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \end{pmatrix} \text{ subject to } \begin{pmatrix} \mathbf{A} & -\mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \end{pmatrix} = \mathbf{y} \text{ and } \begin{pmatrix} \mathbf{x}^+ \\ \mathbf{x}^- \end{pmatrix} \geq \mathbf{0} \tag{4.3}$$

The numerical solution of (4.3) is easily accomplished using the simplex method, or an interior point method for linear programming.

We illustrate noiseless $\ell_1$ minimization on a simple numerical example. We use a random overcomplete basis, where each element of a $10 \times 40$ matrix $\mathbf{A}$ is a zero-mean unit-variance Gaussian random variable independent of other elements. Figure 4.1 (a) illustrates that the procedure is able to recover the underlying coefficients exactly (the original and the recovered signals match). However, when small amounts of noise are added ($\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $n_i$ are i.i.d. Gaussian with $\sigma = .02$), the reconstruction breaks down, and bears little information about the underlying signal, see Figure 4.1 (b).

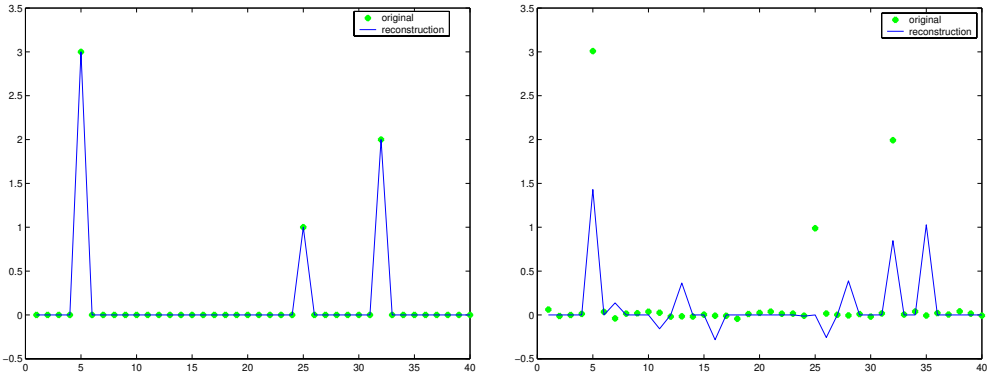



**Figure 4.1.** Noiseless formulation of $\ell_1$ regularization: real-valued data. (a) No noise is present. Solution is exact. (b) Data corrupted by noise. Solution breaks.

We show another aspect of $\ell_1$ penalization in Figure 4.2. This can be called super-resolution. In plot (a) we continue our random basis example, except we bring two of the peaks of the original signal close together. In this plot we compare the result of solving the noiseless $\ell_1$ problem, e.g. minimizing $\ell_1$ norm subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$ against the result of doing a plain pseudo-inverse. The pseudo-inverse solution is not sparse; in addition it also loses the third peak and distorts the amplitude of the other two which happen to be above the floor of spurious peaks. The $\ell_1$ solution on the other hand

perfectly recovers the signals[2].

In plot (b) of Figure 4.2 we use an overcomplete discrete Fourier basis instead of the random one. Now the data is complex. The dimensions of $\mathbf{A}$ are again $10 \times 40$. In this context $\mathbf{y} \in \mathbb{C}^M$ can be viewed as the time-domain signal, and $\mathbf{x} \in \mathbb{C}^N$ as its overcomplete Fourier representation. Let $\mathbf{F} \in \mathbb{C}^{N \times M}$ be a matrix with the following entries: $\mathbf{F}_{n,m} = \exp(-j2\pi \frac{n-1}{N} m)$. It corresponds to taking the first $M$ columns of the usual $N \times N$ DFT matrix $\mathbf{F}_N$. Let $\mathbf{A} = \frac{1}{N}\mathbf{F}'$. Then we have our usual inverse problem, $\mathbf{y} = \mathbf{A}\mathbf{x}$. The pseudo-inverse solution $\mathbf{x}_{PINV} = \mathbf{A}^\dagger \mathbf{y}$ is nothing but the spectrum of $\mathbf{y}$ obtained by zero-padding: $\mathbf{x}_{PINV} = \mathbf{x}_{ZPD} = \mathbf{F}_N \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}$. All the columns of $\mathbf{F}_N$ that are not present in $\mathbf{F}$ are multiplied by zeros. The real part of $\mathbf{x}_{PINV}$ is shown in plot (b)2. The result is smooth and not sparse. Two of the original peaks which are close together are not resolved by the pseudo-inverse spectrum. This is an example of the fundamental limit on resolution using linear operators. If instead of taking the pseudo-inverse we use the $\ell_1$ approach, we obtain the original $\mathbf{x}$ exactly (also shown in plot (b)). Thus in a sense we are able to go beyond the limits of resolution, which is possible due to the fact that the number of entries in $\mathbf{x}$ is small. This example has direct connection with the source localization problem. The overcomplete DFT basis is more suitable for explaining superresolution, because columns of $\mathbf{A}$ which are close in terms of column-index are also close in terms of Euclidean distance. This leads to smooth pseudo-inverse spectra. When $\mathbf{A}$ is a random overcomplete basis this is no longer the case, and that is the reason why the pseudo-inverse solution in plot (a) appears to have little structure.

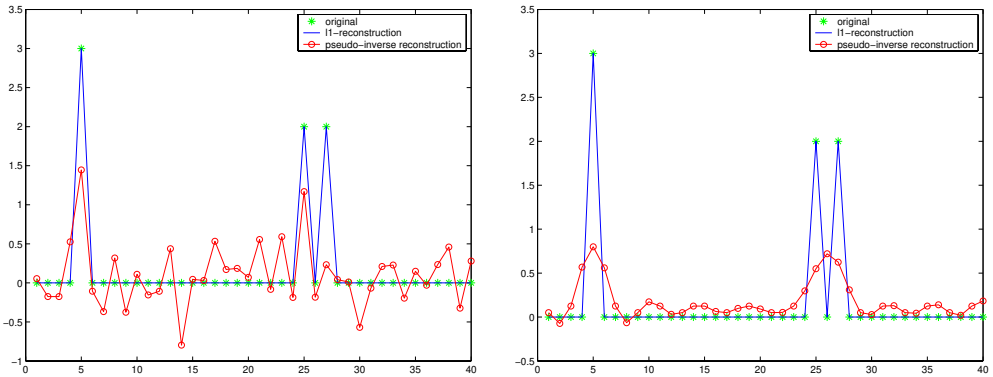



**Figure 4.2.** Comparison of noiseless $\ell_1$ penalization with pseudo-inverse solutions: (a) Overcomplete $10 \times 40$ random basis (real data). (b) Overcomplete $10 \times 40$ DFT basis (complex data).

So far we have discussed the numerical solution of the real-valued scenario only, but in our array processing application (and for the example in plot (b) of Figure

[2]The basis in this example is random, and on rare occasions it does not satisfy the conditions for exact solutions from Chapter 7. Then the solution minimizing the $\ell_1$-norm will not be the same as the original signal.

4.2) we are forced to deal with complex-valued data. All the formulations (noiseless and all the noisy ones in the next section) remain the same, but the $\ell_1$ norm can no longer be represented as a linear function of the arguments (of their real and imaginary parts): $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} \sqrt{Re(x_i)^2 + Im(x_i)^2}$, where $Re()$ and $Im()$ represent the real and imaginary parts. Notice that even if we square the $\ell_1$ norm we still get terms which involve square roots, thus none of the problems can be reformulated as quadratic programming. The use of complex-valued data compels us to enter the domain of second order cone (SOC) programming. Fortunately, SOC problems have efficient globally convergent algorithms implemented in the framework of Interior Point Methods (IPM). We discuss SOC later on in this chapter, and IPM in Appendix B.

### ■ 4.1.2 Handling noise

An equivalent form of the noiseless problem (4.2) is

$$\min \ \|\mathbf{x}\|_1, \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 = 0, \tag{4.4}$$

which right away suggests how to accommodate noise:

$$\min \ \|\mathbf{x}\|_1, \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \beta^2, \tag{4.5}$$

where $\beta$ is a regularization parameter which sets the amount of noise that we wish to allow. Also, several alternative equivalent formulations can be considered. The most widely considered formulation (in particular it is used in [29,33]) is the penalized form:

$$\min \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{4.6}$$

And, if we switch the constraints with the objective, then we arrive at the third form:

$$\min \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, \text{ subject to } \|\mathbf{x}\|_1 \leq \delta \tag{4.7}$$

Notice that in (4.5), we can rewrite the constraint $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \leq \beta^2$ as $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \beta$, which leads to an equivalent penalized form:

$$\min \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 + \tilde{\lambda}\|\mathbf{x}\|_1 \tag{4.8}$$

The difference from (4.6) lies in the absence of the square for $\ell_2$ norm term. This version can be represented more readily in the SOC framework.

For convenience of referring to the different formulations we introduce the following labels for the problems: we refer to the first constrained form, (4.5), which minimizes an $\ell_1$ norm subject to an $\ell_2$ constraint as "ML1". The other constrained form, (4.7) which minimizes an $\ell_2$ norm subject to an $\ell_1$ constraint is labeled "ML2". And the two unconstrained problems, (4.8), (4.6), which minimize a joint cost function, we call MLJ and MLJSQ respectively (SQ is for square of the $\ell_2$ norm). Also, note that in ML1 and ML2 problems the square on the constraint and on the objective respectively can be removed without changing the problem. Thus we also refer

to the problem min $\|\mathbf{x}\|_1$ subject to $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \beta$ as ML1, and to the problem min $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ subject to $\|\mathbf{x}\|_1 \leq \delta$ as ML2.

All of the formulations, ML1, ML2, MLJ, and MLJSQ are equivalent in the sense that the sets of solutions corresponding to all possible choices of the regularization parameters are the same for all the formulations. That means that going from one formulation to another we only have to properly map the corresponding regularization parameters (this is not trivial to do, but such a mapping exists, although possibly nonlinear, and discontinuous).

All of the above formulations may have their applications, but for the source localization method we mainly use ML1, MLJ, and MLJSQ. The numerical solution is obtained for all of them in a second order cone framework (Section 4.1.3) via an interior point implementation. We present numerical examples and comparison of the virtues of different versions in Section 4.1.5. Next we discuss second order cone programming.

### ■ 4.1.3 Second order cone programming

We remind the reader that the need to consider SOC arose from the use of $\ell_1$ norm with complex data. For $\mathbf{x} \in \mathcal{C}^N$, $\|\mathbf{x}\|_1$ is neither a linear nor a quadratic function of the real and imaginary components, and cannot be rewritten as one: $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} \sqrt{Re(x_i)^2 + Im(x_i)^2}$.

SOC programming explicitly deals with constraints of the form $\|x_2, ..., x_n\|_2 \leq x_1$, which bear the name second order cone, or Lorentz cone. To represent $\ell_1$ norms of complex data we only need to consider direct products of several second order cones. We delay the details of representation until Section 4.1.4. It turns out that SOC programming has many favorable properties, efficient algorithms for computation, and substantial theoretical foundation.

We briefly summarize some of the reasons why SOC is endowed with so many benefits [10]. It can be easily verified that second order cone is a closed pointed convex cone which possesses a non-empty interior [3]. As such it falls under the realm of convex conic programming (CP). As we illustrate, conic programming may be regarded as a generalization of linear programming (LP), and many important attributes of LP carry over to CP.

LP problems can be written in the following form:

$$\min \quad \mathbf{c}'\mathbf{x} \quad \text{such that} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \tag{4.9}$$

The constraint $\mathbf{A}\mathbf{x} \geq \mathbf{b}$ is equivalent to $\mathbf{A}\mathbf{x} - \mathbf{b} \geq 0$, or $\mathbf{A}\mathbf{x} - \mathbf{b} \in \mathbb{R}^N_+$, where $\mathbb{R}^N_+ = \{\mathbf{x} \in \mathbb{R}^N | x_i \geq 0\}$, the positive quadrant. Hence, the order relation $\mathbf{a} \geq \mathbf{b}$ is equivalent to $\mathbf{a} - \mathbf{b} \in \mathbb{R}^N_+$. $\mathbb{R}^N_+$ is a closed pointed convex cone, and $\mathbb{R}^N_+$ induces a partial ordering[4] on $\mathbb{R}^N$. This partial ordering of $\mathbb{R}^N$ is not unique, and in fact it can be shown that any pointed convex cone $\mathbf{K}$ induces a partial ordering in the same fashion. Using the

[3] Closed means that it contains all its limit points, [34], and pointed means that it contains no lines ($+\mathbf{x}$ and $-\mathbf{x}$ cannot both belong to a pointed cone unless $\mathbf{x} = 0$).

[4] A partial order is basically an order relation where some elements may not be comparable.

notation from [10], $\mathbf{a} - \mathbf{b} \in \mathbf{K}$ induces a partial ordering $\mathbf{a} \geq_{\mathbf{K}} \mathbf{b}$, i.e. $\mathbf{K}$ induces a new partial ordering on $\mathbb{R}^N$, denoted by "$\geq_{\mathbf{K}}$". We can consider a new class of optimization problems:

$$\min \quad \mathbf{c}'\mathbf{x} \quad \text{such that} \quad \mathbf{A}\mathbf{x} \geq_{\mathbf{K}} \mathbf{b} \tag{4.10}$$

If we impose the additional restrictions that $\mathbf{K}$ is closed, and possesses non-empty interior, (thus eliminating some possible degenerate partial orderings), the new class of optimization problems not only looks similar to the formulation of LP in (4.9), but also shares many of its properties. The depth of the similarity between LP and CP can be seen by looking at duality. Duality is a very strong result in LP theory, and it concerns the problem dual[5] to (4.9) (the derivation can be found in any linear programming book):

$$\min \quad \mathbf{b}'\mathbf{y} \quad \text{such that} \quad \mathbf{A}'\mathbf{y} = \mathbf{c}, \mathbf{y} \geq 0 \tag{4.11}$$

The weak LP duality theorem states that any $\mathbf{y}$ feasible for the dual problem has the objective value $\mathbf{b}'\mathbf{y}$ lower than or equal to the optimal objective value of the primal problem. The strong LP Duality theorem, a central result in LP theory states that if either the primal (4.9) or the dual (4.11) are solvable (feasible and bounded above), then the primal and the dual objective values are equal (i.e. there is no duality gap).

A similar result holds for conic programming. First we define the dual cone, $\mathbf{K}_*$:

$$\mathbf{K}_* = \{\mathbf{y} \in \mathbb{R}^N | \mathbf{y}'\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbf{K}\} \tag{4.12}$$

If $\mathbf{K}$ is a closed convex pointed cone with non-empty interior, then so is $\mathbf{K}_*$, and additionally the cone dual to $\mathbf{K}_*$ is $\mathbf{K}$. The problem dual to the conic programming program in (4.10) takes the following form [10]:

$$\min \quad \mathbf{b}'\mathbf{y} \quad \text{such that} \quad \mathbf{A}'\mathbf{y} = \mathbf{c}, \mathbf{y} \geq_{\mathbf{K}_*} 0 \tag{4.13}$$

As in LP case, the problem dual to the dual problem is the primal problem. The weak duality theorem holds exactly as for LP, and the strong duality also holds but with some extra assumptions on non-degeneracy of solutions.

If we take $\mathbf{K}$ to be a second order cone, then we get a second order cone programming problem. It immediately inherits all the properties of conic programming, but also has a multitude of its own. For example, second order cones are self dual, i.e. $\mathbf{K} = \mathbf{K}_*$. Another extremely rich and useful subclass of cone programming is semidefinite programming (SDP), where $\mathbf{K}$ is the cone of positive semidefinite matrices. Both SOC programming and SDP allow very efficient solutions using an interior point approach.

[5]Define the Lagrangian function: $L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}'\mathbf{x} - \boldsymbol{\lambda}'(\mathbf{A}\mathbf{x} - \mathbf{b})$. Define the dual function: $q(\boldsymbol{\lambda}) = \inf_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$. Then the dual problem is $\max q(\boldsymbol{\lambda})$ subject to $\boldsymbol{\lambda} \geq 0$. See [35] for more general definitions.

### ■ 4.1.4 Representing $\ell_1$ problems with complex data in SOC framework

We now describe how to pose the $\ell_1$ optimization problems described in Sections 4.1.1 and 4.1.2 in the form of second order cone programming. SOC representation of the problems allows us to use an implementation [36] of a path-following interior point method for optimization over symmetric cones for their solution. We touch upon the topic of interior point methods in Appendix B.

The general form of a second order cone problem is:

$$\begin{aligned} &\min\ \mathbf{c}'\mathbf{x} \\ &\text{such that}\ \ \mathbf{A}\mathbf{x} = \mathbf{b},\ \text{and}\ \mathbf{x} \in \mathbf{K} \end{aligned}$$

where $\mathbf{K} = \mathbb{R}^N_+ \times \mathbf{L}_1 ... \times \mathbf{L}_{N_L}$. $\mathbb{R}^N_+$ is the $N$-dimensional positive orthant cone, and $\mathbf{L}_1, ..., \mathbf{L}_{N_L}$ are second order cones (Lorentz cones). Second order cone of dimension $n$ ($n$ does not have to equal to $N$) has the following definition:

$$\mathbf{L} = \{\mathbf{x} : x_1 \geq \|x_2, ..., x_n\|_2\} \tag{4.14}$$

We are interested in representing problems (4.2, 4.5, 4.6, 4.7, 4.8) with real and complex data in terms of SOC constraints.

#### Noiseless $\ell_1$ problem

The most basic $\ell_1$ problem with complex data is (4.2), which we restate here for convenience:

$$\min\ \|\mathbf{x}\|_1,\ \text{subject to}\ \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{4.15}$$

Since in the formulation of SOC we cannot have nonlinear objective functions, we rewrite it as

$$\min\ t,\ \text{such that}\ \ \|\mathbf{x}\|_1 \leq t,\ \ \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{4.16}$$

Furthermore, the inequality with the $\ell_1$-norm of $\mathbf{x} \in \mathbb{C}^n$ can be restated as a direct product of $N$ second order cones of dimension 3:

$$\|\mathbf{x}\|_1 \leq t \quad \Leftrightarrow \quad \|Re(x_i), Im(x_i)\|_2 \leq t_i,\ \text{for}\ i \in \{1, .., N\}\ \text{and}\ t = \sum_{i=1}^{N} t_i \tag{4.17}$$

Using the above two steps, the SOC formulation of (4.2) becomes:

$$\min \mathbf{1}'\mathbf{t} \tag{4.18}$$

$$\text{subject to}\ \|Re(x_i), Im(x_i)\|_2 \leq t_i, i \in \{1, .., N\}, \tag{4.19}$$

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{4.20}$$

Each of the $N$ constraints $\|Re(x_i), Im(x_i)\|_2 \leq t_i$ can be written as $(t_i, Re(x_i), Im(x_i)) \in \mathbf{L}_i$, meaning that the triple belongs to a second order cone. The reformulation of the other problems proceeds in a similar fashion.

**Constrained noisy $\ell_1$ problems, ML1 and ML2**

Consider the ML1 problem first: min $\|\mathbf{x}\|_1$ subject to $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2 \leq \beta$. This is the form without squaring $\beta$ and the $\ell_2$-norm of the residual; it fits easier into SOC framework without the squares. We introduce a new variable, $\mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x}$. The problem is transformed into the following:

$$\min \mathbf{1}'\mathbf{t} \tag{4.21}$$

$$\text{subject to } \|Re(x_i), Im(x_i)\|_2 \leq t_i, i \in \{1,..,N\}, \tag{4.22}$$

$$\mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x} \text{ and } \|\mathbf{z}\|_2 \leq \beta \tag{4.23}$$

For the representation we use a new second order cone, $\|\mathbf{z}\|_2 \leq \beta$, i.e. $(\beta, \mathbf{z}) \in \mathbf{L}$. This new cone has dimension $2M+1$, since $\mathbf{z}$ is a complex M-dimensional vector, i.e. the second order cone constraint can be expanded to $(\beta, Re(\mathbf{z}), Im(\mathbf{z})) \in \mathbf{L}$.

The ML2 problem, min $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$, subject to $\|\mathbf{x}\|_1 \leq \delta$, has a very similar representation:

$$\min s \tag{4.24}$$

$$\text{subject to } \mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x} \text{ and } \|\mathbf{z}\|_2 \leq s \tag{4.25}$$

$$\|Re(x_i), Im(x_i)\|_2 \leq t_i, i \in \{1,..,N\}, \text{ and } \mathbf{1}'\mathbf{t} \leq \delta \tag{4.26}$$

**Joint noisy $\ell_1$ problems, MLJ and MLJSQ**

The representation of the MLJ problem, min $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1$, differs slightly from the previous two:

$$\min s + \lambda \mathbf{1}'\mathbf{t} \tag{4.27}$$

$$\text{subject to } \mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x} \text{ and } \|\mathbf{z}\|_2 \leq s \tag{4.28}$$

$$\text{and } \|Re(x_i), Im(x_i)\|_2 \leq t_i, \quad i \in \{1,..,N\}, \tag{4.29}$$

Some difficulty arises with the MLJSQ problem, since the norm in $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ is squared. We repeat that (4.8) and (4.6) are equivalent up to a nonlinear mapping of the regularization parameter, but since the latter form is the one that is used in LASSO and Basis Pursuit, it is worthwhile to have an implementation for comparison's sake. It is surprising that the trouble is caused by the simple squared $\ell_2$ norm constraint, but fortunately, there is a simple trick that can be used to represent the squared $\ell_2$ norm inequalities in terms of second order cones [10].

We again have $\mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x}$. We need to minimize $\|\mathbf{z}\|_2^2$, or $\mathbf{z}'\mathbf{z}$. This can also be expressed as minimizing $s$, such that $\mathbf{z}'\mathbf{z} \leq s$. Now, it is a fact that $s = \frac{(s+1)^2}{4} - \frac{(s-1)^2}{4}$, thus the above inequality is the same as $\mathbf{z}'\mathbf{z} + \frac{(s-1)^2}{4} \leq \frac{(s+1)^2}{4}$, which can be expressed

as $\|\mathbf{z}, (s-1/2)\|_2 \leq (s+1)/2$. Let $u = (s+1)/2$, and $v = (s-1)/2$, then the constraint becomes $\|\mathbf{z}, v\|_2 \leq u$, which is in SOC form. The full representation of the MLJSQ version is:

$$\min s + \lambda \mathbf{1}'\mathbf{t} \tag{4.30}$$

$$\text{subject to } \mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x} \text{ and } \|\mathbf{z}, v\|_2 \leq u \tag{4.31}$$

$$u = \frac{s+1}{2}, \quad v = \frac{s-1}{2}, \quad s \geq 0 \text{ and } \|Re(x_i), Im(x_i)\|_2 \leq t_i, \quad i \in \{1, .., N\}, \tag{4.32}$$

#### Other problems representable in SOC framework

To illustrate the power of SOC constraints, we list some other functions and sets which can be represented in terms of SOC constraints using more sophisticated procedures of the same flavor as was done for the square of $\ell_2$ norm in (4.32), [10]. The set $\{(t, s) \in \mathbb{R}^2 | ts \geq 1, t > 0\}$, as well as $\{(x_1, x_2, t) \in \mathbb{R}^3 | x_1, x_2 \geq 0, t \leq \sqrt{x_1 x_2}\}$ can be reformulated as SOCs. Also, quite surprisingly, constraints with p-norm, $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$, where $p \geq 1$, and $p$ being a rational number can also be represented. (However, this is only practical for fractions with numerators and denominators which are small integers). Hence, the use of efficient algorithms induced by SOC representation is applicable to a much wider set of problems than is readily apparent at first sight.

#### Optimizing SOC problems by an interior point method

Several implementations of SOC programming by interior point methods (IPM) have been developed in the optimization community, and we use a package for optimization over self-dual homogeneous cones (which includes direct products of the positive orthant-constraints, SOC constraints and semidefinite cone constraints), SeDuMi [36], developed by J. Sturm at Tilburg University. Other non-commercial interior point alternatives which allow SOC constraints include SDPT3 [37].

The topic of interior point methods is an exciting recent development in optimization. The original role of the interior point approach to optimization was that of a competitor to the simplex method for linear programming problems. Nowadays it is well understood that the scope of problems efficiently tackled by the interior point approach is much wider. For linear programming some classes of problems can be solved by interior point methods much more efficiently than using the simplex method [38]. For second order cone programming and semidefinite programming, the importance of interior point methods is even greater, since there are no other efficient alternatives. We give a brief introduction of interior point methods in Appendix B. However, for the purposes of the thesis interior point methods are simply a convenient efficient tool.

The general form of a problem solved by SeDuMi is:

$$\begin{aligned} &\min \ \mathbf{c}'\mathbf{x} \\ &\text{such that } \ \mathbf{A}\mathbf{x} = \mathbf{b}, \text{ and } \mathbf{x} \in \mathbf{K} \end{aligned}$$

where $\mathbf{K} = \mathbb{R}^N_+ \times \mathbf{L}_1 ... \times \mathbf{L}_{N_L} \times \mathbf{S}_1 ... \times \mathbf{S}_{N_S}$. $\mathbb{R}^N_+$ is the $N$-dimensional positive orthant cone, $\mathbf{L}_1, ..., \mathbf{L}_{N_L}$ are second order (Lorentz) cones, and $\mathbf{S}_1, ..., \mathbf{S}_{N_S}$ are semidefinite cones. The representative power of second order cones suffices for our tasks, thus we do not use the option of more powerful SDP constrains. We already have representations of all our $\ell_1$ problems in this form, thus using SeDuMi to solve them poses no additional difficulties.

#### Semidefinite representation of SOC constraints

Alternatively, it is also possible to represent SOC constraints as semidefinite constraints. The overhead in rewriting SOC constraints in terms of the semidefinite cone is substantial; it is more efficient to use a SOC solver if one is available. However, semidefinite programming is more emphasized in current optimization research; software implementations of SDP are more readily available and may evolve at a faster pace than SOC programming. For that reason we show how to represent SOC constraints as semidefinite cone constraints.

First, the semidefinite cone is defined for the space of symmetric matrices, $\mathbf{X} \in \mathbb{R}^{M \times M}$, which is equivalent to a plain $M(M+1)/2$-dimensional vector space. The semidefinite cone constraint requires that the matrix $\mathbf{X}$ is positive semidefinite, i.e. $\mathbf{y'Xy} \geq 0 \quad \forall \mathbf{y}$, which is the same as requiring that the eigenspectrum of $\mathbf{X}$ is non-negative. The second order cone constraint $t \geq \|\mathbf{x}\|_2$, where $t \in \mathbb{R}^1, \mathbf{x} \in \mathbb{R}^N$, is equivalent to positive semidefiniteness of the following matrix: $\left( \begin{smallmatrix} t\mathbf{I}_N & \mathbf{x} \\ \mathbf{x}' & t \end{smallmatrix} \right)$, where $\mathbf{I}_N$ is an identity matrix. Needless to say, the representation is not efficient, since we have gone from an $N+1$-dimensional space to $(N+1)(N+2)/2$-dimensional space. Nevertheless, when an SOC solver is not available, this may be a good idea. To pay the dues to the power of SDP, it has to be said that SOC problems are but a small subset of the possible problems that can be handled using SDP constraints. Some other examples include eigenvalue and singular-value problems, as well as relaxations of many combinatorial optimization problems.

### ■ 4.1.5 Numerical examples of $\ell_1$ regularization

Armed with an implementation of a path-following interior point method for second order cone optimization we undertake several numerical experiments with $\ell_1$ penalization. First we consider the joint unconstrained formulation of the problem, (4.6). We take $\mathbf{A}$ to be a random-Gaussian matrix (all $\mathbf{A}_{i,j}$ are i.i.d. standard normal random numbers), $\mathbf{A} \in \mathbb{R}^{10 \times 40}$, $\mathbf{x}$ is sparse, and $\mathbf{y} = \mathbf{Ax} + \mathbf{n}$, where $\mathbf{n}$ is an i.i.d. Gaussian random vector with standard deviation $\sigma = 0.02$. This is the same problem as the one considered in Section (4.1.1), which could not be handled using the noiseless version of $\ell_1$ penalization. It can be seen from Figure 4.3 (a) that with a proper choice of the regularization parameter $\lambda$ (we discuss it later, in Chapter 5) the noisy formulation of (4.6) has no difficulty with the added noise, and we recover a very close approximation to the original signal. The crucial assumption for this scheme to work is the sparsity of the unknown signal $\mathbf{x}$. In Figure 4.3 (b), we see the outcome when the assumption

does not hold. In this example, $\mathbf{x}$ has 5 non-zero entries, which is half of the dimension of the embedding column space of $\mathbf{A}$, and is no longer sparse. The recovered signal has little resemblance to the original hidden signal. The topic of the required sparsity is very challenging, and we discuss it in more detail in Chapter 7.

In Figure 4.4 we return to the overcomplete $10 \times 40$ DFT basis, and compare the $\ell_1$ reconstruction with the $\ell_2$ reconstruction. We use the same sparse signal $\mathbf{x}$ with 3-nonzero elements as in Figure 4.3, plot (a). It can be seen that with the noise added the $\ell_1$ reconstruction is not exact, but is a good approximation to the original signal. However, the $\ell_2$ reconstruction bears little resemblance to the original signal.

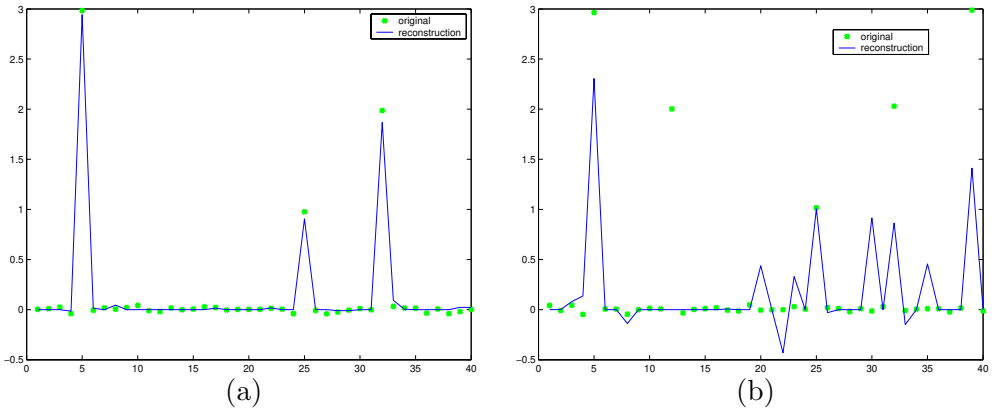



**Figure 4.3.** Data corrupted by noise, random overcomplete basis, joint $\ell_1$ formulation MLJSQ. (a) $\mathbf{x}$ is sparse, and the method works well. (b) $\mathbf{x}$ is not sparse, method breaks.

According to the previous examples, the joint formulation (4.6) appears quite promising for practical linear inverse problems with sparse signals, and the reader may question the need to consider the other equivalents. We discuss the answer to this question next.

Each version has a scalar regularization parameter, and we plot the set of solutions over the grid of possible regularization parameters, along with the $\ell_1$-norms of the solutions, and the residuals, $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ in Figure 4.5. The true underlying signal is the same as the one in Figure 4.3 (a). It has three nonzero entries: $x_5 = 3, x_{25} = 1, x_{32} = 2$. ML1 version is in plot (a), ML2 in plot (b), MLJ in plot (c), and MLJSQ in plot (d). The top subplot in each plot has the set of solutions for a grid of parameters, displayed as an intensity map. In plot (a), vertical coordinate is the index of the vector, and horizontal is the relevant regularization parameter. The middle and bottom subplots contain $\|\mathbf{x}\|_1$ and $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ as a function of the relevant regularization parameter[6].

The main attraction of ML1 and ML2 formulations is in their ability to directly control the resulting $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$, and $\|\mathbf{x}\|_1$ respectively. Plots (a) and (b) show that

[6]In MLJ and MLJSQ problems we added a factor $1 - \lambda$ in front of $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ in order to limit the useful grid of $\lambda$ to the range $[0, 1]$. Compared to the original formulation this leads to an invertible nonlinear continuous scaling of the $\lambda$-axis.

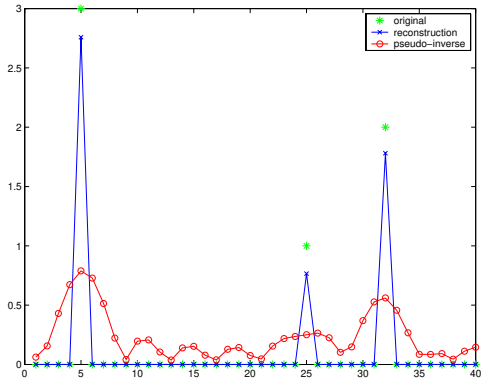



**Figure 4.4.** Data corrupted by noise, overcomplete DFT basis, joint $\ell_1$ formulation MLJSQ. The underlying $\mathbf{x}$ is sparse, and $\ell_1$ gives good sparse solutions but the pseudo-inverse gives poor blurred solutions with notable sidelobes.

as we vary $\beta$ in the ML1 problem, the resulting norm of the residual of the optimal solution, $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ is equal to $\beta$ (until $\beta = \|\mathbf{y}\|_2$). Similarly as we vary $\delta$ in ML2, $\|\mathbf{x}\|_1$ follows it closely up to $\|\mathbf{x}_N\|_1$, where $\mathbf{x}_N$ is the solution of the noiseless problem min $\|\mathbf{x}\|_1$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$.

Unlike ML1 and ML2, the joint formulations MLJ and MLJSQ have a very erratic behavior of the residual $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$, and the $\ell_1$ norm of the optimal solution, as a function of $\lambda$ (plots (c) and (d)). The curves for MLJSQ appear to be smooth, but a closer look (by zooming in) uncovers discontinuities of the derivative. This erratic behavior has advantages as well as drawbacks. On the positive side, the choice of regularization parameter is fairly robust due to the long flat regions in both curves. In ML1 and ML2, on the contrary, any small change of the regularization parameter leads to a commensurate change in the optimal solution, and therefore $\beta$ and $\delta$ must be chosen with care.

On the other hand, since $\lambda$ is not directly linked to either $\|\mathbf{x}\|_1$, or $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$, it is hard to predict *apriori* a good choice for $\lambda$. The dependence of $\lambda$ on $\beta$ is highly nonlinear, and there is no known way to predict the proper choice for $\lambda$ with the knowledge of $\beta$ or $\delta$ in the ML1 and ML2 formulations.

Another small point to make is the difference between MLJ and MLJSQ problems. MLJSQ is the standard formulation considered in statistics and signal representation communities, but MLJ is a little more efficient computationally using the SOC framework (square of the norm in MLJSQ is accommodated using a trick). Also MLJ appears to be more robust to the choice of regularization parameter, but further investigation is necessary to support this claim.

Finally, we include a plot of $\lambda\|\mathbf{x}\|_1 + (1-\lambda)\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ for MLJ problem and a plot of $\lambda\|\mathbf{x}\|_1 + (1-\lambda)\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ (extra square) for MLJSQ problem in Figure 4.6. This weighted sum of the jagged curves $\|\mathbf{x}\|_1$ and $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$ from Figure 4.5 plots (c) and (d) appears to be smooth. These plots pose several interesting questions, which

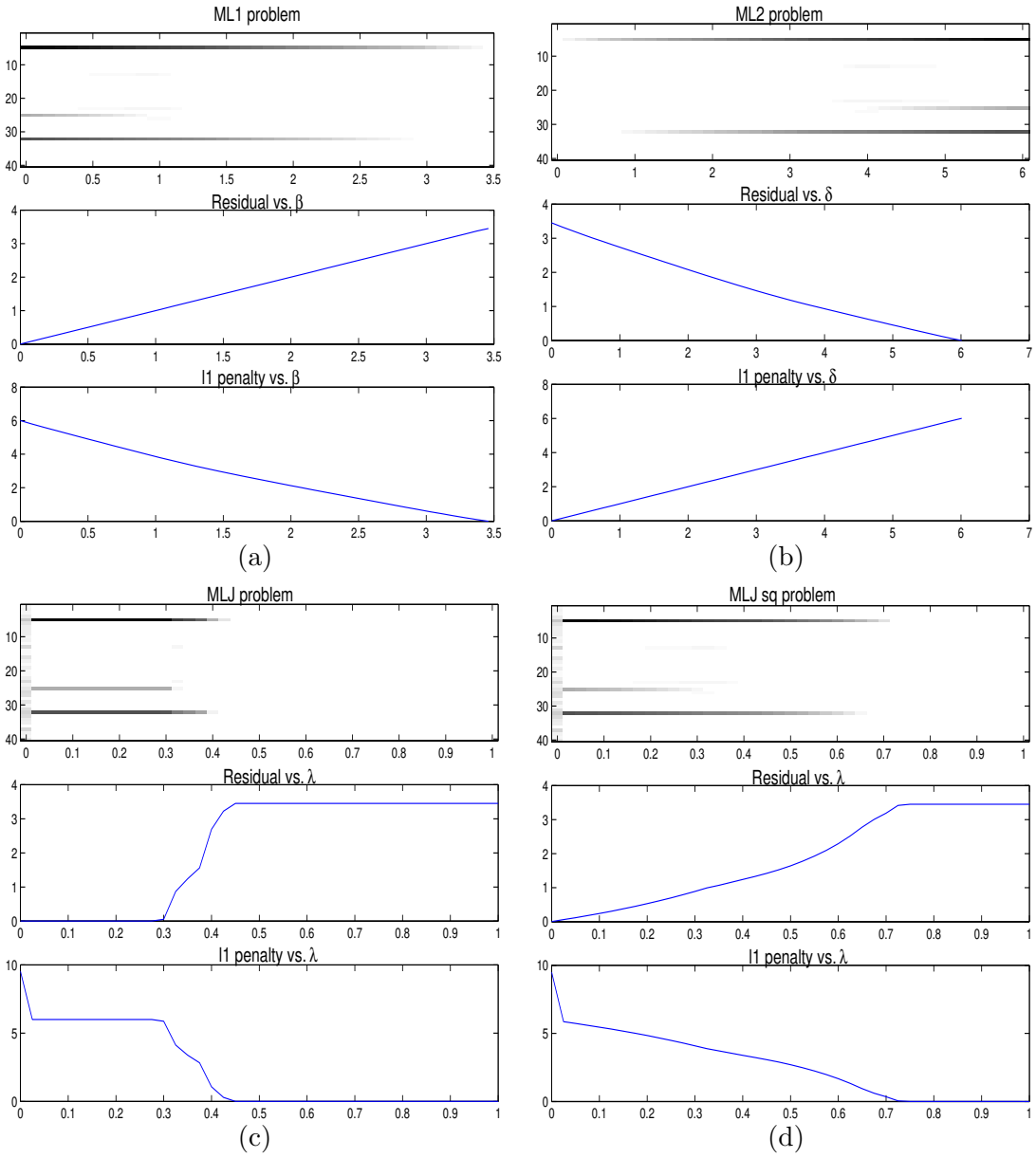



**Figure 4.5.** Versions of $\ell_1$. Top: Solutions vs. reg param, Middle: $\|\mathbf{x}\|_1$, Bottom: $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$. (a) ML1 (b) ML2 (c) MLJ (d) MLJSQ

may or may not be useful practically, but are definitely interesting from a theoretical perspective. Some of the questions include: the nature of the maximum of the curve $\lambda\|\mathbf{x}\|_1 + (1-\lambda)\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$, also whether the curve has properties such as smoothness, concavity, etc. Next, in order to better understand the dependence of the solutions of MLJSQ on $\lambda$ we solve a small problem analytically using nonsmooth optimization

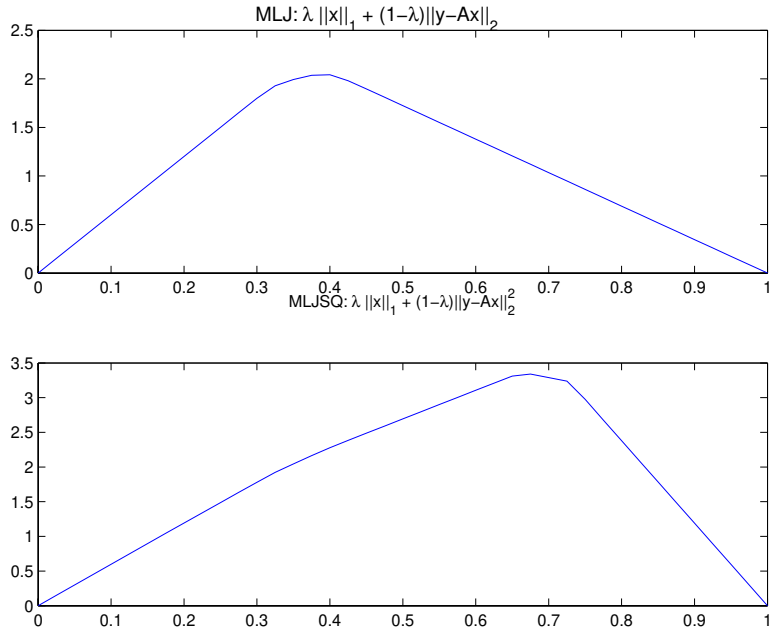



**Figure 4.6.** Top: MLJ, $\lambda\|\mathbf{x}\|_1 + (1-\lambda)\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2$ vs. $\lambda$. Bottom: MLJSQ, $\lambda\|\mathbf{x}\|_1 + (1-\lambda)\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ vs. $\lambda$.

theory.

### ■ 4.1.6 Analytical solution of a small problem

We consider a small real-valued instance of the MLJSQ problem (4.6), for which we can find an analytic form of the solution using optimality conditions for convex non-differentiable unconstrained optimization. Refer to Appendix C for a short overview of the mathematics involved. The value of considering this problem is the insight that it gives on the dependence of the solutions of MLJSQ on $\lambda$. Let

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 1.5 \end{pmatrix}, \text{ and } \quad y = \begin{pmatrix} 6 \\ 6 \end{pmatrix}$$

The MLJSQ problem is

$$\min \quad f(\mathbf{x}) \quad = \quad \min \ \|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 \tag{4.33}$$

This is an unconstrained convex minimization problem. The first term, $\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2$ is convex, and the second term $\lambda\|\mathbf{x}\|_1$ is also convex, thus the total cost function is convex. In fact, it has one global optimum $\forall\ \lambda > 0$. The difficulty with the optimality conditions for this problem lies in the fact that $\|\mathbf{x}\|_1$ is not differentiable at 0. Thus we have to use the optimality conditions from non-smooth convex optimization. They

state that the subdifferential of $f$ at $\mathbf{x}$ has to contain the $\mathbf{0}$-vector for $f$ to achieve a global minimum at $\mathbf{x}$. Appendix C reviews the terminology.

The subdifferential of $g(\mathbf{x}) = \|\mathbf{x}\|_1$ is the following set:

$$\partial g = \left\{ \mathbf{u} \text{ such that } \begin{cases} u_i = 1 & \text{if } x_i > 0 \\ u_i = -1 & \text{if } x_i < 0 \\ u_i \in [-1, .., 1] & \text{if } x_i = 0 \end{cases} \right\} \tag{4.34}$$

The interesting part of this subdifferential is when some of the coordinates are equal to 0, where $g$ is non-differentiable.

The subdifferential of the MLJSQ function, $f = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$ is the set

$$\partial f = \left\{ 2\mathbf{A}'(\mathbf{A}\mathbf{x} - \mathbf{y}) + \lambda \mathbf{u}(\mathbf{x}) \right\} \tag{4.35}$$

where $\mathbf{u}(\mathbf{x})$ is defined above in (4.34). For our example this translates into:

$$\partial f = \left\{ 2 \begin{pmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 1.5 \end{pmatrix} \left[ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 1.5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \begin{pmatrix} 6 \\ 6 \end{pmatrix} \right] + \lambda \begin{pmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \end{pmatrix} \right\} \tag{4.36}$$

Note that the product $\mathbf{A}'\mathbf{A}$ is rank-2, thus the first row can be expressed as a linear combination of the other two. The coefficients are 1/4 and 1/6. Thus rewriting the equation we get:

$$\partial f = 2 \begin{pmatrix} 1/4(5x_1 + 13x_2 + 10.5x_3 - 30) + 1/6(4.5x_1 + 10.5x_2 + 11.25x_3 - 27) \\ 5x_1 + 13x_2 + 10.5x_3 - 30 \\ 4.5x_1 + 10.5x_2 + 11.25x_3 - 27 \end{pmatrix} + \lambda \begin{pmatrix} u(x_1) \\ u(x_2) \\ u(x_3) \end{pmatrix} \tag{4.37}$$

The optimality condition states that $\mathbf{0} \in \partial f$ at the optimum. Since we do not know which indices of $\mathbf{x}$ are positive, negative or zero, in general it is necessary to try out all the cases, and see which ones have a non-empty solution set. For example, if $\lambda > 0$, then the case when all $x_i > 0$ does not have solutions: otherwise all $u(x_i) = 1$, and if the second and the third rows are equal to zero, then the first one is $7\lambda/12$. Thus $\lambda = 0$, and the set of solutions is $\begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix} + \mathbf{n}$, where $\mathbf{n} \in Null(\mathbf{A})$, any vector in the nullspace of $\mathbf{A}$.

The regions which do have solutions for some $\lambda > 0$ are $0 \times \mathbb{R}^+ \times \mathbb{R}^+$, and $0 \times \mathbb{R}^+ \times 0$. For the first region we have:

$$\begin{pmatrix} 2(\frac{1}{4}p + \frac{1}{6}q) + \lambda a \\ 2p + \lambda \\ 2q + \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.38}$$

where $p = 5x_1 + 13x_2 + 10.5x_3 - 30$, and $q = 4.5x_1 + 10.5x_2 + 11.25x_3 - 27$. Thus $p = q = -\lambda/2$, and $a = -5/12$. Solving for $x_2$ and $x_3$ as a function of $\lambda$, we get $x_2 = \frac{3}{2} - \frac{\lambda}{96}$, and $x_3 = 1 - \frac{5\lambda}{144}$. The solutions satisfy the assumed positivity constraints as long as $0 < \lambda \leq 144/5 = 28.8$.

Similarly, for the second region, $x_2 = \frac{30}{13} - \frac{\lambda}{26}$, and $28.8 \leq \lambda \leq 60$. For $\lambda \geq 60$ the solution is $\mathbf{x} = \mathbf{0}$. We illustrate the solution path as a function of $\lambda$ in Figure 4.7. The two bold triangles are the two planes corresponding to the two rows of $\mathbf{A}$. The nullspace of $\mathbf{A}$ goes through $\mathbf{0}$ parallel to their intersection, the line connecting $(6, 0, 0)$, and $(0, 1.5, 1)$. When $\lambda$ goes from zero to a very small value, the solution to the problem jumps right away to $(0, 1.5, 1)$, which is the point satisfying $\mathbf{y} = \mathbf{A}\mathbf{x}$ with minimum $\|\mathbf{x}\|_1$. Whenever $\lambda > 0$, no matter how small, the optimum solution no longer satisfies $\mathbf{y} = \mathbf{A}\mathbf{x}$, but is still very close to the line where this is true. The broken two-segment line originating from $(0, 1.5, 1)$, and ending at $(0, 0, 0)$ is the solution path as a function of $\lambda$.

When instead we consider the convex combination formulation with $(1 - \lambda)$ in front of the $\ell_2$ norm of the residuals, we have a very similar situation[7]. Now instead of being a function of $\lambda$, the coordinates of the optimal solution are a function of $\frac{\lambda}{1-\lambda}$. The intervals for the new $\lambda$ for which the optimal solution falls into the two regions that we considered above are $(0, 0.9664]$, and $[0.9664, 0.9836]$. For $\lambda \geq 0.9836$, the optimal solution is $\mathbf{0}$. For comparison, the solution obtained using the pseudo-inverse is $\mathbf{x}_{PINV} = [78/157, 216/157, 144/157]' \approx [0.4968, 1.3758, 0.9172]'$. In contrast to $\ell_1$ solutions with appropriate $\lambda$, the pseudo-inverse solution is not sparse.

This small example is not useful for practical purposes; yet it demonstrates some of the properties of the solution set that are useful to know for practical $\ell_1$ optimization problems described in previous sections. For example, the first formulation (the one without the $(1-\lambda)$ term) always leads to piecewise linear solutions paths, and whenever $\lambda \geq 0$, the solution does not satisfy $\mathbf{y} = \mathbf{A}\mathbf{x}$. Similar analysis can be done for the MLJ problem. This will allow to compare virtues of the two formulations analytically.

### ■ 4.1.7 Sign patterns of solutions, noiseless version

We conclude the part of this chapter devoted to $\ell_1$ optimization by describing a very interesting observation that we have made regarding the optimal solutions to the noiseless $\ell_1$ penalization, (4.2). We consider the case when the matrix $\mathbf{A}$ and the level of sparsity of the signals do not guarantee that the underlying signal can be perfectly recovered [8]. However, exact reconstructions can still be obtained for a subset of such signals. It appears that the signals which do yield correct reconstructions using noiseless $\ell_1$-penalty minimization can be determined *apriori*, by looking at a finite number of test cases and their sign patterns. The observation applies to the real-valued case ($\mathbf{x}$ is real-valued, and $\mathbf{A}$ may still be complex). For complex $\mathbf{x}$ case no extensions have

[7] The reason to introduce $(1 - \lambda)$ is to limit the possible set of $\lambda$ to the interval $[0, 1]$. Otherwise an upper bound for $\lambda$ is hard to find.

[8] These conditions are described in Chapter 7.

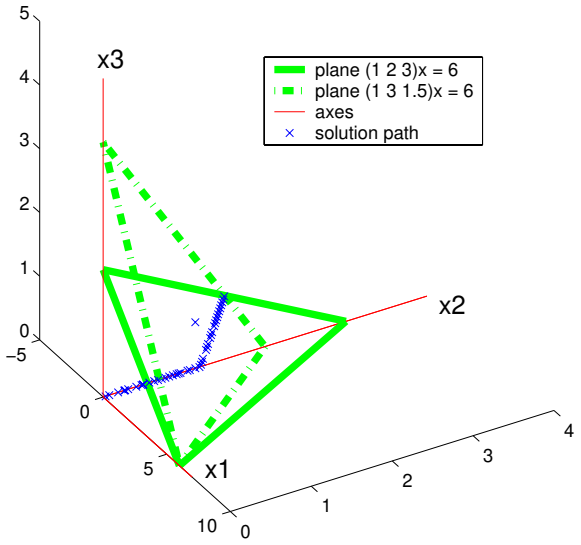



**Figure 4.7.** Solution path $\hat{\mathbf{x}}(\lambda)$. Two bold triangles are the two planes corresponding to the rows of $\mathbf{A}$. The solution path is the broken line that goes to $(0, 0, 0)$.

been found so far.

Define the sparsity profile as the set of nonzero indices of $\mathbf{x}$. Also define the sign pattern of $\mathbf{x}$ as the vector $\mathbf{s}$ with elements equal to $+1$ for positive indices of $\mathbf{x}$, $-1$ for negative indices, and 0 otherwise. Thus $s_i = \begin{cases} 1, & \text{if } x_i > 0 \\ -1, & \text{if } x_i < 0 \\ 0, & \text{if } x_i = 0 \end{cases}$. We also normalize $\mathbf{s}$ so that the first non-zero index is equal to 1 (by multiplying by $-1$ if necessary). This is done to remove an ambiguity, since negating $\mathbf{x}$ leads to the negation of $\mathbf{y}$, with no effect on the $\ell_1$ norm and the residual.

Let an overcomplete $\mathbf{A}$ be fixed, and $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{x}$ is not sparse enough to allow exact solutions using the $\ell_1$ approach. (If it is sparse enough then all signals allow exact reconstructions). We try to find $\mathbf{x}$ given $\mathbf{A}$ and $\mathbf{y}$. We have empirically discovered the following property: if $\mathbf{x}$ having a specified sparsity profile has a particular sign pattern, and the $\ell_1$ reconstruction uncovers it exactly, then any $\tilde{\mathbf{x}}$ with the same sparsity profile, and the same sign pattern will also produce the correct result under $\ell_1$ optimization. Similarly, if $\mathbf{x}$ leads to an incorrect reconstruction, then all $\tilde{\mathbf{x}}$ with the same sign pattern and sparsity profiles will not produce an exact answer. That means that the property of allowing exact reconstructions using $\ell_1$ penalization is invariant under multiplying each element of $\mathbf{x}$ with arbitrary positive scalars.

For a fixed sparsity profile we can run $\ell_1$ optimization for signals $\mathbf{x}$ with all possible

sign patterns, and make a note whether they resulted in exact answers or not. Then we can determine whether any signal $\mathbf{x}$ will lead to an exact reconstruction simply by looking at its sign pattern. Making a loop over all possible sparsity profiles we will be able to classify recoverability of all possible signals.

To illustrate this behavior, we include several examples with $\mathbf{A}$ a random basis (each element being an independent realization of a standard normal random variable with mean 0, and variance 1). We illustrate the case when $\|\mathbf{x}\|_0^0 = 3$, but the observation holds for any number of non-zero elements of $\mathbf{x}$. (We have conducted a large number of experiments to test this surprising behavior, so although we do not have a theoretical proof, we have high confidence in its veracity). The sparsity profile (the support of $\mathbf{x}$) is $(10, 20, 30)$. The maximum number of non-zero elements which is allowed to have exact solutions using $\ell_1$ penalization for this realization of $\mathbf{A}$ is less than three. Some of the signals are reconstructed exactly, while others are not.

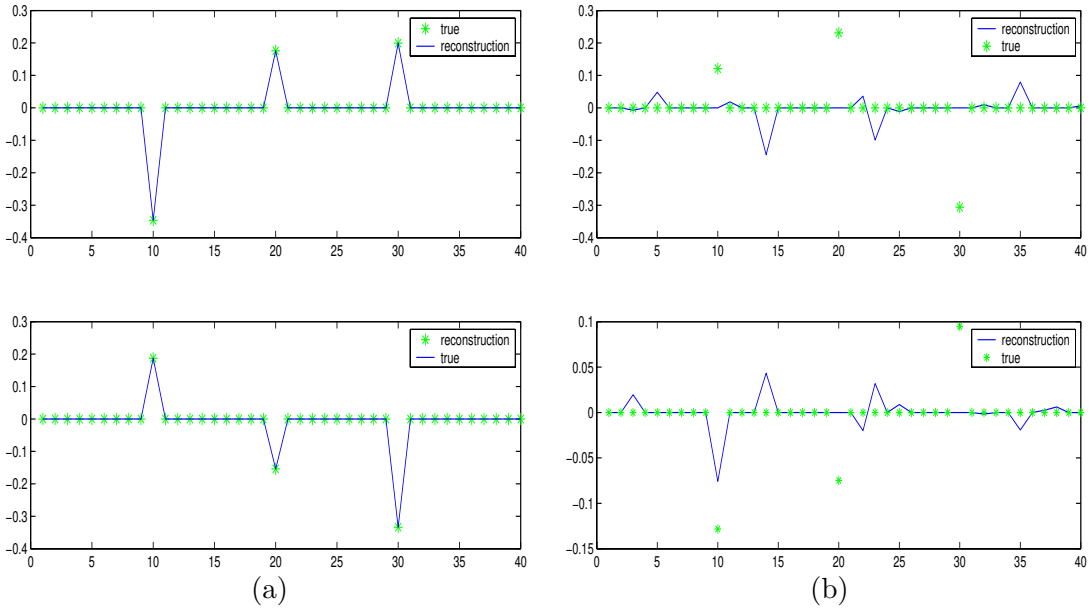



**Figure 4.8.** Sign patterns of exact solutions. Sparsity profile (support of $\mathbf{x}$) is (10, 20, 30). (a) Sign pattern for exact solutions: $(+,-,-)$. Two signals along with exact reconstructions. (b) Sign pattern for wrong solutions: $(+,+,-)$. Two signals, and the corresponding wrong reconstructions.

For the specific case that we considered the sign pattern that yields correct reconstruction is $(+,-,-)$, and the other sign patterns $(+,+,-)$, $(+,-,+)$, and $(+,+,+)$ yield incorrect reconstructions ($+$ and $-$ correspond to $+1$ and $-1$ respectively). In Figure 4.8 we plot the original signal, $x_i$, and its reconstruction, $\hat{x}_i$, vs. $i$, for $i \in \{1,..,40\}$. We include two examples of signals with sign patterns that lead to exact solutions in plot (a), and two examples of signals with sign patterns that do not get exact answers in plot(b). The first sign pattern is $(+,-,-)$, and the second is $(+,+,-)$.

Considering a random basis is not representative of other bases, in particular those that are associated with our source localization problem. In general, the fact that a particular solution does not yield the reconstruction exactly does not mean that the reconstruction carries no information about the underlying $\mathbf{x}$. For our source localization application overcomplete bases $\mathbf{A}$ have a strong structure (columns of $\mathbf{A}$ are samples from a parameterized manifold). Nearby columns of $\mathbf{A}$ (in terms of their index) are also very close in terms of the Euclidean distance. When the solution is not exact, it is usually very close in terms of the column number to the exact one. The solution is still very useful even if it does not lead to the exact true answer. We describe this more in Chapters 6 and 7. Nevertheless, it is definitely useful to know when we obtain exact solutions.

## ■ 4.2 $\ell_p$ Regularization

The numerical method for noisy $\ell_p$ optimization,

$$\min \ J(\mathbf{x}) = \min \ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p. \tag{4.39}$$

that we present in this section was developed by Cetin in [11], and some previous work was done in [39]. The method has very good performance when $p < 1$, (comparable to that of $\ell_1$). The use of $\ell_p$ techniques with $p < 1$ has been limited in the literature due to the many challenges involved.

The $\ell_p$-norm is not convex for $p < 1$, (and, even its level sets are not convex) so we are not guaranteed to achieve global minima by local optimization methods. Instead of turning to computationally very demanding global optimization methods, we choose to stay with local optimization methods, but rely on having a good starting point [9].

The use of $\ell_p$ norms with $p < 1$ to enforce sparsity can be justified in the same way as we did it for $\ell_1$. In fact, when $p < 1$ the necessary conditions for the global optimum of the noiseless $\ell_p$ problem ($\min \|\mathbf{x}\|_p^p$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$) to be equal to the global optimum of the $\ell_0$ problem ($\min \|\mathbf{x}\|_0^0$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$) are less stringent (see Section 7.3). However, finding the global solution of the noiseless $\ell_p$ problem for $p < 1$ is more difficult than for the case when $p = 1$ (the noiseless $\ell_p$ problem is non-convex as well). Our theoretical results about $\ell_p$ regularization in Chapter 7 are based on global optima and cannot be directly applied to local solutions obtained by the numerical method we describe here. We skip the noiseless version and go directly to the noisy formulation, $J(\mathbf{x}) = \min \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_p^p$.

Apart from non-convexity, the other difficulty arising with using the $\ell_p$ cost function, is that it is not differentiable at $\mathbf{0}$. Actually, the cost functions are not differentiable for both $p = 1$ and $p < 1$, but for $p = 1$ there is a way to go around this difficulty: in the

[9] Looking ahead, a good starting point in the context of array processing does not mean that we have to have the peaks corresponding to the sources resolved (which is necessary for maximum likelihood methods). We get away with the beamforming solution and sometimes even with a non-zero constant signal as a good starting point. More information on this topic can be found in Chapter 5.

real data case we look at $\mathbf{x}^+$ and $\mathbf{x}^-$, as defined in Section 4.1.1, and get an equivalent linear programming problem; in the complex case we use SOC representation. For $p < 1$ this is not possible, and instead we consider differentiable approximations of the $\ell_p$ cost function; one might use such approximations for the $\ell_1$ cost as well. Differentiable approximations typically have a parameter which controls the trade-off between the smoothness of the approximation and the closeness to the non-differentiable function which is being approximated. In other words, a differentiable approximation is a family of functions.

For $p = 1$, a well-known approximation is:

$$\|\mathbf{x}\|_1 \approx \sum_{i=1}^{N} \rho(x_i), \quad \text{where} \tag{4.40}$$

$$\rho(x) = \begin{cases} x_i^2, & \text{if } |x_i| < \gamma \\ |x_i| - \gamma + \gamma^2, & \text{if } |x_i| \geq \gamma \end{cases} \tag{4.41}$$

where $\gamma$ is the parameter. As $\gamma \to 0$, the approximation converges uniformly to the $\ell_1$ norm. However, for any $\gamma > 0$, it is differentiable. This approximation can actually be reformulated as a quadratic program (or SOC in complex case) [40]. However, we are much more interested in the case where $p < 1$. Another approximation [39] which works for $p < 1$ as well as for $p = 1$ is:

$$\|\mathbf{x}\|_p^p \approx \sum_{i=1}^{N} (|x_i|^2 + \epsilon)^{p/2} \tag{4.42}$$

Now $\epsilon \geq 0$ is the smoothing parameter. This differentiable approximation suits our purposes well, and we get the following modified cost function:

$$J(\mathbf{x}) \approx J_\epsilon(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \sum_{i=1}^{N} (|x_i|^2 + \epsilon)^{p/2} \tag{4.43}$$

Note that $J_\epsilon(\mathbf{x}) \to J(\mathbf{x})$ as $\epsilon \to 0$. For our applications we usually set $\epsilon$ to be a small constant. The minimization of $J_\epsilon(\mathbf{x})$ does not yield a closed-form solution in general, so numerical optimization techniques must be used.

For the solution of this optimization problem, Cetin [11] uses the half-quadratic regularization method of [41], which was briefly mentioned in Chapter 3. Half-quadratic regularization converts a non-quadratic optimization problem into a series of quadratic problems. We skip the details, since for our purposes it suffices to view the algorithm as a quasi-Newton's method. However, the proof of local convergence from any starting point that we refer to later is based on the half-quadratic roots of the algorithm.

We use the following iterative algorithm:

$$\mathbf{H}\left(\hat{\mathbf{x}}^{(n)}\right)\ \hat{\mathbf{x}}^{(n+1)} = 2\mathbf{A}^H\mathbf{y} \tag{4.44}$$

where $n$ denotes the iteration number, and:

$$\mathbf{H}(\mathbf{x}) \triangleq 2\mathbf{A}^H\mathbf{A} + \lambda\Lambda(\mathbf{x}) \tag{4.45}$$

$$\Lambda(\mathbf{x}) \triangleq \text{diag}\left\{\frac{p}{(|x_i|^2+\epsilon)^{1-p/2}}\right\}$$

where diag$\{\cdot\}$ is a diagonal matrix whose $i$-th diagonal element is given by the expression inside the brackets. The important difference from methods such as ridge regression [42] lies in the dependence of $\mathbf{H}$ on the iterates through $\Lambda(\mathbf{x})$, which is a spatially varying penalty.

The method can also be interpreted as a quasi-Newton method with Hessian approximation $\mathbf{H}$. The full Hessian is given by

$$\nabla_{\mathbf{xx}}J_\epsilon = \mathbf{H}(\mathbf{x}) + \lambda\text{diag}\left\{\frac{p(p-2)|x_i|^2}{(|x_i|^2+\epsilon)^{2-p/2}}\right\} \tag{4.46}$$

For $p < 2$ the second term is always negative and may make the Hessian indefinite. By keeping the first part only, we get a positive definite approximation to the Hessian. Also, for $p = 2$, the approximation becomes exact.

The quasi-Newton iteration has the following form:

$$\hat{\mathbf{x}}^{(n+1)} = \hat{\mathbf{x}}^{(n)} - \beta\mathbf{H}\left(\hat{\mathbf{x}}^{(n)}\right)^{-1}\nabla_{\mathbf{x}}J_\epsilon\left(\hat{\mathbf{x}}^{(n)}\right) \tag{4.47}$$

By choosing stepsize $\beta$ to be equal to 1, a cancellation of the terms in the right hand side reduces this iteration to (4.44). The gradient of $J_\epsilon(\mathbf{x})$ is

$$\nabla_{\mathbf{x}}J_\epsilon = 2\mathbf{A}^H(\mathbf{A}\mathbf{x}-\mathbf{y}) + \lambda\ \text{vec}\left\{\frac{px_i}{(|x_i|^2+\epsilon)^{1-p/2}}\right\} = \tag{4.48}$$

$$= (2\mathbf{A}^H\mathbf{A}\mathbf{x} + \lambda\text{diag}\left\{\frac{p}{(|x_i|^2+\epsilon)^{1-p/2}}\right\})\mathbf{x} - 2\mathbf{A}^H\mathbf{y} = \mathbf{H}(\mathbf{x})\mathbf{x} - 2\mathbf{A}^H\mathbf{y}. \tag{4.49}$$

Hence, we get

$$\hat{\mathbf{x}}^{(n+1)} = \hat{\mathbf{x}}^{(n)} - \mathbf{H}\left(\hat{\mathbf{x}}^{(n)}\right)^{-1}\nabla_{\mathbf{x}}J_\epsilon\left(\hat{\mathbf{x}}^{(n)}\right) = \tag{4.50}$$

$$2\mathbf{H}\left(\hat{\mathbf{x}}^{(n)}\right)^{-1}\mathbf{A}^H\mathbf{y}. \tag{4.51}$$

which leads to (4.44).

We run the iteration in (4.44) until $\frac{\|\hat{\mathbf{x}}^{(n+1)}-\hat{\mathbf{x}}^{(n)}\|_2^2}{\|\hat{\mathbf{x}}^{(n)}\|_2^2} < \delta$, where $\delta > 0$ is a small constant. Compared to standard optimization tools, the above scheme yields an efficient method matched to the structure of our optimization problem. Convergence properties of algorithms of this type have been analyzed, and convergence from any initialization to a local minimum is guaranteed [43, 44].

### ■ 4.2.1 Solution of positive definite linear systems

The solution of the linear equation (4.44) at each iteration has not been specified. It is of course possible to solve the equation by plain Gaussian elimination, but we can do a little bit better. The two components of $\mathbf{H}\left(\hat{\mathbf{x}}^{(n)}\right)$ are both symmetric and the sum is positive definite (p.d.): $2\mathbf{A}^H\mathbf{A}$ is p.d. when $\mathbf{A}$ is invertible (or positive semidefinite when this is not the case), and $\lambda\Lambda(\mathbf{x})$ is a diagonal matrix with positive elements, hence p.d. Thus instead of Gaussian elimination, we can replace it with Cholesky decomposition followed by the solution of two linear equations with upper and lower triangular matrices. The Cholesky decomposition of a positive definite matrix is $\mathbf{Q} = \mathbf{G}^H\mathbf{G}$, where $\mathbf{G}$ is upper triangular. Hence to solve $\mathbf{Q}\mathbf{x} = \mathbf{y}$, we solve $\mathbf{G}^H\mathbf{z} = \mathbf{y}$, and $\mathbf{G}\mathbf{x} = \mathbf{z}$. This results in some savings, but the procedure is still of order of $N^3$.

More important savings come when we realize that the algorithm in (4.44) is iterative, and so each iteration does not have to be solved exactly. This allows us to use faster approximate methods for the solution of the linear system. We outline the idea of one of the most prominent iterative methods, Preconditioned Conjugate Gradients in Appendix D. In our implementation of $\ell_p$ optimization, we use the conjugate gradient algorithm at each iteration.

Chapter 5

# Sparse-Regularization Framework for Source Localization

Now comes the time to unveil the practical power of inverse problems with sparsity regularization, by explaining how to apply them to the source localization problem. In Chapter 4 we described a family of procedures for solving the ill-posed inverse problem $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}$, with $\mathbf{A}$ overcomplete, using the knowledge that $\mathbf{s}$ is sparse. The source localization problem for array processing does not immediately come in this form. We described the narrowband model in Chapter 2. The basic model is[1]

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t) + \mathbf{n}(t),\ t \in \{1, .., T\} \tag{5.1}$$

where $\boldsymbol{\theta} = [\theta_1, ..., \theta_K]$ is the vector of unknown source locations. It is clear that some work has to be done to transform the problem into the sparse regularization framework, since the dependence of $\mathbf{y}$ on $\boldsymbol{\theta}$ is nonlinear, and the matrix $\mathbf{A}(\boldsymbol{\theta})$ is unknown. We start by describing how to mold the source localization problem for a single time sample (i.e. $T = 1$) into our sparse regularization framework in Section 5.1.1.

When multiple samples are present (and usually $T \gg 1$), the problem becomes more interesting. A naive approach is to treat each time sample independently, which is discussed in Section 5.1.3. Using the time samples independently has many drawbacks, and we investigate several possibilities for combining the time samples prior to solving the inverse problem.

A simple solution comes for the practically important case where $\mathbf{u}(t)$ has a strong non-zero temporal mean. These kinds of problems show up in localization of moving vehicles. In this case data for multiple time samples can be combined by taking a temporal average. Section 5.1.4 furnishes the details. For other array processing applications $\mathbf{u}(t)$ may be a zero-mean random process. A prime example is wireless communications. If the signal $\mathbf{u}(t)$ has components which are not correlated or just weakly correlated amongst themselves, there is an approximate way to combine the time samples by looking at a beamspace representation of the data. We describe it in full detail in Section 5.1.5.

[1] We may use an alternative time indexing where appropriate, $\{t_1, .., t_T\}$, when we wish to stress time dependence. In that case, we mean that $t_1 = 1$, $t_2 = 2$, .., $t_T = T$.

If the sources are zero-mean and correlated, then neither of these two ways of combining the data performs satisfactory. An approach to combine multiple time samples that does not make any assumptions on the sources is to merge the subproblems for different time samples into a single larger inverse problem. However, if the number of time samples is large, then the amount of computation becomes significant. More information is available in Section 5.1.6.

The most promising general approach which lowers dramatically the computational effort of the previous approach is obtained through the use of the singular value decomposition (SVD) of the matrix of the observed array outputs. Again, all the data are combined into one inverse problem. The SVD version is reasonable computationally and can handle all of the types of $\mathbf{u}(t)$ as well. Section 5.1.7 deals with this approach.

This summarizes all the different cases of narrowband farfield model that we have considered. The narrowband model applies to both the farfield and the nearfield cases. We mention the latter in Section 5.1.8.

The wideband model differs from that of narrowband due to the fact that time delays can no longer be represented as simple phase shifts. Instead, the signal spectrum is partitioned into several narrow frequency ranges, where the narrowband model applies. There are several possibilities of what to do with the multiple frequency subproblems, and we explore them in Section 5.2.

In Section 5.3 we describe how to get rid of the effects of the grid by using an adaptive grid refinement procedure. And in Section 5.4 we propose a method for the selection of the regularization parameter based on the well-known discrepancy principle from inverse problems.

## ■ 5.1 Narrowband problem

### ■ 5.1.1 Representation for one time sample

For simplicity we first consider the case where we have a single time sample of data at the sensors, or $T = 1$ in (5.1). Even for one time sample the problem is not in the linear inverse form of (3.7), because the forward operator $\mathbf{A}(\boldsymbol{\theta})$ depends on the unknown source locations $\boldsymbol{\theta}$. In order to go around this difficulty, we introduce an overcomplete representation $\mathbf{A}$ in terms of all possible source locations. Let $\{\tilde{\theta}_1, ..., \tilde{\theta}_{N_\theta}\}$ be a sampling grid of all source locations of interest. In the farfield case, $\tilde{\theta}_n$'s are scalars representing the directions of arrival (DOA), whereas in the nearfield case, $\tilde{\theta}_n$'s are vectors containing range and bearing information. The forward operator takes the following form: $\mathbf{A} = [\mathbf{a}(\tilde{\theta}_1), \mathbf{a}(\tilde{\theta}_2), ..., \mathbf{a}(\tilde{\theta}_{N_\theta})]$.

We represent the signal field by an $N_\theta \times 1$ vector $\mathbf{s}(t)$, where the $n$-th element $s_n(t)$ is nonzero and equal to $u_k(t)$ if source $k$ comes from $\tilde{\theta}_n$, for some $k$. For a single time sample the problem is reduced to

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}, \tag{5.2}$$

The $M \times N_\theta$ matrix $\mathbf{A}$ is composed of steering vectors corresponding to each poten-

tial source location as its columns. To emphasize, $\mathbf{A}$ differs from the steering matrix representation $\mathbf{A}(\boldsymbol{\theta})$ used in many array processing methods in the sense that it contains steering vectors for *all* potential locations, rather than only the (unknown) source locations. Hence, in our framework $\mathbf{A}$ is known and does not depend on the actual source locations. The reason behind using such redundancy in the representation is our desire to formulate the problem in a sparse signal reconstruction framework, which we discussed in Section 3.2. Our matrix $\mathbf{A}$ in this terminology is an overcomplete basis, where each basis vector corresponds to an array manifold vector for a sampling grid of locations.

As in numerous non-parametric source localization techniques, our approach consists of forming an estimate of the signal energy as a function of location, which ideally contains dominant peaks at the source locations. We need to obtain an estimate of the signal field $\mathbf{s}$ through the sensor observations $\mathbf{y}$, by solving (5.2) which is an ill-posed inverse problem, as we describe in Chapter 3. The central assumption in our approach is that the sources can be viewed as point sources, and their number is small. With this assumption the underlying spatial spectrum is sparse. Thus we can solve this inverse problem via regularizing it by favoring sparse signal fields using the methodology developed in Chapter 4. We can use any of the noisy $\ell_1$ methods (ML1, ML2, MLJ, MLJSQ), or the noisy $\ell_p$ method, (4.39), to solve the problem in (5.2). A discussion of which of these procedures to choose appears in Section 6.1. Apart from the case where ML1 is preferred for automatic regularization parameter selection (see Section 5.4.2) in our experience the choice of a procedure is not a critical issue.

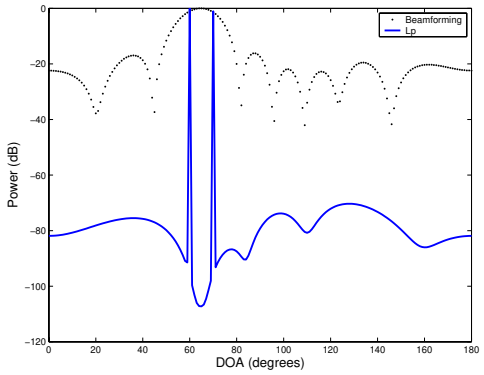



**Figure 5.1.** Single sample source localization with $\ell_p$: spatial spectra of two sources with DOAs of $60^\circ$ and $70^\circ$, (SNR = 20 dB).

We present an example of using $\ell_p$ regularization with $p = 0.1$ for single sample source localization in Figure 5.1. We consider a uniform linear array of $M = 8$ sensors separated by half a wavelength of the actual narrowband source signals. We consider two narrowband signals in the far-field impinging upon this array from DOA's $60^\circ$

and 70°, which are closer together then the Rayleigh limit, and the SNR is 20 dB. The regularization parameter in this example is chosen by subjective assessment, but automatic alternatives can be used as discussed in Section 5.4. Using beamforming the two peaks of the spectrum are merged, but using our sparse regularization approach they are well resolved, and the sidelobes are suppressed almost to zero. Apart from the asymptotic bias, which we discuss in Section 6.3, this spectrum estimate is an example of what superresolution source localization methods aim to achieve.

### ■ 5.1.2 Treating multiple time samples

Single snapshot processing may have its own applications, but source localization with multiple snapshots is of greater practical importance. When we bring time into the picture, our overcomplete representation is easily extended. The general narrowband source localization problem with multiple snapshots reformulated using an overcomplete representation has the following form:

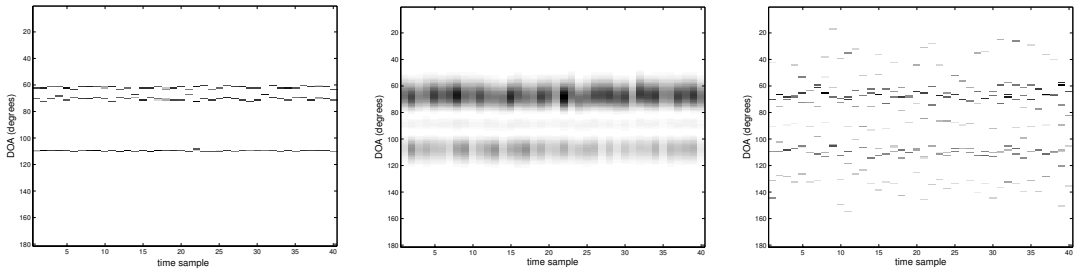
$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t),\ t \in \{1, ..T\} \tag{5.3}$$

Next few sections present how to deal with the availability of multiple time samples.

### ■ 5.1.3 Treating each time index separately

The first thought that comes to mind when we switch from one time sample to several time samples is to solve each problem indexed by $t$ separately. Each problem can be solved as if it was a single snapshot problem, in the exact same way as was done in Section 5.1.1. We will have a set of $T$ solutions, $\hat{\mathbf{s}}(t)$. There are several ways to get one representative estimate of source locations from them. Two simple possibilities include taking the mean and finding the peaks, or using one of the many schemes of clustering analysis. The computing effort for the whole task is dominated by the solution of the $T$ inverse problems, and is linearly proportional to $T$. This approach is very simple, and is especially useful in dynamic scenarios, where the locations of the sources are evolving with time.

In Figure 5.2 we continue with our experimental setup from Section 5.1.1, increasing the number of time samples to $T = 40$, and adding another source at 108°. In plot (a) little noise is added to the sensors, SNR = 20dB. We use $\ell_p$ regularization with $p = 0.1$ for each time sample. The plot shows the solutions of the subproblems for all of the $T$ snapshots as an intensity map. The horizontal axis is time, and the vertical one is the DOA. We clearly see three distinct lines around the correct DOAs, 60°, 70° and 108°. In plot (b) we conduct sample by sample processing using beamforming, and the result is blurry; the first two sources are not resolved, and sidelobes appear as spurious sources. Thus for high SNR sample by sample processing $\ell_p$ regularization is superior to beamforming. However, when we decrease the SNR to 3 dB, and again use $\ell_p$ processing, the variance of the estimates of source locations for each individual problem is too high, and it is not possible to distinguish the sources anymore.




(a) (b) (c)

**Figure 5.2.** Sample by sample processing: sources at $60^\circ$, $70^\circ$ and $108^\circ$. (a)$\ell_p$ processing with $p = 0.1$, SNR = 20 dB. (b) Sample by sample beamforming: SNR = 20 dB. (c) $\ell_p$ processing: SNR = 3 dB.

The method is appropriate at modest SNR for dynamic scenarios, but for stationary cases, where the locations of the sources do not change noticeably for a period of several time samples, it is not a good solution. The main drawback of treating each time sample separately lies in the fact that there is no cooperation between the subproblems for different time samples. For example changing the indices of support of $\mathbf{s}(t_1)$ to arbitrary values has no direct influence on the corresponding indices of support of $\mathbf{s}(t_2)$. The result is that the approach suffers from sensitivity to SNR. Small perturbations in data lead to small perturbations in solutions to individual problems, and we can expect to correct them when we combine solutions for different time samples. However, moderate to strong perturbations may lead to completely useless solutions of individual problems (only one time sample is available for each subproblem), and attempting to combine them turns out to be futile. This motivates us to consider combining the data for different time samples prior to solving inverse problems.

### ■ 5.1.4 Non-zero mean processing

A simple way to combine different time samples exists for the case where $\mathbf{u}(t)$ has a strong non-zero temporal mean, (the power spectrum at the 0-frequency dominates all other frequencies). These kinds of signals appear in a number of passive sensor array processing tasks, including the localization of moving vehicles with acoustic sensors. For example, tracks of a tank in motion produce noise containing strong harmonics, which upon demodulation take the required non-zero mean form. If $\mathbf{u}(t)$ has a strong non-zero temporal mean, then so does $\mathbf{s}(t)$ on the indices which correspond to the locations of the sources. This motivates combining the data by taking a temporal average: $\bar{\mathbf{y}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t)$, and similarly for $\bar{\mathbf{s}}$ and $\bar{\mathbf{n}}$. This allows us to look at just one problem:

$$\bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{s}} + \bar{\mathbf{n}} \tag{5.4}$$

instead of $T$ problems as in (5.3). Taking the mean instead of just taking one particular time instance (e.g. $t = 1$) reduces the variance of the noise by $T$. This is the simplest way to translate multiple snapshot source localization into sparse regularization. Once

we have (5.4), we are in the same situation as in the single time sample case, and we use exactly the same methods to solve the problem.

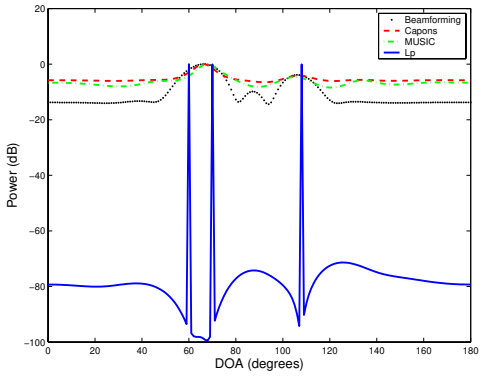



**Figure 5.3.** Non-zero mean signals: time sample combination by averaging followed by $\ell_p$ processing. Sources are at 60°, 70° and 108°, and SNR = 3 dB.

In Figure 5.3 we continue our experimental setup from the previous section. The sources are now non-zero mean, (they are modeled as $u_i(t) = 1 + \nu_i(t)$, where $\nu_i(t)$ are independent normal random variables with zero mean and standard deviation $\sigma = 0.2$). We combine different time samples by averaging and use $\ell_p$ regularization with $p = 0.1$. We compare our results against beamforming, Capon's and MUSIC spectra. We take $T = 200$ samples so that MUSIC and Capon's methods have good estimates of the sensor covariance matrix $\mathbf{R}$. The correct DOAs are again 60°, 70° and 108°. When we have a strong temporal mean our technique has an advantage over MUSIC and Capon's methods which are developed for zero-mean signals and are not able to take full advantage of the non-zero mean. In the plot it is clear that $\ell_p$ provides a better spectrum estimate than beamforming, MUSIC and Capon's methods. We present more simulations results for this scenario including bias and variance analysis in Chapter 6.

### ■ 5.1.5 Zero-mean beamspace processing

The transformation into beamspace domain [13, 45, 46] is a commonly used tool in array processing and in source localization in particular. It can be used to improve the computational complexity of source localization by reducing the dimensionality of the data, improve resolution, and reduce sensitivity to sensor position uncertainty [46]. The basic idea is to take the data at the sensors and to form beams (steer the array by applying appropriate weights) in several directions by using beamforming. These beams are the new data instead of the sensor outputs. There are many possibilities for how many beams to use, which steering directions to select, and whether or not to use the full array or subarrays for some of the beams. We do not go into these details.

We start by transforming the data into the beamspace domain. The steered beams

are sampled on a grid of locations, and the corresponding collection of manifold vectors is kept as columns in matrix $\mathbf{B}$. Then the beamspace data is:

$$\mathbf{z}(t) = \mathbf{B}^H \mathbf{y}(t) = (\mathbf{B}^H \mathbf{A})\mathbf{s}(t) + \mathbf{B}^H \mathbf{n}(t) \tag{5.5}$$

Next, we combine the squared amplitudes of the time samples of the beam outputs. A similar idea has been proposed in [13]. Denote the $(i,j)$-th entry of $\mathbf{B}^H\mathbf{A}$ by $v_{i,j}$, and $\mathbf{B}^H\mathbf{n}(t)$ by $\tilde{\mathbf{n}}(t)$. Then we have:

$$|z_i(t)|^2 = |\sum_j s_j(t) v_{i,j} + \tilde{n}_i(t)|^2$$

$$= \sum_j |s_j(t)|^2 |v_{i,j}|^2 + 2 \sum_{j_1 \neq j_2} Re[s_{j_1}(t)^* s_{j_2}(t) v_{i,j_1}^* v_{i,j_2}] + 2 \sum_j Re[s_j(t)^* v_{i,j}^* \tilde{n}_i(t)] + |\tilde{n}_i(t)|^2$$

When the sources are uncorrelated and have a zero temporal mean, and the noise is zero mean, the cross terms are all zero mean, and their temporal sums are negligible. Hence the average squared beamspace data $\overline{|z_i|^2} = \frac{1}{T}\sum_{t=1}^{T} |z_i(t)|^2$ can be represented well by a linear transformation of the element-by-element square of $\mathbf{B}^H\mathbf{A}$, denoted by $\mathbf{V}$, with average squared noise $\overline{|\tilde{n}_i|^2} = \frac{1}{T}\sum_{t=1}^{T} |\tilde{n}_i(t)|^2$ added:

$$\overline{|\mathbf{z}|^2} = \mathbf{V}\overline{|\mathbf{s}|^2} + \overline{|\tilde{\mathbf{n}}|^2} \tag{5.6}$$

The unknown in the equation is not $\mathbf{s}$ but rather $\overline{|\mathbf{s}|^2}$, since the peaks of the spectrum of $\overline{|\mathbf{s}|^2}$ appear for the same locations as for $\mathbf{s}$. If we let $\check{\mathbf{z}} = \overline{|\mathbf{z}|^2}$, and similar for $\check{\mathbf{s}}$, and $\check{\mathbf{n}}$ then we can write the equation (5.6) as $\check{\mathbf{z}} = \mathbf{V}\check{\mathbf{s}} + \check{\mathbf{n}}$. Now this equation is in the same form as (5.2), and can be solved by the same methods. For now we ignore the fact that the noise after our transformation is not Gaussian (it actually is one-sided), and use the same $\ell_2$ penalty $\|\check{\mathbf{z}} - \mathbf{V}\check{\mathbf{s}}\|_2^2$ for the data-fidelity term. Also, despite the assumptions of zero-mean and uncorrelatedness in the development, on simulated examples the spectra exhibit peaks in the vicinity of the correct source locations even when these conditions are not met exactly.

In Figure 5.4 we present simulation results for beamspace processing with $\ell_1$ regularized solution of the inverse problem (5.6). The three sources now have zero temporal mean. The number of time samples is $T = 200$. In the figure we compare the spectra of beamforming, Capon's and MUSIC methods against the spectrum obtained using $\ell_1$ regularization of the beamspace formulation. The correct DOAs are again 60°, 70° and 108°. The SNR is moderately high, 10 dB, so all the techniques except beamforming are able to resolve the three sources.

### ■ 5.1.6 Joint-time inverse problem

The methods of combining the samples in previous two sections (nonzero mean and beamspace) make assumptions on the type of signals that they can handle, and they work poorly when these assumptions are noticeably violated. We are interested in

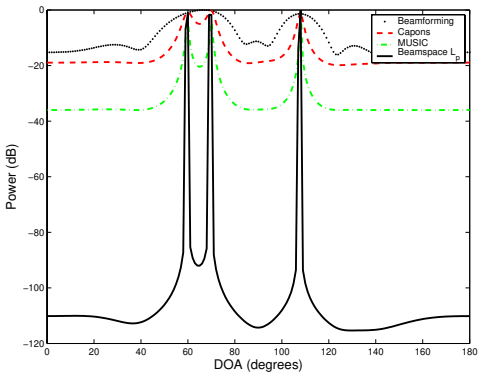



**Figure 5.4.** Beamspace processing of zero-mean uncorrelated sources with $\ell_1$: spatial spectra of three sources with DOAs of 60°, 70° and 108°, (SNR = 10 dB).

methods which on the one hand do not make any assumptions on the type of signals, and on the other hand use different time samples in synergy, unlike sample by sample processing. In this section we describe one method which meets both of these goals. The starting point for the method is to combine the multiple time-problems in (5.3) by stacking the data and signal vectors over time:

$$\begin{aligned}\breve{\mathbf{y}} &= [\mathbf{y}(t_1)',\ \mathbf{y}(t_2)', ..., \mathbf{y}(t_T)']',\\ \breve{\mathbf{s}} &= [\mathbf{s}(t_1)',\ \mathbf{s}(t_2)', ..., \mathbf{s}(t_T)']',\\ \breve{\mathbf{n}} &= [\mathbf{n}(t_1)',\ \mathbf{n}(t_2)', ..., \mathbf{n}(t_T)']'\end{aligned}$$

Then the matrix linking the data to the unknowns is block-diagonal, with copies of $\mathbf{A}$ repeated $T$ times:

$$\breve{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & & \\ & \mathbf{A} & \\ & & \ddots \\ & & & \mathbf{A} \end{pmatrix} \tag{5.7}$$

The resulting inverse problem takes the exact same form as (5.2):

$$\breve{\mathbf{y}} = \breve{\mathbf{A}}\breve{\mathbf{s}} + \breve{\mathbf{n}} \tag{5.8}$$

Without any additional thinking one might try to solve it in the same fashion as the previous problems, by imposing sparsity on $\breve{\mathbf{s}}$. This has the drawback that sparsity is enforced in both the spatial dimension and in time (sparsity of all the entries is penalized, and originally we have a 2-D grid of space and time). Enforcing sparsity in time domain in our case is not appropriate, but it may be useful for some particular types of signals, such as shot noise, which is itself sparse in time. Additionally, there is no constructive cooperation between the different subproblems, similarly to the processing in 5.1.3. Again, changing the indices of support of $\mathbf{s}(t_1)$ to arbitrary values has no direct

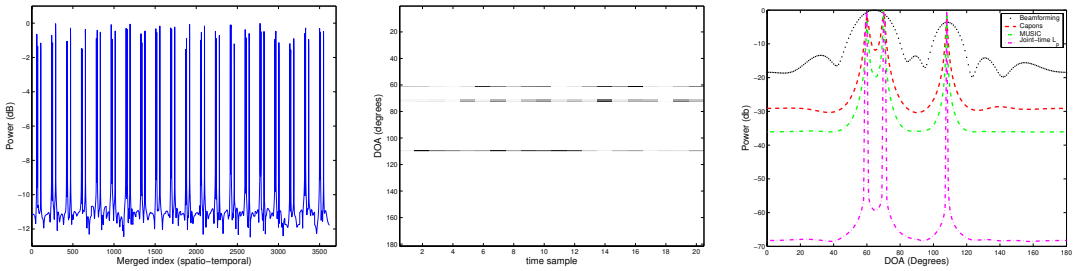



**Figure 5.5.** Joint-time processing with $\ell_1$: spatial spectra of three sources with DOAs of 60°, 70° and 108°, (SNR = 10 dB) (a) Full spatio-temporal spectrum as a vector . (b) Full spectrum as an image (horizontal axis is time, and vertical axis is space (DOA)). (c) Combined spectrum.

influence on the corresponding indices of support of $\mathbf{s}(t_2)$ (indices of support correspond to the estimates of the source locations).

A much better way to proceed is to impose a different prior, one that requires sparsity in the spatial dimension, but does not require sparsity in time. This can be done by first computing the $\ell_2$-norm of all time-samples of a particular spatial index of $\mathbf{s}$, e.g. $s_i^{(\ell_2)} = \|[s_i(t_1),\ s_i(t_2),\ ...,\ s_i(t_T)]\|_2$, and penalizing the $\ell_1$-norm of $\mathbf{s}^{(\ell_2)} = \left[s_1^{(\ell_2)}, ..., s_{N_\theta}^{(\ell_2)}\right]$. An $\ell_p$ norm can also be used instead of $\ell_1$. The cost function becomes

$$\|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 + \lambda\|\mathbf{s}^{(\ell_2)}\|_1 \tag{5.9}$$

This cost function is representable in SOC form. The optimization is performed over $\breve{\mathbf{s}}$; $\mathbf{s}^{(\ell_2)}$ is a function of $\breve{\mathbf{s}}$. The time samples are combined using the 2-norm which has no sparsifying effects. The spatial samples are combined using the $\ell_1$ norm which does enforce sparsity. This scheme will not prefer to have one time sample with most of energy to many time samples with energy equally distributed among them. Also, the different time-indices of $\mathbf{s}$ reinforce each other, since the penalty is much higher if the supports of $\mathbf{s}(t)$ for different $t$ do not line up exactly. Once $\breve{\mathbf{s}}$ is computed using the new cost function, it can be used to get the source location estimates by splitting $\breve{\mathbf{s}}$ into the corresponding $\mathbf{s}(t_i)$, and taking the mean over time, and finding its peaks (similar to sample by sample processing).

The main drawback of this technique is its computational cost. The size of the inverse problem increases linearly with $T$ and the computational effort required to solve it increases superlinearly with $T$. Thus when $T$ is large the approach is not viable for the solution of the real-time source localization problem.

In Figure 5.5 we include the reconstructions of the full spatio-temporal spectra $\breve{\mathbf{s}}$ (as a vector in plot (a) and as an image in plot (b)) as well as the agglomerated spectrum which uses simple averaging. The sources are zero mean, and we take few samples $T = 20$ to lessen the computational burden. The SNR is 10 dB. It can be seen that the resulting spectrum is very sharp. In the full spatio-temporal spectrum displayed as an

image clear advantages over the sample-by sample processing can be seen: this appears to be a better reconstruction then the one in Figure 5.2, plot (a), obtained by sample by sample processing for a *higher* SNR; peaks for all the time samples are aligned in the joint-time version. We use the information available in all the time samples to produce an estimate at each time point, so all the peaks are aligned. In addition, the spectrum is not sparse in the temporal domain since we explicitly made an adjustment to the cost function to avoid it.

### ■ 5.1.7 SVD-lp processing

The technique that we present in this section is based on the singular value decomposition (SVD) of the matrix of sensor output data, and it appears to be the overall best adaptation of the regularization framework to the source localization problem. The technique does not make any assumptions on the sources, and it is applicable to zero and non-zero mean signals, correlated and uncorrelated. The computational complexity is much lower than that involved in joint-time processing from last section; this is due to the fact that the dimension of the new problem does not increase with the number of time samples. Also, all time samples are combined together prior to solving the inverse problems unlike the sample by sample processing. We in fact present two versions of using the SVD for combining time samples. The first method is described mainly for historical purposes; it is simpler, but may have poor performance for a particular realization of the signals. We discuss why this poor performance occurs, and motivate the second version, which works very well and has consistent performance for all realizations. When we refer to $\ell_p$-SVD and $\ell_1$-SVD processing elsewhere in the thesis, we have in mind the second version described in this section.

The basic idea is to find a few vectors which summarize all the information (or at least as much as possible) about the cloud of points $\mathbf{y}(t),\ t \in \{1, .., T\}$. In the first version (the faulty one) we find just one vector which is a linear combination of the singular vectors of the sensor output matrix. In the second version we keep several singular vectors and combine them in the same fashion as was done for the joint-time problem of the last section.

#### Linear combination of the singular vectors

An interesting possibility for summarizing the sensor outputs is a linear combination of the left singular vectors of the sensor-data matrix $\mathbf{Y} = [\mathbf{y}(t_1),\ \mathbf{y}(t_1), ..., \mathbf{y}(t_T)]$. If we define $\mathbf{N}$ and $\mathbf{S}$ similar to $\mathbf{Y}$, then we can write (5.3) as $\mathbf{Y} = \mathbf{AS} + \mathbf{N}$. Consider the singular value decomposition of $\mathbf{Y}$:

$$\mathbf{Y} = \mathbf{U}\Lambda\mathbf{V}' \tag{5.10}$$

Note that at the moment we are not dealing with a decomposition of the data-covariance matrix, which gets used in subspace-based source localization techniques. We will relate our development with it later on in the section. When no noise is present, matrix $\mathbf{Y}$

has rank $K$, where $K$ is the number of sources. The range space of $\mathbf{Y}$ is spanned by the first $K$ columns of $\mathbf{U}$, corresponding to the non-zero singular values. When we add noise, $\mathbf{Y}$ becomes full-rank (as long as $T \geq N$), and the largest $K$ singular vectors will correspond to signal plus noise subspace, whereas the rest will be due to noise alone. A meaningful linear combination that would summarize the cloud of points is a combination of the signal subspace singular vectors multiplied by the signal subspace singular values. Let $\mathbf{1}_K \in \mathbb{R}^T$ be a vector with its first $K$ entries being ones, and the others zeros. Then the proposed linear combination can be written as:

$$\mathbf{y}_s = \mathbf{U}\Lambda\mathbf{1}_K \tag{5.11}$$

The motivation is that even with noise added, the signal subspace singular vectors will be a reasonable representation of the range space of $\mathbf{Y}$. We weigh the different singular vectors according to the energy that they contain to get a single vector representing the signal subspace. If noise is reasonably small, it is possible to determine which singular values correspond to which subspace by looking at their magnitudes. Noise singular values will be smaller.

The resulting vector $\mathbf{y}_s$ belongs to the range space of $\mathbf{Y}$; it is nothing but a linear combination of $\mathbf{y}(t)$. In fact, $\mathbf{y}_s = \mathbf{U}\Lambda\mathbf{1}_K = \mathbf{Y}\mathbf{V}\mathbf{1}_K$. Let $\mathbf{s}_s = \mathbf{S}\mathbf{V}\mathbf{1}_K$, and $\mathbf{n}_s = \mathbf{N}\mathbf{V}\mathbf{1}_K$, then we get a linear model describing the transformed data:

$$\mathbf{y}_s = \mathbf{A}\mathbf{s}_s + \mathbf{n}_s \tag{5.12}$$

Any linear combination of $\mathbf{s}(t)$ will have the same sparsity profile (support) as the underlying spatial spectrum, therefore the locations of the sources can be determined from the indices of support of $\mathbf{s}_s$.

The method typically has very good performance, but it has a drawback of occasional strong outliers, i.e. for some rare realizations of $\mathbf{u}$ it consistently converges to a wrong solution for all noise realizations. At first the author thought that the outliers are artifacts of poor convergence of the numerical algorithms. This is not the case. Upon computing the bias plot for two sources versus their angular separation, the bias was very well behaved except for sharp narrow rises (which looked continuously differentiable after zooming by using a finer grid) at unpredictable locations. The reason was later found to be the unequal distribution of power among the sources in the transformed domain. For 2 signals, even if $u_1(t)$ and $u_2(t)$ [2] have the same the same power, the transformed signals $u_1^s(t)$ and $u_2^s(t)$ [3] may turn out to have very different powers. This inhibits the performance of sparsity regularization techniques, since the estimate of the smaller of the elements has a larger variance and may even be judged as noise and removed. When the difference in power is not very large (up to around 10 dB), the effects are nearly transparent. The SVD transformation on rare occasions leads to

[2] Recall that in our notation $\mathbf{s}$ is the overcomplete representation of $\mathbf{u}$, i.e. $\mathbf{u}(t)$ are the elements of $\mathbf{s}(t)$ at the indices corresponding to the locations of the signals.

[3] $\mathbf{u}^s$ corresponds to non-zero elements of $\mathbf{s}_s$ (we use a superscript to allow a second index).

much larger discrepancies. The reason this happens is that the linear combination of the singular vectors, $\mathbf{y}_s$, may get aligned much closer to one of the steering vectors than to other ones, effectively falling into a smaller dimensional subspace. This alignment depends on the particular realization of $\mathbf{u}(t)$, and is an important flaw in the processing.

#### Joint SVD processing

Trouble with the previous version arrives when we combine the left singular vectors, by taking $\mathbf{y}_s = \mathbf{U}\Lambda\mathbf{1}_K$. There is an alternative inspired by the joint-time processing of Section 5.1.6. Instead of taking a linear combination of the singular vectors we consider merging the singular vectors into a larger inverse problem.

Let $\mathbf{Y}_{SV} = \mathbf{U}\Lambda\mathbf{D}_K = \mathbf{Y}\mathbf{V}\mathbf{D}_K$, where $\mathbf{D}_K = [\mathbf{I_K}\ \ \mathbf{0}]'$. Here $\mathbf{I_K}$ is a $K \times K$ identity matrix, and $\mathbf{0}$ is a $K \times (T-K)$ matrix of zeros[4]. We multiply the singular vectors by the singular values similar to the previous version, but now instead of adding the vectors together, we keep them separate ($\mathbf{Y}_{SV}$ is a $M \times K$ matrix). Let $\mathbf{S}_{SV} = \mathbf{S}\mathbf{V}\mathbf{D}_K$, and $\mathbf{N}_{SV} = \mathbf{N}\mathbf{V}\mathbf{D}_K$, to obtain $\mathbf{Y}_{SV} = \mathbf{A}\mathbf{S}_{SV} + \mathbf{N}_{SV}$. Now let us consider each column (corresponding to each singular vector) of this equation separately: $\mathbf{y}^{SV}(k) = \mathbf{A}\mathbf{s}^{SV}(k) + \mathbf{n}^{SV}(k),\ \ k = 1,..,K$. If $K > 1$, then we have several subproblems and we can combine them into a single one by stacking. Let $\breve{\mathbf{y}} = \mathrm{vec}(\mathbf{Y}_{SV})$ (i.e. stack all the columns into a column vector $\breve{\mathbf{y}}$). Define $\breve{\mathbf{s}}$, and $\breve{\mathbf{n}}$ similarly. Also, let $\breve{\mathbf{A}} = \begin{pmatrix} \mathbf{A} & & \\ & \ddots & \\ & & \mathbf{A} \end{pmatrix}$, i.e. $\breve{\mathbf{A}}$ is block diagonal with K replicas of $\mathbf{A}$. Finally we get $\breve{\mathbf{y}} = \breve{\mathbf{A}}\breve{\mathbf{s}} + \breve{\mathbf{n}}$ which is in the form of (5.2).

Now we have a similar situation as in the joint-time processing case, but we have reduced the number of subproblems dramatically from $T$ to $K$. Most importantly, $K$ is fixed and is not a function of $T$. By using the SVD to reduce the dimensionality of our observation space we got rid of the most important drawback of joint-time processing, the dependence of the size of the problem on $T$ while still summarizing the information from all the $T$ samples. We can use the remaining steps from joint-time processing without change to solve the new SVD-domain set of problems.

The vector $\breve{\mathbf{s}}$ has been constructed by stacking $\mathbf{s}^{SV}(k)$ for all the signal subspace singular vectors, $k = 1,..,K$. Every spatial index $i$ appears for each of the singular vectors. We want to impose sparsity in $\breve{\mathbf{s}}$ only spatially (in terms of $i$), and not in terms of the singular vector index $k$. We use an $\ell_2$ norm to combine the samples in terms of singular vector index, and a sparsifying $\ell_1$ penalty for the spatial coordinate. Let $\breve{s}_i^{(\ell_2)} = \sqrt{\sum_{k=1}^{K}(s_i^{SV}(k))^2}$, $\forall i$. Then sparsity of the resulting $N_\theta \times 1$ vector $\breve{\mathbf{s}}^{(\ell_2)}$ corresponds to the sparsity of the spatial spectrum. Hence we can find the spatial spectrum of $\breve{\mathbf{s}}$ by minimizing

$$\|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 + \lambda\|\breve{\mathbf{s}}^{(\ell_2)}\|_1 \tag{5.13}$$

[4]If $T < K$, or if the sources are coherent, we use the number of signal subspace singular values instead of $K$ in forming $\mathbf{D}_K$.

The cost function is representable in SOC form. A similar development can be also done using $\ell_p$ for $p < 1$ instead of $\ell_1$.

The SVD of the matrix of sensor outputs has a very close relation with the eigen-decomposition used in subspace methods. If we compute the correlation matrix of the data, $\mathbf{R} = \frac{1}{T}\mathbf{Y}\mathbf{Y}'$, then the eigen-decomposition is given by

$$\mathbf{R} = \frac{1}{T}\mathbf{U}\Lambda\mathbf{V}'\mathbf{V}\Lambda'\mathbf{U}' = \frac{1}{T}\mathbf{U}\Lambda^2\mathbf{U}' \tag{5.14}$$

Hence the eigen-decomposition of the data-correlation matrix $\mathbf{R}$ has left singular vectors of $\mathbf{Y}$ as the eigenvectors, and squares of the singular values of $\mathbf{Y}$ divided by $T$ as the eigenvalues. This justifies our discussion of splitting the range space of $\mathbf{Y}$ into signal subspace and noise subspace, since subspace methods have been relying on it for quite some time (in the zero mean case).

Instead of scaling the singular vectors by the singular values, while forming $\mathbf{Y}_{SV}$, we may scale them by the squares of singular values, to parallel the singular value decomposition of $\mathbf{R}$. Let $\mathbf{Y}_{SV} = \mathbf{U}\Lambda^2\mathbf{D}_K = \mathbf{Y}\mathbf{V}\tilde{\mathbf{D}}_K$, where $\tilde{\mathbf{D}}_K = \Lambda\,[\mathbf{I_K}\ \ \mathbf{0}]'$. The rest of the technique follows exactly what is done originally with $\tilde{\mathbf{D}}_K$ replacing $\mathbf{D}_K$. This modification was empirically observed to notably reduce bias, as we describe in Section 6.4. However, since we noticed this superiority of alternative scaling factors very late in the process of thesis writing, most of the experiments elsewhere in the thesis use the previous version scaled by non-squared $\Lambda$.

Similar to the subspace methods our formulation uses information about the number of sources $K$. However, we empirically observe that incorrect determination of the number of sources in our framework has no catastrophic consequences (such as complete disappearance of some of the sources as may happen with MUSIC), since we are not relying on the structural assumptions of the orthogonality of the signal and noise subspaces. Underestimating or overestimating $K$ manifests itself mainly in the loss of SNR[5].

We present a simulation using our $\ell_1$-SVD processing for zero-mean signals in Figure 5.6. The setup of the experiment is the same as in the last section, except we take $T = 200$ time samples, and lower the SNR to $-8$ dB. MUSIC and Capon's methods have trouble with such amounts of noise, but the spectrum obtained using $\ell_1$-SVD still clearly shows all the three sources. We present more extensive experimental results using this technique in Chapter 6.

[5]There are additional effects, such as a reduction in the number of resolvable sources. When the number of signals is estimated correctly, the number of resolvable sources is around $M - 1$, where $M$ is the number of sensors. However if we severely underestimate the number of sources and take only one singular vector, then the resulting spectrum will still be useful to find multiple sources, but the maximum number of sources that can be resolved would be less than $M - 1$. This reduction is milder than that of the MUSIC method, where any underestimates of the number of sources lead to the disapperance of the peaks.

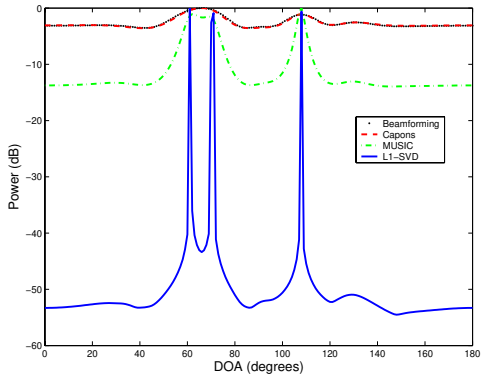
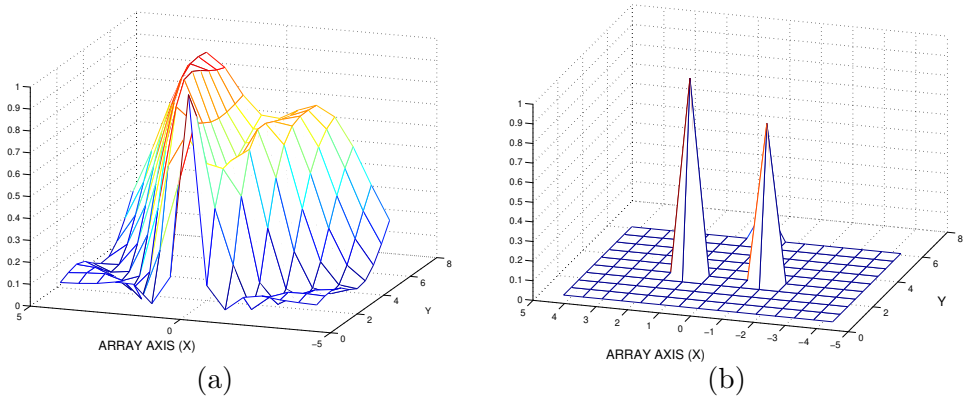



**Figure 5.6.** SVD-$\ell_1$: Spatial spectra of three sources with DOAs of 60°, 70° and 108° (SNR = -8 dB).

### ■ 5.1.8 Narrowband signals in the nearfield

In order to localize sources in the nearfield of the array the manifold is parameterized by both range and bearing, as described in Section 2.1. By taking a grid over range and bearing and stacking it into a vector we get the same form of the problem as for the farfield case, (5.3). The spatial field is sparse in both range and bearing, thus sparsity has to be enforced over all elements of the stacked data. This can be accomplished simply by using an $\ell_1$ or $\ell_p$ penalty. All the methods described in this chapter can be extended in this fashion to handle the nearfield scenario. An important drawback of these methods applied to the nearfield is the need to construct a two dimensional sampling grid over both range and bearing. The dimension of the resulting inverse problems grows quadratically with the fineness of the grid.




(a) (b)

**Figure 5.7.** Nearfield narrowband example: two non-zero mean signals. Source locations are parameterized by distances along and perpendicular to the array axis. Array element spacing is 0.6 meters. (a) Conventional beamforming. (b) Time-sample averaging followed by $\ell_p$ processing.

We present an example of localizing two nearfield sources in Figure 5.7. We look at a non-zero mean case for simplicity, and use averaging to combine multiple snapshots. The spatial locations are parameterized by distance along the array axis, and distance perpendicular to the array axis, from the center of the array. This is equivalent to parameterization by range and bearing. In plot (a) we show a result of using plain beamforming, whereas in plot (b) we use $\ell_p$ processing with $p = 0.1$. The spectrum obtained using $\ell_p$ method is much sparser than the one using beamforming, and peaks due to both of the sources can be clearly seen.

## ■ 5.2 Wideband scenario

The main difficulty which arises when wideband signals are considered is the impossibility to represent the delays by simple phase shifts. A way to deal with this issue is to separate the signal spectrum into several narrowband regions, each of which yields to narrowband processing described in the last section. In general, when the sources have wide frequency spectra, then we are interested not only in the source locations, but also in the frequency composition of each source. We present two approaches for wideband processing. The first one, described next in Section 5.2.1, treats each frequency band independently, which leads to computational simplicity. The second approach in Section 5.2.2 attempts to get a better source location estimate by joint processing of data at different frequency bands.

### ■ 5.2.1 Independent processing in each frequency band

To separate the spectrum into narrowband regions it is possible to use a filterbank, $h_1(t), h_2(t), ..., h_W(t)$, in which each filter $h_k(t)$ has a small spectral support around the central frequency $w_k$, satisfying the narrowband assumption. After filtering the outputs of each sensor with each filter the result is a set of $W$ time-domain problems of the form of (5.3):

$$\mathbf{y}_k(t) = \mathbf{A}(\omega_k)\mathbf{s}_k(t) + \mathbf{n}_k(t) \tag{5.15}$$

We can solve each one using one of the narrowband methods described in the Section 5.1. Once we solve each of the narrowband subproblems we get a spatio-frequency spectrum of the sources.

A better alternative to the filter-bank approach is a frequency domain representation (which can be thought of as a filter bank in its own right). We transform the sensor data into the frequency domain resulting in the following set of linear models:

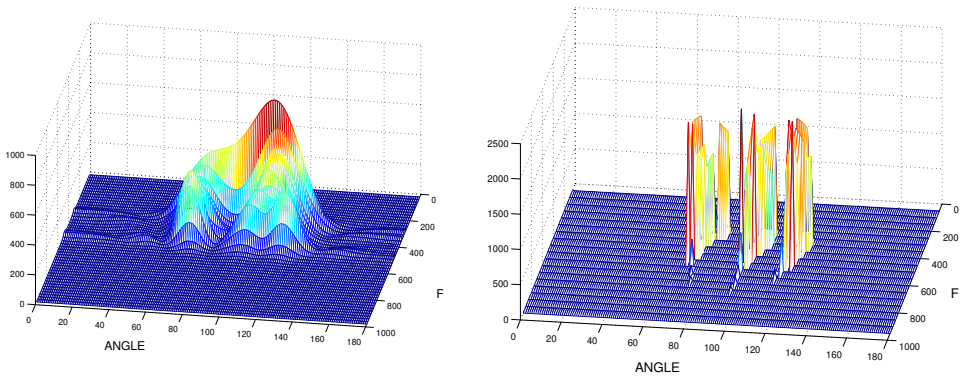
$$\mathbf{y}(\omega) = \mathbf{A}(\omega)\mathbf{s}(\omega) + \mathbf{n}(\omega), \quad \omega = 2\pi k/T, k \in \{0, .., T-1\} \tag{5.16}$$

The set of frequencies here includes every DFT-frequency corresponding to the sampling period, and is much larger then the set of center frequencies used in the filter-bank version of the algorithm, $T \gg W$. We can reduce the number of inverse problems

by invoking the narrowband approximation. We split the frequency support of the sensor outputs into several regions $(\check{w}_0, \hat{w}_0), (\check{w}_1, \hat{w}_1), ..., (\check{w}_W, \hat{w}_W)$. Then we use the steering matrix $\mathbf{A}(w_k)$, where the center frequency of $k$-th narrowband region is $w_k = (\check{w}_k + \hat{w}_k)/2$, as an approximation to the problems for every DFT-frequency falling in the region. This way we again get a set of $W$ problems, but the number of data points in each subproblem is reduced by at least $W$ in the process, compared to the plain filterbank version. The $k$-th problem has the following form:

$$\mathbf{y}(\omega) = \mathbf{A}(\omega_k)\mathbf{s}(\omega) + \mathbf{n}(\omega), \quad \omega \in (\check{w}_k, \hat{w}_k) \tag{5.17}$$

The main improvement over the original DFT model in (5.16) is that only one steering matrix $\mathbf{A}(\omega_k)$ is used for all of the DFT-frequencies in the region $(\check{w}_k, \hat{w}_k)$. Now we can either transform each region to the time domain (shifting first to center the region at 0-frequency for demodulation purposes), or we can work directly in the frequency domain. In both cases we are faced with the same issues of how to treat multiple time/frequency snapshots[6] and we can use one of the approaches for narrowband processing. The most practical and versatile approach for this case is again the $\ell_p$-SVD technique described in Section 5.1.7. Solving all $W$ problems leads to the desired spatio-frequency spectrum.




(a) (b)

**Figure 5.8.** Wideband example: 3 chirps, DOAs 70°, 98°, and 120°. Frequencies are processed independently. (a) Conventional beamforming. (b) $\ell_1$-SVD processing.

In Figure 5.8 we present an example of using the same 8-element uniform linear array as the one used throughout this chapter, but the signals are now wideband. We consider three chirps with DOAs 70°, 98°, and 120° with frequency span from 250 Hz to 500 Hz, and $T = 500$ time samples. Using conventional beamforming the spatio-frequency spectra of the chirps are merged and cannot be easily separated (plot (a)) (especially in lower frequency ranges), whereas using $\ell_1$-SVD (plot (b)) they can

[6] A frequency snapshot is a vector $\mathbf{y}(\omega)$ where $\omega \in (\check{w}_k, \hat{w}_k)$, so that the linear problem is approximated using $\mathbf{A}(\omega_k)$. This way all such $\mathbf{y}(\omega)$ can be treated similar to time snapshots for the corresponding time-domain problem.

be easily distinguished throughout their support. The frequency spectrum of each chirp in $\ell_1$-SVD reconstruction has a jagged shape due to the fact that we treat each frequency independently. Using joint-frequency processing (next section) we can use the information that the spectrum of a chirp is smooth and impose an additional penalty on smoothness in frequency domain to take care of this jagged appearance.

### ■ 5.2.2 Joint-frequency processing

At this point we have several different inverse problems for each frequency band, and we may want to attempt something which parallels the joint-time discussion, in Section 5.1.6. We stack all the frequency vectors, and stack the steering matrices in a similar fashion:

$$\begin{aligned}\breve{\mathbf{y}} &= [\mathbf{y}_s(w_1)' \ \mathbf{y}_s(w_2)' \ ... \ \mathbf{y}_s(w_W)']',\\ \breve{\mathbf{s}} &= [\mathbf{s}_s(w_1)' \ \mathbf{s}_s(w_2)' \ ... \ \mathbf{s}_s(w_W)']',\\ \breve{\mathbf{n}} &= [\mathbf{n}_s(w_1)' \ \mathbf{n}_s(w_2)' \ ... \ \mathbf{n}_s(w_W)']'\end{aligned}$$

The block-diagonal elements are now all different, with $\mathbf{A}_k = \mathbf{A}(\omega_k)$:

$$\breve{\mathbf{A}} = \begin{pmatrix} \mathbf{A}_1 & & & \\ & \mathbf{A}_2 & & \\ & & \cdots & \\ & & & \mathbf{A}_W \end{pmatrix} \tag{5.18}$$

The form of the inverse problem is the same as for the joint-time case:

$$\breve{\mathbf{y}} = \breve{\mathbf{A}}\breve{\mathbf{s}} + \breve{\mathbf{n}} \tag{5.19}$$

Solving the joint-frequency model by imposing an $\ell_1$ or a general $\ell_p$ penalty on $\breve{\mathbf{s}}$, i.e. $\min \ \ \|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 + \lambda\|\breve{\mathbf{s}}\|_p^p$, enforces sparsity both spatially and in frequency. In general it is not a good idea, but if the signals of interest are composed of superpositions of harmonics, then it leads to a desirable sharpness of both the spatial and frequency spectra.

In Figure 5.9 we look at two wideband signals consisting of two harmonics each. The array is again an 8-element ULA. Harmonics have frequencies 200 and 500 Hz at DOA 80°, and 200 and 400 Hz at 110°. Plot (a) shows results using conventional beamforming, and plot (b) uses a joint model with $\ell_p$ sparsity penalty on both DOA and on frequency. The exponent in $\ell_p$ is set to $p = 0.1$. The results are displayed as an intensity map on a 2-D grid of angle and frequency. Conventional beamforming is unable to separate the two spatial peaks for frequency 200 Hz, whereas the joint-frequency $\ell_p$ processing produces four sharp peaks corresponding to each spatio-frequency component.

When the situation is more general, and frequency spectra of the sources are not sparse, penalizing sparsity in frequency is not useful. Instead we can parallel what was done for the joint-time processing in Section 5.1.6. This translates into penalizing the $\ell_1$-norm of the $\ell_2$-norm of all the frequency components for a particular spatial location.

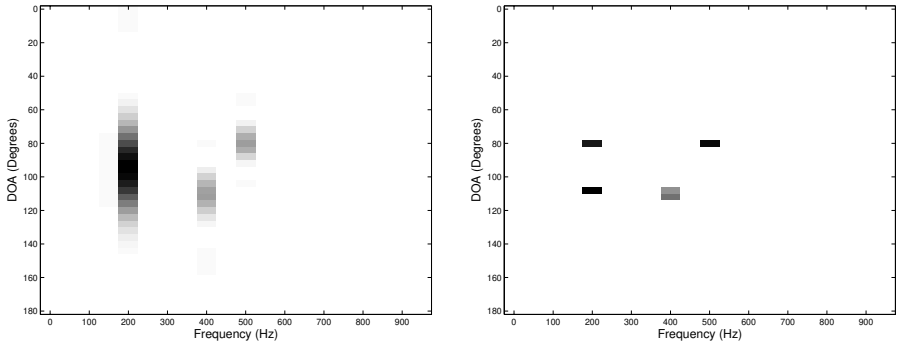



**Figure 5.9.** Wideband example: wideband harmonics, DOA 80°: frequencies 200 and 500 Hz and DOA 110°: frequencies 200 and 400 Hz. (a) Conventional beamforming at each frequency. (b) Joint-frequency $\ell_p$ processing.

We compute the $\ell_2$-norm of all frequency-samples of a particular spatial index $i$ of $\mathbf{s}$, i.e. $s_i^{(\ell_2)} = \|[s_i(\omega_1), s_i(\omega_2), ..., s_i(\omega_T)]\|_2$. We penalize the $\ell_1$-norm of $\mathbf{s}^{(\ell_2)} = \left[s_1^{(\ell_2)}, ..., s_{N_\theta}^{(\ell_2)}\right]$. The cost function becomes

$$\|\check{\mathbf{y}} - \check{\mathbf{A}}\check{\mathbf{s}}\|_2^2 + \lambda\|\mathbf{s}^{(\ell_2)}\|_1 \tag{5.20}$$

This leads to the synergy of the different frequency components for the purpose of getting better spatial spectra. Alternatively, if we have additional knowledge about the unknown spectra, such as smoothness, then it may be beneficial to impose a relevant prior in the frequency domain (instead of an $\ell_2$ prior) along with a sparsity prior in the spatial domain. This is applicable for example to superpositions of several chirp signals, which have smooth spectra.

### ■ 5.2.3 Wideband focusing matrices

The number of narrowband regions depends on the allowable bandwidth which still satisfies the narrowband assumption. The amount of work that has to be carried out is linearly proportional (if the models are not joined into a single inverse problem) to the number of narrowband regions. An interesting idea based on the so-called "focusing matrices" has been developed in [47] which allows to extend the frequency ranges by suitable orthogonal transformations of the data.

The main idea of the method is to find a matrix $\mathbf{Q}$ such that the range space of $\mathbf{QA}(\boldsymbol{\theta}, \omega_k)$ is the same as, or is a good approximation to that of $\mathbf{A}(\boldsymbol{\theta}, \omega_j)$. The steering matrices $\mathbf{A}(\boldsymbol{\theta}, \omega)$ that we refer to correspond to the unknown locations $\boldsymbol{\theta}$, and they are not the overcomplete versions. Denote $\tilde{\mathbf{y}}(\omega_k) = \mathbf{Qy}(\omega_k)$, and $\tilde{\mathbf{n}}(\omega_k) = \mathbf{Qn}(\omega_k)$. Signals at frequency $j$ do not get changed, hence $\tilde{\mathbf{y}}(\omega_j) = \mathbf{y}(\omega_j)$, and $\tilde{\mathbf{n}}(\omega_j) = \mathbf{n}(\omega_j)$. Then the problems

$$\tilde{\mathbf{y}}(\omega_k) = \mathbf{Qy}(\omega_k) = (\mathbf{QA}(\boldsymbol{\theta}, \omega_k))\,\mathbf{s}(\omega_k) + \mathbf{Qn}(\omega_k) \approx \mathbf{A}(\boldsymbol{\theta}, \omega_j)\mathbf{s}(\omega_k) + \tilde{\mathbf{n}}(\omega_k) \tag{5.21}$$

$$\text{and } \tilde{\mathbf{y}}(\omega_j) = \mathbf{y}(\omega_j) = \mathbf{A}(\boldsymbol{\theta}, \omega_j)\mathbf{s}(\omega_j) + \tilde{\mathbf{n}}(\omega_j) \tag{5.22}$$

can be used coherently.

Given a method to generate such a matrix $\mathbf{Q}$, we can proceed to solve the wideband source localization problem. First we take the center frequency of the region of interest, $\omega_c$, and fix the corresponding steering matrix $\mathbf{A}(\boldsymbol{\theta}, \omega_c)$. Next we compute the set of focusing matrices $\mathbf{Q}(w)$ for each of the remaining frequencies in the region of interest, and apply them to the sensor observations, $\tilde{\mathbf{y}}(\omega_k) = \mathbf{Q}(\omega_k)\mathbf{y}(\omega_k)$. The tricky part is that we need to know $\boldsymbol{\theta}$ in order to compute $\mathbf{Q}(\omega)$, and $\boldsymbol{\theta}$ is the unknown that we are trying to estimate in the first place. However, for the purpose of getting a good approximation, we only need to know roughly the regions where there are signals (closely-spaced signals do not have to be resolved, and their number needs not be determined). We can find $\mathbf{Q}(\omega)$ based on the estimates of $\theta$'s using plain beamforming.

We are led to the following set of problems for all $w_k$:

$$\tilde{\mathbf{y}}(\omega_k) = \mathbf{A}(\omega_c)\mathbf{s}(\omega_k) + \mathbf{n}(\omega_k) \tag{5.23}$$

The $\mathbf{A}(\omega_c)$ matrix is now overcomplete, and $\tilde{\mathbf{y}}(\omega_c) = \mathbf{y}(\omega_c)$. Now we have exactly the same set of problems as in the narrowband region approximation (Section 5.2.1), and we can solve them by using $\ell_p$-SVD as well. In practice, if there is a large number of frequencies, instead of computing $\mathbf{Q}(\omega)$ for every one of them, it is more efficient to use the same $\mathbf{Q}(\omega)$ for several nearby frequencies. However, the quality of approximation deteriorates if we consider signals with very wide spectra (selecting $\mathbf{Q}$ which produces good approximations is not possible towards the outer limits of a wide spectrum). Thus, the approach is not a panacea but can be thought of as an extension of the notion of narrowband.

Multiple ideas exist for the selection of a suitable focusing matrix $\mathbf{Q}$. The authors of [47] argue that in order to preserve the information content of the data, $\mathbf{Q}$ has to be orthogonal (for example, this leads to having the same covariance matrix for the noise for the transformed data). With the requirement of orthogonality, they propose to use the following cost function:

$$min\ \|\mathbf{A}(\boldsymbol{\theta}, \omega_c) - \mathbf{Q}(\omega_k)\mathbf{A}(\boldsymbol{\theta}, \omega_k)\|_F, \text{ subject to } \mathbf{Q}(\omega_k)'\mathbf{Q}(\omega_k) = \mathbf{I} \tag{5.24}$$

This problem is well-known under the name of orthogonal Procrustes, [48], and has an efficient global solution using the singular value decomposition.

Our main point here is that the idea of wide-band focusing matrices could be used in our framework to extend the notion of "narrowband" in the context of the techniques of Sections 5.2.1 and 5.2.2.

## ■ 5.3 Multi-resolution grid refinement and zooming

One of the limitations of the techniques as we presented them is their inherent limitation to a grid of steering locations. In order to achieve greater accuracy than allowed by the grid, but at the same time avoid the substantial increase in computation by considering a dense fine grid, we explore the idea of adaptively refining the grid.

Instead of having a universally fine grid, we make the grid fine only around the places where the signals are present. This requires the knowledge of the source locations, which can be obtained by using a coarse grid first. We can also refine the grid several times to get very accurate estimates. The algorithm is the following:

1. Create a rough grid of potential source locations $\tilde{\theta}_i^{(0)}$, for $i = 1, .., N_\theta$. Set $r = 0$.

2. Form $\mathbf{A}_r = \mathbf{A}(\tilde{\boldsymbol{\theta}}^{(r)})$, where $\tilde{\boldsymbol{\theta}}^{(r)} = [\tilde{\theta}_1^{(r)},\ \tilde{\theta}_2^{(r)}, ...,\ \tilde{\theta}_{N_\theta}^{(r)}]$. Use $\ell_1$ regularization[7] (any applicable version, e.g. $\ell_1$-SVD) to get the estimates of the source locations, $\hat{\theta}_j^{(r)}$, $j = 1, .., K$, and set $r = r + 1$.

3. Get a refined grid $\tilde{\theta}_i^{(r)}$ around the locations of the peaks, $\hat{\theta}_j^{(r-1)}$. We specify how this is done below.

4. Return to step 2 until resolution of the grid is fine enough.

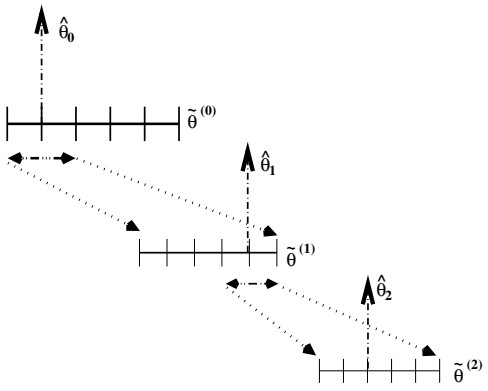



**Figure 5.10.** Illustration of adaptive grid refinement. Coarse grid source localization is followed by refining the grid around the peak locations.

Now we clarify some of the details of the algorithm. In step 1 the rough grid of potential source locations, $\theta_i^{(0)}$ has to be rather fine at the start, not to introduce notable bias. A 1° or 2° uniform sampling, (or similar uniform sampling of the cosine of the angle), usually suffices. The next comment is about grid refinement in step 3. Many different ways to refine the grid can be imagined; we choose simple equi-spaced grid refinement. Suppose we have a locally uniform grid (piecewise uniform), and at step $r$ the spacing of the grid is $\delta_r$. We pick an interval around the $j$-th peak which includes two grid spacings to either side, i.e. $[\hat{\theta}_j^{(r)} - 2\delta_r, \hat{\theta}_j^{(r)} + 2\delta_r]$, for $j = 1, .., K$. In the intervals

[7] In theory it is possible to use $\ell_p$ regularization with small $p$ instead of $\ell_1$. However, there appear to be numerical difficulties with $\ell_p$ when we reach a fine grid size; they are still under investigation.

around the peaks we select the new grid which has spacing a fraction of the old one, $\delta_{r+1} = \frac{\delta_r}{\gamma}$. In our simulations we choose $\gamma = 3$. It is possible to achieve fine grids either by rapidly shrinking $\delta_r$ for few refinement levels, or by shrinking it slowly using more refinement levels. We find that the latter approach is more numerically stable, that is why we set $\gamma = 3$, a relatively small number. After a few (e.g. 5) iterations of refining the grid, it becomes small enough that its effects are almost transparent. We illustrate the idea of grid refinement in Figure 5.10. In our experience the procedure works very well. We use it to get accurate bias-plots and CRB plots in Chapter 6.

This multiresolution method refines the grid around *each* of the peaks of the spectra at previous resolution. However, situations may arise where we are only interested in an accurate estimate of a particular peak, say $\hat{\theta}_1$. This task may be called zooming. Unfortunately, we cannot ignore the presence of other peaks and refine the grid only around $\hat{\theta}_1$, since in order to have $\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2$ small, we must account for all the components of $\mathbf{y}$ which are created by the corresponding sources. Without taking them into account the procedure will create multiple spurious peaks in the refined grid. What may be done is either subtracting the irrelevant components of $\mathbf{y}$ corresponding to the peaks near $\theta_2, ..., \theta_M$, or projecting $\mathbf{y}$ onto the orthogonal complement of the matrix $\mathbf{A}([\theta_2, ..., \theta_K])$, getting rid of the spurious peaks.

## ■ 5.4 Regularization parameter selection

A crucial part of our source localization framework is the choice of the regularization parameter, $\lambda$, which balances the fit of the solution to the data versus the sparsity prior. The same issue arises in all inverse problems where regularization is used, for example in some machine learning tasks, where the fit to the data is balanced versus the complexity of the model. If we define the complexity of the underlying spatial spectrum as the number of nonzero elements in the discretized spatial spectrum, then we are dealing with a very similar problem. Small regularization parameters correspond to good fits to the data and high model complexity (for our case that corresponds to having wide mainlobes or spurious peaks), with consequent overfitting, while too much regularization makes the models over simplistic and fails to explain the data well. The proper choice lives somewhere in between the two extremes.

Over the years many schemes for the selection of the regularization parameter have been developed in the inverse problems, machine learning and statistics communities. The discrepancy principle, an established paradigm for regularization parameter selection, appears to be a good match for our problem. In Section 5.4.2 we propose our adaptation of the discrepancy principle which allows a very efficient implementation. We have had less success or experience with some other approaches; we discuss the issues related to their use for our problem in Appendix F.

### ■ 5.4.1 Discrepancy principle

A very natural idea is to select the regularization parameter (we will refer to it as $\lambda$ from now on, for simplicity) such that the residuals of the solution obtained using $\lambda$ match some known statistics of the noise. This comes under the name of discrepancy principle [26,49]. For example, if the noise is white Gaussian (spatially and temporally) with a known standard deviation, then we can select $\lambda$ such that $\|\mathbf{y}-\mathbf{A}\mathbf{s}\|_2^2 \approx E[\|\mathbf{n}\|_2^2]$. How well this works depends on the spread of the distribution of $\|\mathbf{n}\|_2^2$ around its mean, and the sensitivity of the inverse problem to the corresponding choice of $\lambda$. For quadratic regularization problems, the calculation of $\lambda$ to match the residual to the noise can be efficiently carried out numerically. However, if the problem is non-quadratic, (or even worse non-convex), then ferreting out this $\lambda$ requires substantial work. The only general procedure is an adaptive search, where we first guess a value of $\lambda$, solve the corresponding inverse problem, compare the resulting residual to noise, adjust $\lambda$ accordingly, and repeat until a good candidate is found. If the possible range of $\lambda$ is not known, then this may require a large number of iterations (and inverse problem solutions). Next we propose a considerably more efficient (avoiding iterations) approach based on the constrained version of $\ell_1$ regularization, ML1.

### ■ 5.4.2 Discrepancy principle in $\ell_1$ constrained form

We present a new practical method which avoids the need to search over different $\lambda$ altogether. So far we have developed it for non-zero mean case, and for low-noise $\ell_1$-SVD problems. An efficient procedure is available for the $\ell_1$ version of the techniques only.

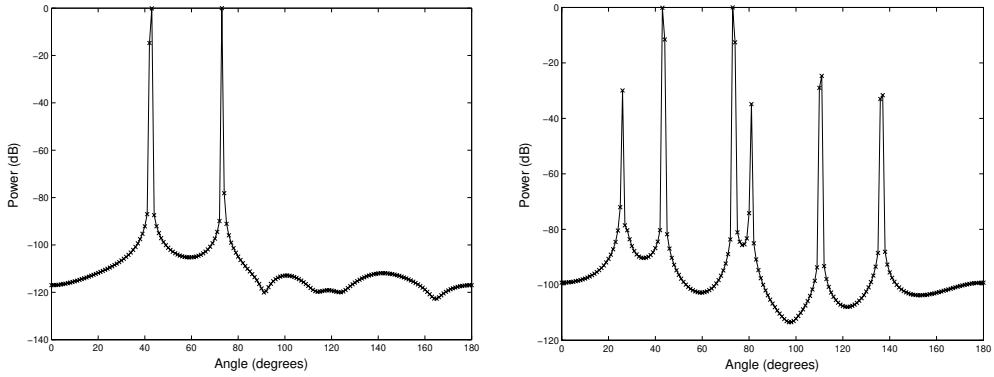
The idea is quite simple. Recall that the set of solutions of the joint version of the noisy $\ell_1$ problem MLJ, min $\|\mathbf{y}-\mathbf{A}\mathbf{s}\|_2^2+\lambda\|\mathbf{s}\|_1$, which we used in this chapter, and the constrained versions, ML1 in particular, min $\|\mathbf{s}\|_1$ subject to $\|\mathbf{y}-\mathbf{A}\mathbf{s}\|_2^2 \le \beta^2$, are equivalent in the sense that the sets of solutions $\hat{\mathbf{s}}(\lambda)$, and $\hat{\mathbf{s}}(\beta)$, over all $\lambda$ and $\beta$ are the same. Going from one version to another is just a question of reparameterization. The difficulty is that we cannot predict the value of $\lambda$ for MLJ such that the MLJ cost function function will have $\hat{\mathbf{s}}(\beta)$ as the optimal solution. Fortunately, there is no real need to find this mapping, since we can solve ML1 problem with about the same effort as solving MLJ using constrained quadratic programming for real data or Second Order Cone (SOC) programming for complex data, as described in Chapter 4.

For the ML1 problem the task of choosing $\beta$ such that we have the residual $\|\mathbf{y}-\mathbf{A}\mathbf{s}\|_2^2$ match the expected value of the norm of the noise is considerably easier. We just choose $\beta$ high enough so that the probability that $\|\mathbf{n}\|_2^2 \ge \beta$ is small. We cannot simply set $\beta = E[\|\mathbf{n}\|_2^2]$, since it is quite likely that a particular realization will have $\|\mathbf{n}\|_2^2 \ge \beta$, and this makes the true solution (the one that would have been obtained in the noiseless case) fall outside the feasible region. In practice that manifests itself in spurious peaks due to noise that are necessary to drive $\|\mathbf{y}-\mathbf{A}\mathbf{s}\|_2$ down to values smaller than $\beta$.

If the distribution of the noise is known, e.g. i.i.d. Gaussian, then computing an

upper bound for $\|\mathbf{n}\|_2^2$ is not very challenging. In the white i.i.d. Gaussian case with variance $\sigma^2$, if we have $\mathbf{n} \in \mathbb{R}^n$, then $(1/\sigma^2)\|\mathbf{n}\|_2^2 \sim \chi_n^2$, the $\chi^2$ distribution with $n$ degrees of freedom.

In practice we choose $\beta$ so that the confidence interval $[0, \beta]$ integrates to 0.999 probability. This rule for the selection of $\beta$ appears to be a good choice down to very low SNR. For very high SNR, $\beta$ computed using the confidence interval may be a very small number, and it is necessary to limit $\beta$ from below to some moderately small fixed value so that the errors due to finite precision of the constrained $\ell_1$ optimization are ignored.




(a) (b)

**Figure 5.11.** The use of the constrained version of the discrepancy principle. SNR = 20 dB, DOAs of 43° and 73° (a) $\beta$ selected at the top of a .999 confidence interval . (b) $\beta \ll \|\mathbf{n}\|_2^2$ .

For illustrative purposes, we include two plots of using constrained $\ell_1$ optimization. In figure 5.11 (a) $\beta$ is selected by the rule that we proposed, and it yields a good reconstruction. The true source locations are at angles 43° and 73° with respect to the array axis, and the $\ell_1$ spectrum exhibits strong peaks at these locations. However, in (b), $\beta$ is chosen much smaller than the norm of the particular noise realization. As we described, spurious peaks appear in order to drive down the residual to $\beta$. It is still possible to determine the correct DOA's, since the spurious peaks are about 30 dB lower, but the first spectrum is visually much more pleasing.

#### Noise norm prediction for $\ell_1$-SVD

A similar procedure can be applied to the $\ell_1$-SVD version of our technique [8]. The objective function for $\ell_1$-SVD is $\min \ \|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 + \lambda\|\breve{\mathbf{s}}^{(\ell_2)}\|_1$. We can instead solve $\min \ \|\breve{\mathbf{s}}^{(\ell_2)}\|_1$ subject to $\|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 \leq \beta^2$. The two problems are related by an unknown transformation from $\lambda$ to $\beta$. Again, we do not need to find this transformation, since

[8] We have in fact proposed two different versions of the $\ell_1$-SVD technique, one with scaling by $\Lambda$, and another with scaling by $\Lambda^2$. The latter version was conceived much later than the first one, so current discussion of parameter choice applies to the first one only. Extensions are possible.

the second problem (constrained form) can be solved with about the same complexity as the unconstrained one. However, a proper choice of the regularization parameter for the constrained version is easier to find. Recall that

$$\|\breve{\mathbf{y}} - \breve{\mathbf{A}}\breve{\mathbf{s}}\|_2^2 = \|\mathrm{vec}(\mathbf{Y}_{SV}) - \begin{pmatrix} \mathbf{A} & & \\ & \ddots & \\ & & \mathbf{A} \end{pmatrix} \mathrm{vec}(\mathbf{S}_{SV})\|_2^2 \tag{5.25}$$

$$= \sum_k \|\mathbf{y}^{SV}(k) - \mathbf{A}\mathbf{s}^{SV}(k)\|_2^2 = \|\mathbf{Y}_{SV} - \mathbf{A}\mathbf{S}_{SV}\|_{fro}^2 = \|\mathbf{Y}\mathbf{V}\mathbf{D}_K - \mathbf{A}\mathbf{S}\mathbf{V}\mathbf{D}_K\|_{fro}^2 \tag{5.26}$$

$$= \|(\mathbf{Y} - \mathbf{A}\mathbf{S})\mathbf{V}\mathbf{D}_K\|_{fro}^2 = \|\mathbf{N}\mathbf{V}\mathbf{D}_K\|_{fro}^2 \tag{5.27}$$

The signal vector $\mathbf{S}$ and all the related quantities, $\breve{\mathbf{s}}$, $\mathbf{S}_{SV}$, and $\mathbf{S}$, correspond to the true signals, and not to the reconstructed ones. Denote $\mathbf{V}_K = \mathbf{V}\mathbf{D}_K$, then $\mathbf{V}_K$ is the set of first $K$ right singular vectors, i.e. an orthonormal set. Hence it is not difficult to get a confidence interval for $\|\mathbf{N}\mathbf{V}\mathbf{D}_K\|_{fro}^2 = \|\mathbf{N}\mathbf{V}_K\|_{fro}^2$, which is a sum of $MK$ squares of normal random variables with mean zero and standard deviation $\sigma$, the same as the original sensor noise. This is a $\chi^2$ distribution with $MK$ degrees of freedom.

This statement in fact is only approximately correct. The singular value decomposition $\mathbf{Y} = \mathbf{A}\mathbf{S}+\mathbf{N} = \mathbf{U}\mathbf{S}\mathbf{V}'$ depends on the particular realization of noise, and hence $\mathbf{V}_K$ is a function of $\mathbf{N}$. However, when noise is small, the term $\mathbf{A}\mathbf{S}$ dominates the singular value decomposition and the change due to the addition of $\mathbf{N}$ is small. In simulations we observe that confidence intervals for the norm of the noise based on ignoring the dependence of $\mathbf{V}_K$ on $\mathbf{N}$ are very accurate up to moderate amounts of noise. In order to use the same scheme for lower SNR, either the dependence has to be recorded based on simulations, or by more intricate analysis it may be possible to predict the variance of the norm of the noise taking into account its influence on $\mathbf{V}_K$.

Chapter 6

# Practical Issues and Performance Analysis

Having introduced the techniques, we have left out some important details concerning their implementation and behavior for the reason of clarity. Looking at the figures appearing in Chapter 5 and reading through the brief commentary which accompanies them, many questions immediately come to mind. How is the regularization parameter chosen? How do we initialize the techniques? Which method is better, $\ell_p$ or $\ell_1$, and if both work equally well then why do we need to consider them both? What is the effect of the grid, and what happens if the true sources have locations in between the grid points? How many sources can be resolved? We answer or at least discuss all of the questions above in Section 6.1.

Another question of primary importance is that of performance. As we have seen, a great number of practical source localization methods have already been developed and investigated. Thus a new method has to justify its existence by having some favorable properties. In the case of $\ell_1/\ell_p$ regularized inverse problem source localization the benefits are: resolving closely-spaced sources; the robustness to low SNR (lower threshold region), [1] correlated sources, and low number of snapshots; as well as absence of the need for accurate initialization (which is essential for Maximum Likelihood methods). We explicate these benefits and compare the performance of our techniques to existing source localization methods in Section 6.2.

Finally, we devote two sections to the study of bias (Section 6.3) and variance (Section 6.4) of the estimates of source locations using our techniques. We look specifically at the non-zero mean scenario with time-sample combination by averaging, and the $\ell_1$-SVD version in the zero-mean case. In the section discussing variance, a comparison with the CRB is provided for parameter regions where our technique is unbiased.

There are quite a few different versions of our source localization scheme, and there are many questions to be explored. Throughout this chapter, instead of considering every possible combinations of versions and questions, we note that many of the char-

[1] Threshold region means the breakdown region of an estimator, where the performance suddenly exhibits a sharp rise in variance, and departs from the CRB. This occurs commonly in nonlinear estimation problems.

acteristics are very similar for all the versions of the techniques. When important differences arise we stop to point them out. Otherwise, we take the freedom to switch freely between the different versions of $\ell_1$ and $\ell_p$. In all the experiments in this chapter we consider only the farfield narrowband problem.

## ■ 6.1 Details of the techniques and their implementation

We now discuss some of the details of the proposed sparse source localization framework. The particular questions that we address in this section are the effects of the grid, comparison of $\ell_p$ for general $p$ and $\ell_1$, initialization of the techniques and selection of some parameters for the techniques, and the number of resolvable sources.

### ■ 6.1.1 Effects of the grid

Recall that the spatial spectrum obtained using our source localization scheme is inherently limited to a grid of spatial locations. As we discuss in Section 5.3, this can be mitigated with the use of a multi-resolution approach, but even there at each resolution the true signal location is most likely to be between the grid points. Fortunately, it turns out that both $\ell_p$ and $\ell_1$ behave very reasonably under such circumstances. Figure 6.1 illustrates what typically happens. The true source is located at 40.7° with respect to the array axis, whereas the two closest points on the sampling grid are 40° and 41°.

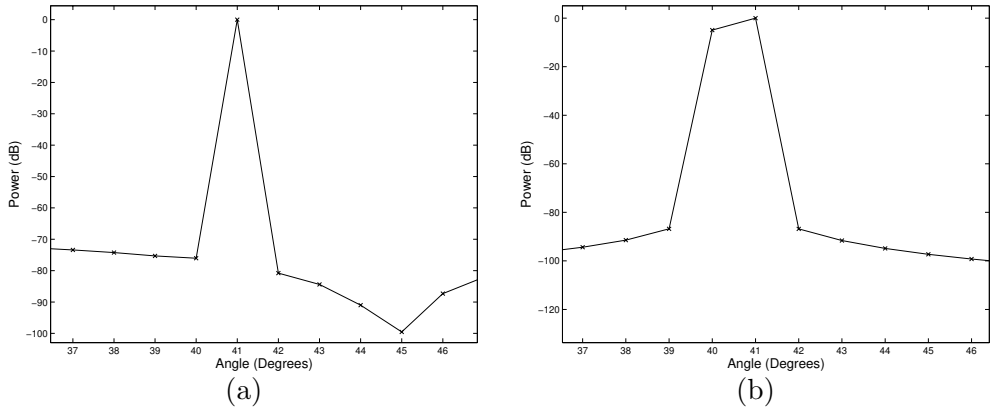



**Figure 6.1.** Source in between the grid points: DOA of 40.7°. SNR = 20 dB. (a) $\ell_p$ . (b) $\ell_1$ .

In the plots we zoom in on the region of interest, and it can be seen clearly that the $\ell_p$ spectrum with $p = 0.1$ has a peak at 41°, whereas the $\ell_1$ spectrum has a wider main lobe which includes both of the grid points. The explanation lies in the fact that $\ell_p$ with $p = 0.1$ puts a much higher penalty on the lack of sparsity, and thus it strongly prefers one non-zero index instead of two. For the $\ell_1$ method sparsity penalty is milder, and the cost of reduced sparsity is balanced by the improvement in the fit to the data. Note the observed outcomes are not universal, and under particular realizations $\ell_1$ may

turn out to have just one peak, and similarly $\ell_p$ has wider peaks from time to time. Nevertheless, both outcomes are favorable for the task of source localization, (it would be much worse if instead spurious peaks appeared in unexpected places) and hence the grid is not a major hindrance, as long as it is not very coarse.

### ■ 6.1.2 $\ell_p$ vs. $\ell_1$

An interesting question that we are faced with is the choice of which cost function to use, $\ell_p$ with $p < 1$, or $\ell_1$. From our theoretical analysis (Section 7), $\ell_p$ is expected to have much better resolution capabilities as $p \to 0$, provided we converge to a global minimum of the cost function (4.39). However, we use local optimization methods, and we are only guaranteed to converge to local minima. Penalization with $\ell_1$, on the other hand, involves a convex cost function, and is guaranteed to converge to a global minimum. Hence, theory still leaves the question without an answer.

In practice, we observe that usually the solutions using the two cost functions (with the Interior Point Method (IPM) implementation of $\ell_1$ in one of three forms ML1, ML2, or MLJ from Section 4.1.2, and our iterative procedure for $\ell_p$ with small $p$ from Section 4.2) are remarkably similar. Although occasionally, the $\ell_p$ method seems to have a little better sparsity (as in the case of sources in between the grid points). One possible explanation is that the $\ell_1$ minimum is very close to a local minimum of the $\ell_p$ cost function, and using our initialization scheme we consistently converge to it. An investigation of the structure of the local minima of the $\ell_p$ cost function appears to be very difficult, and has not been done. Also, the IPM implementation of $\ell_1$ has much better performance than the iterative procedure for $\ell_p$ optimization for problems with high matrix condition numbers. These problems arise in our multiresolution approach, where the condition number can increase dramatically for the refined grid, and for problems with very closely-spaced sources. In these cases $\ell_p$ either converges to poor solutions, or has numerical difficulties, and the use of $\ell_1$ is preferred.

In terms of the computational complexity the two versions are similar (all the different versions of $\ell_1$ have about the same running time). The number of iterations of the Quasi-Newton method for $\ell_p$ is typically between 10 and 20 (unless the tolerances are set very high), and the number of iterations of the interior point method is also about the same. The time per iteration is in both cases dependent upon a solution of a linear system of equations of similar dimensions, so it is also comparable[2].

One clear benefit of $\ell_1$ is that we do not have to worry about convergence to bad local minima (due to the lack of bad local minima altogether). Yet, initializing the $\ell_p$ technique with the beamforming spectrum seems to consistently converge to very good local minima, which are as useful for the task of source localization as the global minima of $\ell_1$.

Using the $\ell_p$ iterative method we are not limited to small $p$, we can also set $p = 1$

[2]For reference, the time required to solve an instance of the source localization problem with an $8 \times 180$ matrix $\mathbf{A}$ using either $\ell_1$ or $\ell_p$ with $p = 0.1$ with a Matlab implementation of the code on Linux on a computer with an 800 MHz Pentium 3 processor is roughly 2 seconds.

without any change in the algorithm. This way we achieve a smooth approximation to the $\ell_1$ cost function. One would expect that this has similar minima as those of pure $\ell_1$, but the resulting spectra have much wider main-lobes than those using pure $\ell_1$ via IPM, or $\ell_p$ with small $p$. A plot of a typical spectrum is included in Figure 6.2. This phenomenon of widening the peaks can be eliminated by setting the smoothing parameter $\epsilon$ in the $\ell_p$ approximation (4.42) to a very small value, and setting PCG tolerance for convergence to a very small value. By using $10^{-10}$ for both[3] we get about the same level of sharpeness in the spectra as for $\ell_p$ with $p = 0.1$. However, the number of iterations required for convergence, and the running time of each PCG iteration increase substantially. From here on, whenever we speak of $\ell_1$ source localization, we refer to the IPM implementation with the exact $\ell_1$ cost function.

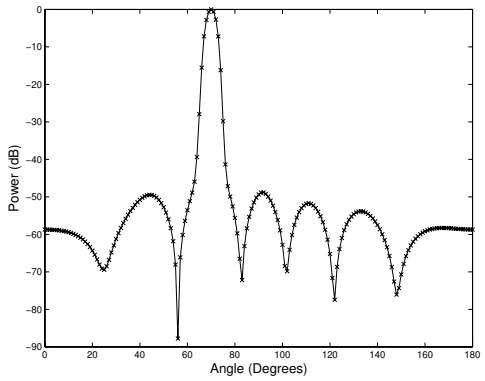



**Figure 6.2.** Iterative $\ell_p$ procedure with $p = 1$. The smoothing parameter $\epsilon$ and the PCG tolerance are kept the same as for $\ell_p$ with $p = 0.1$. DOA: $70^\circ$ (SNR = 20 dB).

And, last but not least, currently there is an important reason of preferring $\ell_1$ over $\ell_p$ with $p < 1$, since it may allow an efficient one-step choice of the regularization parameter by turning to the constrained version (ML1). It is also possible to formulate a constrained version of the $\ell_p$ cost function, i.e. min $\|\mathbf{s}\|_p^p$ subject to $\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2 \leq \beta$, and in fact we have implemented it using the log-barrier approach. However, the speed of convergence is extremely slow. The reason quite likely lies in our poor choice of the points along the central path of the problem, and a carefully implemented IPM is likely to have the same speed of convergence as the corresponding $\ell_1$ problem.

### ■ 6.1.3 Initialization

One of the important benefits of our approach to source localization is that we do not require an initialization by an already very accurate spectrum, unlike ML techniques. In

[3]For comparison, for $\ell_p$ with $p = 0.1$, we typically set both $\epsilon$ and the PCG tolerance somewhere in between $10^{-3}$ and $10^{-5}$.

fact any of the $\ell_1$ techniques have global convergence independent of the initialization, so they are completely insensitive. For the $\ell_p$ counterpart initialization does make a difference, and we have found that starting from a beamforming solution we converge to good local minima. Figure 6.3 shows how the iterative procedure improves with each iteration and finally obtains a sharp spectrum resolving the two sources.

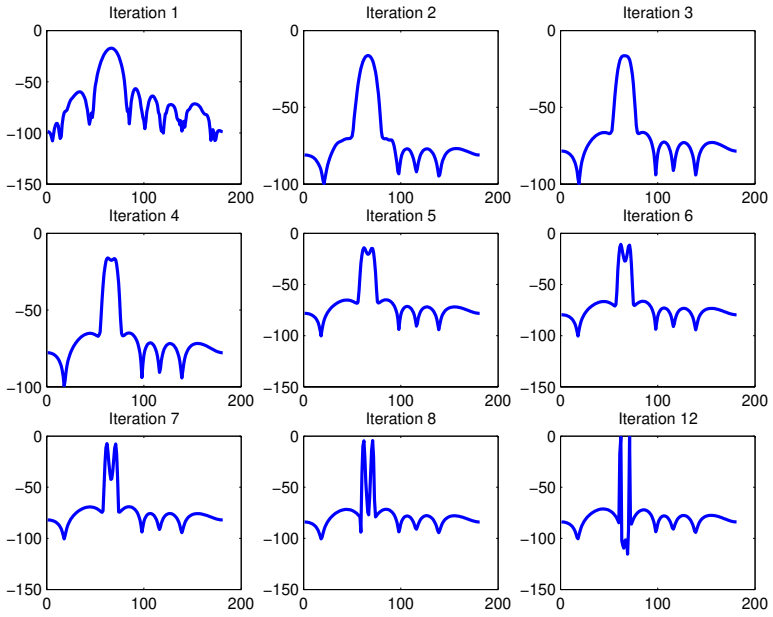



**Figure 6.3.** $\ell_p$: Progress with iterations staring from beamforming solution

We illustrate the effect of choosing a different initialization. Figure 6.4 (a) shows what happens when we initialize the iterations by Gaussian noise. The method converges to a poor solution. In plot (b) we initialize by the MUSIC spectrum. The method converges to peaks which are unbiased. When a spectrum obtained using $\ell_p$ exhibits a bias (see Section 6.3), it may be possible to correct it using an initialization with another super-resolution technique, such as MUSIC (provided the signals are not correlated, and the SNR is high enough). The practical applications of this are not very clear, since sparse regularization framework is beneficial in the regions with low SNR, where MUSIC fails. Hence if MUSIC can be used to achieve an accurate spectrum there is no reason whatsoever to run $\ell_p$. The example does show that we may achieve much better source localization through the investigation of other possibilities for initialization, or by global optimization techniques.

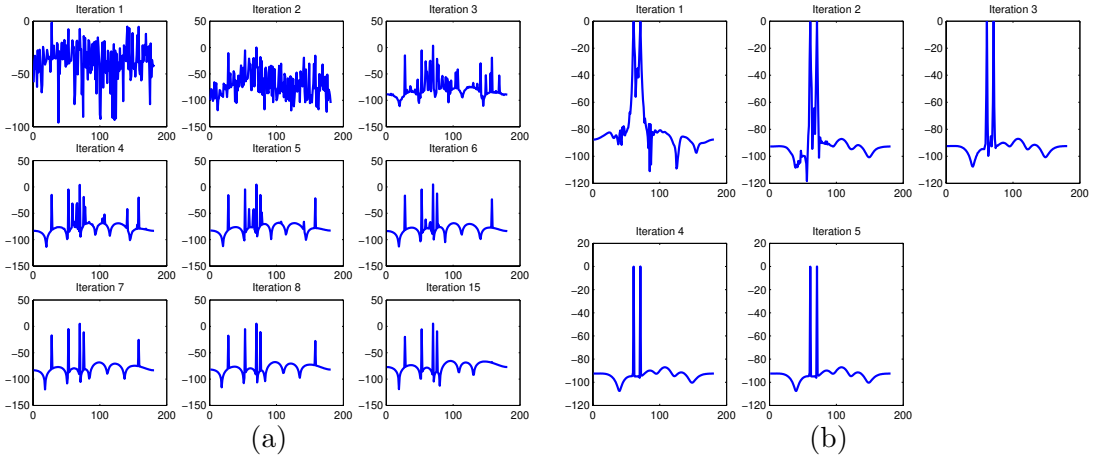



**Figure 6.4.** Progress of iterations for $\ell_p$ with different initializations: (a) Random initialization. (b) Initialization by MUSIC spectrum.

### ■ 6.1.4 Parameter selection

No specifications of an algorithm is complete without a discussion of how to select various parameters. We are lucky to have only a few parameters, so the discussion is not very long. In the $\ell_1$ technique, the only parameter of interest is the regularization parameter. We can also consider the choice of a particular version, ML1, ML2, or MLJ as a ternary-valued parameter. In the $\ell_p$ processing case there are two additional parameters, namely the power, $p$, and $\epsilon$, the smoothing factor for the differentiable approximation in (4.42).

We have discussed the constrained $\ell_1$ version of the discrepancy principle for choosing the regularization parameter in Section 5.4. There are some other possibilities for automatic selection, which we discuss in Appendix F, but it is not yet clear whether they can be successfully used in practice. So far, we have developed the constrained $\ell_1$ approach for some limited scenarios only, so for other scenarios we have no fast methods. In these cases we usually set the parameters by subjective assessment of the resulting spectra through trial and error. Moderate changes of the SNR, and the positions of the sources do not require reselection of the parameter, so this approach is feasible for our purposes. In practice, manual choice would be useful only in very controlled source localization problems, and we keep looking for a more general fast automatic rule.

Choosing an appropriate version of $\ell_1$ is much easier. The ML2 version has no practical significance, since the $\ell_1$ norm of the incoming signal is typically unknown. The MLJ is used whenever the estimation of the variance of the noise is difficult, or when it is completely unknown (for example: beamspace, unknown noise fields, and problems with array model errors). In cases where noise power is known or can be predicted, ML1 is the best bet.

Selecting the parameters associated with the $\ell_p$ technique does not pose a major difficulty. For $p$ in the range from around 0.01 to 0.9, the results do not change no-

ticeably. Shifting $p$ within this region has very little effect on the solution. In the rest of the manuscript, we set $p = 0.1$. When $p$ is around 1, then as we described, with loose tolerances we observe widening of the lobes, with the associated resolution loss, whereas with tight tolerances the running time to convergence increases substantially. When $p$ is very close to zero, difficulties with convergence arise. The insensitivity of solutions to the choice of $p$ within a large region is contrary to what we expect from our theoretical analysis, namely that lower values of $p$ would lead to better solutions.

The smoothing parameter $\epsilon$ appears in the differentiable approximation to the $\ell_p$ norm [4] :

$$\|\mathbf{s}\|_p^p \approx \sum_{i=1}^{N} (|\mathbf{s}_i|^2 + \epsilon)^{p/2} \tag{6.1}$$

where $N$ is the dimension of the vector $\mathbf{s}$. When $\epsilon$ is too large, the approximation is not a good one, and the solutions are overly smooth, exhibiting wide mainlobes. When $\epsilon$ is very small, the number of iterations required for convergence increases drastically. If we stop the iterations prior to convergence, then the solutions we achieve are again very smooth and with wide mainlobes [5]. We have found empirically that a choice of $\epsilon$ which does not require very many iterations, and yet converges to very sharp solutions is around $10^{-3}$ to $10^{-5}$ for our data.

A parameter which has notable importance for $\ell_p$ with small $p$ but which we have not defined explicitly is the scaling of the variables. Using $\ell_p$ penalization it makes a large difference to have $\mathbf{y}$ versus $\mathbf{y}/10$. As $p \to 0$, the norm of the gradient of $\ell_p$-penalty increases indefinitely, whereas as $p \to \infty$ it approaches zero. Hence if we scale the data so that $\mathbf{s}$ is very small, the $\ell_p$ penalty has a very strong preference for sparsity. When $\mathbf{s}$ is large, the penalty is milder. Reasons for such behavior become apparent when we take a close look at the $\ell_p$ cost function for scaled data. Suppose we have $\tilde{\mathbf{y}} = 0.1\ \mathbf{y}$, and $\tilde{\mathbf{s}} = 0.1\ \mathbf{s}$. Also, suppose that $\mathbf{s}$ has a few large coefficients, and all the others zeros. Then the original cost is $\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_p^p$. When $p$ is very small, the cost of the scaled data is approximately $0.1^2\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_p^p$. The $p$-norm term is not scaled since for small $p$ it essentially counts the number of nonzero coefficients, which we do not change by moderate scaling. In order to have the same solution for the scaled problem, we need to adjust the regularization parameter by $0.1^2$. Additional difficulties arise since we may also need to change $\epsilon$ in the approximation of the $\ell_p$ norm. We find that we avoid difficulties with overpenalization if the data is scaled so that the norm of $\mathbf{y}$ is of the order of magnitude of the number of non-zero elements of $\mathbf{s}$, or greater, and $\mathbf{A}$ has columns normalized to unity.

[4] We use the term $\ell_p$-norm, but in fact the cost function has $\ell_p$-norm to the p-th power. We continue with this practice to avoid obfuscation of the sentences.

[5] The number of iterations can serve as regularization on smoothness - the less iterations, the smoother the solution. This is in fact the motivation for the Landweber regularization method [26].

### ■ 6.1.5 Number of resolvable sources

In order to characterize the number of resolvable sources we have several relevant theoretical results which will be described in Chapter 7. These results are developed for the noiseless regularization, but they motivate what occurs in the noisy case as well. The basic result in Theorem 1 translated to fit our array processing problem states that if the array manifold $\mathbf{a}(\boldsymbol{\theta})$ is unambiguous, then the noiseless $\ell_0$ problem, $\min \|\mathbf{s}\|_0^0$ such that $\mathbf{y} = \mathbf{A}\mathbf{s}$, has a solution $\hat{\mathbf{s}}$ which is unique if $\|\hat{\mathbf{s}}\|_0^0 < (M+1)/2$. In particular, that means that if some signal $\mathbf{s}^*$ satisfies $\|\mathbf{s}^*\|_0^0 < (M+1)/2$ and $\mathbf{y} = \mathbf{A}\mathbf{s}^*$, then $\mathbf{s}^*$ is the $\ell_0$ solution.

Also, signals which are not sparse enough are not guaranteed to equal the $\ell_0$ solution. In fact, there exist vectors $\tilde{\mathbf{s}}$ with $\|\tilde{\mathbf{s}}\|_0^0 = L \geq (M+1)/2$ and $\mathbf{y} = \mathbf{A}\tilde{\mathbf{s}}$, such that the $\ell_0$ solution $\hat{\mathbf{s}}$ may have a lower sparsity, or have the same sparsity but be nonunique. That is to say, if the true spatial spectrum is not sparse enough, then we cannot guarantee that the optimal solution to the $\ell_0$ problem will get it right. Finally, for all $\tilde{\mathbf{s}}$ such that $\|\tilde{\mathbf{s}}\|_0^0 \geq M$ there is $\hat{\mathbf{s}}$ satisfying $\mathbf{y} = \mathbf{A}\hat{\mathbf{s}}$ and $\|\tilde{\mathbf{s}}\|_0^0 \geq \|\hat{\mathbf{s}}\|_0^0$. That means that when the number of sources is greater than or equal to the number of sensors then we are guaranteed that we will not get the spatial spectrum correct by solving the noiseless $\ell_0$ problem.

How does it relate to $\ell_1$ and $\ell_p$ regularization? Also in Chapter 7 we have a number of results relating the noiseless $\ell_0$ problem with the noiseless $\ell_1$ and noiseless $\ell_p$ problems. Recall that originally we considered $\ell_p$ and $\ell_1$ penalties as approximations to the non-differentiable $\ell_0$ penalty. Our theoretical results are very general; they apply to any overcomplete basis. However, for the source localization application the basis is a parameterized manifold, so array manifold vectors corresponding to nearby source locations are also close in terms of the Euclidean distance. What often happens is that even if the equivalence conditions are not met, and the $\ell_1$ solution does not give the correct answer to the noiseless problem, it gives a very close approximation. For an overcomplete basis where basis elements are not related with their nearby neighbors such behavior does not occur. Our theoretical results give some motivation for the number of resolvable sources in the practical problem, but to get a reliable estimate we have to measure it empirically.

For the single time sample problem (5.2) and for the non-zero mean processing with averaging (5.4) we found by simulations that the number of resolvable sources [6] for an unambiguous array with $M$ sensors is $M/2$. That is to say that if we have more signals than $M/2$, then the spectrum quality deteriorates dramatically, and some sources may not yield peaks in the spectrum, and spurious peaks may appear[7]. We illustrate this in

[6] The arrays that we used in these simulations are uniform and linear, with half-wavelength sensor spacing, so these estimates may not hold for very different array geometries. Also sources in these experiments were uncorrelated, so potentially the number of resolvable sources may decrease with correlated sources.

[7] If some sources are very close, then it may not be possible to resolve $M/2$ sources using $\ell_p$ with small $p$ at any SNR due to the asymptotic bias in the $\ell_p$ technique.

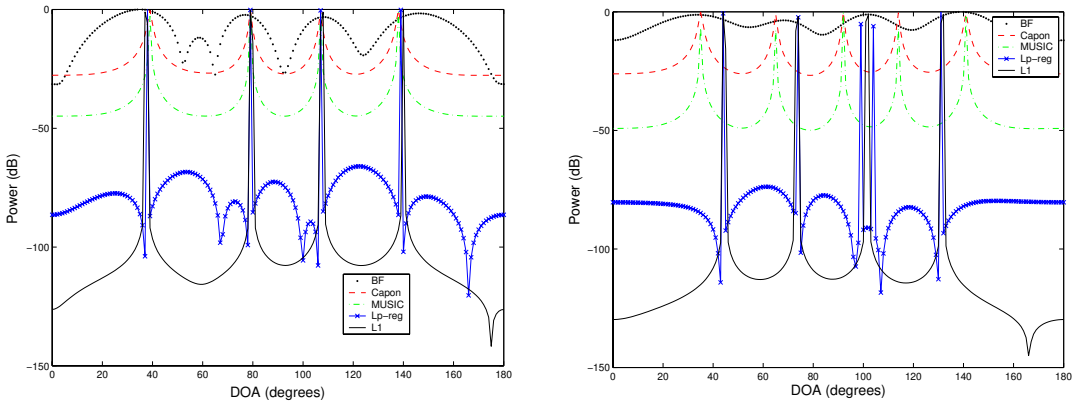



**Figure 6.5.** Number of resolvable sources for the averaging technique. (a) 4 sources are resolved. (b) 5 sources are not resolved. $\ell_p$ and $\ell_1$ spectrum are not useful indicators of source locations.

Figure 6.5. The array has 8 sensors, and the number of sources in plot (a) is 4. The $\ell_p$ and $\ell_1$ spectra have peaks at the source locations. We also included spectra of Capon's and MUSIC methods, and since the SNR is high (SNR=30 dB), these spectra also exhibit peaks at the correct locations. In plot (b) the number of sources is increased to 5, and the peaks of $\ell_1$ and $\ell_p$ spectra no longer correspond to the source locations. This can be seen since they are different from the peaks of MUSIC and Capon's spectra which allow higher number of resolvable sources, and show correct DOAs.

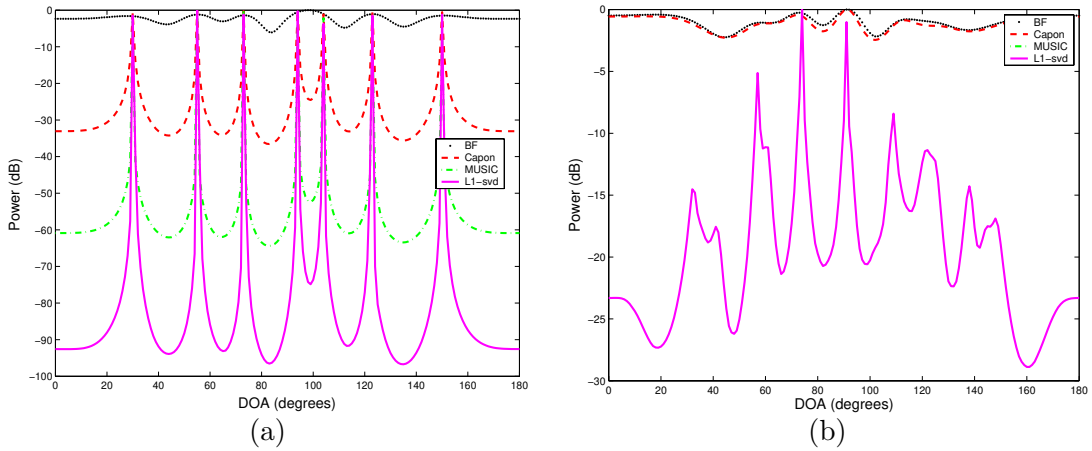



**Figure 6.6.** Number of resolvable sources for the $\ell_1$-SVD technique. (a) 7 sources are resolved. (b) 8 sources are not resolved. $\ell_1$-SVD spectrum is not a useful indicator of source locations.

For the $\ell_1$-SVD technique the number of resolvable sources is higher. Recall that we merge the problems for different singular values into a single larger inverse problem.

We have not analyzed theoretically what this will imply, but empirically the number of sources that the joint $\ell_1$-SVD technique with multiple time samples can resolve is $M-1$ for an unambiguous array with $M$ sensors. This is illustrated in Figure 6.6. The number of sensors in the array is again $M = 8$, but the number of sources in plot (a) is 7. All three techniques ($\ell_1$-SVD, MUSIC, and Capon's method) exhibit peaks at the source locations. When we increase the number of sources to 8 in plot (b) none of the spectra have peaks at correct locations. When the number of time samples is much less than the number of sources, then the number of resolvable sources may decrease. We have not fully characterized this dependence, but even with 1 time sample multiple sources can be resolved.

## ■ 6.2 Benefits of using the sparse regularization framework

### ■ 6.2.1 Superresolution and robustness to noise

A strong feature of our framework is its ability to resolve closely-spaced sources and its good robustness to noise. In this discussion we join these two features together because for superresolution methods resolution depends on the SNR. This is not true for conventional beamforming where resolution has an upper bound, the Rayleigh resolution limit, independent of the SNR. Other superresolution methods such as MUSIC and Capon's exhibit excellent resolution when SNR is high, but once the noise becomes significant their resolution begins to decrease. This also happens to our method as well, but according to our simulations our techniques can withstand higher levels of noise. Maximum Likelihood methods work well with good initialization, but since the initialization is typically performed by MUSIC, it has similar troubles with robustness to low SNR.

First we take a look at non-zero mean signals and the averaging version of our technique. We consider a uniform linear array of $M = 8$ sensors separated by half a wavelength of the actual narrowband source signals. We consider two narrowband signals in the far-field impinging upon this array. The total number of snapshots is $T = 200$. The objective function that we use for the plots has $\ell_p$ penalization with $p = 0.1$. We consider the case when the two sources lie within a Rayleigh resolution cell. Figure 6.7 contains results for SNRs of 20 dB and 5 dB. Beamforming spectrum merges the two peaks. At 20 dB, Capon's, MUSIC and $\ell_p$ are all able to separate the two sources. However, when the SNR is lowered to 5 dB, MUSIC and Capon produce spectra where the peaks are merged, whereas the $\ell_p$ spectrum still exhibits two distinct peaks. These plots demonstrate the relatively superior robustness of the $\ell_p$ method to high levels of noise in the non-zero mean case.

This example was based on a single trial. Now we characterize the performance of the $\ell_p$ method over 200 independent trials, as a function of SNR. We consider two performance metrics. The first one is the probability of detecting the two sources with $1^\circ$ accuracy. The second one is the root-mean-squared-error (in angles) in locating the sources. The two measures convey very similar information, and the experimental

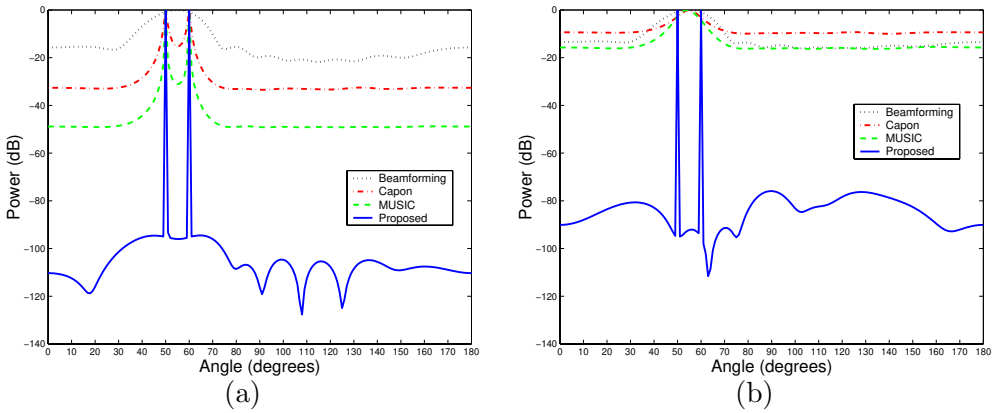



**Figure 6.7.** Spatial spectra of two sources with DOAs of 50° and 60°. (a) SNR = 20 dB. (b) SNR = 5 dB.

results are very similar, so we present only the probability of detection results here. Figure 6.8 presents results for the case when the sources are separated by 15°. This plot confirms that the proposed method has a significantly better probability of correct detection than Capon's method and MUSIC at low SNR values.

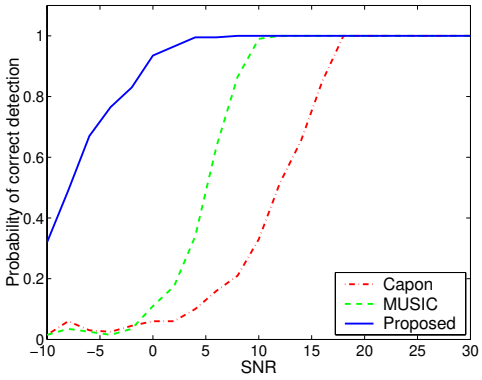



**Figure 6.8.** Probability of correct detection for two sources as a function of SNR. DOAs: 50° and 65°

Robustness to noise and superresolution occurs not only for the non-zero mean version of the technique, but for other versions as well. We take a look at the $\ell_1$-SVD version to justify this claim. In fact for zero-mean sources we lose an important advantage of having a non-zero mean which is not fully exploited by either Capon's or MUSIC methods, according to our simulations. But even without this advantage the $\ell_1$-SVD technique still appears to have very good robustness properties to noise. We illustrate this behavior in Figure 6.9. In plot (a) SNR = 10 dB, and all three techniques are able to resolve the two closely-spaced sources, at 65° and 70°. However, when we lower the SNR to −3 dB, MUSIC and Capon's methods are no longer able to resolve

the two sources and the two peaks merge into a single one. Some additional analysis of resolvability and robustness to noise appears in the bias and CRB sections, 6.3 and 6.4.

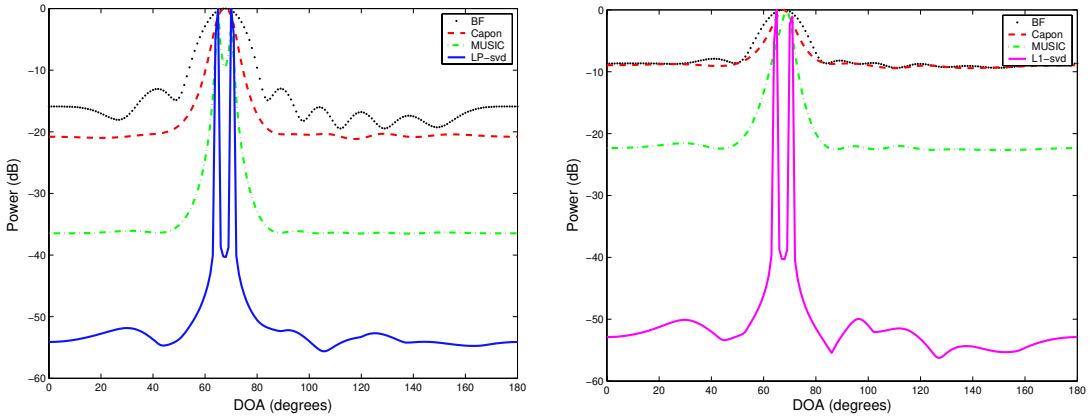



**Figure 6.9.** Spatial spectra of two sources with DOAs of 65° and 70° using $\ell_1$-SVD. (a) SNR = 10 dB. (b) SNR = -3 dB.

### ■ 6.2.2 Robustness to limited number of samples

Robustness to limited number of samples is another important benefit of our approach. Recall that the initial version of our technique from Section 5.1.1 is developed for a single sample, the extreme case of limited number of samples. Other versions such as averaging, beamspace, joint-time processing, and $\ell_1$-SVD are based on the single-sample version, and when only one time sample is available they in fact reduce to the single-sample version. So all our techniques can resolve multiple sources despite having just one time sample[8]. This is not possible for MUSIC and Capon's methods. For Maximum Likelihood methods it can be done, but as always, provided that a good initialization is chosen. When multiple sources are present good initialization for single-sample processing is not an easy task. Beamforming also is able to resolve multiple sources using a single time sample, but the resolution is limited. For our methods the resolution using a single time sample is comparable to that when multiple time samples are available (if the SNR is not very low), and initialization is not an issue as in the multiple time sample case. In Figure 6.10 we illustrate the behavior for a uniform linear array with 8 sensors when only one time sample is available. The $\ell_p$-technique is able to resolve both sources, whereas Capon's method and MUSIC miss the second source entirely.

[8]Of course, in the single snapshot case, if one of the signals is very close to zero, then this source cannot be localized. This is not particular to our method - if the source is silent, then we can regard it

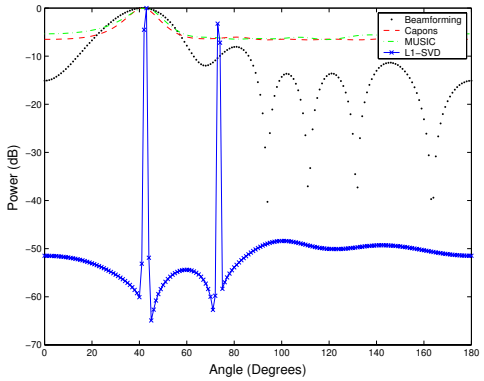



**Figure 6.10.** One time sample, SNR=20 dB. DOAs: 42.83° and 73.33°

Ability to use one time sample to localize multiple sources is just one facet of robustness to limited number of time samples. The other facet is that by reducing the number of time samples the performance of all superresolution techniques decreases. Using the probability of correct detection criterion from last section, our technique appears to handle the decrease in the number of time samples better. We illustrate this in Figure 6.11. Capon's method seems to reach a limit on the probability of detection which does not improve with the number of snapshots at about 40 snapshots. MUSIC does improve to reach a unity correct detection probability, but that takes almost 200 time samples. The averaging version of $\ell_p$ regularization reaches a unity probability of correct detection in about 50 to 60 samples, and it has a considerably better probability of detection throughout the region.

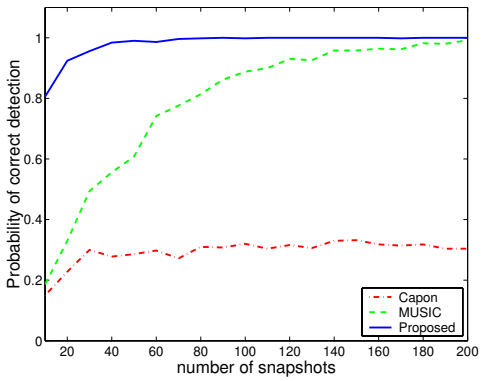



**Figure 6.11.** Probability of correct detection for two sources with DOAs of 50° and 65° as a function of the number of snapshots (SNR = 10 dB).

as nonexistent.

### ■ 6.2.3 Robustness to correlated sources

Due to the nature of the averaging technique for the non-zero mean case it is insensitive to correlated sources, since we are only dealing with the temporal mean, and we do not take the spatial covariance matrix into account. Figure 6.12 shows that the coherence of the signals poses a serious difficulty to the standard implementations of MUSIC and Capon's methods [9], but the resolution capabilities of $\ell_p$-technique are affected very little. Note that at this SNR and spacing, Capon's method and MUSIC would resolve the sources if they were uncorrelated (see Figure 6.7).

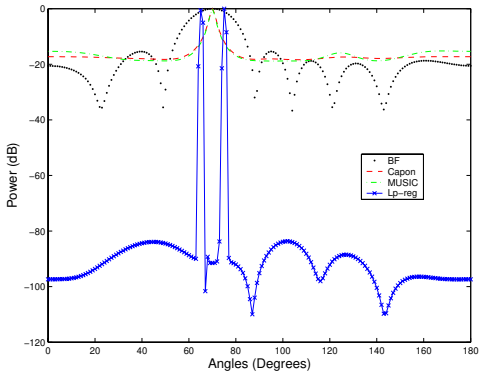



**Figure 6.12.** Robustness to correlated signals. DOA's: $65^\circ$ and $75^\circ$ (SNR = 20 dB).

The $\ell_1$-SVD version also has better robustness to low SNR and better resolution when the sources are correlated. This occurs for a different reason than in the non-zero mean case with the averaging technique. As we explained in Section 5.1.7, when we have correlated sources, source localization does not break down, since even a single singular vector is sufficient to represent a multi-dimensional signal subspace. When we have multiple singular vectors we gain robustness to noise. In the case of MUSIC, when two sources are perfectly correlated, a signal subspace eigenvector moves to the noise subspace, and the corresponding source location cannot be found. An example appears in Figure 6.21, in Section 6.4.

### ■ 6.2.4 Lack of need for accurate initialization

The need to initialize ML techniques by an already accurate estimate of the location of the sources, or alternatively using global optimization techniques is their major drawback. When source signals are correlated, or when only one time sample is available and there are multiple sources, accurate initialization is especially challenging. Our set of techniques on the other hand does not suffer from problems with initialization. All $\ell_1$

[9] There exist versions of MUSIC with better robustness properties to correlated sources, but robustness comes at a price of reduced resolution.

techniques are globally convergent from any initialization, and $\ell_p$ techniques converge to very good solutions starting from beamforming solution (or even from a constant solution), with no need to resolve the sources at the starting point. We discuss initialization of the techniques in more detail in Section 6.1.3.

## ■ 6.3 Bias

An important downside in using the $\ell_1$ / $\ell_p$ regularization techniques for array processing is the bias. After discussing bias in general, we first characterize the bias for the averaging version of our technique with nonzero mean signals, and next for $\ell_1$-SVD with zero-mean sources.

One source of bias which immediately comes to mind is due to the fact that our estimates are limited to a grid. If the source location resides strictly in between two consequent grid points, then the estimate will fall on one of these two points. At least in the very low noise case (when the estimates fall consistently on the same grid points), there will be a bias due to the grid. This bias can be efficiently eliminated by our multiresolution approach (up to any required precision). Unfortunately there is another source of bias inherent in the nature of our sparsity enforcing functionals ($\ell_1$, $\ell_p$), which cannot be easily removed. The reason for the second type of bias is not known fully, but the two ingredients which may explain it are regularization, and the approximation of sparsity by $\ell_p/\ell_1$ functionals. The visual appearance of this bias is illustrated in Figure 6.13 (a), and a close up view in (b). Note that we only show a single trial, but since the SNR is very high (SNR=42dB), the same shift in peak locations is observed for all realizations. Even if we remove noise altogether, peaks do not align with the true values.

By looking at cost functionals of the form $J(\mathbf{s}) = J_1(\mathbf{s}) + \lambda J_2(\mathbf{s})$, where $J_1$ is the data-fidelity term, and $J_2$ is the regularizer, we allow solutions which have a higher model-fit residual (lower data-fidelity) if they have a lower regularizing term. In general, this may force the distribution of the estimates for repeated trials (with different noise realizations) to be non-symmetric around the true values, and the mean values being different from the true values. For example, consider a simple inverse problem $y = kx + n$, where $n$ is scalar 0-mean Gaussian, and $x$ is treated as deterministic (non-random) unknown quantity. The Tikhonov regularization cost function is $(y-kx)^2+\lambda x^2$, and the solution is $\hat{x} = \frac{y}{k+\lambda/k}$. Hence $E[\hat{x}] = E[\frac{kx+n}{k+\lambda/k}] = x\frac{k}{k+\lambda/k}$. When $x \neq 0$ and $\lambda$ is strictly positive, the regularized estimate is biased. Analytical expressions for cost functions involving $\ell_1$ or $\ell_p$ are much harder to obtain, thus we rely on computer simulations.

The other part of the explanation is that we are using an approximate measure of sparsity. Recall that imposing a penalty on sparsity (the number of non-zero elements) is not computationally tractable, thus we are forced to use approximations. The $\ell_1$ approximation, while having many favorable properties for sparse source localization, results in solutions which are in general different from those obtained using sparsity. In our theoretical analysis (Section 7)) we show that under some assumptions on the num-

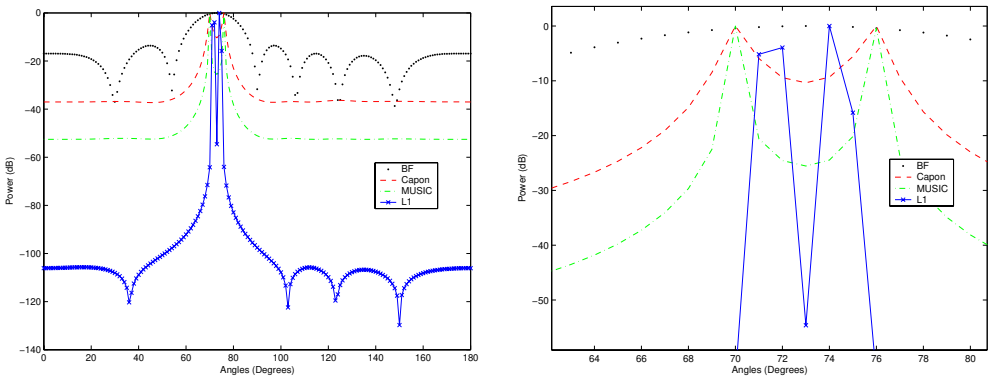



**Figure 6.13.** Bias: two sources with DOAs of 70° and 76°. SNR = 42 dB. (a) Full plot . (b) Detail of a .

ber of sources and their separation and with no noise present, the $\ell_1$ solution exactly matches that of $\ell_0$, which in turn exactly equals the true source locations (provided source locations are on the grid). However, in array processing applications these conditions are often not met, resulting in solutions which are biased even when no noise is present and when the true positions of the sources fall exactly on the considered grid of locations.

The $\ell_p$ approximation with $p \ll 1$ is a much better approximation than $\ell_1$, but the resulting cost function is not convex. We cannot develop a reasonable procedure for finding the global minimum, but, as we mentioned already, it appears that the local minima which are obtained with our iterative procedure starting from the beamforming solution have peaks which fall very close to the true source locations. In Chapter 7 we provide a result stating what conditions are necessary for the global optimum of the $\ell_p$ cost function to be equal to the true source locations. The required conditions are milder then those of $\ell_1$, but the global optimality is lost, so the bias may now appear due to the convergence to a local minimum.

#### Bias for non-zero mean signals with data combination by averaging

We include several plots to illustrate the discussion. We start with the non-zero mean case and the averagig version of our technique. Figure 6.13 (a) shows the spectra of $\ell_p$, beamforming, and MUSIC and Capon's methods. The SNR has been set very high (SNR = 40 dB) so that both MUSIC and Capon are able to resolve the two sources. The array is uniform and linear with $M = 7$ sensors. The separation between the sources is 6°, and the $\ell_1$ technique converges to a biased answer, as is readily apparent from the magnified portion of the plot.

A difficulty with investigating bias is that it depends on the regularization parameter selection. If a particular regularization parameter leads to unbiased estimates of two fairly closely-spaced sources, then by increasing the regularization parameter sooner or later the two estimates become biased or merge into a single estimate. The question

of the choice of the regularization parameter is very difficult, and we have an efficient method for its automatic choice only for constrained version of $\ell_1$ regularization. In general, we have observed that all our techniques are biased for a fixed non-zero regularization parameter (and zero regularization parameter is not useful since it leads to no regularization and an ill-posed problem when $\mathbf{A}$ is overcomplete). However, for the case of $\ell_1$ we observed that if $\lambda$ is decreased to zero as SNR approaches $\infty$ then the technique is asymptotically unbiased, provided that the number of sources is resolvable by the array[10]. When $\lambda$ is selected automatically in the constrained $\ell_1$ formulation by the discrepancy principle, $\lambda$ indeed decreases to zero as SNR increases. For low SNR, $\lambda$ is chosen quite large to suppress noise, and leads to more notable bias. With high SNR $\lambda$ is selected very low, and causes little bias. However, the issue of asymptotic unbiasedness is also dependent on the numerical stability and convergence properties of the optimization techniques involved, so tolerances have to be selected appropriately to observe the lack of asymptotic bias. The $\ell_p$ counterpart on the other hand is asymptotically biased. This may happen either due to the nonconvexity of the cost function and convergence to local minima, or due to numerical convergence properties of our implementation of the iterative half-quadratic algorithm.

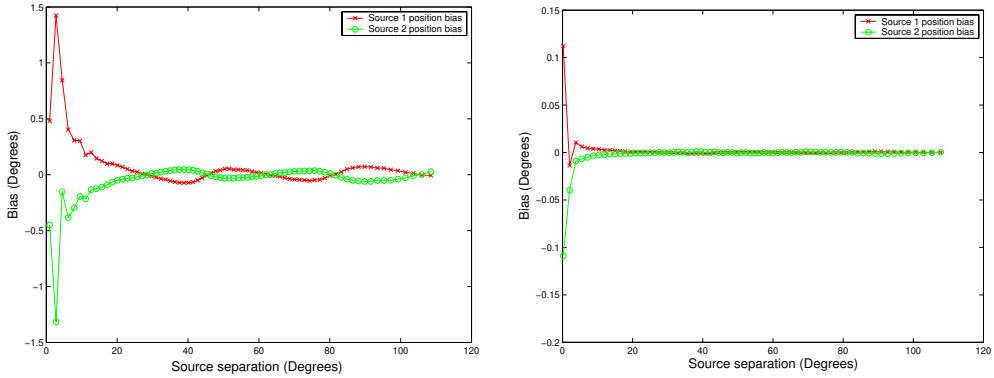



**Figure 6.14.** Bias vs. separation for non-zero mean $\ell_1$ with averaging. (a) SNR =20 dB (b) SNR = 60 dB.

What we see in Figure 6.13 is the second type of bias only (i.e. the inherent bias, and not the grid bias), since the sources have been conveniently placed on the grid locations. Placing the sources in between the grid points usually produces estimates on the nearmost grid points. In Figure 6.14 we get rid of the grid bias in a different way, by using the multi-scale procedure described in Section 5.3, which refines the grid with each scale. After several scales the grid becomes fine enough so that its bias effects can be ignored. To produce the figure we used the constrained version of $\ell_1$ processing with an automatic selection of the regularization parameter. The number of sources

[10]See 6.1.5 for a discussion on the number of resolvable sources.

in the plot is two. We plot the bias of the two sources versus the angular separation between the two sources. One source is fixed, and the second one is placed at various separations from the first one. For each separation the experiment is repeated 50 times with different noise realizations. We plot the biases of the estimates for both sources.

The bias curves in Figure 6.14 appear to have a region where bias is most pronounced for closely-spaced sources, and a region for well-separated sources where the bias has an oscillatory pattern (zooming in is necessary to see it in plot (b)). As we mentioned, we choose the regularization parameter by the discrepancy principle in the $\ell_1$ constrained form and $\lambda$ decreases to zero as SNR increases; thus the technique is asymptotically unbiased. The amplitude of the peaks in the oscillatory portion of the bias appears to have a linear dependence on the standard deviation of noise. It is at first surprising that in plot (a) at very low separations (under about 5°) the bias increases with separation. There is a simple explanation. Since the SNR is not very high, the effect of the regularization parameter cannot be neglected, and the technique is biased. Under 5°, the two peaks are merged into a single one, in between the two true source locations. Hence, at first, by increasing the separation we increase the distance from the arithmetic mean to the true source locations which equals the bias. Beyond 5° separation, the two sources are resolved, and the estimates follow the source locations. The problem at higher separations is easier then at lower ones, hence the bias becomes smaller.

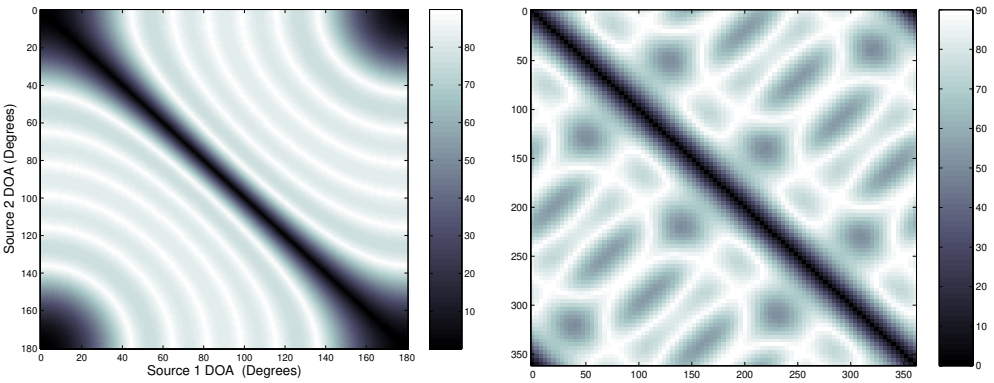



**Figure 6.15.** Angles between steering vectors corresponding to DOAS $\theta_1$ and $\theta_2$. (a) ULA with 8 sensors. (b) Cross array with 8 sensors.

Some insight into the shape of the bias curve comes from considering the plot of angles between the steering vectors associated with the true positions of the sources. For two sources with DOA's $\theta_1$ and $\theta_2$, with the associated steering vectors $\mathbf{a}(\theta_1)$ and $\mathbf{a}(\theta_2)$, we are interested in the angle between them $\angle(\mathbf{a}(\theta_1), \mathbf{a}(\theta_2))$. Actually, the angle between two complex vectors is not properly defined, since the definition in Euclidean spaces does not carry over: if $\mathbf{x}, \mathbf{y} \in \mathbb{C}^N$ then $\frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2}$ is a complex number. We are

interested in the ratio of the norm of the projection of $\mathbf{x}$ on $\mathbf{y}$ to the norm of $\mathbf{x}$. Thus we attach the following meaning to the notation: $\angle(\mathbf{x}_1, \mathbf{x}_2) = \cos^{-1}(\frac{|\mathbf{x}'\mathbf{y}|}{\|\mathbf{x}\|_2\|\mathbf{y}\|_2})$. By the Schwartz inequality $|\mathbf{x}'\mathbf{y}| \leq \|\mathbf{x}\|_2\|\mathbf{y}\|_2$, hence the notation is well defined, and takes values in $[0, .., \pi/2]$. For two sources, the angle $\phi = \angle(\mathbf{a}(\theta_1), \mathbf{a}(\theta_2))$ can be thought of as a measure of difficulty of the problem. If $\phi$ is close to $\pi/2$, then the two steering vectors are nearly orthogonal, and do not interfere with each other (for example in beamforming, the contribution of power from the source at $\theta_1$ to the steered beam at $\theta_2$ is close to zero. On the other hand, if $\phi$ is small, then the power measured by the array steered at either $\theta_1$ or $\theta_2$ depends on both of the signals, and in beamforming this corresponds to sidelobe interaction.

Hence a plot of $\phi$ for all possible pairwise combinations of $\theta_1$ and $\theta_2$ is of great interest, and we include it for a uniform linear array in Figure 6.15 (a). As we expect, for closely-spaced sources (near the main diagonal of the image), the steering vectors are nearly collinear, hence $\angle(\mathbf{a}(\theta_1), \mathbf{a}(\theta_2))$ is small. For well separated sources, the angle exhibits an oscillatory pattern near the value of 90°. This may have direct connection to the observed structure of the bias as function of source separation. A similar analysis can be done for more than two sources, but then we have to face defining a measure of distance between higher dimensional subspaces. One possibility is through the use of principal angles between subspaces. For nonuniform or nonlinear arrays plots of angles between pairs of steering vectors sometime have very interesting visual appearance. In plot (b) we make the plot for a cross array with 8 sensors. The cross array in the plot is composed of two perpendicular uniform linear arrays intersecting in their centers.

#### Bias for $\ell_1$-SVD with zero-mean signals

When we switch to zero-mean sources, we can no longer use the averaging formulation. We now switch to the $\ell_1$-SVD formulation described in Section 5.1.7. The zero-mean version has several important differences from the non-zero mean version and deserves a separate discussion.

The $\ell_1$-SVD version is biased as well. However, the structure of the bias is not the same as for the non-zero mean signals with averaging. Recall that the bias of our techniques depends on the regularization parameter selection. For the experiments in this section we selected the regularization parameter manually by subjective assessment at high SNR and held it fixed for all SNR. This was done due to the fact that our automatic selection method for $\ell_1$-SVD has not been fully investigated, and theoretical development applies to the low-noise case only. In the end of Section 5.1.7 we proposed a different scaling for the singular vectors, $\Lambda^2$ instead of $\Lambda$. We investigate the bias for both possibilities, since they have notable differences.

First we take a look at our original version of $\ell_1$-SVD with scaling by $\Lambda$ in Figure 6.16. The two signals are zero-mean and the SNR=10 dB in plot (a), and SNR=40 dB in plot (b). We plot the bias of the estimates of source locations in degrees versus the separation of the source locations, also in degrees. The setup of the experiment is the same as for non-zero mean signals from the previous section, except for the source

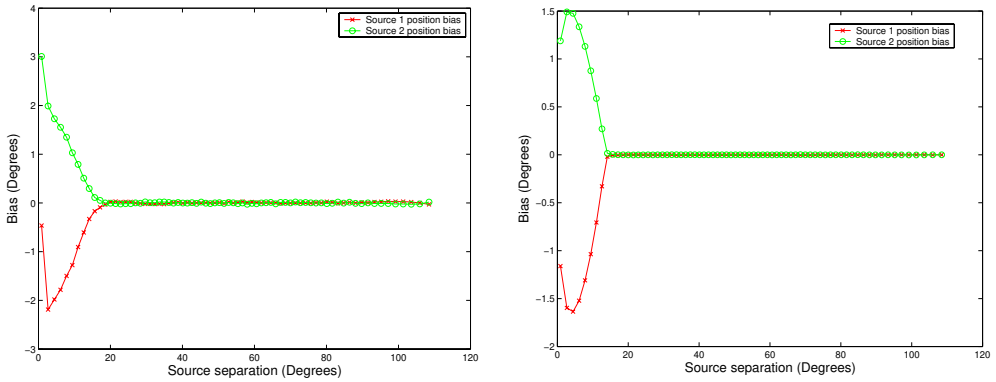



**Figure 6.16.** Bias vs. separation, $\ell_1$-SVD, scaling by $\Lambda$. (a) SNR =10 dB (b) SNR = 40 dB.

signals, which now have a zero temporal mean. We see in the figure that there is considerable bias for closely-spaced sources, which appears both at low and high SNRs. The amplitude does not decrease dramatically for higher SNR, since we are not changing the regularization parameter. For well separated sources there is a notable difference from the non-zero mean version, the technique appears to be unbiased at all SNR. The oscillatory pattern is absent.

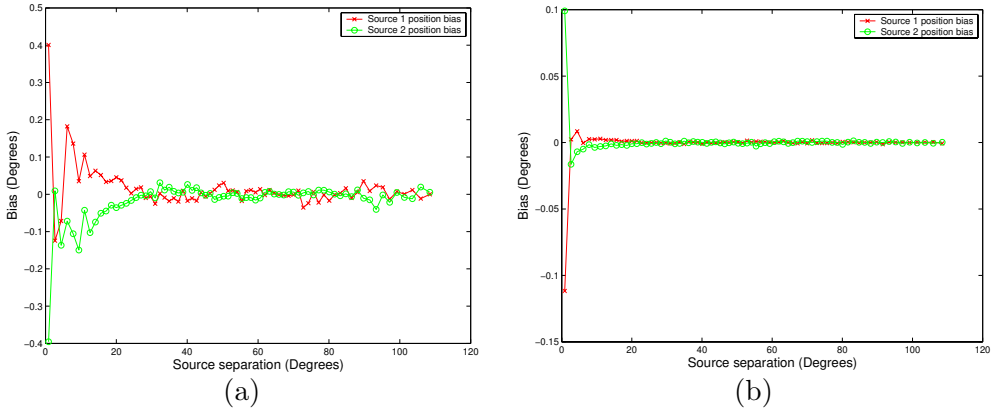



**Figure 6.17.** Bias vs. separation, $\ell_1$-SVD, proposed scaling by $\Lambda^2$. (a) SNR =10 dB (b) SNR = 40 dB.

Now we consider our proposed modification, where the scaling of the singular vectors is by $\Lambda^2$, and not by $\Lambda$. The results appear in Figure 6.17. To our great surprise, for high SNR in plot(b) most of the bias disappears. There is a small biased region for very closely-spaced sources, but the magnitude of this bias is smaller than the corresponding one for scaling by $\Lambda$. In plot (a), SNR is 10 dB, and some bias appears for closely-spaced sources. However, again the amplitude is smaller than the corresponding one for scaling $\Lambda$. For well-separated sources the new scaling also appears to produce no bias in the

estimates for all SNR. Further investigation of these issues will follow the completion of the thesis.

## ■ 6.4 Variance and the CRB

### Nonzero mean signals

The comparison with the Cramer-Rao Bound (CRB) has become an important ingredient of the analysis of any source localization method for array processing. The CRB puts a lower limit on the variance of any unbiased estimator. When an estimator meets the CRB, it is called efficient, but there may not exist any efficient estimators. Many of the existing estimators of source location are biased, but they have the property of asymptotic unbiasedness, as either the number of snapshots or the number of sensors or both approach infinity. There exists an extension of the CRB which deals with biased estimators, but it is generally agreed that it has limited practical value. In our case, it is difficult to even compute it, since it requires the estimation of the derivative of the bias. For a review of the CRB refer to Appendix A, and for a derivation of the CRB for source localization refer to [21].

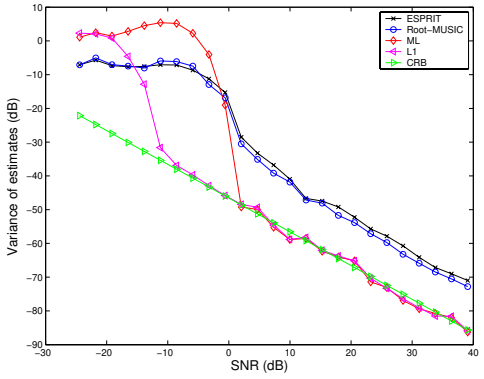



**Figure 6.18.** CRB for non-zero mean sources, comparison with variances of ESPRIT, Root-MUSIC, ML, and $\ell_1$

We start our analysis of variance with nonzero mean signals using the averaging version of our technique. Instead of trying to estimate the derivative of the bias needed for the biased CRB analysis, we note that for particular values of the separation, our estimates are in fact unbiased for all SNR based on our simulations. Instead of the global CRB analysis we consider local properties of the variance for particular separations of the sources. In Figure 6.14, the separations between the two sources for which our estimator is unbiased correspond to the zero crossings of the two curves. The derivation of the analytical expressions for the variance of the $\ell_1$ technique is not an easy task, and we instead conduct an empirical analysis using computer simulations.

Figure 6.18 shows the variance of the $\ell_1$ estimator in the constrained form with automatic selection of the regularization parameter and multi-scale grid refinement (to eliminate grid bias). The two sources are at 42.83° and 88.33°, with the separation being 45.5° (this corresponds to an unbiased estimate)[11]. We compare the variance of $\ell_1$ source localization to that of ESPRIT, Root-MUSIC, Maximum Likelihood, and to the CRB. The number of trials for each value of SNR is 50, and the curve plotted for each estimator corresponds to the variance of the first source; the variance of the second source is very similar, and is removed to avoid obfuscation of the figure. There are several important observations about the picture. First of all, the variance of the $\ell_1$ technique follows the CRB closely above -10 dB SNR, and the bias at the considered source separation is zero for all SNR, hence locally the technique is efficient for non-zero mean signals. The ESPRIT and Root-MUSIC methods have been developed for zero mean signals, and they are not able to take advantage of the portion of signal power which is contained in the mean [12], hence the variance falls above the CRB by about 15 dB. Finally, the breakdown zone for the $\ell_1$ technique appears at much lower SNR than the one for ML, ESPRIT, and Root-MUSIC. This supports the claim that sparse regularization leads to higher robustness to noise in the non-zero mean case.

It is important to remember that the sparsity regularization framework is suitable for sparse signal fields only, where the number of sources is considerably less then the number of sensors in the array. When we consider number of sources comparable to the number of sensors, the performance degrades, and there may not exist any unbiased regions, or the technique may even produce completely unusable spectra.

#### $\ell_1$-SVD processing in the zero-mean case

Similar to the non-zero mean case, in order to compute the CRB we choose a separation of the two sources for which the estimation is unbiased. In the zero-mean case with $\ell_1$-SVD this choice is much easier to make, since for all large separations the technique is unbiased at all SNR. In the experiment a single regularization parameter is chosen by subjective assessment and is kept across all SNRs.

Figure 6.19 compares the variance of the $\ell_1$-SVD technique [13] with the variances of ESPRIT, Root-MUSIC, Deterministic Maximum Likelihood (DML), and the CRB. It can be seen that for zero-mean sources the technique again meets the CRB at high SNR. However, at low SNR the threshold region appears at the same SNR as that of other techniques. The reason for the difference with the non-zero mean case is that now we do not have the advantage of being able to use the information present in the mean

[11] The important parameter is not the difference of the DOA's in degrees, but the difference of cosines of the two angles, since for uniform linear arrays the beampattern is proportional to the spatial DFT of sensor measurements parameterized by the cosines of DOA's, and resolution depends on separation of the sources in terms of differences of cosines of DOAs

[12] The signals are modeled as $u_i(t) = 1 + \nu_i(t)$, where $\nu_i(t)$ are independent normal random variables with zero mean and standard deviation $\sigma = 0.2$.

[13] We are using the version of $\ell_1$-SVD which has scaling by $\Lambda$, and not by $\Lambda^2$, but according to our limited simulations the variances of the two are the same.

(which ESPRIT, and Root-MUSIC cannot take advantage of even when the mean is present).

In order to compare the variance with the CRB we have selected the two sources to be well-separated. However, as we described previously, our technique is especially useful at low SNR in its biased regions, when the separation between the sources is small. We illustrated this advantage on single experiments in Section 5.1.7. Our CRB analysis shows that additionally, in the unbiased regions, $\ell_1$-SVD performs as well as the other techniques.

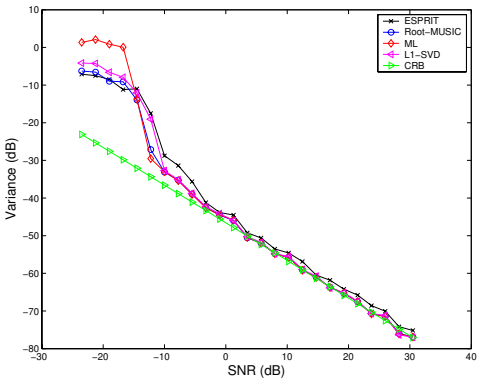



**Figure 6.19.** CRB for zero mean *uncorrelated* sources, comparison with variances of ESPRIT, Root-MUSIC, ML, and $\ell_1$, DOAs 42.83° and 73.33°

There is however an advantage of using our framework in unbiased regions as well, and it comes up when we consider correlated sources. It is well known that the performance of MUSIC and ESPRIT suffers from the correlation of the sources. From our discussion in Section 5.1.7 we expect that correlated sources will not lead to a major loss in performance for our approach, since we are not affected as much by vectors in the noise subspace being taken into the signal subspace.

For the experiment, everything is left as before, except that the sources are now correlated with the mixing matrix $\left[\begin{smallmatrix} 1 & 0.9 \\ 1 & 1 \end{smallmatrix}\right]$ normalized so that the columns have unit norm. Also, due to the fact that there is only one singular value remaining in the signal subspace, we take $\mathbf{Y}_{SV}$ from Section 5.1.7 as the first singular vector only (or in general as a matrix of all signal subspace singular vectors scaled by the corresponding singular values). Figure 6.20 shows the outcome of the experiment. As expected, the variance of MUSIC and ESPRIT are well above the CRB. Maximum Likelihood is not affected by the correlated sources as long as the initialization by MUSIC is reasonably close to the true source locations. However, the threshold drop of performance appears much earlier for MUSIC when the sources are correlated, leading to a similar early performance drop for ML. The variance of $\ell_1$-SVD matches the CRB fairly closely at high SNR, and at lower SNR it starts to deviate from the CRB, and finally exhibits a sharp rise in variance.

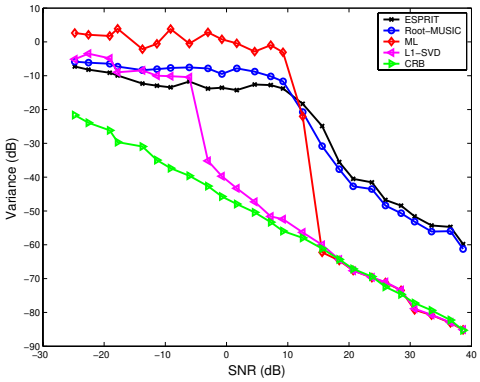



**Figure 6.20.** CRB for zero mean *correlated* sources, comparison with variances of ESPRIT, Root-MUSIC, ML, and $\ell_1$, DOAs: 42.83° and 73.33°.

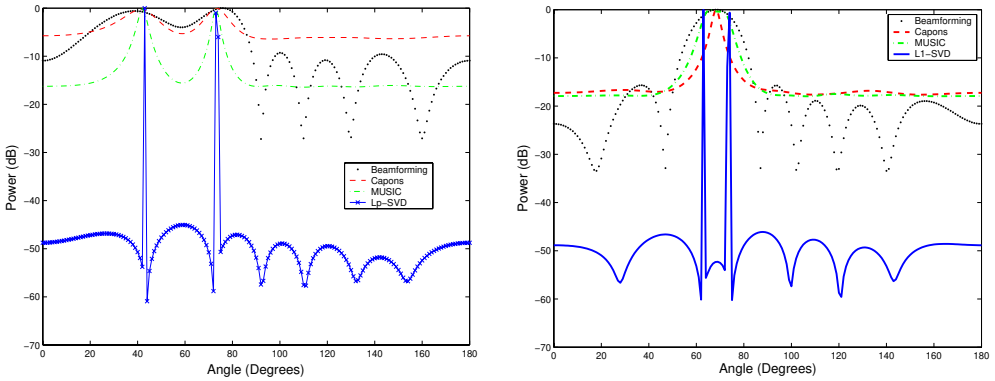



**Figure 6.21.** Spectra for correlated sources, $SNR = 20dB$: (a) DOAs: 42.83° and 73.33°. (b) DOAs: 62.83° and 73.33°.

There is a noticeable gain over Maximum Likelihood with MUSIC initialization in this case. However it is even greater for closely-spaced sources (although then our technique is biased). This happens due to the widening of the mainlobe of the MUSIC spectrum for correlated sources. We illustrate this for single trials in Figure 6.21. The covariance matrix of the two sources is again $\left[\begin{smallmatrix} 1 & 0.9 \\ 1 & 1 \end{smallmatrix}\right]$, normalized to unity. The separation in plot (a) is about 30°, and the plot illustrates what we mean by the widening of the mainlobes of the MUSIC spectrum (same happens for Capon's method as well). No such widening occurs for our technique. In plot (b) the sources are brought much closer, to a separation of about 10°, and we observe that neither Capon's method nor MUSIC are able to resolve the two peaks. The $\ell_1$-SVD method, although it is biased, is able to resolve them well.

Chapter 7

# Theoretical Analysis: solving the $\ell_0$ problem by $\ell_p$ and related topics

On numerous occasions we referred to the fact that the very hard problem of $\ell_0$ regularization can be solved exactly via $\ell_p$ and $\ell_1$ regularization under certain conditions. Readers interested in these conditions opened the manuscript on the right page, we describe (and prove) them in this chapter.

We briefly review the problem that we set out to analyze. Recall that in the context of signal representation using sparse bases we arrived at the following problem: $\min \|\mathbf{x}\|_0^0$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$. [1] In plain English, the goal of the problem is to find the sparsest representation of $\mathbf{y}$ in terms of an overcomplete basis $\mathbf{A}$. We also have a related problem, $\min \|\mathbf{x}\|_p^p$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$, for $p \leq 1$. The claim is that the solution $\hat{\mathbf{x}}$ to the former problem ($\ell_0$) is also the solution to the latter problem ($\ell_p$) if $\hat{\mathbf{x}}$ is sparse enough with respect to $\mathbf{A}$. For the case $p = 1$, this is very surprising since the $\ell_1$ problem is convex, and can be efficiently solved using linear programming or second order cone programming (for real and complex data respectively) as described in Section 4.1.1. For general $p$, finding the global minimum is considerably harder, but in return the sparsity requirements for $\hat{\mathbf{x}}$ are lower, especially for $p$ close to 0. The equivalence results are for the case when there is no noise, but they serve as a strong supporting argument for the use of the noisy $\ell_p$ penalization as well.

Before relating the $\ell_0$ problem to $\ell_1$ and $\ell_p$ problems, we first address the question of uniqueness of the $\ell_0$ problem, to make sure that $\ell_0$ solutions are useful. We start the chapter by introducing the notion of rank-K unambiguity, which leads to a necessary[2] and sufficient condition for the uniqueness of solutions to the $\ell_0$ problem. Then we introduce a different measure, maximum absolute dot-product of pairs of columns of $\mathbf{A}$, $M(\mathbf{A})$, and relate it to rank-K unambiguity. This leads to another sufficient uniqueness condition for the $\ell_0$ problem. The reason that we introduce a second measure, $M(\mathbf{A})$, is that we use it next to prove the equivalence between $\ell_1$ and $\ell_0$ problems if $\hat{\mathbf{x}}$ is sparse

[1] Recall that $\|\mathbf{x}\|_0^0$ refers to the number of nonzero elements of $\mathbf{x}$.

[2] We should be careful with the meaning of "necessary" that is used here. The condition is necessary to guarantee that given an arbitrary signal $\mathbf{y}$, if the solution $\hat{\mathbf{x}}$ to the $\ell_0$ problem is sparse enough, then it is the unique solution. However, for a fixed $\mathbf{y}$ (other $\mathbf{y}$'s are not considered) this condition may be too restrictive and not necessary.

enough with respect to $\mathbf{A}$. We finish the exposition of $\ell_1$ penalization by exploring the dependence of $M(\mathbf{A})$ on the size of $\mathbf{A}$ using results from the theory of spherical codes.

Next, we switch to a general $p \leq 1$. We bring forward two more measures of $\mathbf{A}$ which in turn lead to the equivalence of $\ell_p$ and $\ell_0$ problems. The second measure is especially interesting, since it shows that as $p \to 0$, the condition for the equivalence approaches the condition for the uniqueness of solutions of the $\ell_0$ problem.

Lastly, we discuss a preliminary analysis of the noisy version of $\ell_1$ penalization. The result that we prove is rather interesting, but its purpose is mainly to stimulate further analysis of the noisy problem.

The analysis in this chapter was mainly incited by two papers, [8] and [9]. They consider an important special case when $\mathbf{A}$ is composed of two orthogonal bases and under this assumption prove the equivalence of $\ell_0$ and $\ell_1$ problems. The main contribution of our work is the extension of their result to any overcomplete basis $\mathbf{A}$, as well as connecting $M(\mathbf{A})$ with the more direct measure of rank-K unambiguity. In addition we consider conditions for the equivalence of $\ell_0$ and $\ell_p$, and look at noisy $\ell_1$ penalization. Also, all the results of this chapter are valid in general for complex-valued quantities, except the spherical code discussion in Sections 7.2.2 and 7.2.3 which applies to the real case only.

## ■ 7.1 $\ell_0$ conditions

Before starting to prove the equivalence conditions for the $\ell_p$ and $\ell_0$ problems, first we would like to find conditions such that the $\ell_0$ cost function has a unique solution. The general form of such conditions is that if the optimal solution $\hat{\mathbf{x}}$ is sparse enough with respect to $\mathbf{A}$ then the solution is unique. We propose the use of rank-K unambiguity (which we define in the next section) which leads to a condition on the sparsity of $\hat{\mathbf{x}}$ for unique solutions (minima) of the $\ell_0$ cost function. In order to eventually relate $\ell_0$ and $\ell_1$ problems we describe another measure, $M(\mathbf{A})$, which tells how well separated the columns of $\mathbf{A}$ are. Then we prove a bound on $M(\mathbf{A})$ relating it to rank-K unambiguity. This leads to an alternative sufficient condition for the uniqueness of solutions to the $\ell_0$ cost function in terms of $M(\mathbf{A})$.

### ■ 7.1.1 Definition of rank-$K$ unambiguity

Take $\mathbf{A} \in \mathbb{C}^{M \times N}$ with columns $\mathbf{a}_i$, $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_N]$. We call $\mathbf{A}$ rank-$K$ unambiguous[3] if any set of $K$ columns of $\mathbf{A}$ is linearly independent, but this is not true for $K+1$ (i.e. either $K = N$, and no additional columns exist, or there exists a set of $K+1$ columns which are linearly dependent). If columns of $\mathbf{A}$ are linearly dependent (i.e. $\mathbf{A}$ does not have full column rank), then $K$ can also be defined as the cardinality of a linearly dependent set of columns of $\mathbf{A}$ with the smallest number of columns, minus 1. If $\mathbf{A}$ has full column rank then $K = N$. Let $R$ be the rank of $\mathbf{A}$, then the following holds

[3] Our definition is motivated by rank-ambiguity of [50].

(assuming $\mathbf{A}$ has at least one non-zero entry, and $M$, $N \geq 1$):

$$1 \leq K \leq R \leq \min(M, N) \tag{7.1}$$

When the set of columns is linearly dependent, $K$ can take any value from 1 to $R$. For example if a vector consisting of all zeros belongs to the set of columns of $\mathbf{A}$, then $K = 1$ independent of $R$, $M$, and $N$. On the other hand, for a random matrix with each element being a standard Gaussian random variable, if $N > M$, then $K = M = R$ with[4] probability 1. A matrix $\mathbf{A}$ having the value of $K$ in between 1 and $R$ can be easily constructed by appending a column which is a linear combination of $K$ other linearly independent columns to the previously described random matrix.

What we have defined can be called a weak notion of rank-$K$ unambiguity, but it is also possible to define a strong notion of rank-$K$-unambiguity by requiring weak rank-$K$ unambiguity, and in addition the property that if the columns are linearly dependent, then *all* sets of $K + 1$ columns are linearly dependent (as opposed to at least one). Strong rank-$K$ unambiguity implies that $\mathbf{A}$ has rank $K$. This can be easily shown by considering sets $\Phi_i = \{\mathbf{a}_1, ...\mathbf{a}_K, \mathbf{a}_{K+i} : i \in \{1, .., N-K\}\}$. All $\Phi_i$ are linearly dependent, thus each $\mathbf{a}_{K+i}$ belongs to the $K$-dimensional span of $\{\mathbf{a}_1, ..., \mathbf{a}_K\}$, thus all the columns of the matrix belong to it, and the rank is $K$. We are mainly concerned with the weak notion of rank-$K$ unambiguity, but situations with the strong notion are common in source localization applications, and may provide some further insight into the question of uniqueness that we consider.

### ■ 7.1.2 Uniqueness of $\ell_0$ regularization

Consider the $\ell_0$ problem

$$\min \ \|\mathbf{x}\|_0^0 \ \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{7.2}$$

where, as before, $\|\mathbf{x}\|_0^0$ is the count of nonzero entries of $\mathbf{x}$. The matrix $\mathbf{A}$ is $M \times N$, and throughout this chapter we consider the case where $N > M$, (in fact, for the array-processing application in the rest of the thesis, typically $N \gg M$). Suppose that $\mathbf{y}$ is a sparse combination of the columns of $\mathbf{A}$ with some coefficients $\mathbf{x}^*$, i.e. $\mathbf{y} = \mathbf{A}\mathbf{x}^*$, and $\|\mathbf{x}^*\|_0^0 = L$. We are interested in the conditions on the sparsity of $\mathbf{x}^*$ (i.e. the number of non-zero coefficients, $L$), such that the solution $\hat{\mathbf{x}}$ to (7.2) is unique, has $L$ non-zero elements, and is achieved at $\mathbf{x}^*$. Alternatively, if we do not know the underlying signal $\mathbf{x}^*$, then if we have a solution $\hat{\mathbf{x}}$ to (7.2), we are interested in its uniqueness.

**Theorem 1 (uniqueness of solutions to the $\ell_0$ cost function).** *Assume that $\mathbf{A}$ is rank-$K$ unambiguous, and has $N > M$ columns. Also suppose that for some $\mathbf{x}^*$, $\mathbf{y} = \mathbf{A}\mathbf{x}^*$, and $\|\mathbf{x}^*\|_0^0 = L$. Then (7.2) has a unique solution $\hat{\mathbf{x}} = \mathbf{x}^*$ for all such $\mathbf{y}$ if and only if $L < (K + 1)/2$.*

[4]The set where this does not hold has measure zero.

*Proof.* Since $N > M$, and $\mathbf{A}$ is rank-K unambiguous, there exists a linearly dependent set of $K+1$ columns of $\mathbf{A}$. If $2L > K$, then there exists a set of $2L$ column vectors, $\mathbf{a}_1, .., \mathbf{a}_{2L}$, which is linearly dependent. That means that there exists a set of coefficients $\alpha_1, .., \alpha_{2L}$ such that $\sum_{i=1}^{2L} \alpha_i \mathbf{a}_i = 0$, or $\sum_{i=1}^{L} \alpha_i \mathbf{a}_i = \sum_{i=L+1}^{2L} -\alpha_i \mathbf{a}_i$. That means that any signal $\mathbf{y}$ lying on the line $\gamma \sum_{i=1}^{L} \alpha_i \mathbf{a}_i,\ \gamma \in \mathbb{C}$, admits two possible representations with sparsity $L$ in terms of the overcomplete basis $\mathbf{A}$. This proves that the condition is necessary. The term necessary requires a clarification: it means necessary for the uniqueness of *all* $\mathbf{x}^*$ with sparsity less than or equal to $L$. That is to say, it is necessary to prevent the existence of $\mathbf{x}^*$ with sparsity $L$, such that a different vector $\hat{\mathbf{x}}$ with the same or lower sparsity also satisfies $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}$. Yet, for a particular signal $\mathbf{x}^*$ the condition may be overly restrictive and not necessary.

If $L \leq K/2$, then $\sum_{i=1}^{2L} \alpha_i \mathbf{a}_i = 0$ has $\alpha_i = 0, \forall i$ as its only solution, since the smallest cardinality among linearly dependent sets is $K+1$. This means that no two distinct signals with sparsity less than or equal to $\frac{K}{2}$ can yield the same $\mathbf{y}$. Thus $L \leq K/2$, or $L < (K+1)/2$ is also a sufficient condition. ■

We would like to say more about the necessary condition. We proved that if the condition is not satisfied then there exist signals $\mathbf{x}^*$ which have non-unique sparse representations. However, for a particular signal $\mathbf{x}^*$ the representation can be unique. Yet, when $N \gg M$, the number of lines (one-dimensional subspaces) of the form $\gamma \sum_{i=1}^{L} \alpha_i \mathbf{a}_i,\ \gamma \in \mathbb{C}$ allowing ambiguous sparse representations explodes combinatorially (consider the number of ways to select $2L$ columns out of $N$ possible ones). Practically that may mean that when we allow deviations from $\mathbf{y}$ to account for noise, then non-uniqueness becomes a likely outcome. Lines have measure zero, but if we allow uncertainty, then the total measure of all lines may become significant.

Also, when $K = M$, then if $L \geq M$ then the sparse representation is not unique for all $\mathbf{y}$ (all underlying $\mathbf{x}^*$). This happens due to the fact that any $M$ columns form a basis. Hence any signal can be represented as a linear combination of any $M$ columns of $\mathbf{A}$. When $\mathbf{A}$ has more columns than rows this translates to lack of uniqueness for every signal $\mathbf{y}$.

### ■ 7.1.3 Connection of rank-$K$ unambiguity with maximum dot-product of columns of A

Using rank-K unambiguity we arrive at a condition for uniqueness of solutions to (7.2). However, the measure of $\mathbf{A}$ that we use, K, (from rank-K unambiguity) depends discontinuously on the entries of $\mathbf{A}$. Linear dependence can be destroyed by an infinitesimal change of just one entry of $\mathbf{A}$. We now introduce a different measure of $\mathbf{A}$, which depends continuously on the entries of $\mathbf{A}$, and which will be used in Section (7.2) to connect the $\ell_0$ and $\ell_1$ problems. For the rest of the chapter, we assume that $\|\mathbf{a}_i\|_2^2 = 1$, i.e. all columns of $\mathbf{A}$ are normalized to unity. Define

$$M(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j|, \quad \text{where } \|\mathbf{a}_k\|_2 = 1, \forall k. \tag{7.3}$$

$M(\mathbf{A})$ measures how spread-out the columns of $\mathbf{A}$ are, and whether or not any two columns are almost collinear. Due to Schwartz inequality, $0 \leq M(\mathbf{A}) \leq 1$, and $M(\mathbf{A}) = 0$ if and only if $\mathbf{A}$ has orthogonal columns. Small values of $M(\mathbf{A})$ mean that the columns are almost orthogonal, whereas values close to unity mean that there are at least two columns separated by a very small angle.

Although $M(\mathbf{A})$ takes into account only the relation between pairs of columns of $\mathbf{A}$, it has a strong tie with linear dependence structure of larger sets of columns, and in particular with the cardinality of the minimum linearly dependent set of $\mathbf{A}$, $K+1$, where $K$ is as defined in Section 7.1.1. The following theorem relates the two.

**Theorem 2 (Relation of rank-$K$ unambiguity to $M(\mathbf{A})$).** *If the smallest cardinality among linearly dependent sets of columns of $\mathbf{A}$ is $K+1$, $K > 0$ (i.e. $\mathbf{A}$ is rank-$K$ unambiguous and not full column rank; the linearly independent case $K = N$ is excluded), then:*

$$M(\mathbf{A}) \geq \frac{1}{K} \tag{7.4}$$

*Proof.* First of all, $\mathbf{A}$ is rank-$K$ unambiguous, and as we stated in Section 7.1.2, we consider the case $N > M$, so there exists a set of $K+1$ linearly dependent columns of $\mathbf{A}$, $\{\mathbf{a}_{i_1}, ...\mathbf{a}_{i_{K+1}}\}$, which we collect into a new matrix $\tilde{\mathbf{A}} \in \mathbb{C}^{M\times(K+1)}$. Since we are reducing the set of possible columns, then the maximum absolute dot-product can not increase, hence $M(\tilde{\mathbf{A}}) \leq M(\mathbf{A})$. The rank of $\tilde{\mathbf{A}}$ is $K$, thus by an appropriate orthogonal transformation (which keeps all the pairwise dot-products invariant) we can rotate $\tilde{\mathbf{A}}$ to a matrix which has rows $K+1$ through $M$ as zeros. These rows do not change the dot products and can be ignored for our purposes. Let the singular value decomposition be $\tilde{\mathbf{A}} = \mathbf{U}\Lambda\mathbf{V}'$, then the required orthogonal transformation is $\mathbf{U}'$. The desired rotated matrix with the rows of zeros is $\mathbf{U}'\tilde{\mathbf{A}}$. Let $\bar{\mathbf{A}}$ denote $\mathbf{U}'\tilde{\mathbf{A}}$ with the irrelevant rows of zeros removed. Then the problem of bounding $M(\mathbf{A})$ reduces to minimizing $M(\bar{\mathbf{A}})$ for a matrix $\bar{\mathbf{A}}$ which is constrained to lie in $\mathbb{C}^{K\times(K+1)}$, and which has rank-$K$ unambiguity. That means that $M(\mathbf{A}) \geq M(\tilde{\mathbf{A}}) = M(\bar{\mathbf{A}})$. What remains to be found is the following:

$$\min M(\bar{\mathbf{A}}) = \min_{\mathbf{A}} \max_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|, \text{ where} \|\mathbf{a}_k\|_2 = 1, \forall k, \text{ and } \mathbf{A} \in \mathbb{C}^{K\times(K+1)}. \tag{7.5}$$

In Theorem 3 we prove that the optimal value of this problem is $\frac{1}{K}$. This proves the current theorem (Theorem 2), since $M(\mathbf{A}) \geq M(\bar{\mathbf{A}}) \geq \frac{1}{K}$. ■

Let us return to the problem in (7.5). A well-known result in the geometry of polytopes [51], communication theory, and sphere-packing on the Euclidean sphere is that

$$\min_{\mathbf{A}} \max_{i\neq j} \mathbf{a}_i'\mathbf{a}_j = \frac{-1}{K} \tag{7.6}$$

with the same conditions on $\mathbf{A}$ as in (7.5). The difference of this result from our problem lies in the absence of the absolute value on the dot-products of the columns. The result also states that the optimal polytope achieving this value is the regular simplex centered at the origin. (Here, we represent a polytope by a matrix which has the coordinates of its vertices as columns. We restrict the class of polytopes to have the vertices on the unit sphere). The problem of minimizing the maximum dot-product of the columns (7.6) is equivalent to maximizing the minimum angle between any of the two columns of $\mathbf{A}$. For the 2D case, the regular simplex is an equilateral triangle, having the central angle 120°, as shown in Figure 7.1.

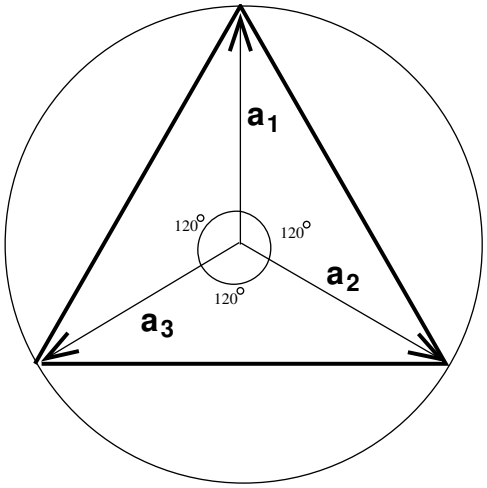



**Figure 7.1.** Regular simplex in $\mathbb{R}^2$: it maximizes the minimum pairwise angle between the vectors $\mathbf{a}_i$, $i = 1, 2, 3$.

Next we show that the regular simplex is also optimal in terms of $\min_{\mathbf{A}} \max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j|$ (with the absolute value), and the optimal value achieved is $\frac{1}{K}$. The proof was described to the author by Dr. Shor [52] and independently by R. Blume-Kohout [53]. The author would also like to thank W. Sun [54] for helpful discussions on the subject[5].

Returning to our discussion, first, to motivate the optimality of the regular simplex it is easy to see first that if $\mathbf{A}$ is a regular simplex then $M(\mathbf{A}) = |\frac{-1}{K}| = \frac{1}{K}$, and that all the pairwise dot-products are equal. Also, the same $M(\mathbf{A})$ is achieved for several polytopes related to the simplex: by multiplying any column $\mathbf{a}_i$ by $-1$ (i.e. reflecting any vertex with respect to the origin), we do not change $M(\mathbf{A})$. Likewise, by rotating the simplex with respect to the origin, $M(\mathbf{A})$ is left unaltered. Now we formally prove the theorem of simplex optimality for $M(\mathbf{A})$.

[5]The author also became aware of very interesting work in the field of packing of Grassmanian manifolds [55], which discusses not only minimizing $M(\mathbf{A})$ (which is equivalent to line-packing), but also packing planes, and higher-dimensional subspaces.

**Theorem 3 (Optimality of the simplex for line packing in $\mathbb{C}^K$).** *Let $\mathbf{A} \in \mathbb{C}^{K\times(K+1)}$. Then $M(\mathbf{A}) \geq \frac{1}{K}$. The equality is achieved for the regular simplex (allowing rotations and reflections of vertices around the origin).*

*Proof.* The optimum solution $\mathbf{A}$ for the problem, $\min_{\mathbf{A}} \max_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|$ is the same as the one for $\min_{\mathbf{A}} \max_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|^2$ (with the extra-square). The maximum of $K(K+1)$ numbers is always greater than their average, hence $\max_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|^2 \geq \frac{1}{K(K+1)} \sum_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|^2$. (There are $K(K+1)$ dot-products $\mathbf{a}_i'\mathbf{a}_j$, where $i \neq j$). This quantity is easier to handle. In particular,

$$\max_{i\neq j} |\mathbf{a}_i'\mathbf{a}_j|^2 \geq \frac{1}{K(K+1)} \sum_{i\neq j, i=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_j|^2 = \tag{7.7}$$

$$= \frac{1}{K(K+1)} \left( \sum_{i,j=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_j|^2 - \sum_{i=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_i|^2 \right) = \frac{1}{K(K+1)} \left( \sum_{i,j=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_j|^2 - (K+1) \right) \tag{7.8}$$

The last equality comes from the fact that each $\mathbf{a}_i$ has unit norm. Now what we are left with is putting a lower bound on $\sum_{i,j=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_j|^2$. The sum of all squared dot-products is the Frobenius norm of $\mathbf{A}'\mathbf{A}$, which is equal to the Frobenius norm of $\mathbf{A}\mathbf{A}'$, which in turn is equal to the sum of squares of eigenvalues $\lambda_i$ of $(\mathbf{A}\mathbf{A}')$, i.e.

$$\sum_{i,j=1}^{K+1} |\mathbf{a}_i'\mathbf{a}_j|^2 = \|\mathbf{A}'\mathbf{A}\|_{fro}^2 = \|\mathbf{A}\mathbf{A}'\|_{fro}^2 = \sum_{i=1}^{K+1} \lambda_i^2 \tag{7.9}$$

The matrix $\mathbf{A}\mathbf{A}'$ is Hermitian positive semi-definite, (take any $\mathbf{x} \in \mathbb{C}^K$, then $\mathbf{x}'(\mathbf{A}\mathbf{A}')\mathbf{x} = \|\mathbf{A}'\mathbf{x}\|_2^2 \geq 0$), so all its eigenvalues are real and positive, $\lambda_i \geq 0$.

Write $\mathbf{A}\mathbf{A}'$ as a sum of outer products of columns of $\mathbf{A}$: $\mathbf{A}\mathbf{A}' = \sum_{i=1}^{K+1} \mathbf{a}_i\mathbf{a}_i'$. Each $\mathbf{a}_i\mathbf{a}_i'$ is a projection matrix (recall that we assume throughout that $\|\mathbf{a}_i\|_2^2 = 1$), so trace $(\mathbf{a}_i\mathbf{a}_i') = 1$, for $i = 1, .., K+1$. That means that

$$\text{trace } (\mathbf{A}\mathbf{A}') = \text{ trace } (\sum_{i=1}^{K+1} \mathbf{a}_i\mathbf{a}_i') = \sum_{i=1}^{K+1} \text{ trace } (\mathbf{a}_i\mathbf{a}_i') = K+1 \tag{7.10}$$

The trace of a matrix is also the sum of its eigenvalues, so $\sum_{i=1}^{K} \lambda_i = \text{ trace } (\mathbf{A}\mathbf{A}') = K+1$. Now we want to minimize $\|\mathbf{A}\mathbf{A}'\|_{fro}^2 = \sum_{i=1}^{K} \lambda_i^2$, with the constraint that $\sum_{i=1}^{K} \lambda_i = K+1$, and $\lambda_i \geq 0$ for all $i$. The minimum is achieved when all the eigenvalues are equal[6], i.e. $\lambda_i = \frac{K+1}{K}$.

[6]We prove the claim that $\sum_{i=1}^{N} \lambda_i^2$ subject to $\sum_{i=1}^{N} \lambda_i = 1$ and $\lambda_i \geq 0$ is minimized when $\lambda_i = \frac{1}{N}$ for all $i$. The fact that $\lambda_i$ are eigenvalues is irrelevant, we treat them as plain good old reals for this proof.

We have that the optimal value of $\sum_{i=1}^{K} \lambda_i^2$ is $\sum_{i=1}^{K} (\frac{K+1}{K})^2 = \frac{(K+1)^2}{K}$, which implies that $\sum_{i,j=1}^{K+1} |\mathbf{a}_i' \mathbf{a}_j|^2 \geq \frac{(K+1)^2}{K}$. Returning to our original quantities, that means that

$$\max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j|^2 \geq \frac{1}{K(K+1)} \left( \sum_{i,j=1}^{K+1} |\mathbf{a}_i' \mathbf{a}_j|^2 - (K+1) \right) \geq \tag{7.11}$$

$$\geq \frac{1}{K(K+1)} \left( \frac{(K+1)^2}{K} - (K+1) \right) = \frac{1}{K^2} \tag{7.12}$$

This immediately leads to the desired result, $M(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j| \geq \frac{1}{K}$. ■

An interesting observation is that for most of the proof the fact that we have $K+1$ vectors is inconsequential, so the following generalization is also true: suppose $\mathbf{A} \in \mathbb{C}^{M \times N}$, then $M(\mathbf{A}) \geq \sqrt{\frac{N-M}{(N-1)M}}$.

### ■ 7.1.4 Another condition for the uniqueness of $\ell_0$ regularization

The condition for the uniqueness of the $\ell_0$ problem, Theorem 1, in conjunction with the result relating rank-K unambiguity with $M(\mathbf{A})$ gives rise to another sufficient condition:

**Theorem 4 (Uniqueness of solutions for $\ell_0$ optimization through $M(\mathbf{A})$).** *If $L < \frac{1/M(\mathbf{A})+1}{2}$ then if there exists a vector $\hat{\mathbf{x}}$ with $\|\hat{\mathbf{x}}\|_0^0 = L$ satisfying $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}$, then it is the unique solution of the $\ell_0$ problem (7.2).*

*Proof.* This holds due to the fact that $1/M(\mathbf{A}) \leq K$, thus $(1/M(\mathbf{A}) + 1)/2 \leq (K+1)/2$, thus the new condition implies the previous sufficient condition in Theorem 1.

In effect we derive a sufficient condition which is less tight. The reason that we derived it lies in the fact that we use it later on in Section 7.2 to connect the $\ell_0$ and $\ell_1$ problems. Also, the first condition depends discontinuously on the elements of $\mathbf{A}$ (since it relies on rank-$K$ unambiguity of $\mathbf{A}$), but the new condition relies on $M(\mathbf{A})$, which is continuous with respect to the elements of $\mathbf{A}$. ■

Donoho and Huo in [8] have a similar result of uniqueness of solutions to the $\ell_0$ problem (but a completely different proof) for a special case when $\mathbf{A} \in \mathbb{C}^{M \times 2M}$ is composed of two orthogonal bases: $\mathbf{A} = [\Phi \ \Psi]$, where $\Phi'\Phi = \mathbf{I}$, and $\Psi'\Psi = \mathbf{I}$, and $\mathbf{I}$ is an identity matrix.

We have an optimization problem, and we find the optimal value using Lagrange multipliers. We ignore the positivity constraint for now, since even without it we show that the optimal values come out strictly positive. Let $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_N]'$. Also, let $\mathbf{1}$ be a column vector consisting of ones. Then the Lagrangian for the problem is $\boldsymbol{\lambda}'\boldsymbol{\lambda} + \alpha(\mathbf{1}'\boldsymbol{\lambda} - 1)$, where $\alpha$ is the Lagrange multiplier. The gradient is $2\boldsymbol{\lambda} + \alpha\mathbf{1}$. The Lagrangian is convex, hence the minimum value is achieved at $\boldsymbol{\lambda} = -\frac{\alpha}{2}\mathbf{1}$. Putting it back into the constraint, we have $\frac{\alpha}{2}\mathbf{1}'\mathbf{1} = \frac{\alpha}{2}N$, hence $\alpha = -\frac{2}{N}$. We get the optimal solution is $\boldsymbol{\lambda} = \frac{1}{N}\mathbf{1}$, i.e. all $\lambda_i = \frac{1}{N}$. This proves the claim.

They use a slightly different measure, $\tilde{M}(\mathbf{A}) = \tilde{M}(\Phi, \Psi) = \max_{i,j} |\phi_i' \psi_j|$, where $\Phi = [\phi_1, .., \phi_M]$, and $\Psi = [\psi_1, .., \psi_M]$. They arrive at the same result, stating that for $\ell_0$ uniqueness of $\mathbf{x}$ with $L$ non-zero elements, it is sufficient that $L < \frac{1/\tilde{M}(\mathbf{A})+1}{2}$. A better result for such a class of $\mathbf{A}$ is derived by Elad and Bruckstein [9]. They found that a tighter sufficient condition is:

$$L < \frac{1}{\tilde{M}(\mathbf{A})} \tag{7.13}$$

Since our results apply to more general scenarios, and include the case of two orthogonal bases, it is worthwhile to compare their results to ours. First of all, when $\mathbf{A}$ is indeed composed of two orthogonal bases, then our measure $M(\mathbf{A})$ is exactly the same as $\tilde{M}(\mathbf{A})$, since all dot products of two distinct vectors within an orthogonal basis are zero. Thus our condition reduces to that of Donoho and Huo for the special case when $\mathbf{A} = [\Phi\ \Psi]$. The fact that the sufficient condition of Elad and Bruckstein is tighter can be explained by the additional structure that is imposed on $\mathbf{A}$. Their result no longer holds when $\mathbf{A}$ is relaxed to be any overcomplete basis[7]. Also, when restricted to two merged orthogonal bases, their condition does not contradict our first condition based on rank-K unambiguity: using a corollary of Theorem 3, $M(\mathbf{A}) \geq \sqrt{\frac{N-M}{(N-1)M}}$. For the case of $\mathbf{A} \in \mathbb{C}^{M \times 2M}$, (i.e. $N = 2M$), $M(\mathbf{A}) \geq \sqrt{\frac{1}{(2M-1)}}$. However, the fact that $\mathbf{A} = [\Phi\ \Psi]$, where $\Phi$ and $\Psi$ are orthonormal can be used to get a tighter bound. Donoho and Huo [8] proved that for such case $M(\mathbf{A}) \geq \sqrt{\frac{1}{M}}$. Then, using the condition of Elad and Bruckstein, we obtain $L < \frac{1}{M(\mathbf{A})} \leq \sqrt{M} \leq \frac{1+M}{2}$. Also, $K = M$, since $\Phi$ and $\Psi$ are orthogonal. Thus satisfying Elad and Bruckstein's sufficient condition implies satisfying our necessary and sufficient condition with rank-K unambiguity, Theorem 1. (It is easy to see that $\sqrt{M} \leq \frac{1+M}{2}$, since $1 - 2\sqrt{M} + M = (1 - \sqrt{M})^2 \geq 0$).

The paper of Donoho and Huo [8] suggests an extension of their measure $\tilde{M}(\mathbf{A})$ to a basis composed of two invertible bases. We argue in Section 7.2 that our definition of $M(\mathbf{A})$ is more natural, and much easier to apply to the problem of relating $\ell_1$ and $\ell_0$ minimizations. In fact we use our $M(\mathbf{A})$ to prove equivalence of $\ell_0$ and $\ell_1$ problems not only for $\mathbf{A}$ composed of pairs of invertible bases, but for any overcomplete basis $\mathbf{A}$.

The last comment about our new condition is that it does not directly take into account $N$, the number of columns of $\mathbf{A}$. The only property of the matrix used in the derivation is rank-$K$ unambiguity, which has to satisfy only one relation with $N$: $K \leq \min(M, N)$. Thus for matrices $\mathbf{A}$ which have considerably more columns than rows, $N \gg M$, $M(\mathbf{A})$ is quite large, and the bound only guarantees the uniqueness of solutions with a very low number of non-zero coefficients (the requirements are too

[7] A simple counterexample is $\mathbf{A}$ representing a regular 5-simplex in $\mathbb{C}^4$. $M(\mathbf{A})$ in this case equals $\frac{1}{4}$, and $\frac{1}{M(\mathbf{A})} = 4$. So $L = 3$ would be sparse enough to achieve unique solutions using Elad and Bruckstein's sufficient condition. This is impossible, since $3 \geq \frac{4+1}{2}$ which violates our necessary condition (Theorem 1) based on rank-K unambiguity.

high). However, rank-$K$ unambiguity may be as large as $M$ no matter how small $M(\mathbf{A})$ is (as long as $M(\mathbf{A}) > 0$).

We are interested in bounds for $M(\mathbf{A})$ for $\mathbf{A} \in \mathbb{C}^{M \times N}$, as a function of $M$ and $N$, to be able to characterize what information on the required sparsity can be extracted based on the knowledge of $M(\mathbf{A})$ for given $M$ and $N$. That means that if $N$ is much larger than $M$, then we may know that $M(\mathbf{A})$ will be too large without having to calculate it. We describe the bounds in some detail (for the real case) in Sections 7.2.2 and 7.2.3.

## ■ 7.2 Solving the $\ell_0$ problem by $\ell_1$

The two before-mentioned papers (by Donoho and Huo [8], and Elad and Bruckstein [9]), consider the question of when the problem of minimizing the $\ell_0$-norm of the representation coefficients of a signal in an overcomplete basis is equivalent[8] to the problem of minimizing their $\ell_1$-norm. The question is of significant practical value since the $\ell_1$ problem can be solved with the help of linear optimization or second-order cone programming, whereas the $\ell_0$ problem can be tackled directly only by computationally expensive combinatorial optimization. However, as mentioned before, their work is concerned with the case when the overcomplete basis $\mathbf{A}$ is composed of two merged orthogonal bases. We are considering the problem where no assumptions about the structure of $\mathbf{A}$ are made. Dimensions are arbitrary (any $N > M$ is allowed, not just $N = 2M$ as in the case of two merged minimal bases), and the only parameter describing $\mathbf{A}$ on which we are relying is $M(\mathbf{A})$. Donoho's paper suggested that in order to extend their results to a merge of two non-orthogonal (but still invertible) bases, $\mathbf{A} = [\Phi\ \Psi]$, one should consider a measure

$$\tilde{M}(\Phi, \Psi) = \max\ [sup_{i,j}|\Phi^{-1}\Psi|_{i,j},\ sup_{i,j}|\Psi^{-1}\Phi|_{i,j}] \tag{7.14}$$

We argue that this is not the best way to generalize, and provide a different alternative which works much better. A major difficulty is that $\tilde{M}$ depends on a particular partition of $\mathbf{A}$ into $\Phi$ and $\Psi$, thus for a general $M \times 2M$ matrix $\mathbf{A}$ one would need to take into account all possible partitions and find the minimum $\tilde{M}$, to get the best results. For a general $M \times N$ matrix $\mathbf{A}$ the situation is even more complicated. Donoho and Huo do not propose this, but we take the freedom to conjecture an extrapolation of their definition. To extend the proposed measure $\tilde{M}$ it would be necessary to split $\mathbf{A}$ into an invertible $M \times M$ matrix $\Phi$, and the remainder $M \times (N - M)$ matrix $\Psi$, and consider $\tilde{M}(\Phi, \Psi) = sup_{i,j}|\Phi^{-1}\Psi|_{i,j}$ as the measure. To get a bound on sufficient sparsity of $\mathbf{x}$ it is necessary to find the maximum $\tilde{M}$ over all partitions in a set of partitions $\mathcal{P}$ which has the property that $\Phi_p$ cover $\mathbf{A}$ (i.e. the union of columns of $\Phi_p$ over all partitions $p \in \mathcal{P}$ includes the set of columns of $\mathbf{A}$). Then to find a tight bound it is necessary to find the minimum over all such sets of partitions $\mathcal{X}$ of the maximum

[8] There is a recent paper on the same subject by Feuer and Nemirovski, [56], proving that Elad and Bruckstein's sufficient condition for equivalence in merged orthogonal bases is in fact also necessary.

of $\tilde{M}$ for all partitions in each set. Computing such a bound numerically or simplifying it analytically appears to be a nontrivial task.

We go in a different route and use the extension of $M(\mathbf{A})$ from a merge of two orthogonal bases to a general overcomplete basis as described in previous sections.

$$M(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j|, \quad \text{such that } \|\mathbf{a}_k\|_2 = 1, \forall k. \tag{7.15}$$

This definition is much simpler than the generalization suggested by Donoho and Huo in [8], and it is equivalent to their definition for the case of two merged orthogonal bases. The similarity allows us to follow some of the steps taken by Donoho and Huo in their proof of the equivalence of $\ell_0$ and $\ell_1$ for $\mathbf{A}$ composed of two orthogonal bases in the beginning stages of our proof for the more general overcomplete $\mathbf{A}$. The proof is the subject of the next section.

### ■ 7.2.1 Sufficient condition for equivalence of $\ell_0$ and $\ell_1$ problems

First, we restate the problems that we wish to relate. The problem which we refer to as the $\ell_0$-problem has the following form:

$$\min \ \|\mathbf{x}\|_0^0 \ \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{7.16}$$

The $\ell_1$ problem has the same form but with $\ell_1$ norm instead of the $\ell_0$ norm:

$$\min \ \|\mathbf{x}\|_1 \ \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{7.17}$$

In both of these problems we assume that $N > M$ in order for the problem to be of interest. Next, we prove the following sufficient condition:

**Theorem 5 (Equivalence of $\ell_0$ and $\ell_1$ problems for general overcomplete A).** *Suppose that the $\ell_0$ problem (7.16) has a unique solution $\hat{\mathbf{x}}$ with sparsity equal to $L$, e.g. $\|\hat{\mathbf{x}}\|_0^0 = L$. If $L < \frac{1+1/(M(\mathbf{A}))}{2}$, then the solution of the $\ell_1$ problem in (7.17) is $\hat{\mathbf{x}}$. In other words we can get the solution to the $\ell_0$ problem by solving the $\ell_1$ problem.*

*Proof.* In the beginning stages, the structure of the proof follows that of [8] and [9], generalizing some of the notions so that a general overcomplete basis $\mathbf{A}$ could be handled. A novel aspect of the proof is a derivation of a bound on $Q(\mathbf{A})$ for a general basis, in Theorem 6, see below.

Suppose that $\hat{\mathbf{x}}$ is the optimal solution to (7.16). To satisfy $\mathbf{y} = \mathbf{A}\tilde{\mathbf{x}}$, any other candidate must have the form $\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \boldsymbol{\delta}$, where $\boldsymbol{\delta} \in Null(\mathbf{A})$, nullspace of $\mathbf{A}$. In order for $\hat{\mathbf{x}}$ to be the optimal $\ell_1$ solution as well, we need:

$$\|\hat{\mathbf{x}} + \boldsymbol{\delta}\|_1 > \|\hat{\mathbf{x}}\|_1 \text{ for any } \boldsymbol{\delta} \in Null(\mathbf{A}), \ \boldsymbol{\delta} \neq \mathbf{0} \tag{7.18}$$

Alternatively, $\|\hat{\mathbf{x}} + \boldsymbol{\delta}\|_1 - \|\hat{\mathbf{x}}\|_1 > 0$ for all $\boldsymbol{\delta} \in Null(\mathbf{A}), \ \boldsymbol{\delta} \neq \mathbf{0}$.

Let $\mathcal{I}_x$ denote the set of indices where the optimal $\ell_0$ solution $\hat{\mathbf{x}}$ has non-zero values (the support of $\hat{\mathbf{x}}$). Also its complement, the set of zero-valued indices of $\hat{\mathbf{x}}$, is denoted

by $\mathcal{I}_x^C$. We can divide the $\ell_1$ norm into the components on and off the support of $\hat{\mathbf{x}}$ and then use the triangle inequality to manipulate (7.18) as follows:

$$\|\hat{\mathbf{x}}+\boldsymbol{\delta}\|_1 - \|\hat{\mathbf{x}}\|_1 = \left(\sum_{i\in\mathcal{I}_x}|\hat{x}_i+\delta_i| + \sum_{i\in\mathcal{I}_x^C}|\delta_i|\right) - \sum_{i\in\mathcal{I}_x}|\hat{x}_i| = \tag{7.19}$$

$$= \sum_{i\in\mathcal{I}_x}(|\hat{x}_i+\delta_i| - |\hat{x}_i|) + \sum_{i\in\mathcal{I}_x^C}|\delta_i| \geq \sum_{i\in\mathcal{I}_x^C}|\delta_i| - \sum_{i\in\mathcal{I}_x}|\delta_i| \tag{7.20}$$

The first equality uses the fact that $\hat{x}_i = 0$ for $i \in \mathcal{I}_x^C$, and the triangle inequality is used in the form $|a+b| - |a| \geq -|b|$. Note that the bound in (7.20) does not depend on $\hat{\mathbf{x}}$, only on $\|\hat{\mathbf{x}}\|_0^0$ and the structure of the nullspace of $\mathbf{A}$. Starting from $\sum_{i\in\mathcal{I}_x^C}|\delta_i| - \sum_{i\in\mathcal{I}_x}|\delta_i| > 0$, adding $2\sum_{i\in\mathcal{I}_x}|\delta_i|$ on both sides, and using the fact that $\sum_{i\in\mathcal{I}_x}|\delta_i| + \sum_{i\in\mathcal{I}_x^C}|\delta_i| = \sum_i|\delta_i| = \|\boldsymbol{\delta}\|_1$ we get the following condition for equivalence:

$$\frac{\sum_{i\in\mathcal{I}_x}|\delta_i|}{\|\boldsymbol{\delta}\|_1} < \frac{1}{2}, \text{ where } \boldsymbol{\delta} \in Null(\mathbf{A}),\ \boldsymbol{\delta} \neq \mathbf{0} \tag{7.21}$$

This is a great start, but unfortunately, hard to test numerically. In order to move further from (7.21), we consider the following family of problems indexed by $i$:

$$\min\ \|\boldsymbol{\delta}\|_1 \text{ subject to } \boldsymbol{\delta} \in Null(\mathbf{A}) \text{ and } \delta_i = 1 \tag{7.22}$$

What we are trying to do here is to find the minimum possible $\ell_1$-norm of $\boldsymbol{\delta} \in Null(\mathbf{A})$, when we fix $\delta_i = 1$. Suppose the minimum value is $Q_i$, when index $i$ is fixed. Define $Q(\mathbf{A}) = \min_i\ Q_i$. Next, we use Theorem 6, which we prove right after this one, which states that $Q(\mathbf{A}) \geq (1 + \frac{1}{M(\mathbf{A})})$,

Consider the condition that we are trying to prove, (7.21):

$$\frac{\sum_{i\in\mathcal{I}_x}|\delta_i|}{\|\boldsymbol{\delta}\|_1} = \sum_{i\in\mathcal{I}_x}\frac{|\delta_i|}{\|\boldsymbol{\delta}\|_1} \leq \sum_{i\in\mathcal{I}_x}\frac{1}{Q_i} \leq \sum_{i\in\mathcal{I}_x}\frac{1}{Q(\mathbf{A})} = \|\hat{\mathbf{x}}\|_0^0\frac{1}{Q(\mathbf{A})} \leq \|\hat{\mathbf{x}}\|_0^0(1+\frac{1}{M(\mathbf{A})})^{-1} \tag{7.23}$$

We need to have $\frac{\sum_{i\in\mathcal{I}_x}|\delta_i|}{\|\boldsymbol{\delta}\|_1} < \frac{1}{2}$. In order for that to happen, it is sufficient that $\|\hat{\mathbf{x}}\|_0^0 < \frac{1}{2}(1+\frac{1}{M(\mathbf{A})})$. This proves the theorem. ■

In the proof we used Theorem 6, which we prove next.

**Theorem 6 (Bound on $Q(\mathbf{A})$ for a general overcomplete A).** *$Q(\mathbf{A})$, the minimum optimum value of problems (7.22) over all $i$, can be bounded by $Q(\mathbf{A}) \geq (1+\frac{1}{M(\mathbf{A})})$.*

*Proof.* Without loss of generality we assume that $Q_i$ is minimized at $i = 1$, so we have $\delta_1 = 1$. (This does not impose any restrictions since we can always rearrange the columns of $\mathbf{A}$ and the indices of $\boldsymbol{\delta}$ such that this is true). Also, recall that the columns of $\mathbf{A}$ are normalized to have unit Euclidean distance.

Split $\mathbf{A}$ into two parts: $\mathbf{A} = [\mathbf{b}\ \mathbf{C}]$, where $\mathbf{b} \in \mathbb{C}^{M\times 1}$, and $\mathbf{C} \in \mathbb{C}^{M\times(N-1)}$. Then $\mathbf{A}\boldsymbol{\delta} = [\mathbf{b}\ \mathbf{C}]\boldsymbol{\delta} = \mathbf{b}\delta^b + \mathbf{C}\boldsymbol{\delta}^C = 0$. Thus $\mathbf{b}\delta^b = -\mathbf{C}\boldsymbol{\delta}^C$. Here $\delta^b$ is the 1st index of $\boldsymbol{\delta}$, i.e. $\delta^b = \delta_1$, and $\boldsymbol{\delta}^C$ is the rest of the vector $\boldsymbol{\delta}$. By assumption (or reindexing) $\delta^b = 1$.

Apply $\mathbf{b}'$ to both sides: $\mathbf{b}'\mathbf{b}\delta^b = 1 = -\mathbf{b}'\mathbf{C}\boldsymbol{\delta}^C$. But by the definition of $M(\mathbf{A})$, vector $\mathbf{b}'\mathbf{C}$ has every component less than $M(\mathbf{A})$ in absolute value, thus $1 = |\mathbf{b}'\mathbf{C}\boldsymbol{\delta}^C| \leq |\mathbf{b}'\mathbf{C}||\boldsymbol{\delta}^C| \leq M(\mathbf{A})\mathbf{1}'|\boldsymbol{\delta}^C| = M(\mathbf{A})\ \|\boldsymbol{\delta}^C\|_1$. Consequently[9]

$$\|\boldsymbol{\delta}\|_1 = |\boldsymbol{\delta}^b| + \|\boldsymbol{\delta}^C\|_1 = 1 + \|\boldsymbol{\delta}^C\|_1 \geq 1 + \frac{1}{M(\mathbf{A})} \tag{7.24}$$

This proves the theorem. ■

Next we describe how to get a feel for $M(\mathbf{A})$ as a function of the dimensions of $\mathbf{A}$, $M$ and $N$, for the case when $\mathbf{A}$ is real-valued.

### ■ 7.2.2 The insight into $M(\mathbf{A})$ from the theory of spherical codes

The problem of finding $\mathbf{A} \in \mathbb{R}^{M\times N}$ to minimize $M(\mathbf{A})$ is very similar to the problem of designing spherical codes[10]. We use results from the theory of spherical codes to relate $M(\mathbf{A})$ to the dimensions of $\mathbf{A}$, e.g. $M$ and $N$. We borrow the information about spherical codes from the book "Codes on Euclidean Spheres" [57]. A code on a Euclidean sphere is a finite set of unit norm vectors: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ...., \mathbf{x}_N : \mathbf{x}_i \in \mathbb{R}^M, \|\mathbf{x}_i\|_2 = 1\}$. $N$ is the number of codewords. Alternatively, we can represent a code by matrix $\mathbf{X} = [\mathbf{x}_1, .., \mathbf{x}_N]$, keeping in mind that the order of the columns is inconsequential. An important measure associated with a spherical code is the squared minimum distance: $\rho = \min\{\|x - y\|_2^2 : x, y \in \mathcal{X}, x \neq y\}$. The goals of code design are maximizing $\rho$, minimizing $M$ and maximizing $N$. These goals are contradictory, and the main problem is of characterizing the regions of achievable $(\rho, M, N)$, and developing codes as close as possible to the boundaries of this region.

Two functions associated with a code are used to describe the achievable region. $N_M(\rho)$, is the largest possible size of a spherical code with dimension $M$, and a minimum squared distance of at least $\rho$. The other one is $\rho_M(N)$, the minimum possible squared distance for a code with dimension $M$, and $N$ codewords. Formally,

$$N_M(\rho) = \max\{|\mathcal{X}| : \rho(\mathcal{X}) \geq \rho, dim(\mathcal{X}) = M\}, \tag{7.25}$$

where $\rho(\mathcal{X})$ is the minimum Euclidean distance for any two distinct codewords in $\mathcal{X}$, and the $|\mathcal{X}|$ notation for sets denotes the cardinality of the set, or the number of elements

[9] We use $\mathbf{1}$ to represent a vector of ones.

[10] We have not extended these results to $\mathbb{C}^{M\times N}$, so everything related to spherical codes deals with real quantities.

in the code. Similarly,

$$\rho_M(N) = \max\{\rho(\mathcal{X}) : |\mathcal{X}| \geq N, dim(\mathcal{X}) = M\}. \tag{7.26}$$

These two functions $N_M(\rho)$, and $\rho_M(N)$ are not fully known for dimension $M$ higher than 3, but several bounds have been constructed. In Section 7.2.3 we describe one of these bounds and also present an easy extension which allows to put a bound on $M(\mathbf{A})$.

#### Relation of $\rho_M(N)$ to $M(\mathbf{A})$

Before describing the sphere packing bound which partially characterizes achievable $\rho_M(N)$, we need to relate $\rho_M(N)$ to $M(\mathbf{A})$, the quantity that we are ultimately interested in. The problem of maximizing $\rho(\mathcal{X})$, the minimum square distance between any two codewords, is identical to that of maximizing the minimum angle between the two codewords, or equivalently minimizing the maximum[11] pairwise dot-product:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 = \mathbf{x}_i'\mathbf{x}_i + \mathbf{x}_j'\mathbf{x}_j - 2\mathbf{x}_i'\mathbf{x}_j = 2 - 2\mathbf{x}_i'\mathbf{x}_j \tag{7.27}$$

The second equality is obtained since all $\mathbf{x} \in \mathcal{X}$ have $\|\mathbf{x}\|_2^2 = 1$. This relates $\rho$ to the angle $\theta$ between $\mathbf{x}_i$ and $\mathbf{x}_j$ since for $\mathbf{x}_i \in \mathbb{R}^M$, $\mathbf{x}_i'\mathbf{x}_j = cos(\theta)$, which is a monotonically decreasing function on $[0, \pi]$, which leads to another switch of maximization to minimization.

For a spherical code maximizing the minimum $\rho(\mathcal{X})$ is equivalent to $\min\max \mathbf{x}_i'\mathbf{x}_j$, $i \neq j$, $i, j = 1, .., N$. However, our measure $M(\mathbf{X})$ (now we have $\mathbf{X}$ instead of $\mathbf{A}$, so we use $M(\mathbf{X})$), is somewhat different: $\max |\mathbf{x}_i'\mathbf{x}_j|$, $i \neq j$, $i, j = 1, .., N$. An extra absolute value of the dot product is taken. The absolute value appears in our problem making our problem equivalent to packing lines on a Euclidean sphere, whereas the spherical code problem is equivalent to packing rays. In our case, each column of $\mathbf{A}$ (or $\mathbf{X}$) represents a subspace, and each $\mathbf{a}_i$ can be represented as a dipole centered at the origin, $\mathbf{a}_i$ and $-\mathbf{a}_i$. We are looking for the minimum angle between lines (i.e. 1-subspaces),[12] and not between vectors. We illustrate this difference in Figure 7.2. Next we prove a bound for $N_M(\rho)$, and in a similar fashion a bound for $M(\mathbf{A})$.

### ■ 7.2.3 Sphere-packing bound

The sphere packing bound puts an upper limit on $N$ as a function of $\rho$ and $M$, i.e. it bounds $N_M(\rho)$. The idea is quite simple. If $\mathcal{X}$ has minimum pairwise distance $\rho$, or equivalently minimum pairwise angle $\theta$, then the conic region with angle $\theta$ around each codeword $\mathbf{x}_i$ has to be empty of other codewords. Alternatively, conic regions with angle $\frac{\theta}{2}$ around each codeword have to be disjoint. In order for that to happen the surface areas of the spherical caps of the cones (intersection of the cones with the unit sphere) for all $N$ codewords have to sum to a value which is less than the surface area of

[11] Note the switch of the order of min and max!

[12] The angle between two lines is the smaller of the 2 angles at the intersection. The lines that we are concerned with are linear subspaces so they intersect at the origin.

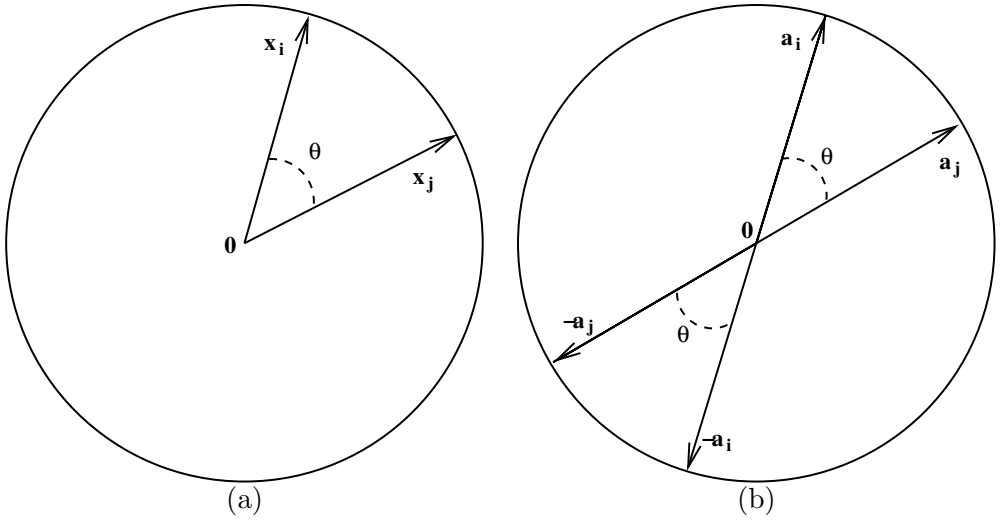



**Figure 7.2.** Comparison of ray packing and line packing on the 2-sphere. (a) Ray packing of $\mathbf{x}_i$. (b) Line packing of $\mathbf{a}_i$.

the unit sphere of dimension $M$. Otherwise the surface caps would overlap. The sphere packing bound is the largest possible $N$ which allows the sum of the areas of these spherical caps to be less than the surface area of the unit sphere. The condition on the sum of the areas is not sufficient for the existence of the code; in fact spherical caps for a sphere of dimension $M > 2$ cannot cover the sphere and be disjoint. Fortunately, we are not interested in the problem of code design, only in characterizing the feasible region. The problem of code design is very hard and optimal solutions have been found only for a small set of pairs $M$ and $N$. We are content with just having a bound.

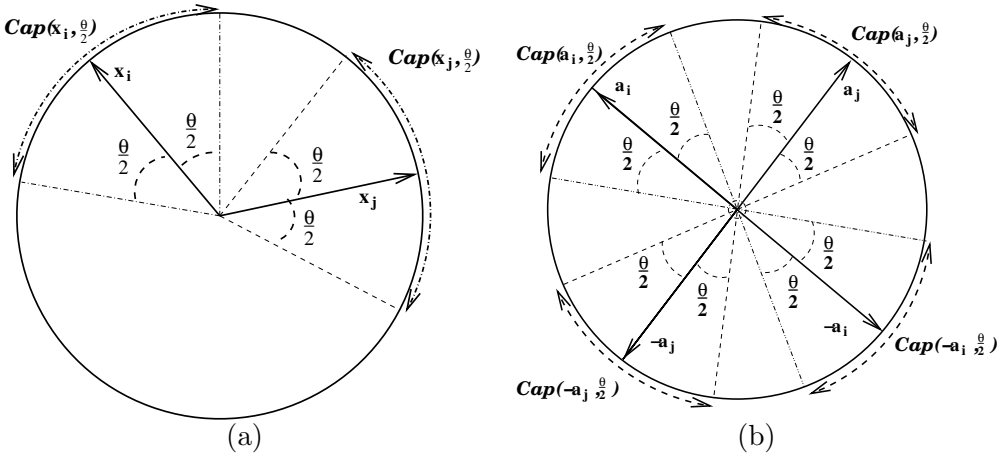



**Figure 7.3.** Comparison of ray caps and line caps on the 2-sphere. (a) Ray caps for $\mathbf{x}_i$ and $\mathbf{x}_j$. (b) Line caps for $\mathbf{a}_i$ and $\mathbf{a}_j$.

Now we put the preceding informal discussion into formulas. For notational convenience, define the unit sphere $\Omega_M = \{\mathbf{x} \in \mathbb{R}^M : \|\mathbf{x}\|_2^2 = 1\}$, and the spherical cap around $\mathbf{x}$ with angle $\theta$, $Cap(\mathbf{x}, \theta) = \{\mathbf{y} \in \Omega_M : \mathbf{x}'\mathbf{y} \geq cos(\theta)\}$. An illustration appears in Figure 7.3. Denote the surface area of $Cap(\mathbf{x}, \theta)$ by $C_M(\theta)$ (it does not depend on $\mathbf{x}$). Define $Y_M(r)$ to be the surface area of an $M$-dimensional sphere with radius $r$.

We summarize some of the expressions for the quantities defined above (without proofs). For a more detailed explanation refer to [57].

$$Y_M(r) = k_M r^{M-1}, \text{where} \tag{7.28}$$

$$k_M = \begin{cases} \frac{(2\pi)^{M/2}}{(M-2)!!}, & \text{for } M = 2,\ 4,\ ... \\ 2\frac{(2\pi)^{(M-1)/2}}{(M-2)!!}, & \text{for } M = 3,\ 5,\ ... \end{cases} \tag{7.29}$$

The double-factorial notation stands for

$$m!! = \begin{cases} m(m-2)(m-4)...(3)(1), & \text{for odd } m, \\ m(m-2)(m-4)...(4)(2), & \text{for even } m \\ 1, & \text{for } m = 0, \text{ and } m = 1. \end{cases} \tag{7.30}$$

$$C_M(\theta) = \int_0^\theta Y_{M-1}(sin(\beta))d\beta = k_{M-1} \int_0^\theta sin^{M-2}(\alpha)d\alpha. \tag{7.31}$$

With these definitions we are now in a position to state the upper and lower bounds for $N_M(\rho)$ for spherical codes (i.e. for ray-packing).

**Theorem 7 (Sphere packing bound).** *Let $\mathcal{X}$ be any spherical code with parameters ($\rho$, $M$, $N$). Then the following inequality holds:*

$$N \leq N_M(\rho) \leq \left\lfloor \frac{C_M(\pi)}{C_M(\theta/2)} \right\rfloor \tag{7.32}$$

*where $\rho = 2 - 2cos(\theta)$, and $\lfloor \bullet \rfloor$ denotes the largest integer below $\bullet$.*

We are ultimately interested in characterizing $M(\mathbf{A})$. Luckily, most of the work has already been done. It remains to note that since we have lines instead of rays, then each column $\mathbf{a}_i$ of $\mathbf{A}$ corresponds to two conical regions, one around $\mathbf{a}_i$, and another around $-\mathbf{a}_i$, (see Figure 7.3). Again in order for the $2N$ conical regions to be non-overlapping, the sum of their surface areas has to be below the surface area of the unit sphere. This leads to the following

**Theorem 8 (Bound on N as a function of $M(\mathbf{A})$).** *Let $\mathbf{A} \in \mathbb{R}^{M\times N}$. Then*

$$N \leq \left\lfloor \frac{C_M(\pi)}{2C_M(\theta/2)} \right\rfloor \tag{7.33}$$

*where $M(\mathbf{A}) = cos(\theta)$, $0 \leq \theta \leq \pi/2$, and $\lfloor \bullet \rfloor$ denotes the largest integer below $\bullet$.*

This bound can be used to test whether it is possible at all to achieve a desired value of $M(\mathbf{A})$ for given $M$ and $N$. This concludes our discussion of the $\ell_1$ problem. Next we move on to the case of general $p \leq 1$.

## ■ 7.3 Conditions for the equivalence of $\ell_p$ and $\ell_0$ problems

In this section we present two sufficient conditions for the equivalence of $\ell_0$ (7.16) and $\ell_p$ (7.34) problems for $p \leq 1$. The first condition can be easily tested numerically, but it does not give preference to lower $p$. The second condition is harder to test numerically, but it gives a strong preference for smaller $p$, and in particular it shows that as $p \to 0$, then the sufficient condition for equivalence of $\ell_p$ and $\ell_0$ problems approaches the necessary and sufficient condition for the uniqueness of solutions of the $\ell_0$ problem.

Practically, for $p < 1$ these results are not as important as the results for $\ell_1$ equivalence from Section 7.2, since finding the global optimum of the non-convex $\ell_p$ cost function is a much harder task. In particular, the iterative method for $\ell_p$ minimization that we presented only finds a local minimum. However, these results may stimulate further work in global $\ell_p$ optimization, for example by homotopy continuation methods.

### Generic conditions

We consider the $\ell_p$ problem

$$\min \ \|\mathbf{x}\|_p^p \ \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{7.34}$$

Matrix $\mathbf{A}$ is $M$ by $N$, with $N > M$, and $p \leq 1$. Suppose that the $\ell_0$ problem $\min \ \|\mathbf{x}\|_0^0$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$ has a unique solution $\hat{\mathbf{x}}$, and $\|\hat{\mathbf{x}}\|_0^0 = L$. $\ell_0$ uniqueness means that there exists a single vector $\hat{\mathbf{x}}$ satisfying the constraints, which has at most $L$ non-zero elements, where $L < \frac{K+1}{2}$, and matrix $\mathbf{A}$ is rank-$K$ unambiguous. The purpose of the following analysis is to find out under what conditions on $\mathbf{A}$, $L$, and $p$, the solution of (7.34) is also the solution of the $\ell_0$ problem, $\hat{\mathbf{x}}$.

We start in a similar fashion to what was done for the $\ell_1$ case in Section 7.2. In order for $\hat{\mathbf{x}}$ to be the unique solution of (7.34), it must be true that $\|\hat{\mathbf{x}}\|_p^p < \|\hat{\mathbf{x}} + \boldsymbol{\delta}\|_p^p$ for any $\boldsymbol{\delta} \in Null(\mathbf{A})$, $\boldsymbol{\delta} \neq \mathbf{0}$. Let $\mathcal{I}_x$ denote the set of indices where the optimal solution $\hat{\mathbf{x}}$ has non-zero values (the support of $\hat{\mathbf{x}}$). Let its complement, the set of zero-valued indices of $\hat{\mathbf{x}}$, be denoted by $\mathcal{I}_x^C$. Then

$$\|\hat{\mathbf{x}} + \boldsymbol{\delta}\|_p^p - \|\hat{\mathbf{x}}\|_p^p = \sum_{i\in\mathcal{I}_x} (|\hat{x}_i + \delta_i|^p - |\hat{x}_i|^p) + \sum_{i\in\mathcal{I}_x^C} |\delta_i|^p \tag{7.35}$$

This can be simplified using the fact that the $\| \bullet \|_p^p$ functional satisfies the triangle inequality for $p \leq 1$ [13]:

$$\|\mathbf{z}\|_p^p - \|\mathbf{y}\|_p^p \leq \|\mathbf{z}+\mathbf{y}\|_p^p \leq \|\mathbf{z}\|_p^p + \|\mathbf{y}\|_p^p \tag{7.36}$$

The triangle inequality leads to the following (we use it in the form $|a+b|^p - |a|^p \geq -|b|^p$):

$$\sum_{i\in\mathcal{I}_x} (|\hat{x}_i + \delta_i|^p - |\hat{x}_i|^p) + \sum_{i\in\mathcal{I}_x^C} |\delta_i|^p \geq -\sum_{i\in\mathcal{I}_x} |\delta_i|^p + \sum_{i\in\mathcal{I}_x^C} |\delta_i|^p \tag{7.37}$$

Thus the condition for getting the $\ell_0$ solution at the global optimum of the $\ell_p$ cost function is to have the sum of $|\delta_i|^p$ on the support of the optimal solution to be less than the sum of $|\delta_i|^p$ on its complement for every element $\boldsymbol{\delta} \in Null(\mathbf{A}),\ \boldsymbol{\delta} \neq \mathbf{0}$:

$$\sum_{i\in\mathcal{I}_x^C} |\delta_i|^p - \sum_{i\in\mathcal{I}_x} |\delta_i|^p > 0 \tag{7.38}$$

Similarly to the $\ell_1$ case, this condition is not easy to use. Next we present two related conditions. The first of them can be readily computed numerically. The second condition does not have this benefit, but it is tighter for small $p$, and in particular explicitly shows that the sparsity requirements are reduced as $p \to 0$.

### ■ 7.3.1 First condition for equivalence of $\ell_0$ and $\ell_p$ for $p \leq 1$

In order to be able to simplify the condition further, we examine $\mathbf{A}$ more closely. We introduce a new measure of $\mathbf{A}$, $L_1(\mathbf{A})$, and use it to get an equivalence relation between $\ell_p$ and $\ell_0$ problems. Define

$$L_1(\mathbf{A}) = \min \|\boldsymbol{\delta}\|_1 \text{ subject to } \|\boldsymbol{\delta}\|_\infty = 1, \text{ and } \boldsymbol{\delta} \in Null(\mathbf{A}) \tag{7.39}$$

For the real-data case, this functional of $\mathbf{A}$ can be readily found by $N$ linear problems, as we describe in Appendix E.

Now consider the condition from the previous section:

$$\sum_{i\in\mathcal{I}_x^C} |\delta_i|^p - \sum_{i\in\mathcal{I}_x} |\delta_i|^p = \|\boldsymbol{\delta}\|_\infty \left( \sum_{i\in\mathcal{I}_x^C} \frac{|\delta_i|^p}{\|\boldsymbol{\delta}\|_\infty} - \sum_{i\in\mathcal{I}_x} \frac{|\delta_i|^p}{\|\boldsymbol{\delta}\|_\infty} \right) > 0 \tag{7.40}$$

Now all the indices of $\frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_\infty}$ are normalized such that $\|\frac{\boldsymbol{\delta}}{\|\boldsymbol{\delta}\|_\infty}\|_\infty = 1$. It suffices to show that $\sum_{i\in\mathcal{I}_x^C} |\delta_i|^p - \sum_{i\in\mathcal{I}_x} |\delta_i|^p > 0$ when $\|\boldsymbol{\delta}\|_\infty = 1$, and $\boldsymbol{\delta} \in Null(\mathbf{A})$. Now comes the time to use our new measure, $L_1(\mathbf{A})$. Since $\boldsymbol{\delta}$ is normalized to 1 in $\infty$-norm, that

[13] Note the presence of the $p$-th power. The triangle inequality fails for the $\ell_p$-quasi-norm, $\| \bullet \|_p$ for $p < 1$ (without raising to $p$-th power). This is why $\ell_p$ is not a norm for $p < 1$, and just a quasi-norm.

means that $\sum_{i\in\mathcal{I}_x}|\delta_i|^p \leq L$, the number of nonzero elements of $\hat{\mathbf{x}}$. For $p \leq 1$, and $|\delta_i| \leq 1$, $|\delta_i|^p \geq |\delta_i|$. Hence,

$$\sum_{i\in\mathcal{I}_x^C}|\delta_i|^p = \|\boldsymbol{\delta}\|_p^p - \sum_{i\in\mathcal{I}_x}|\delta_i|^p \geq \|\boldsymbol{\delta}\|_1 - \sum_{i\in\mathcal{I}_x}|\delta_i|^p \geq \tag{7.41}$$

$$\geq L_1(\mathbf{A}) - \sum_{i\in\mathcal{I}_x}|\delta_i|^p \geq L_1(\mathbf{A}) - L \tag{7.42}$$

Putting it all together, we have that:

$$\sum_{i\in\mathcal{I}_x^C}|\delta_i|^p - \sum_{i\in\mathcal{I}_x}(|\delta_i|^p) \geq L_1(\mathbf{A}) - 2L \tag{7.43}$$

So a sufficient condition for the equivalence of the $\ell_0$ problem and $\ell_p$ problem with $p \leq 1$ is the following:

**Theorem 9 (Equivalence of $\ell_0$ and $\ell_p$ with $p \leq 1$, first sufficient condition).** *Suppose that the $\ell_0$ problem (7.16) has a unique solution $\hat{\mathbf{x}}$ with sparsity equal to $L$, e.g. $\|\hat{\mathbf{x}}\|_0^0 = L$. If $L < \frac{L_1(\mathbf{A})}{2}$, then the solution of the $\ell_p$ problem in (7.34) is $\hat{\mathbf{x}}$.*

The sufficient condition that we presented does not depend on $p$ except for the fact that $p \leq 1$. In particular it does not favor smaller $p$. The reason that this is the case is that we use an $\ell_1$-based measure on $\mathbf{A}$. We can change this by considering $L_p(\mathbf{A})$ as the minimum $\ell_p$ norm for all $\boldsymbol{\delta} \in Null(\mathbf{A})$ subject to $\|\boldsymbol{\delta}\|_\infty = 1$. This however does not have the benefit of a tractable numerical solution. Without further analysis it is not immediately clear that smaller $p$ would be preferred. Instead of following up on $L_p(\mathbf{A})$ we choose to follow a different path and introduce another measure of $\mathbf{A}$ next.

### ■ 7.3.2 Another equivalence condition for $\ell_p$ and $\ell_0$ problems, $p \leq 1$

A very interesting equivalence condition comes up when we consider order statistics of the elements of the nullspace of $\mathbf{A}$. Define $\tilde{\boldsymbol{\delta}}$ to be a permutation of $\boldsymbol{\delta}$ in which the absolute values of the coordinates $\delta_i$ appear sorted in decreasing order. Thus, $\tilde{\delta}_1 = \max_i\ |\delta_i|$, $\tilde{\delta}_N = \min_i\ |\delta_i|$. If several indices of $\boldsymbol{\delta}$ have the same value then their ordering is immaterial for our purposes. Suppose matrix $\mathbf{A}$ is rank-K unambiguous, i.e. the minimum linearly dependent set of columns of $\mathbf{A}$ contains $K+1$ elements. We define $S(\mathbf{A})$ as follows:

$$S(\mathbf{A}) = \min\ \tilde{\delta}_{K+1} \text{ over all } \boldsymbol{\delta} \in Null(\mathbf{A}),\ \text{with } \|\boldsymbol{\delta}\|_\infty = 1 \tag{7.44}$$

Since $\mathbf{A}$ is rank-K unambiguous, $S(\mathbf{A})$ is greater than zero (otherwise there would exist a set of $K$ linearly dependent column-vectors of $\mathbf{A}$). Also $S(\mathbf{A}) \leq 1$, since $\|\boldsymbol{\delta}\|_\infty = \tilde{\delta}_1 = 1$. Having defined $S(\mathbf{A})$ we now use it to simplify the condition in (7.38). To do that we look for a lower bound on $\sum_{i\in\mathcal{I}_x^C}|\delta_i|^p$ and an upper bound on $\sum_{i\in\mathcal{I}_x}|\delta_i|^p$. Same as in the last section, $\sum_{i\in\mathcal{I}_x}|\delta_i|^p \leq L$, where $L$ is the number of nonzero elements of $\hat{\mathbf{x}}$, ($\hat{\mathbf{x}}$

is the solution to the $\ell_0$ problem (7.16)). At least $K+1$ elements in $\boldsymbol{\delta} \in Null(\mathbf{A})$ have absolute values greater than or equal to $S(\mathbf{A})$. We have assigned at most $L$ of them already to the support of $\hat{\mathbf{x}}$, $\mathcal{I}_x$ , so at least $K+1-L$ remain off the support. Hence, $\sum_{i \in \mathcal{I}_x^C} |\delta_i|^p \geq S(\mathbf{A})^p (K+1-L)$. Using these two inequalities together we get:

$$\sum_{i \in \mathcal{I}_x^C} |\delta_i|^p - \sum_{i \in \mathcal{I}_x} |\delta_i|^p \geq S(\mathbf{A})^p (K+1-L) - L \tag{7.45}$$

In order for (7.38) to be satisfied, a sufficient condition is: $S(\mathbf{A})^p (K+1-L) - L > 0$. For this condition to hold, $L$ must satisfy: $L < \frac{S(\mathbf{A})^p (K+1)}{1+S(\mathbf{A})^p}$.

**Theorem 10 (Equivalence of $\ell_0$ and $\ell_p$ with $p \leq 1$, second sufficient condition).** *Suppose that the $\ell_0$ problem (7.16) has a unique solution $\hat{\mathbf{x}}$ with sparsity equal to $L$, e.g. $\|\hat{\mathbf{x}}\|_0^0 = L$. Also, $\mathbf{A}$ is rank-K unambiguous. If $L < \frac{S(\mathbf{A})^p (K+1)}{1+S(\mathbf{A})^p}$, then the solution of the $\ell_p$ problem in (7.34) is $\hat{\mathbf{x}}$.*

Now let us take a look at the new condition. Since $S(\mathbf{A})$ is below unity except for degenerate cases, (and typically significantly below unity), so when $p < 1$ the bound is less restrictive than when $p = 1$. As $p$ goes to zero, the condition approaches $(K+1)/2$, as long as $S(\mathbf{A})$ is non-zero. But this is the necessary and sufficient condition for uniqueness of solutions to the $\ell_0$ problem which we have derived in Theorem 1! When $p$ is sufficiently close to 0, the equivalence of $\ell_p$ and $\ell_0$ problems requires the same sparsity of $\mathbf{x}$ as it is necessary for $\ell_0$ problem to have a unique solution. This is very interesting theoretically.

However, practically this has limited value due to the fact that the global minimum of the $\ell_p$ problem cannot be easily found using current optimization techniques. Also, as $p \to 0$, the $\ell_p$ problem becomes "increasingly" non-convex, also increasing the difficulty of finding the global optimum. One direction of research that may lead to efficient methods to find the global optimum of the $\ell_p$ problem is homotopy continuation. The idea is that we start from the solution to the $\ell_1$ problem, and decrease $p$ slowly, finding the solution of the $\ell_p$ problem for each $p$.

## ■ 7.4 Sparsity regularization: a sensitivity result for the noisy version.

So far in this chapter we have considered the noiseless problem, $\mathbf{y} = \mathbf{A}\mathbf{x}$. In practice there is always some noise, so it is worthwhile to analyze the noisy problem as well (also recall that the noisy problem is the focus of the rest of the thesis): $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$. Assume that $\mathbf{A}$ is rank-M unambiguous and there is an underlying true signal $\mathbf{x}_1$, such that $\mathbf{y} = \mathbf{A}\mathbf{x}_1 + \mathbf{n}$, and $\|\mathbf{x}_1\|_0^0 = L < (M+1)/2$. Since noise is present, we are interested in what kinds of sparse representations of $\mathbf{y}$ are possible when we allow some deviation from $\mathbf{y}$ to account for noise. That is to say we are looking for sparse signals $\mathbf{x}_2$, such that $\|\mathbf{x}_2\|_0^0 = L < (M+1)/2$, and $\mathbf{A}\mathbf{x}_2$ is not too far from $\mathbf{A}\mathbf{x}_1$. The question of what method to use to find such $\mathbf{x}_2$ is immaterial; we are just interested in the existence and properties of such $\mathbf{x}_2$.

An important question is the sensitivity of sparse representations to noise, or to the amount of deviation from $\mathbf{y}$ that we can tolerate. One way to quantify that is by trying to answer whether it is possible to have $\mathbf{x}_1$ very different from $\mathbf{x}_2$ while $\mathbf{A}\mathbf{x}_1$ is almost the same as $\mathbf{A}\mathbf{x}_2$ [14]. The definition of "$\mathbf{x}_1$ very far from $\mathbf{x}_2$" that we use for our analysis is the following. Let the support of $\mathbf{x}_i$ be denoted by $\mathcal{I}_i$, $i = 1, 2$. Denote the matrices composed of columns of $\mathbf{A}$ which correspond to the non-zero indices of $\mathbf{x}_i$ by $\mathbf{A}_i$. For example, if $\mathbf{x}_1$ is non-zero on indices $\{1, 4, 5\}$, then $\mathbf{A}_1 = [\mathbf{a}_1\ \mathbf{a}_4\ \mathbf{a}_5]$, where $\mathbf{a}_k$ is the $k$-th column of $\mathbf{A}$. We call $\mathbf{x}_1$ very different from $\mathbf{x}_2$ if the columns corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ are well-separated[15]: that means that the matrix composed of the columns corresponding to the support of both $\mathbf{x}_1$ and $\mathbf{x}_2$, $\mathbf{A}_{\mathcal{I}} = [\mathbf{A}_1\ \mathbf{A}_2]$, is full rank and $M(\mathbf{A}_{\mathcal{I}}) \leq J$, for some small $J$. Recall that $M(\mathbf{A}) = \max_{i \neq j} |\mathbf{a}_i' \mathbf{a}_j|$, which was defined in (7.3). This definition of well-separatedness using $M(\mathbf{A}_{\mathcal{I}})$ is not exactly what we are looking for, since it also requires elements within $\mathbf{x}_1$ and $\mathbf{x}_2$ to be well-separated. However, it is appropriate as a starting point.

Let $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$, then we are in effect trying to solve the following problem:

$$\min\ \|\mathbf{A}\mathbf{x}\|_2^2,\ \text{subject to}\ \|\mathbf{x}\|_0^0 = 2L\ \ \text{and}\ \|\mathbf{x}\|_2^2 = 1 \tag{7.46}$$

The $\ell_2$ norm of $\mathbf{x}$ is set to 1 for normalization, to prevent trivial solutions. If we can find $\mathbf{x}$ such that $\|\mathbf{A}\mathbf{x}\|_2^2$ is very small, then any method of sparse solutions of the noisy problem may be overly sensitive to small noise. We would like to put a lower bound on $\|\mathbf{A}\mathbf{x}\|_2^2$ as a function of $J$ and $L$.

Since the columns of $\mathbf{A}$ corresponding to zero-valued indices of $\mathbf{x}$ do not affect the minimization, we can discard them. Denote the vector of nonzero elements of $\mathbf{x}$ by $\tilde{\mathbf{x}}$, then $\tilde{\mathbf{x}} \in \mathbb{C}^{2L}$. We get the following equivalent problem:

$$\min\ \|\mathbf{A}_{\mathcal{I}}\tilde{\mathbf{x}}\|_2^2,\ \text{subject to}\ \|\tilde{\mathbf{x}}\|_2^2 = 1 \tag{7.47}$$

By the Rayleigh quotient theorem [58] $\|\mathbf{A}_{\mathcal{I}}\tilde{\mathbf{x}}\|_2^2 \geq \lambda_{\min}\|\tilde{\mathbf{x}}\|_2^2$, where $\lambda_{\min}$ is the minimum eigenvalue of $\mathbf{A}_{\mathcal{I}}'\mathbf{A}_{\mathcal{I}}$. Matrix $\mathbf{A}_{\mathcal{I}}'\mathbf{A}_{\mathcal{I}}$ is positive definite, since all the columns are assumed linearly independent (we have $L < (M+1)/2$, so $2L \leq M$, and $\mathbf{A}$ is rank-M unambiguous). Hence all its eigenvalues $\lambda_i > 0$. We would like to say more than that. In particular we would like to bound $\lambda_{\min}$ using $J$. We know that $M(\mathbf{A}_{\mathcal{I}}) \leq J$, hence all the off-diagonal elements of $\mathbf{A}_{\mathcal{I}}'\mathbf{A}_{\mathcal{I}}$ are smaller than or equal to $J$ in magnitude. All the diagonal elements are 1 because columns of $\mathbf{A}$ are normalized to unity. Let $\mathbf{B} = \mathbf{A}_{\mathcal{I}}'\mathbf{A}_{\mathcal{I}}$; then $\mathbf{B}_{i,i} = 1$, and $|\mathbf{B}_{i,j}| \leq J$, when $i \neq j$. When $J$ is notably smaller than unity, we can use the Gersgorian eigenvalue perturbation theory [58] to put a bound on $\lambda_{\min}$. We use the famous Gersgorian discs theorem [58]:

[14] This question only makes sense since we imposed the sparsity requirement on both $\mathbf{x}_1$ and $\mathbf{x}_2$. Otherwise $\mathbf{x}_1$ and $\mathbf{x}_2$ can be arbitrarily far apart while $\mathbf{A}\mathbf{x}_1 = \mathbf{A}\mathbf{x}_2$, if the difference $\mathbf{x}_1 - \mathbf{x}_2$ belongs to the nullspace of $\mathbf{A}$.

[15] This definition is useful for the array processing application since we are ultimately interested not in the values of $\mathbf{x}_1$, but in the indices of support of $\mathbf{x}_1$, which correspond to the estimates of the source locations.

**Theorem 11 (Gersgorian).** *Let* $\mathbf{B} = [b_{i,j}] \in \mathbb{C}^{N\times N}$*, and let* $R_i(\mathbf{B})$ *denote the deleted absolute row sums of* $\mathbf{B}$*:*

$$R_i(\mathbf{B}) = \sum_{j=1, j\neq i}^{N} |b_{i,j}|, \quad 1 \leq i \leq N \tag{7.48}$$

*Then all the eigenvalues of* $\mathbf{B}$ *are located in the union of* $N$ *disks:*

$$\bigcup_{i=1}^{N} \{\mathbf{z} \in \mathbb{C} : |z - b_{i,i}| \leq R_i(\mathbf{B})\} \tag{7.49}$$

We have a very special structure for our $\mathbf{B} = \mathbf{A}'_{\mathcal{I}}\mathbf{A}_{\mathcal{I}}$, which allows to simplify the statement of the theorem considerably. First, all $b_{i,i} = 1$. Second, all $R_i(\mathbf{B})$ can be bounded by the same number, $(2L-1)J$. (Recall that our $\mathbf{B} \in \mathbb{C}^{2L\times 2L}$). Hence $\bigcup_{i=1}^{2L}\{\mathbf{z} \in \mathbb{C} : |z - b_{ii}| \leq R_i(\mathbf{B})\} \subset \{\mathbf{z} \in \mathbb{C} : |z-1| \leq (2L-1)J\}$. Finally, recall that $\mathbf{B}$ is hermitian, so all its eigenvalues are real. Looking at the minimum eigenvalue of $\mathbf{B}$, $\lambda_{\min}$, (the one that is of interest to us), we get $|\lambda_i - 1| \leq (2L-1)J$, which leads to $\lambda_{\min} \geq 1 - (2L-1)J$. This translates right away into a bound on $\|\mathbf{Ax}\|_2^2$:

**Theorem 12 (Sensitivity to noise for sparse representation).** *For* $\mathbf{A}$*,* $\mathbf{x}$*,* $L$ *and* $J$ *as defined above,* $\|\mathbf{Ax}\|_2^2 \geq (1 - (2L-1)J)\|\mathbf{x}\|_2^2$.

Practically the meaning of this result is the following. We are interested in the question of existence of dramatically wrong solutions, where the nonzero indices of $\mathbf{x}_2$ are far apart from the nonzero indices of $\mathbf{x}_1$ in terms of $M(\mathbf{A}_{\mathcal{I}})$. This corresponds to having a small $J$. But if $J$ is small enough, then we get a strictly positive bound on the distance between the two vectors, $\mathbf{x}_1$ and $\mathbf{x}_2$. What the bound says, is that if we get a very wrong $\mathbf{x}_2$, (i.e. $J$ is small, and $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$ has a large $\ell_2$-norm) then the corresponding deviation ($\mathbf{Ax} = \mathbf{Ax}_1 - \mathbf{Ax}_2$) must also be large. In particular, it is impossible to get a very bad sparse solution $\mathbf{x}_2$ where $\mathbf{Ax} = \mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2)$ is small.

The result is interesting, but too many conditions are imposed. In particular, we assumed that the columns of $\mathbf{A}$ corresponding to non-zero elements of $\mathbf{x}$ to be well-separated. It is of interest to have milder requirements. For example require that the columns corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{A}_1$ and $\mathbf{A}_2$ are well separated mutually, but the joint matrix $\mathbf{A}_{\mathcal{I}} = [\mathbf{A}_1\ \mathbf{A}_2]$ does not necessarily have all the columns well-separated. That is to say, it is possible to have large $M(\mathbf{A}_1)$, and large $M(\mathbf{A}_2)$, but all we need is a small $\mathbf{A}'_1\mathbf{A}_2$. Then the bound on the error depends only on the mutual separation of the two sets of columns. This turns out to be more difficult and has not been done yet. Many other important questions also remain for the noisy $\ell_1$ regularization. They can be vaguely stated as "when can we expect to be able to find good approximations of the true underlying sparse signal in noisy scenarios". They are the subject of further research.

Chapter 8

# Model Errors and Self-Calibration

The formulation of our source localization methodology as well as the formulations of all other source localization methods described in Chapter 2 depend heavily on the assumption that all the relevant model parameters are known exactly. Source localization model parameters include the positions, gains, and mutual coupling of the sensors, speed of wave propagation, directivity pattern of the sensors, classification of the sources as farfield/nearfield, and many others[1]. In practice, these parameters are known only approximately through measurement. More importantly, even when these model parameters are measured very accurately, they may evolve with time due to aging of the equipment, or change in the environment, and the array becomes uncalibrated.

Model errors deteriorate the performance of source localization methods. The effect of model errors is felt the most in those methods that rely the most on model structure, in particular in super-resolution methods for source localization. In fact, model errors and the high cost of array calibration are the main reasons for limited practical applications of modern source localization methods such as MUSIC and ESPRIT. The performance of many methods in scenarios with model errors has been analyzed theoretically [59–61], and the results show that even moderate errors can sometimes lead to dramatic deterioration of source localization performance.

Accurate calibration of sensor arrays is typically a very costly procedure, and it is of great interest to develop methods which allow to extend the period of time when the array can be used successfully without recalibration. Two directions of research that try to accomplish this goal are robust source localization and self-calibration. Robust source localization methods make the performance of the array less sensitive to model errors. This is achieved by including the possibility of model errors in the formulation of the method, and guaranteeing that the method performs satisfactorily when model parameters are perturbed from their nominal values. Good examples of robust methods are [62], [63], and [64]. Self-calibration refers to simultaneous localization of the unknown sources and the use of these sources to calibrate the array (estimate perturbed model parameters). We restrict our attention to self-calibration methods only. This chapter first presents the self-calibration problem, then discusses several existing

[1]For simplicity in our work we have assumed that all the sensors are omnidirectional, that all the gains are unity, and that there is no crosstalk between the sensors. In practical applications these assumptions may not be justifiable, and have to be dealt with.

self-calibration methods based on block-coordinate descent, and in the end we follow a similar strategy to extend our source localization method to achieve self-calibration. We present preliminary experimental results.

## ■ 8.1 Self-calibration problem formulation

In general the set of possible model errors is very large, and we limit ourselves to the problem of sensor position uncertainties. Self-calibration for other model errors has the same flavor. In addition, we only look at planar arrays, with sources in the farfield and confined to the same plane. Source locations in this case are the directions of arrival (DOA's) of plane waves from the different sources. Suppose that the nominal positions of the sensors are $\mathbf{p}_0$, but the actual positions have migrated to $\mathbf{p} = \mathbf{p}_0 + \Delta\mathbf{p}$, where $\mathbf{p} = [p_1, .., p_M]$. Then the self-calibration problem that we are trying to solve is the following:

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta}, \mathbf{p})\mathbf{u}(t) + \mathbf{n}(t), t \in \{1, .., T\} \tag{8.1}$$

Quantities of interest include $\boldsymbol{\theta}$ and $\mathbf{p}$. The signals $\mathbf{u}(t)$, and the noise $\mathbf{n}(t)$ are unknown, except for the assumption that both are stationary random processes, and the covariance of the noise is $E[\mathbf{n}(t)\mathbf{n}(t)'] = \sigma^2\mathbf{I}$.

The first question that comes to mind is that of observability. Is it at all possible to determine the unknowns given the data? It is fairly obvious that if all sensor positions $p_m$, $m \in \{1, .., M\}$, are unknown, and all the source DOA's $\theta_k$, $k \in \{1, .., K\}$, are also unknown, then any translation and rotation of the array are unobservable. Shifting the array by a fixed vector with respect to the origin will not change the delays (phase shifts), since relative delays between the sensors are unaffected. Similarly, if the array is rotated by angle $\theta_0$ with respect to the phase center, all the DOA's are also changed by $\theta_0$, making this change unobservable.

Rockah and Schultheiss [65] showed that when the location of one sensor and the direction to another are known, and at least three spectrally (or temporally) disjoint sources are present in unknown locations, then a non-linear array can be calibrated as SNR approaches infinity, or as the number of time samples goes to infinity. They also assume that the displacements of the sensors with respect to their nominal positions are small. Weiss and Friedlander [66] claim that spectral or temporal disjointness is not necessary, and that spatial separation is sufficient. The analysis in [65] is based on the Cramer-Rao-Bound (CRB), and that means that only local observability is taken into account. This is the reason that sensor position errors are required to be small relative to the nominal positions. Describing global observability conditions is a very challenging task.

Even when sensor positions are known, the question of array ambiguity, is still unsolved. Lack of array ambiguity is a necessary assumption before attempting self-calibration. Suppose that the array manifold $\mathbf{a}(\theta)$ is known as a function of $\theta$. The array

is called ambiguous if $\mathbf{a}(\tilde{\theta}) \in \text{span}\{\mathbf{a}(\theta_1), ..., \mathbf{a}(\theta_k)\}$, where $k < M$, and $\tilde{\theta} \notin \{\theta_1, ..., \theta_k\}$, for some set $\theta_1, ..., \theta_k$, and some $\tilde{\theta}$. That means that there exists a linearly dependent set of distinct steering vectors with cardinality less than the number of sensors. It is known that a uniform linear array is unambiguous if sensor spacing is below half of the wavelength. There are some results using differential geometry for general linear and nonlinear arrays [67, 68], but in general whether a given array is ambiguous or not is difficult to answer. Global observability conditions for self-calibration are even more difficult since another set of parameters $p_m$ is added.

Instead of attempting to tackle global observability conditions we also assume that perturbed sensor positions are close to their nominal values, and that the geometry of the array is nondegenerate. Under these assumptions results in [65] apply.

## ■ 8.2 Prior work in self-calibration

An attractive idea for self-calibration can be loosely described as block-coordinate descent. Starting from a source localization method (which requires the knowledge of sensor positions), we first estimate source locations from nominal guesses of sensor positions, and then in turn use these estimates of source locations to give better estimates of the positions of the sensors. The procedure is repeated until convergence. These ideas have been used to extend the maximum likelihood and MUSIC source localization methods in [66, 69]. We discuss them briefly.

Leaving out the details, the basic ML source localization method attempts to minimize the following cost function (also see Section 2.2.4):

$$J_{ML}(\boldsymbol{\theta}) = \sum_{t=1}^{T} \|\mathbf{y}(t) - \mathbf{A}(\boldsymbol{\theta})\mathbf{u}(t)\|_2^2 \tag{8.2}$$

Both $\boldsymbol{\theta}$ and $\mathbf{u}(t)$ for all $t$ are unknown. $\mathbf{A}(\boldsymbol{\theta})$ is unknown since it depends on $\boldsymbol{\theta}$, which we are trying to estimate (see Section 2.1). This is not the overcomplete matrix $\mathbf{A}$. Model parameters (the positions of the sensors) are assumed known and are hidden inside $\mathbf{A}(\boldsymbol{\theta})$. In order to extend this technique to also do self-calibration we explicitly take sensor positions into account:

$$J_{ML}(\boldsymbol{\theta}, \mathbf{p}) = \sum_{t=1}^{T} \|\mathbf{y}(t) - \mathbf{A}(\boldsymbol{\theta}, \mathbf{p})\mathbf{u}(t)\|_2^2 \tag{8.3}$$

We have the maximum likelihood method to solve a part of the problem (minimizing $J_{ML}(\boldsymbol{\theta}, \mathbf{p})$ with respect to $\boldsymbol{\theta}$ when $\mathbf{p}$ is fixed), so it is natural to take advantage of it, splitting the optimization into two parts. This leads to the following "block-coordinate descent" procedure for self-calibration:

1. Let $\hat{\mathbf{p}}^{(0)} = \mathbf{p}$, and let $i = 0$.
2. Find $\hat{\boldsymbol{\theta}}^{(i)} = argmin_{\boldsymbol{\theta}} \;\; J_{ML}(\boldsymbol{\theta}, \hat{\mathbf{p}}^{(i)})$

3. Find $\hat{\mathbf{p}}^{(i+1)} = argmin_{\mathbf{p}} \;\; J_{ML}(\hat{\boldsymbol{\theta}}^{(i)}, \mathbf{p})$

4. Set $i = i + 1$ and return to step 2.

The procedure is run until the change in $J_{ML}(\boldsymbol{\theta}, \mathbf{p})$ becomes negligible as the iteration progresses. Step 2 uses a standard maximum likelihood source localization procedure, as described in Chapter 2. Step 3 can be solved using various non-linear optimization methods, or alternatively, the method in [66] uses a first-order approximation to $J_{ML}(\boldsymbol{\theta}, \mathbf{p})$ as a function of $\mathbf{p}$, which admits a closed-form solution.

A similar extension is made in [69] for the MUSIC algorithm. Without accounting for model errors MUSIC finds the estimates of the locations of the sources by minimizing the following cost function (see Section 2.2.3):

$$J_{MUS}(\theta) = \frac{1}{\|\mathbf{U}_n^H \mathbf{a}(\theta)\|_2^2} \tag{8.4}$$

Here, the matrix $\mathbf{U}_n$ contains the noise subspace singular vectors (it comes from the singular value decomposition of the covariance matrix $\mathbf{R}$ of sensor observations, $\mathbf{y}(t)$). To get the estimates of the locations of the sources one finds $K$ ($K$ is the number of sources) largest peaks of $J_{MUS}(\theta)$. Again, to extend MUSIC to do self-calibration, we make the dependence on $\mathbf{p}$ explicit, add one more step to the procedure, and repeat the steps inside a loop until convergence:

1. Let $\hat{\mathbf{p}}^{(0)} = \mathbf{p}$, and let $i = 0$.

2. Find $K$ largest peaks of $J_{MUS}(\theta, \hat{\mathbf{p}}^{(i)})$, and store them in $\hat{\boldsymbol{\theta}}^{(i)} = [\theta_1, ..., \theta_K]$.

3. Find $\hat{\mathbf{p}}^{(i+1)} = argmin_{\mathbf{p}} \;\; \frac{1}{\sum_{k=1}^{K} \|\mathbf{U}_n^H \mathbf{a}(\hat{\theta}_k^{(i)}, \mathbf{p})\|_2^2}$

4. Set $i = i + 1$ and return to step 2.

In the next section we present how block-coordinate descent can be used to extend our $\ell_1/\ell_p$ source localization methods to do self-calibration in a similar way as was done for MUSIC and ML.

## ■ 8.3 Extension of our $\ell_1/\ell_p$ methods to self-calibration

We have multiple ways of solving the set of overcomplete linear equations

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), t \in \{1, ..T\} \tag{8.5}$$

They are described in Chapter 5. Any one of them can be used within a block-coordinate descent[2] approach for self-calibration. We explicitly parameterize the steering matrix

[2]We use the term block-coordinate descent loosely. In fact the procedures do not correspond exactly to the block-coordinate descent method from nonlinear optimization.

by the unknown positions of the sensors, $\mathbf{p}$. During the source localization step positions are kept constant. During the calibration step a submatrix of $\mathbf{A}$ corresponding to the estimated DOA's, $\hat{\boldsymbol{\theta}}$ is used. That means that we try to find $\mathbf{p}$ to minimize $\sum_{t=1}^{T} \|\mathbf{y}(t) - \mathbf{A}(\hat{\boldsymbol{\theta}}, \mathbf{p})\mathbf{u}(t)\|_2^2$.

In order to make this procedure work we have to remember that our source localization procedures described in Chapter 5 have estimates of DOA's limited to the grid. When one tries to use them within a self-calibration application, the limitation to a grid poses a difficulty. We observe that after a few iterations the solution gets stuck at a local minimum unable to change the estimates of the DOA's since a large jump equal to the grid stepsize is required for improvement.

There are two possibilities of combating this difficulty. The first one uses a spectrum obtained by one of our source localization procedures as an initialization for Maximum Likelihood source localization, which is not limited to a grid. The second possibility uses the multi-scale grid refinement idea outlined in Section 5.3. This way the grid stepsize is made small enough to make its effects negligible. We investigate both possibilities. The algorithms for these two procedures are the following:

Using $\ell_1/\ell_p$ as an initialization to ML:

1. Let $\hat{\mathbf{p}}^{(0)} = \mathbf{p}$, and let $i = 0$.

2. Use one of our source localization methods to solve

$$\mathbf{y}(t) = \mathbf{A}(\hat{\mathbf{p}}^{(i)})\mathbf{s}(t) + \mathbf{n}(t), t \in \{1, ..T\} \tag{8.6}$$

   and find the locations of the $K$ largest peaks of the resulting spectrum. Store them in $\hat{\boldsymbol{\theta}}^{(i)} = [\theta_1, ..., \theta_K]$. These peaks appear on the grid. $\mathbf{A}(\hat{\mathbf{p}}^{(i)})$ is the overcomplete matrix.

3. Initialize ML with these estimates and obtain ML-estimates of source locations, $\hat{\boldsymbol{\theta}}_{ML}^{(i)}$.

4. Find $\hat{\mathbf{p}}^{(i+1)} = argmin_{\mathbf{p}} \quad \sum_{t=1}^{T} \|\mathbf{y}(t) - \mathbf{A}(\hat{\boldsymbol{\theta}}_{ML}^{(i)}, \mathbf{p})\mathbf{u}(t)\|_2^2$ using ML-estimates of the source locations. The matrix $\mathbf{A}(\hat{\boldsymbol{\theta}}_{ML}^{(i)}, \mathbf{p})$ is *not* the overcomplete version, since we use only the DOA's of the estimates, and not the whole grid. Also, there is no need to find $\mathbf{u}(t)$, since we are not interested in it. An equivalent cost function which does not involve $\mathbf{u}(t)$ is $argmin_{\mathbf{p}} \sum_t \|\Pi_{\mathbf{A}(\mathbf{p})}^{\perp}\mathbf{x}(t)\|_2^2$, where $\Pi_{\mathbf{A}(\mathbf{p})}^{\perp}$ is the projection matrix onto the orthogonal complement of the range space of the matrix $\mathbf{A}(\hat{\boldsymbol{\theta}}_{ML}^{(i)}, \mathbf{p})$.

5. Set $i = i + 1$ and return to step 2.

Self-calibration with multi-resolution grid refinement.

1. Let $\hat{\mathbf{p}}^{(0)} = \mathbf{p}$, and let $i = 0$.

2. Use the multi-resolution grid refinement procedure with one of our source localization methods to solve

$$\mathbf{y}(t) = \mathbf{A}(\hat{\mathbf{p}}^{(i)})\mathbf{s}(t) + \mathbf{n}(t), t \in \{1, ..T\} \tag{8.7}$$

   and find the locations of the $K$ largest peaks of the resulting spectrum. Store them in $\hat{\boldsymbol{\theta}}^{(i)} = [\theta_1, ..., \theta_K]$.

3. Find $\hat{\mathbf{p}}^{(i+1)} = argmin_{\mathbf{p}} \ \sum_{t=1}^{T} \|\mathbf{y}(t) - \mathbf{A}(\hat{\boldsymbol{\theta}}^{(i)}, \mathbf{p})\mathbf{u}(t)\|_2^2$ using the estimates of the source locations. Similarly to the last algorithm, there is no need to find $\mathbf{u}(t)$.

4. Set $i = i + 1$ and return to step 2.

## ■ 8.4 Examples

Now we present numerical experiments for both of these procedures. We choose the $\ell_1$-SVD version for the source localization method for both experiments. The step of minimization with respect to the positions of the sensors is carried out using simple gradient descent. We use a 1-D uniform linear array with sensor positions known imprecisely. The analysis in [65] tells that a linear array cannot be calibrated, however when a number of sensor locations are known, this ceases to be the case. For that purpose we fix the positions of two of the sensors. Also, since our array is forced to lie on a known 1-dimensional subspace, it appears that having two sources is sufficient for the sensor positions and the source locations to be observable[3]. (If we treat the array as lying in 2-dimensions, then in our case the second coordinate of every sensor is known exactly). The SNR at the outputs of the sensors is set high (to 40 dB) to make accurate self-calibration possible.

The results of the first procedure (using $\ell_1$-SVD as an initialization to Maximum Likelihood) appear in Figure 8.1. Plot (a) shows original sensor position errors (top) and also errors after running 150 iterations of the self-calibration procedure (bottom). It can be seen that after these iterations the errors are noticeably reduced. Plot (b) additionally shows that as a result of calibration the residual $\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2$ and the error in DOA estimates[4] is reduced as well.

We repeat the same experiment with the second self-calibration procedure ($\ell_1$-SVD with iterative grid refinement), and the results appear to be similar, as we can see in Figure 8.2. Again sensor position errors as well as the residual and the DOA errors decrease substantially as a result of applying 150 iterations of the self-calibration procedure.

[3]We make this statement because with such constraints our self-calibration procedures with small amounts of noise converge to the true values of the unknowns.

[4]For the DOA error we use the norm of the vector of differences between the estimates and the corresponding true source locations.

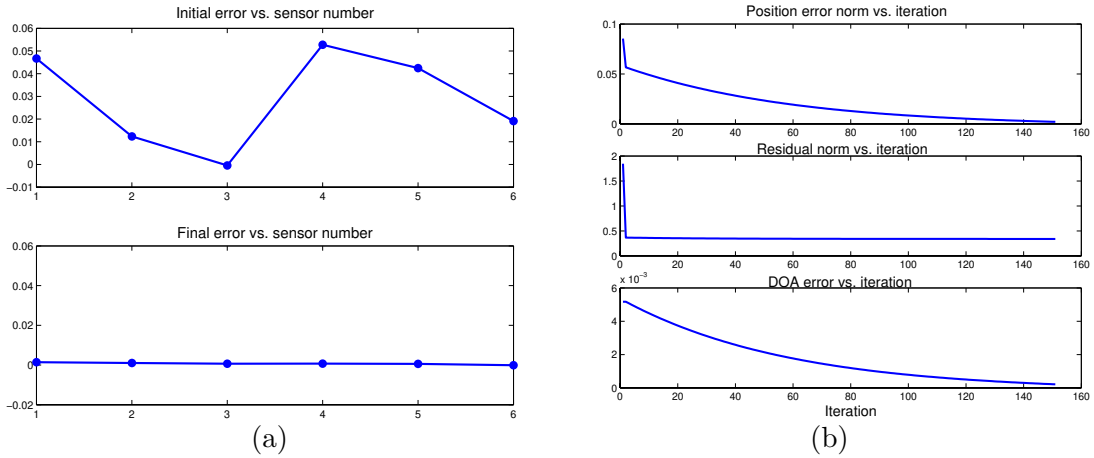



**Figure 8.1.** Self-calibration by ML with $\ell_1$-SVD initialization (a) Original (top) and final (bottom) sensor position errors vs. sensor number. (b) Norm of sensor errors (top), residual $\|\mathbf{y} - \mathbf{A}(\hat{\mathbf{p}}^{(i)})\mathbf{s}\|_2$ (middle), and DOA estimation error (bottom) as a function of iteration number.

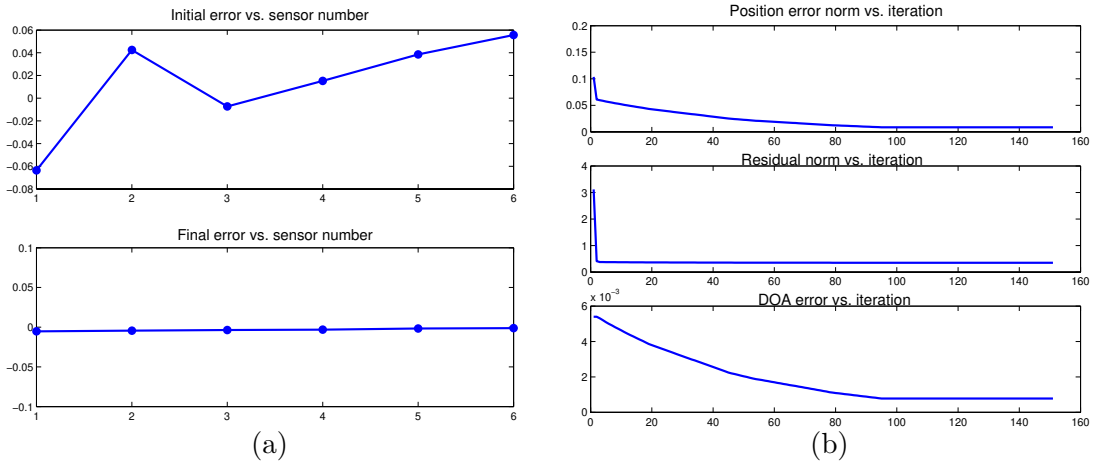



**Figure 8.2.** Self-calibration by $\ell_1$-SVD with multi-level grid refinement. (a) Original (top) and final (bottom) sensor position errors vs. sensor number. (b) Norm of sensor errors (top), residual $\|\mathbf{y} - \mathbf{A}(\hat{\mathbf{p}}^{(i)})\mathbf{s}\|_2$ (middle), and DOA estimation error (bottom) as a function of iteration number.

One additional comment that we need to make is on the usage of the knowledge of the positions of some of the sensors. What we have done in both experiments is to optimize only over the positions of the unknown sensors, and leave the known sensor positions intact. Another approach is to optimize over the positions of all the sensors until convergence, and use the positions of the known sensors at the end to remove the unobservable shift and rotation of the array. We illustrate this procedure in Figure 8.3. Note that in this case the number of iterations necessary to converge can be very small; in fact in the plot most of the change is done during the first iteration. Upon

convergence a shift and a tilt in the array are unobservable, so we use the known sensor positions to remove this ambiguity. For a linear array which has errors limited to the axis of the array, as we have considered in our experiments, this can be done by fitting a straight line to the plot of estimated sensor positions and finding the necessary linear transformation that will make that line go through the known sensor positions (which are themselves on a line). The distribution of errors for estimated sensor positions may have occasional outliers, so a robust line-fitting procedure is preferred. A similar removal of ambiguity can be easily done when sensor errors are not constrained to the array axis, and when the array is non-linear.

In another self-calibration method [70] a similar issue arises and the authors advocate the use of the same approach (optimization over all sensor positions first, and then removing ambiguities). In our experiments we also observe that fixing the position of some of the sensors (the way we have done initially) makes the optimization harder and the sensor position errors at the end have a broken-line pattern (the unknown sensors errors are on a line having a different slope from the line through the fixed sensors)[5]. The approach that first optimizes over the positions of all the sensors and removes unobservabilities later does not seem to lead to such artifacts. Our work with self-calibration is by no means complete, and suggestions for further research are outlined in Chapter 9.

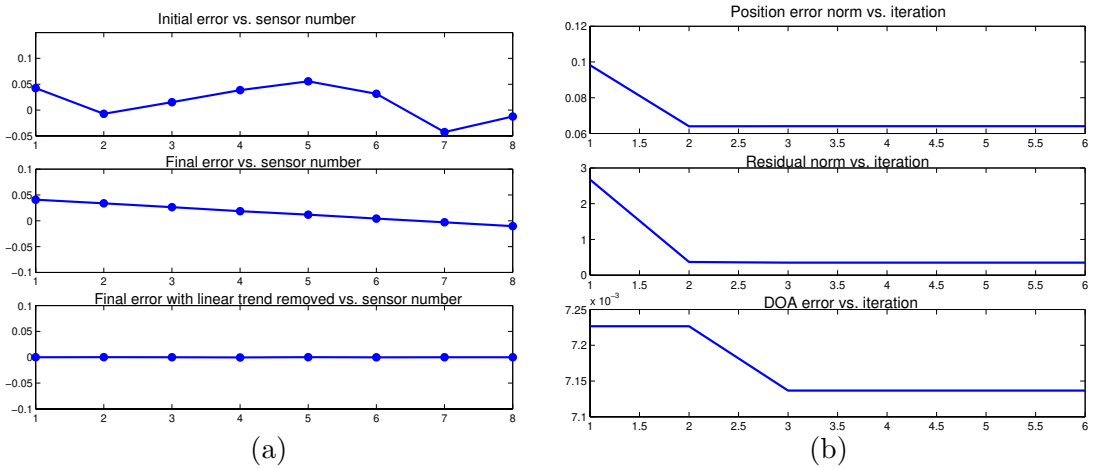



**Figure 8.3.** Alternative way to incorporate the known sensor positions: optimize over *all* sensor positions during the iterative procedure, and upon convergence remove the unobservabilities using the known sensor positions. (a) Original (top), after the iterative procedure (middle), and final, after removing unobservabilities (bottom) sensor position errors vs. sensor number. (b) Norm of sensor errors (top), residual $\|\mathbf{y} - \mathbf{A}(\hat{\mathbf{p}}^{(i)})\mathbf{s}\|_2$ (middle), and DOA estimation error (bottom) as a function of iteration number.

[5] In Figures 8.2 (a) and 8.1 (a) we only show the sensors which are perturbed, thus the effect is not seen.

Chapter 9

# Conclusion

In this chapter we summarize the work done in the thesis, and suggest directions for further research.

## ■ 9.1 Brief summary of the work in the thesis

In this thesis we have considered the problem of sensor array localization of point sources by transforming it into the problem of sparse signal representation using overcomplete bases. This is a very attractive way of looking at source localization because when the sources can be well-modeled as point sources, and their number is small, then the true underlying spatial spectrum is sparse. The problem of signal representation in overcomplete bases is an ill-posed linear inverse problem, and as such, it requires regularization to have unique well-behaved solutions. We are interested in sparse signal representations, so the regularization has to enforce sparsity.

To enforce sparsity we utilized $\ell_p$ penalties with $p \leq 1$. There is an important distinction between $\ell_1$ penalties ($p = 1$), and $\ell_p$ penalties with $p < 1$. For the $\ell_1$ case, the penalty leads to convex optimization problems, whereas for $p < 1$, the associated optimization is nonconvex. For optimization involving $\ell_1$ penalties we used a second order cone programming framework, which has the important benefit of allowing efficient global solutions by interior point methods. When the $\ell_p$ penalty is used, the problem is nonconvex, and we relied on local optimization methods based on half-quadratic regularization.

The sparse signal representation framework is immediately applicable to the single time-sample narrowband source localization problem only. The multiple time-sample narrowband and wideband source localization problems are of greater interest, so we proposed several possibilities to transform the data and to modify the objective functions to be able to handle these problems. In addition, we have considered the questions of removing the limitation of the estimates of source locations to a grid by using adaptive grid refinement, and of automatic choice of regularization parameters inherent in the regularization framework.

We conducted extensive numerical experiments analyzing the behavior of our approach and comparing it to existing source localization methods. We showed that our approach has important advantages such as superresolution, robustness to noise and

limited data, robustness to correlation of the sources, and lack of need for accurate initialization. To address robustness to model errors, we also proposed an extension of the approach to allow self-calibration of sensor position errors by using a procedure similar in spirit to block-coordinate descent.

The second direction of the work done in the thesis is theoretical analysis of the noiseless signal representation problem using overcomplete bases, which is closely related to the noisy signal representation problem that forms the base for our source localization framework. Questions considered in this analysis include the uniqueness of solutions to the noiseless $\ell_0$ problem, and the equivalence of solutions of the $\ell_0$, $\ell_1$ and $\ell_p$ problems. Our results state that if the underlying signal if sprase enough with respect to an ovecomplete basis, then such uniqueness and equivalence of solutions holds. Our results were developed for the case of a general overcomplete basis, where we do not impose any structure, such as being composed of a pair of orthogonal or invertible bases.

A detailed overview of the thesis and a summary of our contributions appears in Chapter 1.

## ■ 9.2 Suggestions for further research

### Choice of the regularization parameter

A very important issue in the framework is the choice of the regularization parameters, $\lambda$ in the joint $\ell_1$ and $\ell_p$ formulations and $\beta$ and $\delta$ in constrained $\ell_1$ formulations from Chapter 4. We have developed an automatic method for the choice of the regularization parameter for the $\ell_1$ constrained form when the distribution of the norm of sensor noise (or the transformed sensor noise) can be well-characterized. It is based on the discrepancy principle from the inverse problems field. An important extension that one could pursue is to the $\ell_1$-SVD case. We have described how to predict the norm of the transformed noise only for scenarios with little noise, whereas moderate and strong levels of noise are of more interest. Additionally, it is of interest to extend the constrained discrepancy principle to the beamspace version of our source localization procedure and to cases with model errors (for self-calibration). Also, recall that although we have developed a procedure for constrained $\ell_p$ optimization paralleling the ML1 version, its performance is very slow. In order to use it successfully a much faster algorithm has to be developed.

It is also worthwhile to continue the investigation of methods for regularization parameter selection from other fields. We briefly discuss the "L-curve" method, the cross-validation method, and universal and min-max regularization parameter selection rules in Appendix F. Some discussion of the viability of these methods for our problem is included, but much more work has to be done to get insights into how to select the regularization parameter for our problem, or to dismiss these methods as inappropriate for our problem.

Another direction is characterizing the dependence of the regularization parameters

for the constrained and unconstrained versions of $\ell_1$ optimization (and also of $\ell_p$). If a good choice of $\lambda$ for the MLJ version can be easily found given a good choice of $\beta$ for ML1 version, then the MLJ cost function has a benefit of lower sensitivity to changes in $\lambda$. Alternatively, if the optimal choice of $\lambda$ can be predicted given the number of sources, then we can use one of the many detectors of the number of sources developed for array processing.

#### Choice of sparsifying regularization functionals

In the thesis we used exclusively $\ell_1/\ell_p$ regularization for enforcing sparsity. However, regularization that favors sparsity is not limited to $\ell_p$-quasi-norms. Many other forms exist, such as Huber regularization, and entropy-based regularization [26]. In particular, the choice of entropy as a regularizing term can be useful for the beamspace formulation in (5.6), since entropy implicitly forces the solutions to be positive. Also, an analysis of the specific features that are necessary for the regularizing term to favor sparsity would provide much insight into the selection of a particular functional. Such analysis has been previously done on some level [28, 43], but deeper understanding can be gained by putting the analysis on firm theoretical grounds and considering much wider sets of regularizing functionals.

Additionally there are issues with $\ell_p$ regularization that remain to be understood. We conjectured that the practical performance of $\ell_p$ with small $p$ is similar to that of $\ell_1$ due to convergence to local minima which are similar in some sense to the global optima of the $\ell_1$ cost function. This phenomenon has to be verified or refuted using more careful analysis. Alternatively, it is worthwhile to develop methods which converge to the global minimum of the $\ell_p$ cost function, as we discuss next.

#### Global solutions to $\ell_p$ or $\ell_0$ problems

The reason that we put more emphasis on $\ell_1$ methods instead of general $\ell_p$ is the convexity of the cost functions associated with the former. However, our theoretical analysis shows that $\ell_p$ has the benefit of requiring lower sparsity to have the global solution of $\ell_p$ cost function match the global solution of $\ell_0$. This motivates the development of global optimization procedures for $\ell_p$ problems.

One idea is based on homotopy continuation. A homotopy is a continuous path in the function space from one continuous function to another. That is to say, we have a parameterized family of functions $f(x,t)$, where $t \in [0,1]$, such that $f(x,0) = f_0(x)$, and $f(x,1) = f_1(x)$, where $f_0(x)$ and $f_1(x)$ are continuous functions of interest. In our case we have a homotopy of $\ell_p$ cost functions parameterized by $p$ from $[p_0, 1]$, where $0 < p_0 < 1$ [1]. This can be used as follows. Given a global solution $\hat{\mathbf{x}}_1$ to the convex $\ell_1$ problem, we decrease $p$ slightly and try to find the global optimum of the $\ell_p$ problem starting with $\hat{\mathbf{x}}_1$ as initialization. We continue decreasing $p$ and using the optimal value from the previous $p$ as initialization until we reach the desired $p_0$. One hopes that the

[1] We have to limit $p$ from below, say by $p_0 = 0.1$, since $\ell_0$ is not continuous.

solution obtained this way is a global optimum of the $\ell_p$ problem for $p = p_0$. There are many difficulties involved in this approach, such as tracking bifurcations, and properly controlling the speed of decrease of $p$. However, even if the global optimum for $p = p_0$ is not found, the procedure may still lead to better local optima than the ones that we obtain using local optimization methods.

Another idea is to try to develop relaxations for the $\ell_0$ problem. Much work has been done in the field of approximations to combinatorial optimization problems, and solutions to many important NP-hard problems can be accurately approximated in an efficient manner using relaxations [71]. An attractive framework for relaxations of combinatorial optimization problems is semidefinite programming.

#### Theoretical analysis

We conducted an analysis of the use of $\ell_1/\ell_p$ penalties to get solutions to the noiseless problem. However, in practice we have to use the noisy formulations instead. So far we have no guarantees that the noisy $\ell_1$ procedure will attain a reasonable solution except for a peculiar result in Section 7.4, which has rather limited applicability. More extensive theory has to be developed for noisy $\ell_1$ and $\ell_0$ optimization. Questions of interest include sensitivity to noise, sensitivity to the choice of the regularization parameter, existence of sparse solutions, and many others. Also one could pursue further an analysis of bias that appears when we use the noisy sparse regularization framework, and a better understanding of the role of structured overcomplete bases (such as our basis $\mathbf{A}$ composed of a grid of samples of the array manifold $\mathbf{a}(\boldsymbol{\theta})$).

#### Analysis of self-calibration and model errors

We presented our initial work on array self-calibration in Chapter 8. More detailed performance analysis of the two proposed methods remains to be carried out. In particular, we would like to compare the performance to that of other self-calibration methods, and to the Cramer-Rao Bound computed for self-calibration in [65]. We considered only the errors in sensor positions; in general, errors in gain, errors in phase, spatial coherence loss, crosstalk between the sensors, and other model errors are also of interest. Another issue worth investigating is the sensitivity of our source localization framework (without self-calibration) to various model errors.

We also would like to consider some alternative self-calibration methods. One possibility inspired from the world of optics is auto-focusing. The idea is that when a photo-camera is out of focus, then the objects in the picture appear blurred, and less sharp. By changing the focus of the camera we change the sharpness of the objects in the picture. Thus in order to focus the camera we can optimize a cost function which measures sharpness of the picture. A similar approach is conceivable for sensor array self-calibration. If the array is uncalibrated, then the spectrum estimate obtained using a particular source localization method may get distorted. By changing the model parameters to minimize this distortion we may be able to calibrate the array. One possibility for a distortion measure is sharpness of the beamforming spectrum. When

the positions of the sensors have parabolic errors, mainlobes of the beamforming spectrum widen. Hence sharpness-enforcing self-calibration is possible. Unfortunately, the distortions do not appear to be limited to simple widening of the mainlobes in the case of general perturbations of sensor positions. In order to use auto-focusing one needs to find practically important families of sensor perturbations and a corresponding measure of distortion of the spectrum of one of the source localization techniques.

#### Miscellaneous

We presented nearfield and wideband extensions of the basic farfield narrowband source localization problem in Chapter 5. However, work remains to be done to analyze their performance and make them computationally more efficient. In particular, we would like to compare the performance of our extensions to conventional wideband and nearfield source localization methods. In addition, for the wideband case we would like to investigate the use of other priors (apart from sparsity) in the frequency domain, for example smoothness. For nearfield, we have to experiment with the multi-resolution grid refinement idea, since sampling in range and in bearing are very different, and a linear 2-D grid is not the optimal strategy.

Many other generalizations are possible which we have not yet addressed. Allowing for more general noise models gives the benefit of applicability to a wider range of practical problems. Some possibilities include non-white (temporally or spatially), non-stationary, and non-Gaussian noise fields. Imposing some structure on the unknown signals, such as cyclostationarity, constant modulus, independence, non-Gaussianity, and temporal structure may allow us to fuse our source localization techniques with other signal processing paradigms, e.g. cyclostationary analysis, blind source separation (BSS) and independent component analysis (ICA).

Another practically significant generalization is the use of non-ideal media exhibiting reverberations and non-ideal sources, e.g. spread sources. This may allow us to tackle problems such as localization of human speakers in small rooms. The topic of distributed sources is a particularly interesting generalization, since a possible solution can be obtained by enforcing sparsity in bases corresponding to the spatial signatures of the sources (our work in the thesis assumes that the spatial signatures are spikes, and we use the corresponding spike basis, i.e. the basis corresponding to the identity matrix). Allowing for non-stationary source positions would extend the scope of applicability of our framework to include multiple-target tracking problems. The robustness of the proposed method to limited number of snapshots could be especially helpful in the tracking context.

Appendix A

# Estimation Theory Concepts and the Cramer Rao Bound

The Cramer Rao inequality puts a lower bound on the variance of any unbiased estimators in the nonrandom unknown parameter estimation problem. Before discussing the CRB, we briefly review some important concepts from estimation theory [72, 73].

A basic estimation problem consists of the following parts: we have a parameter $\boldsymbol{\theta} \in \Theta$, and we have a family of densities parameterized by $\boldsymbol{\theta}$, $p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta})$. We observe a random sample $\mathbf{x}$ from one of the densities, and the goal is to determine $\boldsymbol{\theta}$ based on the observed $\mathbf{x}$. This means constructing a deterministic function $\hat{\boldsymbol{\theta}}(\mathbf{x})$ which furnishes an estimate of $\boldsymbol{\theta}$ for any possible $\mathbf{x}$. There are two important distinctions: if $\boldsymbol{\theta}$ itself has an associated density function $p_{\boldsymbol{\theta}}(\boldsymbol{\theta})$, then this problem is called random parameter estimation, or Bayesian estimation. If on the other hand there is no meaningful prior density that can be assigned to $\boldsymbol{\theta}$, then the problem is that of estimation of a nonrandom but unknown parameter. We focus mainly on the latter case.

We also constrain ourselves by taking into consideration only regular estimation problems. Regularity in this context means satisfying the following set of properties [74]:

- for any $\theta_1 \neq \theta_2$ there is some set $\mathbf{B}$ in the sample space, such that $Pr(\mathbf{x} \in \mathbf{B}; \theta_1) \neq Pr(\mathbf{x} \in \mathbf{B}; \theta_2)$
- All the $p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta})$ have the same support for all $\boldsymbol{\theta}$.
- The parameter space, $\Theta$ is an open interval in $\mathbb{R}^k$.
- First and second order derivatives of $p_{\mathbf{x}}(\mathbf{x}; \boldsymbol{\theta})$ with respect to $\theta$ can be interchanged with integrals over $\mathbf{x}$.

The first requirement is identifiability of the model. If the condition is not met, then it is not possible to distinguish between the two models corresponding to $\theta_1$ and $\theta_2$. The second condition is necessary because we shall be dealing with log-likelihoods, which have to be defined for any $\mathbf{x}$ and $\boldsymbol{\theta}$. The open interval condition avoids the need to deal with differentiability issues on the boundary of the parameter space. The last condition is satisfied for all of the probability densities that we consider in this work.

Constructing an estimator is not a difficult task, the difficulty comes when we try to construct a good estimator. Part of the difficulty lies in choosing an appropriate notion of goodness. Two such notions that are extensively used in the field are the bias, $bias(\boldsymbol{\theta}) = E[\hat{\boldsymbol{\theta}}(\mathbf{x})] - \boldsymbol{\theta}$, and the variance, $Var(\boldsymbol{\theta}) = E[(\hat{\boldsymbol{\theta}}(\mathbf{x}) - \boldsymbol{\theta})^2] - E[(\hat{\boldsymbol{\theta}}(\mathbf{x}) - \boldsymbol{\theta})]^2 = E[(\hat{\boldsymbol{\theta}}(\mathbf{x}) - \boldsymbol{\theta})^2] - bias^2(\boldsymbol{\theta})$. In general one wants to minimize both the bias and the variance, which in general is not possible simultaneously. A common approach is to find the estimator having the minimum variance among the unbiased ones (MVU), but an MVU estimator may not even exist for some problems. Nevertheless, in a large number of cases, the maximum likelihood approach for estimation does yield MVU estimators. The Maximum Likelihood Estimator (MLE) is defined as

$$\hat{\boldsymbol{\theta}}_{ML}(\mathbf{x}) = \arg max_{\boldsymbol{\theta}}\ p_y(\mathbf{x};\boldsymbol{\theta}) \tag{A.1}$$

We shall mention a connection of MLE and the CRB shortly.

The Cramer Rao Bound is used to evaluate potential estimators. If an unbiased estimator meets the CRB with an equality, that means that the task of search for a good estimator is over, we cannot get the variance any lower (of course there is always a possibility of considering other metrics of merit).

**Theorem 13 (Cramer-Rao Inequality).** *If an estimator $\hat{\boldsymbol{\theta}}(x)$ is unbiased ($E[\hat{\boldsymbol{\theta}}(\mathbf{x})] = \boldsymbol{\theta}$), then*

$$Var[\hat{\boldsymbol{\theta}}(\mathbf{x})] \geq \left( E\left[ \left[ \frac{\partial\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]^T \left[ \frac{\partial\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \right] \right)^{-1} = \left( -E\left[ \frac{\partial^2\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial\ \boldsymbol{\theta}^2} \right] \right)^{-1} \tag{A.2}$$

*where the probability $p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})$ is assumed to be strictly positive and twice continuously differentiable.*

The matrix $I_{\mathbf{x}}(\boldsymbol{\theta}) = -E\left[ \frac{\partial^2\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial\ \boldsymbol{\theta}^2} \right]$ is called the expected Fisher information matrix. It is independent of $\mathbf{x}$, but varies with $\boldsymbol{\theta}$, so the bound is a function of the unknown parameter. The notation $\frac{\partial^2\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial\ \boldsymbol{\theta}^2}$ stands for the Jacobian row vector. One corollary of the CRB [72] is that if an efficient estimator exists then it has the form

$$\hat{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta} + \mathbf{I}_{\mathbf{x}}(\boldsymbol{\theta})^{-1} \left[ \frac{\partial\ ln\ p_y(\mathbf{x};\boldsymbol{\theta})}{\partial\ \mathbf{x}} \right]' \tag{A.3}$$

where the dependence on $\boldsymbol{\theta}$ cancels out (otherwise we are left with using the unknown parameter in its own estimation). In fact if (A.3) does not depend on $\boldsymbol{\theta}$, then $\hat{\boldsymbol{\theta}}(\mathbf{x})$ meets the CRB with equality. An estimator meeting the CRB for every value of $\boldsymbol{\theta}$ is labeled an efficient estimator. If an estimator is efficient then it is also the minimum variance unbiased estimator.

Another important ingredient in the theory is the connection between the CRB and the Maximum Likelihood estimation. If an efficient estimator exists then it is the ML

estimator. Unfortunately, if there exists no efficient estimator then ML may be biased, and even if unbiased it may not achieve the minimum variance.

Another property of the CRB which is relevant for array processing is the local and global behavior of estimates for the nonlinear estimation case, for example $\mathbf{x} = \mathbf{h}(\boldsymbol{\theta})+\mathbf{n}$, where $\mathbf{h}$ is non-linear and smooth and $\mathbf{n}$ is jointly Gaussian. In this case no efficient estimators exist [1].

However, if the value of $\boldsymbol{\theta}$ is known to be confined in a sufficiently small region around $\boldsymbol{\theta}_0$ such that $\mathbf{h}(\boldsymbol{\theta})$ can be well-approximated by linearization $\mathbf{h}(\boldsymbol{\theta}_0) + \nabla\mathbf{h}(\boldsymbol{\theta}_0)(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$, then as long as the noise is small, the ML estimate for the linearized problem will follow the CRB very closely. However, once the SNR goes below a certain threshold, CRB becomes a very poor over-optimistic lower bound. This phenomenon is usually termed the threshold behavior.

There is an extension of the CRB to the case when the estimators are biased. Due to the presence of bias, the variance of the estimator is not a proper measure of its quality, and a better one is the mean-squared error, $E[(\hat{\boldsymbol{\theta}}(\mathbf{x}) - \boldsymbol{\theta})^2] = Var[\hat{\boldsymbol{\theta}}(\mathbf{x})] + bias(\boldsymbol{\theta})^2$. It can be shown [73] that the following is true for any estimator:

$$Var[(\hat{\boldsymbol{\theta}}(\mathbf{x}) - \boldsymbol{\theta})] \geq \left( E\left[ \left[ \frac{\partial\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right]^T \left[ \frac{\partial\ ln\ p_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \right] \right)^{-1} \tag{A.4}$$

The Cramer-Rao bound is not the only bound on estimator variance, and in fact several tighter bounds on variance and the mean-squared error have been developed, such as the Barankin bound [75], Bhattacharyya bound [76], Weiss-Weinstein [77], and several others [78]. However, their computation is very complex, and the CRB remains most popular.

[1] Also more generally, if $\hat{\boldsymbol{\theta}}(\mathbf{x})$ is an efficient estimator of $\boldsymbol{\theta}$, then $\hat{\phi}(\mathbf{x}) = \phi(\hat{\boldsymbol{\theta}}(\mathbf{x}))$ is an efficient estimator of $\phi(\boldsymbol{\theta})$ if and only if $\phi$ is a linear function.

Appendix B

# Interior Point Methods

We attempt to briefly motivate the use of interior point methods, and in particular to explain why and how they are applicable to our work. Ideas related to interior point (IP) methods existed for a long time (penalty and barrier methods were investigated in the 60's, and then interest faded due to some difficulties), but in the past two decades they experienced a dramatic resurgence of attention which has lead to major changes in the field of optimization, both in theory and in complexity of optimization tasks which can be readily handled. A great number of methods which can be labeled as interior point have been developed, and successfully used in applications. We limit our discussion to central-path IP methods, which are the most used in practice, and enjoy the most developed theoretical background. IP methods can be applied to various linear and nonlinear programs, but their use is most attractive for special classes of convex programming, namely semidefinite programming (SDP), second order cone (SOC) programming, and linear programming (LP) [1], due to the existence of extensive theoretical results on convergence and complexity. We give a very brief account of central-path following IP methods summarizing the main ideas, and theoretical results relevant to the rest of the manuscript. For a more thorough understanding the reader is referred to [10, 35, 38, 79], from which most of the following (as well as preceding) presentation was borrowed.

A general convex problem has the following form:

$$\min \ \mathbf{c}'\mathbf{x} \tag{B.1}$$

$$\text{such that } \mathbf{x} \in \mathcal{X} \tag{B.2}$$

where $\mathcal{X}$ is a convex region. A nonlinear convex objective function $f(\mathbf{x})$ can be represented in the above form as follows: $\min \ t$ such that $f(\mathbf{x}) \leq t$.

The basic idea of central-path IP methods is to introduce a parameterized family of problems augmented with a barrier function for the convex region. The barrier function has to be well-defined in the feasible region, smooth, strongly convex [2] in the interior

[1] These classes of optimization problems form a nested sequence, (LP $\subset$ SOC $\subset$ SDP) but theoretical results are richer with more restrictions. Thus it is useful to view them separately

[2] $f(\mathbf{x})$ is strongly convex if it is convex, and additionally, $(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))'(\mathbf{x} - \mathbf{y}) \geq \alpha \|\mathbf{x} - \mathbf{y}\|_2^2$, for some $\alpha > 0$, and $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

of $\mathcal{X}$, and increase to infinity as the boundary of $\mathcal{X}$ is approached. For example, for linear programming, with constraint $\mathbf{x} \geq 0$, ($\mathbf{x} \in \mathbb{R}^n$) a valid barrier function is

$$F(\mathbf{x}) = \begin{cases} -\sum_{i=1}^{n} log(\mathbf{x}_i), & \mathbf{x} > 0 \\ \infty, & otherwise. \end{cases} \tag{B.3}$$

By applying a barrier function we transform a convex constrained problem into a family of convex unconstrained problems, (which are infinite outside the feasible region of the original problem). Denote the barrier function for the convex region $\mathcal{X}$ by $F : \mathcal{X} \to \mathbb{R}$. We consider the following family of optimization problems:

$$\min F_t(\mathbf{x}), \quad F_t(\mathbf{x}) = \ t\mathbf{c}'\mathbf{x} + F(\mathbf{x}) \tag{B.4}$$

When $t$ is very large the effect of the barrier is negligible except in a very narrow region around the boundary of $\mathcal{X}$. Also, provided the set $\mathcal{X}$ is bounded, for every $t > 0$ the function $F_t(\mathbf{x})$ attains a minimum $\mathbf{x}^*(t)$ in the interior of $\mathcal{X}$. The curve $\mathbf{x}^*(t)$ is called the central path, and as $t \to \infty$, it approaches the set of optimal solutions of the original constrained problem.

Given this family of functions there are two ways to proceed: the most obvious one is to set $t$ to a very large value, so that the region where the effect of the barrier is felt is very narrow, and we get a very good approximation to the optimal solution $\mathbf{x}^*$ of (B.2) by minimizing $F_t(\mathbf{x})$. Another approach is to start with a small $t_0$, find the optimal solution $\mathbf{x}^*(t_0)$, then increase $t$ to $t_1$, and use $\mathbf{x}^*(t_0)$ as initialization for finding $\mathbf{x}^*(t_1)$, and proceed similarly until $t$ reaches a large value. We stop when $t$ is very large, where as before $\mathbf{x}_t^*$ is close to $\mathbf{x}^*$.

The second approach seems rather strange at first sight, since it requires solving many optimization problems, whereas the first problem requires the solution of just one, of exactly the same form. The reason that the second approach receives a vast amount of attention, and no reasonable practical optimization routine uses the first one, lies in the convergence properties of Newton's method.

Newton's method has the fastest local convergence rate among all smooth unconstrained optimization techniques. As long as we start close enough to the optimal solution the rate of convergence of Newton's method is quadratic. However, far from the optimal point its convergence rate can be very slow. Using other unconstrained optimization techniques, such as gradient descent or conjugate gradients, the rate of convergence may also be very slow. Hence, when attempting to minimize $F_t(\mathbf{x})$ for a large $t$ right away, we will have a slow convergence rate. The idea of the second method is to change $t$ slowly, such that if $t_i$ is sufficiently close to the optimal solution of $F_i$, then going to $t_{i+1}$ we will still be in the region of superlinear convergence for $F_{i+1}$. Also, for very small $t$ the optimal solution is either known (can be easily found analytically from the properties of the convex set and the barrier function), or its numerical computation is very simple and also has superlinear convergence rate, thus initialization does not pose a problem. Thus by turning to the method with incremental $t$-values we get a much faster convergence rate than by solving once with a large $t$.

The only requirements for the next $t$-step to fall into quadratic regions of convergence for Newton's method are that we increase the values of $t$ at a rate that is slow enough, and also to let Newton's method run until the current solution is sufficiently close to the central path. Much research has been directed at quantifying these two requirements, and many step-size rules have been suggested. For the class of convex problems a deep theory of self-concordant barriers has been developed [10], which can be used to prove polynomial-time convergence for all convex problems when self-concordant barriers are used.

The theory of self-concordant barriers is well beyond the scope of this work, and we summarize only the main ideas. If there exists a self-concordant[3] barrier $F(\mathbf{x})$ for the convex constraint region $\mathcal{X}$, with the so-called self-concordance parameter $\theta(F)$, then the proximity of the point to the central path can be meaningfully measured (can be used to quantify the necessary proximity for superlinear convergence of Newton's method) by the local norm induced by the Hessian $H(\mathbf{x})$ of the barrier function, and the progression of $t$-values can be selected as $t_{i+1} = t_i(1 + \frac{0.1}{\sqrt{\theta(F)}})$. For more details refer to [38], and references therein.

Self-concordant barriers are known for only a subset of convex problems, but luckily this subset includes semidefinite programming, second order cone programming, and linear programming. For LP with constraint set $\mathbf{Ax} \leq \mathbf{b}$, the self-concordant barrier is $F(\mathbf{x}) = -\sum_i log(b_i - \mathbf{a}_i'\mathbf{x})$, and for SOC with constraint set $x_k \geq \|(x_1, ..., x_{k-1})\|_2$, the self-concordant barrier is $F(\mathbf{x}) = -log(x_k^2 - x_1^2 - ... - x_{k-1}^2)$. When the convex constraint region $\mathcal{X}$ is a direct product of multiple convex regions $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 .... \times \mathcal{X}_K$, which possess self-concordant barriers $F_k(\mathbf{x}_k)$, with self-concordance parameters $\theta_k$, then $\mathcal{X}$ has a self-concordant barrier $F(\mathbf{x}) = \sum_k F(\mathbf{x}_k)$, with parameter $\theta = \sum_k \theta_k$. Thus, all of the convex problems which are of interest in this work (LP, SOC) possess self-concordant barriers, and hence can be efficiently solved by path-following interior point methods.

There are many practical details associated with implementing path-following IP methods. Two very important details are the trade-offs associated with short step versus long step schemes for path-following, and using the dual problem. The theory of self-concordant barriers provides us with a step-size rule for $t$ which is guaranteed to achieve polynomial time complexity. However, the worst-case results are not representative of the average-case performance which is seen in practice. It has been observed that using a faster step-size rule ($t$'s are increased more rapidly), and using a greater number of Newton's steps leads to much better practical performance. The worst case complexity results for long-step path-following methods however have less guarantees than their short-step counterparts.

Another important implementation issue is the use of the problem dual to (B.2). The dual problem can be analytically computed for LP, SOC, and SDP. Our discussion so far involves a primal interior point scheme, where we are augmenting the primal

[3]Self-concordance means satisfying several properties with respect to the local variability of the Hessian of the barrier function.

problem with the barrier function. The same can be done to the dual problem, and depending on the dimension of the primal set, and the number of constraints, the dual problem can be more efficient. However, the main benefit comes from considering both the primal and the dual problems at the same time, which leads to the so-called primal-dual IP methods. One of the benefits of considering both problems at the same time is that we can find out how close we are to the optimal solution by looking at the duality gap. The theory of IP methods is very rich, but since they are not the focus of our work, we put an end to the exposition here.

Appendix C

# Convex Analysis and Subdifferentials

We review several basic concepts necessary to understand convex non-smooth unconstrained optimality conditions. Consult [35, 80] for an in-depth exposition. First of all, a function $f : \mathbb{R}^N \to \mathbb{R}$ is convex if it satisfies

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y}), \mathbf{x}, \mathbf{y} \in \mathbb{R}^N, \lambda \in [0, .., 1] \tag{C.1}$$

A directional derivative of a function f in direction $\mathbf{u}$ is defined by:

$$f'(\mathbf{x}; \mathbf{u}) = \lim_{t \to 0^+} \frac{f(\mathbf{x} + t\mathbf{u}) - f(\mathbf{x})}{t} \tag{C.2}$$

if the limit exists. A basic result of convex analysis is that if $f$ is a convex function then the limit exists for any $\mathbf{u}$, i.e. $f$ is directionally differentiable in any direction. The subdifferential of a convex $f : \mathbb{R}^N \to \mathbb{R}$ at $\mathbf{x} \in \mathbb{R}^N$ is defined as the following set:

$$\partial f(\mathbf{x}) = \{\xi \in \mathbb{R}^N | f(\mathbf{y}) \geq f(\mathbf{x}) + \xi^T (\mathbf{y} - \mathbf{x}) \ \ \forall \ \mathbf{y} \in \mathbb{R}^N\} \tag{C.3}$$

Each element of $\partial f(\mathbf{x})$ is called a subgradient of $f$ at $\mathbf{x}$. An alternative definition of the subdifferential of a convex function $f$ is:

$$\partial f(\mathbf{x}) = \{\xi \in \mathbb{R}^N | f'(\mathbf{x}; \mathbf{u}) \geq \xi^T \mathbf{u} \ \ \forall \ \ \mathbf{u} \in \mathbb{R}^N\} \tag{C.4}$$

The subdifferential is a generalization of the gradient of $f$. In fact, if $f$ is convex and is also differentiable at a point $\mathbf{x}$ (the directional derivative of $f$ in direction $\mathbf{u}$ is a linear function of $\mathbf{u}$: $f'(\mathbf{x}; \mathbf{u}) = \nabla f(\mathbf{x})^T \mathbf{u}$, $\nabla f(\mathbf{x}) \in \mathbb{R}^N$), then

$$\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\} \tag{C.5}$$

i.e. the subdifferential consists of a single vector, the gradient of $f$ at $\mathbf{x}$. The only subgradient in this case is the gradient. Lastly, we present the unconstrained non-smooth convex optimality conditions:

If $f : \mathbb{R}^N \to \mathbb{R}$ is convex, then $f$ attains a global minimum at $\mathbf{x}$ if and only if $\mathbf{0} \in \partial f(\mathbf{x})$.

Appendix D

# Conjugate Gradients (CG) and Preconditioning

The method of conjugate gradients [35, 48] in its pure form applies to the equations of the form $\mathbf{y} = \mathbf{Q}\mathbf{x}$, where $\mathbf{Q}$ is symmetric and positive definite, which we assume from here on. Extensions are possible for other cases, but we have no need for them, since we have $\mathbf{Q} = \mathbf{H}\left(\hat{\mathbf{s}}^{(n)}\right)$ symmetric and p.d., see (4.44). The problem of finding $\mathbf{x}$ in the linear system $\mathbf{y} = \mathbf{Q}\mathbf{x}$ is equivalent to minimizing the convex quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}'\mathbf{Q}\mathbf{x} - \mathbf{x}'\mathbf{y}$. Solving one also solves the other. Many methods exist for the unconstrained minimization of functions. The simplest one is gradient descent, where we choose an initial point $\mathbf{x}_0$, and then advance in the direction of negative gradient: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k)$. The stepsize, $\alpha$, is chosen to minimize $f$ over the half-line $\{\mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k) | \alpha > 0\}$. The method is very simple, but if the condition number of $\mathbf{Q}$ is very high, the level sets of $f$ are strongly elongated ellipsoids, and the convergence rate of the algorithm is extremely slow.

The method of conjugate gradients has similarities with gradient descent, but it performs considerably better. It is in fact guaranteed to converge to the optimal solution in at most $N$ steps, where $\mathbf{Q} \in \mathbb{R}^{N\times N}$. For large $N$, if $\mathbf{Q}$ is well-conditioned, or if the number of distinct eigenvalues of $\mathbf{Q}$ is small, the method may converge considerably faster than in $N$ steps.

The main constituent of the method is the concept of $\mathbf{Q}$-conjugacy. A set of vectors $\mathbf{d}_1, ...\mathbf{d}_N$ is said to be $\mathbf{Q}$-conjugate if $\mathbf{d}_i'\mathbf{Q}\mathbf{d}_j = 0$, whenever $i \neq j$. To find a set of $\mathbf{Q}$-conjugate vectors starting from $N$ arbitrary linearly independent vectors $\mathbf{c}_i$, a Gramm-Schmidt-like procedure is used. First set $\mathbf{d}_1 = \mathbf{c}_1$. Then, $\mathbf{d}_{k+1} = \mathbf{c}_{k+1} - \sum_{m=0}^{k} \gamma_m^{k+1}\mathbf{d}_m$. The coefficients $\gamma_m^{k+1}$ are chosen such that $\mathbf{d}^{k+1}$ is Q-conjugate to all the $\mathbf{d}_1$, .., $\mathbf{d}_k$. This condition leads to $\gamma_m^{k+1} = \frac{\mathbf{c}_{k+1}'\mathbf{Q}\mathbf{d}_k}{\mathbf{d}_k\mathbf{Q}\mathbf{d}_k}$.

The method of conjugate gradients applies this Gramm-Schmidt process to the set of gradients of $f$ at points $\mathbf{x}_k$, where $\mathbf{x}_0$ is arbitrary, and $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{d}_k$. Similarly to the gradient descent, $\alpha$ is chosen to minimize $f$ over the half line with $\alpha > 0$. In the process of iterations, the method optimizes $f$ over an expanding linear manifold. After step 1, $f$ is minimized over the span$\{\mathbf{d}_1\}$, whereas after step k, $f$ is minimized over span$\{\mathbf{d}_1, ..., \mathbf{d}_k\}$. The set of vectors $\mathbf{d}_1$, ..., $\mathbf{d}_k$ are $\mathbf{Q}$-conjugate, hence linearly

independent. After $N$ iterations the span of $\mathbf{d}_i$'s is the same as the whole space, hence the global optimum is found.

### Preconditioning

As we mentioned, the actual number of iterations that is required for the conjugate gradient algorithm to converge can be considerably smaller than $N$ if $\mathbf{Q}$ has a few distinct eigenvalues, or if it is well-conditioned. Even when $\mathbf{Q}$ does not have these properties it may be possible to find a related linear system, $\tilde{\mathbf{Q}}\tilde{\mathbf{x}} = \tilde{\mathbf{y}}$, which does. The second property of the approximation is that we have to be able to get the optimal solution to the original problem, $\hat{\mathbf{x}}$, easily from the solution of the related problem. We consider the following transformation [48]: $\tilde{\mathbf{Q}} = \mathbf{C}^{-1}\mathbf{Q}\mathbf{C}^{-1}$, $\tilde{\mathbf{x}} = \mathbf{C}\mathbf{x}$, and $\tilde{\mathbf{y}} = \mathbf{C}^{-1}\mathbf{y}$, with $\mathbf{C}$ symmetric positive definite. It is possible to use the optimal solution $\hat{\tilde{\mathbf{x}}}$ to this problem to get $\hat{\mathbf{x}} = \mathbf{C}^{-1}\hat{\tilde{\mathbf{x}}}$, but in practice the algorithm is rewritten in terms of $\mathbf{x}$. In order for this transformation to enhance performance, the matrix $\tilde{\mathbf{Q}}$ has to be well conditioned, *and* the solution of linear systems involving the matrix $\mathbf{M} = \mathbf{C}\mathbf{C}$ has to be very fast (compared to inverting $\mathbf{Q}$). The second condition is satisfied for example if $\mathbf{M}$ is diagonal or block-diagonal.

In practice proper preconditioning can considerably reduce running time of the conjugate gradient method. We have used a diagonal preconditioner, and have not observed noticeable savings. Partially this may be explained by the observation that for our application $\mathbf{Q}$ has a very rapidly decreasing singular value spectrum, and except for a few very large singular values, the rest are very small. Thus, conjugate gradients converges fairly fast to a reasonably accurate solution even without preconditioning.

Appendix E

# Minimizing $\ell_1$ Norm subject to $\ell_\infty$ Constraint

Here we describe how to compute $L_1(\mathbf{A})$ used in Section 7.3.1 with real-valued data. The problem itself is non-linear, but it can be solved by solving a set of linear problems. To restate the definition, we need to find

$$\min \|\boldsymbol{\delta}\|_1 \text{ such that } \|\boldsymbol{\delta}\|_\infty = 1, \text{ and } \boldsymbol{\delta} \in Null(\mathbf{A}) \tag{E.1}$$

Recall that $\boldsymbol{\delta} \in \mathbf{R}^N$. We can partition the feasible set into $N$ regions where in the $i$-th region we have $\delta_i = 1$, and $\delta_j \leq 1$ for all $j$. For each of these regions, $\ell_1$ minimization can be recast into a linear problem by introducing $\mathbf{x}^+$ and $\mathbf{x}^-$, where $x_i^+ = \max\{x_i, 0\}$, $x_i^- = \max\{-x_i, 0\}$, from which $\mathbf{x}$ can be simply recovered as $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$. The $i$-th subproblem looks as follows:

$$\begin{aligned}
\min\ & \mathbf{1}'\mathbf{x}^+ + \mathbf{1}'\mathbf{x}^- && \text{(E.2)}\\
\text{subject to } & [\mathbf{A}\ -\mathbf{A}][\mathbf{x}^+; \mathbf{x}^-] = 0, && \text{(E.3)}\\
& \mathbf{c}_i'\mathbf{x} = 1, && \text{(E.4)}\\
\text{and } & \mathbf{x}^+ > \mathbf{0}, \mathbf{x}^- > \mathbf{0}, \mathbf{x}^+ \leq \mathbf{1}, \mathbf{x}^- \leq \mathbf{1} && \text{(E.5)}\\
& && \text{(E.6)}
\end{aligned}$$

$\mathbf{1}$ is a vector of ones so that $\mathbf{1}'\mathbf{x} = \sum_i x_i$. The vector $\mathbf{c}_i$ is zero everywhere except at the $i$-th coordinate, where it equals 1, so that the $\ell_\infty$ condition is enforced.

Then $L_1(\mathbf{A})$ is found as the minimum value among the solutions for all the subproblems. Some of the subproblems may turn out infeasible due to the fact that $Null(\mathbf{A})$ does not contain a vector with a particular dominant coordinate. These subproblems can be ignored, since they do not affect $L_1(\mathbf{A})$.

Appendix F

# Analysis of the Applicability of Alternative Methods for Automatic Selection of the Regularization Parameter

In Section 5.4 we described an automatic method to choose the regularization parameter based on the the discrepancy principle. In the literature on inverse problems, machine learning, and signal representation, other methods have been proposed for related problems. In this appendix we consider several prominent methods: the "L-curve", cross-validation, and min-max and universal rules. We address the question whether it is possible to use any of these rules for our problem of source localization using the sparse representation framework.

## ■ F.1 L-curve

When the statistics of the noise are not known, the discrepancy principle from Section 5.4.1 can no longer be used. However, several methods have been developed which do not rely on this information. A very popular method in the inverse problem community, especially in applications to image and signal restoration, is the so-called "L-curve" method [24,26]. We discuss the method first, and then give reasons why it does not suit our needs. The name of the method comes from the shape of the curve of the residual versus the regularizer for a family of solutions parameterized by the regularization parameter, $\lambda$. For each $\lambda$ over the possible range we compute the minimizing value $\hat{\mathbf{s}}(\lambda)$ of the cost function $J(\mathbf{s}) = J_1(\mathbf{s}) + \lambda J_2(\mathbf{s})$, and compute the associated residual, $J_1(\mathbf{s})$, and the regularizer, $J_2(\mathbf{s})$. The L-curve is the plot of $J_2(\mathbf{s})$ vs. $J_1(\mathbf{s})$ on the log-log scale. Each point on the curve corresponds to a particular $\lambda$. It has been observed that the shape of the curve bears some resemblance to letter 'L', with the corner of the curve a good choice for the regularization parameter. The authors of the method explain that the reason for the shape of the curve lies in the fact that for low values of $\lambda$ (under-regularized), the solution is dominated by the amplified noise, and the term

$J_2$ is affected much greater by small changes of $\lambda$ than $J_1$. For high values of $\lambda$ the opposite effect occurs, the solution is already over-regularized, so small changes in $\lambda$ lead to small changes in $J_2$, whereas the fit to the data, $J_1$, is affected dramatically.

We observe that one of the necessary constituents for the L-curve to justify its name (have the corresponding shape) is the distribution of singular values of the forward operator, $\mathbf{A}$, in the linear inverse problem $\mathbf{y} = \mathbf{A}\mathbf{s}$. In the problems tackled by the image processing community, the operators usually correspond to smoothing, and the distribution of the singular values is very spread out, leading to high condition numbers. The existence of many small singular values as well as a few large ones leads to the noise amplification effect observed in practical inverse problems. However, in array processing, the resulting overcomplete matrix $\mathbf{A}$ is a Fourier-type operator and typically has a well-behaved singular value distribution (unless there are accumulation points in the grid of $\theta$, which occur in the multi-scale implementation of Section 5.3).

The main reason for using regularization in our approach is not to eliminate discontinuous behavior of the dependence of the solution of the data, but to handle the non-uniqueness of solutions, the other ingredient of ill-posedness. Due to this behavior, the shape of the $J_1$ vs. $J_2$ curve no longer has the same shape as the one observed by researchers in inverse problems such as image processing dealing with widely distributed singular value spectra. In order to demonstrate this claim, we conduct several experiments comparing a typical inverse problem in image processing to that of array processing using the regularization framework.

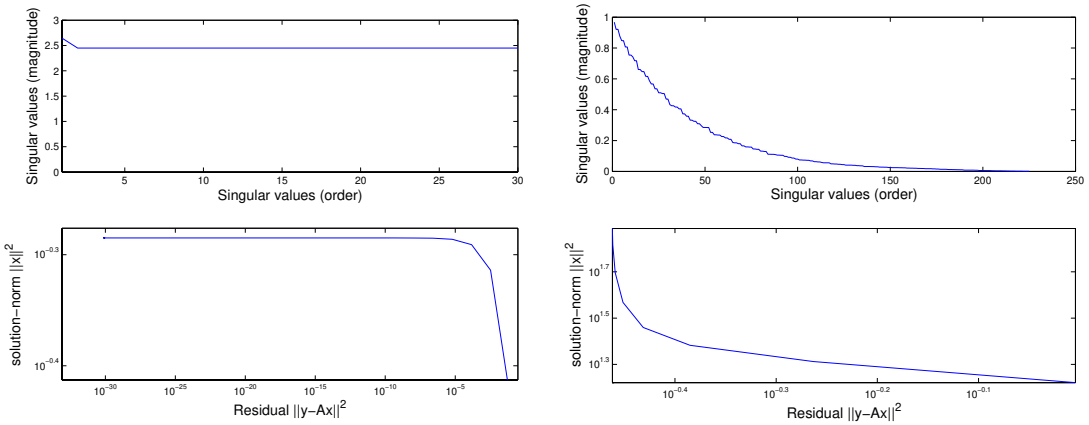



**Figure F.1.** Singular values of $\mathbf{A}$, and the plot of $\|\mathbf{s}(\lambda)\|_2^2$ versus $\|\mathbf{y} - \mathbf{A}\mathbf{s}(\lambda)\|_2^2$, for a range of $\lambda$ (a) Array processing $\mathbf{A}$ . (b) Image processing $\mathbf{A}$ .

In Figure F.1 (b), the top plot shows the distribution of singular values for $\mathbf{A}$ representing a convolution operator for a 2-D Gaussian smoothing filter of size $5 \times 5$, with standard deviation 1. Such filters are commonly used as image blurring kernels. The image size is taken very small, $15 \times 15$, the convolution matrix $\mathbf{A}$ has size $361 \times 225$,

so there are only 225 singular values. Note that there are numerous very small singular values, as well as large ones, and the transition from large to small is very smooth. Since the problem has a small size, the L-curve (lower plot) is not very pronounced (for larger images the condition number is several orders of magnitude higher), but the corner can be defined. However, for the array processing operator $\mathbf{A}$, where the array is a 30-element ULA separated by half-wavelength, and the steering grid is uniform from 0° to 180°, in increments of 1 degree, all the non-zero singular values fall in the interval from 2 to 3 as shown in Figure F.1 (a), top plot, leading to a very good condition number. The resulting curve for the Tikhonov-regularized inverse problem $\|\mathbf{y} - \mathbf{A}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_2^2$ does not have any resemblance to letter 'L', and choosing the regularization by L-curve methods is meaningless. There is no point on the curve which has a small data-fidelity residual as well as a small regularizing term at the same time.

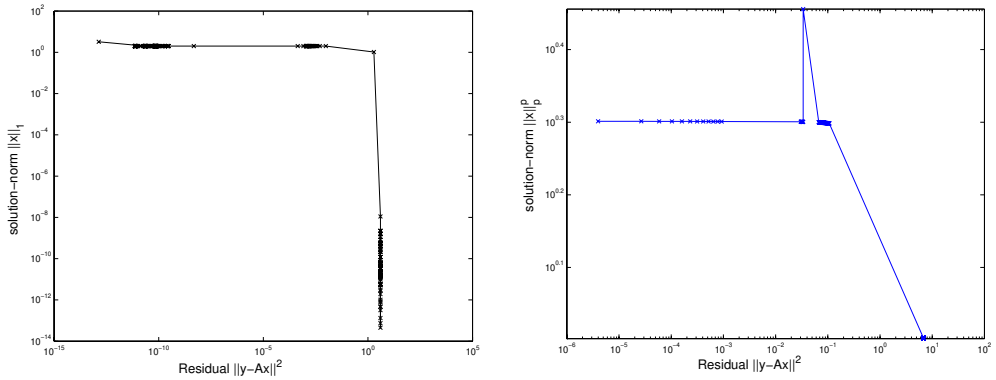



**Figure F.2.** The plot of $\|\mathbf{s}(\lambda)\|_p^p$ versus $\|\mathbf{y} - \mathbf{A}\mathbf{s}(\lambda)\|_2^2$, for a range of $\lambda$ (a) p = 1 . (b) p = 0.1 .

This experiment with Tikhonov regularization foretells that the L-curve method will not be applicable to either of the $\ell_1$ or $\ell_p$ methods (the operator is not changed, so the singular values distribution is the same, just the regularizer is different). As we see from Figure F.2, the shape of the curves for $\ell_p$ and $\ell_1$ is similar to the Tikhonov case ($\ell_2$), with a notable difference. Due to the nature of the $\ell_1$ and $\ell_p$ cost functions some of the transitions do not occur smoothly as in the $\ell_2$ case. This manifests itself in the presence of regions of little change (where the point on the curve are bunched up), and sudden jumps. In the $\ell_p$ case with $p = 0.1$, the cost function is not convex, so occasionally we get stuck at bad local optima (which manifests itself as an outlier on the curve as shown in Figure F.2 (b)). [1]

[1]Note that the iterative algorithm does not reach exact $\ell_p$ solutions, e.g. the indices of $\mathbf{s}$ off the support of $\mathbf{s}$, which have to be zero, are very small but non-zero. Due to the presence of 0.1 power, their effect is quite notable, so in order to get the corresponding L-curve, we need to do hard-thresholding of the solutions at some small value. (Otherwise the L-curve is dominated by rounding error, and has a jagged erratic shape).

## ■ F.2 Ordinary and Generalized Cross Validation

In the past few decades a very rich theory has been developed in the statistical learning and machine learning communities dealing with inference from data. A fundamental question is that of tradeoff between model complexity and fit to the data. By taking models with enough degrees of freedom any data set can be explained perfectly (provided there are no inconsistencies). However, this model only works well for the given data set, and when it is applied to a different one, it performs very poorly. This phenomenon is called overfitting. The ability of a model to explain previously unseen data is called generalization, which has a very close connection to the complexity of the model. The task is to create models which explain the data accurately and at the same time generalize well. These two goals are contradictory, hence we must seek a compromise, and a natural framework for seeking a compromise is by constructing a cost function which is a weighted combination of the two. Looking at machine learning in this way, we get problems which are similar to our interpretation of source localization.

A typical problem in statistical learning is fitting functions to sets of noisy observations. A very well-known instance of this problem is when we limit ourselves to polynomials. The complexity in this case refers to the degree of the polynomial. The problem has the form:

$$y_i = f(s_i) + n_i,\ i \in 1, .., T \tag{F.1}$$

The free variable is $s$, $y$ is the dependent variable, and $n$ is noise. The unknowns are the coefficients of the polynomial, and if we upper bound the degree by $D$, then we have a set of $D+1$ coefficients $\beta_d$. So, $f(s_i, \boldsymbol{\beta}) = \sum \beta_d s_i^d$. (In the process of regularization we would like to reduce the degree further, so many of the coefficients will become zero).

$$y_i = \sum_d \beta_d x_i^d + n_i,\ i \in 1, .., T \tag{F.2}$$

which immediately can be rewritten in the inverse problem form if we take $\mathbf{y} = [y_1, y_2, ...y_T]'$, $\boldsymbol{\beta} = [\beta_1, ..., \beta_D]'$, $[\mathbf{X}]_{i,d} = x_i^d$, and $\mathbf{n} = [n_1, n_2, ...n_T]'$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{n}, \tag{F.3}$$

For our purposes, it is better to look at each observation separately, $y_i = \mathbf{X}_i\beta + n_i, i \in \{1, .., T\}$.

A very old idea to get some handle on the generalization properties of the data is to split the data set into two parts, the training set and the test set. Then the problem is solved using the points in the training set only, and the test set is used to verify that our solution does not overfit to the training data. There are many methods following this philosophy, with the two notable ones being ordinary cross validation (OCV) [2], and generalized cross validation (GCV) [81]. In general when we remove some of the

[2] OCV is also called leave-one-out cross-validation.

points from the training data, the estimate becomes worse, so ordinary cross validation mitigates this by removing only one data point, and averaging over all T possible ways to remove it.

Suppose we are trying to find a suitable parameter for the cost function $J(\boldsymbol{\beta}) = J_1(\boldsymbol{\beta}) + \lambda J_2(\boldsymbol{\beta})$. In the context of polynomial fitting we can take $J_1(\boldsymbol{\beta}) = \sum_i (y_i - f(x_i;\boldsymbol{\beta}))^2$, and $J_2(\boldsymbol{\beta})$ can take various forms, for example the integral of the polynomial over some range, which is linear in the coefficients $\beta_i$. Let us consider the data set with point $k$ removed, i.e. $i \in \{1,..,T\} \setminus \{k\}$. Denote the corresponding cost function $J_k(\boldsymbol{\beta}, \lambda)$, and the optimal solution by $\boldsymbol{\beta}_\lambda^k$. Then the ordinary cross validation function is

$$V_o(\lambda) = \frac{1}{T}\sum_{k=1}^{T}\left(y_k - f(x_k;\boldsymbol{\beta}_\lambda^k)\right)^2 \tag{F.4}$$

If we define $\boldsymbol{\beta}_\lambda$ as the solution to the problem for the full data set (with no points removed), and $a_k = \frac{f(x_k;\boldsymbol{\beta}_\lambda) - f(x_k;\boldsymbol{\beta}_\lambda^k)}{y_k - f(x_k;\boldsymbol{\beta}_\lambda^k)}$, then we can express the ordinary cross validation function as [81]:

$$V_o(\lambda) = \frac{1}{T}\sum_{k=1}^{T}\frac{(y_k - f(x_k;\boldsymbol{\beta}_\lambda))^2}{(1-a_k(\lambda))^2} \tag{F.5}$$

Generalized Cross Validation (GCV) is obtained by replacing each $a_k(\lambda)$ with its average $\mu(\lambda) = \frac{1}{T}\sum_k a_k(\lambda)$, and

$$V_g(\lambda) = \frac{1}{T}\sum_{k=1}^{T}\frac{(y_k - f(x_k;\boldsymbol{\beta}_\lambda))^2}{(1-\mu(\lambda))^2} \tag{F.6}$$

The equation in (F.3) has the same form as our one time sample source localization problem (5.2), and many of our multiple time sample versions where we combine the time samples prior to solving inverse problems[3]. Let us take $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}$ as the general form of the problem that we are solving. Denote the data with entry $k$ removed by $\mathbf{y}^k$, and matrix $\mathbf{A}$ with row $k$ removed by $\mathbf{A}^k$. The cost function for the reduced problem is $J_k(\lambda, \mathbf{s}) = \|\mathbf{y}^k - \mathbf{A}^k\mathbf{s}\|_2^2 + \lambda J_2(\mathbf{s})$. Denote the optimum value of the cost function by $\mathbf{s}_\lambda^k$. Then the ordinary cross validation function for our problem is:

$$V_o(\lambda) = \frac{1}{M}\sum_{k=1}^{M}\left(y_k - [\mathbf{A}]_k\mathbf{s}_\lambda^k)\right)^2 \tag{F.7}$$

Here $y_k$ is the $k$-th entry of the vector $\mathbf{y}$ (the one that was removed), and $[\mathbf{A}]_k$ is the $k$-th row of matrix $\mathbf{A}$, (also the one that was removed to get the solution $\mathbf{s}_\lambda^k$ to

[3]When multiple time-samples (or frequency snapshots, or singular-value subproblems) are present, exactly the same approach can be used, but the data for the $k$-th sensor is removed for all time samples simultaneously.

the reduced problem). In order to find an appropriate $\lambda$ automatically we evaluate the OCV function over a grid of $\lambda$'s and find the minimum value (optimization over $\lambda$ may be very hard to do since $V_o(\lambda)$ may be a non-convex function with multiple local minima). The corresponding GCV value for $\lambda$ is obtained similarly.

The cross-validation approach is very demanding computationally, since we need to solve $M$ inverse problem to evaluate $\mathbf{V}_o(\lambda)$ at each point on the grid. When the statistics of the noise are not known, this remains as the only feasible automatic method of regularization parameter selection applicable to every scenario. It has been observed [82] that OCV tends to produce models which are too complex, which in our case means allowing spurious peaks due to noise in $\mathbf{s}$. There are related cross-validation approaches such as n-fold cross validation where the observations are split into larger groups, and instead of removing one observation we remove the whole group. However, for our application, if the number of sensors is small it may not be possible to remove observations at several sensors since it would notably reduce the number of sources that can be resolved. Partially due to its computational complexity, we do not have extensive experience of using cross-validation for our problem, so its viability remains a topic for further work.

## ■ F.3 Universal and min-max rules

The last set of ideas that we will discuss for automatic selection of the regularization parameters come from the field of function approximation. The universal [27, 83] and min-max [84] parameter selection rules were originally developed for denoising applications with minimum spanning orthogonal bases (not overcomplete). The problem is finding $\mathbf{x}$ (or $\Phi\mathbf{x}$) from

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{n}, \text{ where } \Phi'\Phi = \mathbf{I} \tag{F.8}$$

This has the same form as our problem, but with an important restriction of the basis being orthogonal. Extensions are possible for the invertible case, and Chen [29, 85] claims that the general overcomplete bases can be also tackled using these rules[4]. The rules have been developed originally for thresholding estimators, but an exact link exists between soft thresholding and $\ell_1$ penalization, as shown in [85]. We describe the connection further down.

For simplicity, we start with the easiest possible basis, the standard basis (transparent in the equations). A signal $x(m)$ is corrupted by additive white Gaussian noise $n(m) \sim N(0, \sigma^2)$ to give observations $y(m)$, i.e. $y(m) = x(m)+n(m)$, $m \in \{0, .., M-1\}$, or using vector notation $\mathbf{y} = \mathbf{x} + \mathbf{n}$. We are interested in getting good estimates $\hat{\mathbf{x}}$ in terms of the mean-squared error (MSE), $E[(\hat{\mathbf{x}} - \mathbf{x})^2]$. The signal is sparse but values are unknown, and the noise is uncorrelated. Thus, it is natural to limit the attention to estimators of the form:

$$\hat{x}(m) = \theta(m)y(m) \tag{F.9}$$

[4]However, we have not been able to find any references to such work.

since different indices of $y(m)$ have no interaction, and there is no use in considering joint statistics of pairs or higher numbers of indices. Next we discuss why it is possible to limit the set of estimators even further, and only look at soft and hard thresholding [27]. The choice of the threshold (which is directly related to the choice of regularization parameter) is motivated during the discussion.

The mean-squared error is minimized when $\theta(m) = \frac{x(m)^2}{x(m)^2+\sigma^2}$, but, this estimator is not valid since it depends on $x(m)$, the quantity that we wish to find in the first place. However, it is useful in providing an upper bound on the estimation error. If we restrict the estimator to be a hard threshold (i.e. $\theta(m)$ is 0 or 1, depending on whether we wish to keep or discard $y(m)$), then the optimal threshold is achieved at $\sigma^2$, and the MSE is $\sum_{m=0}^{M-1} \min(x(m)^2, \sigma^2)$. This estimator is likewise unrealizable for the same reasons. The optimal error of the threshold estimator, $\epsilon_t$, is of the same order as that for the general estimator (F.9), $\epsilon_g$, in fact they satisfy $\epsilon_t/2 \leq \epsilon_g \leq \epsilon_t$.

A valid estimator which comes to mind is a threshold estimator where instead of comparing the threshold to $x(m)$, we compare it to $y(m)$. (This estimator is related to the penalized $\ell_1$-norm estimator, as we shall soon describe). For regions where $x(m) \ll \sigma$, or $\sigma \ll x(m)$, the two estimators will behave similarly. The difference is for those regions where $x(m)$ are of the same order of magnitude as $\sigma$. Donoho and Johnstone [83] showed that this valid estimator has MSE within $1+\ln(M)$ of the optimal unrealizable threshold estimator, when the threshold is selected as $T = \sigma\sqrt{2\ln(M)}$. However, if the unknown signal $\mathbf{x}$ is sparse, and has its non-zero values well above the noise floor, then the two errors are much closer together. The reason for the threshold selection is the following: suppose that the signal is sparse (has a few large coefficients and others as zeros). We would like to select the threshold high enough to eliminate all the noise samples with high probability, but no larger, so that we do not also remove the components of the signal. It can be shown that when $T = \sigma\sqrt{2\ln\ M}$,

$$\lim_{M\to\infty} Pr\left(T - \frac{\sigma\ \ln\ \ln M}{\ln\ M} \leq \max_{0\leq m\leq M-1} |n(m)| \leq T\right) = 1. \tag{F.10}$$

This means that asymptotically, as $M \to \infty$, the maximum of $M$ i.i.d. Gaussian samples will fall within the interval $[T - \frac{\sigma\ \ln\ \ln M}{\ln\ M}, T]$. The choice $T = \sigma\sqrt{2\ln\ M}$ is called the universal rule.

The min-max rule for the selection of the threshold [84] is closely related to the universal rule. Asymptotically, as $M \to \infty$, the two give the same values of the thresholds. The min-max rule chooses the threshold which minimizes the worst ratio of the error of the valid threshold estimator to that of the ideal threshold estimator:

$$T^* = \inf_T\ \sup_x\ \frac{E[(y1_{y\geq T} - x)^2]}{\sigma^2/M + \min(\sigma^2, x^2)}, \tag{F.11}$$

where $1_{y\geq T}$ is equal to 1 if $y \geq T$, and 0 otherwise. Asymptotically, the optimal ratio approaches $2\ln M$. However, for small values of $M$, the threshold selected by the

min-max rule is considerably smaller then the one which comes out of the universal rule.

If instead of looking at the standard basis, we look at an orthonormal basis $\Phi$, $\Phi'\Phi = \mathbf{I}$, then the problem becomes:

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{n}, \text{ or } \Phi'\mathbf{y} = \mathbf{x} + \Phi'\mathbf{n} \tag{F.12}$$

By renaming $\tilde{\mathbf{y}} = \Phi'\mathbf{y}$, and $\tilde{\mathbf{n}} = \Phi'\mathbf{n}$, we get a problem of the previous form. The second-order statistics of $\tilde{\mathbf{n}}$ are the same as those of $\mathbf{n}$, since $\mathbf{n}$ is zero-mean, and $\Phi$ is orthonormal.

So far we have looked only at the hard-threshold estimator, $\hat{x}(m) = \theta(m)y(m)$, where $\theta(m) = 1_{y(m)\geq T}$, but it is shown in [83] that the same result applies to the soft threshold estimator, $\hat{x}(m) = \eta(y(m))$, where

$$\eta(y(m)) = \begin{cases} 0, & |y(m)| \leq T \\ y(m) - T, & y(m) \geq T, \\ y(m) + T, & y(m) \leq -T \end{cases} \tag{F.13}$$

The soft threshold estimator has a very close connection with the $\ell_1$-norm penalized estimator, $\hat{\mathbf{x}} = argmin_{\mathbf{x}}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1$. In fact, for 1-dimensional problems, the solution of $1/2\ argmin_x\ (y - x)^2 + \lambda|x|$ yields exactly the soft-thresholding estimator with $T = \lambda$, [85]. For orthogonal bases in (F.8) this also holds, since $\|\tilde{\mathbf{y}} - \mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1 = \sum_m (\tilde{y}(m) - x(m))^2 + \lambda|x(m)|$, which can be solved one coordinate at a time.

It is possible to extend the rules to the scenario where the basis $\mathbf{A}$ is invertible, but not necessarily orthogonal:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \text{ or } \mathbf{A}^{-1}\mathbf{y} = \mathbf{x} + \mathbf{A}^{-1}\mathbf{n} \tag{F.14}$$

Again, by renaming $\tilde{\mathbf{y}} = \mathbf{A}^{-1}\mathbf{y}$, and $\tilde{\mathbf{n}} = \mathbf{A}^{-1}\mathbf{n}$, we get a problem of the original form (with the standard basis). However, contrary to the orthogonal case, with basis $\Phi$, the statistics of $\tilde{\mathbf{n}}$ are no longer the same as those of $\mathbf{n}$. In fact, $\tilde{\mathbf{n}} \sim M(0, (A^{-1})(A^{-1})')$. By constraining the estimator to the form $\hat{x}(m) = \theta(m)y(m)$, we cannot easily address (benefit from) the knowledge of correlation of different indices of the noise. Donoho and Johnstone [83] show that the optimal threshold will depend on the variances of $\tilde{n}(m)$, i.e. we have an index-dependent threshold $T(m) = \tilde{\sigma}(m)\sqrt{2\ln M}$, where $\tilde{\sigma}(m) = Var(\tilde{n}(m))$. This contradicts the suggestions made in [85], that one should still use $\lambda = \sigma\sqrt{2\ln M}$. Now instead of $\lambda\|\mathbf{x}\|_1$, we have to consider a weighted $\ell_1$ norm, where the indices are weighted by the variances of $\tilde{\mathbf{n}}$, i.e. $\sum_m \tilde{\sigma}(n)|x(m)|$.

Our extended discussion of min-max and universal rules serves several purposes. First of all, it appears that neither of the two rules are directly applicable to our problem of regularization parameter selection. This happens due to the fact that for both orthogonal and invertible bases there exists a direct relationship between the noise in observations $\mathbf{y}$, and noise in coefficients. In the overcomplete case this relationship is lost, since many possible coefficients can account for the noise. It is conceivable that

an extension to the overcomplete case is possible, but it is unlikely to be of the same simple form as for orthogonal bases.

Additionally, the results are mainly useful for large $M$, since many of the derivations are based on asymptotic arguments as $M \to \infty$. For our source localization application, $M$ corresponds to the number of sensors and is typically a small number (such as 7 in some of our simulations). The fact that the number of time samples may grow large is not relevant, since in most cases we first transform the data with multiple time samples into a single problem.

Two interesting observations come out of the discussion. First, it may be possible to improve the performance of our technique by considering a weighted $\ell_1$ norm, e.g. for $\mathbf{a} > 0$, $\|\mathbf{x}\|_1^{(\mathbf{a})} = \sum_{i=1}^{M} a_i |x_i|$, where $a_i \geq 0$ for all $i$, instead of the usual $\ell_1$-norm: $\|\mathbf{x}\|_1 = \sum_{i=1}^{M} |x_i|$. At the moment the proper choice of $\mathbf{a}$ is not clear for the reasons of overcompleteness of the bases that we consider. Second observation is that we expect the proper choice of the regularization parameter to scale linearly with $\sigma$, the standard deviation of noise (which we assume to be Gaussian).

To summarize, although we do not use either the minmax or the universal rule for overcomplete bases, we do not rule out the possibility that an extension may be made which will work well for our source localization application. In fact, some research has been done in the selection of the regularization parameter for the case of an overcomplete basis composed of several orthogonal bases [27, 86]. The main difference from our approach is that in these applications the signal is typically represented not as a general linear combination of the elements of the overcomplete basis, but rather as a combination of elements of a particular orthogonal basis contained in the dictionary. This restriction makes the problem much more amenable to analysis.

# Bibliography


[1] H. V. Poor and G. W. Wornell, Eds., *Wireless Communications: Signal Processing Perspectives*, Prentice Hall, 1998.

[2] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1998.

[3] M. I. Skolnik, *Introduction to Radar Systems*, McGraw-Hill, 2000.

[4] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part III: Radar-Sonar Signal Processing and Gaussian Signals in Noise*, John Wiley and Sons, 1968.

[5] R. O. Nielsen, *Sonar Signal Processing*, Artech House, 1991.

[6] E. A. Robinson and S. Treitel, *Geophysical Signal Analysis*, Prentice Hall, 1980.

[7] R. O. Schmidt, *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation*, Ph.D. thesis, Stanford Univ., 1981.

[8] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[9] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Trans. Information Theory*, vol. 48, no. 9, pp. 2558–2567, 2002.

[10] A. Nemirovski A. Ben Tal, *Lectures on Modern Convex Optimization. Analysis, Algorithms and Engineering Applications*, SIAM, 2001.

[11] M. Çetin and W. C. Karl, "Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization," *IEEE Trans. Image Processing*, vol. 10, no. 4, pp. 623–631, Apr. 2001.

[12] B. D. Jeffs, "Sparse inverse solution methods for signal and image processing applications," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998, vol. 3, pp. 1885–1888.

[13] J. J. Fuchs, "Linear programming in spectral estimation. Application to array processing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, vol. 6, pp. 3161–3164.

[14] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[15] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing - Concepts and Techniques*, Prentice Hall, 1993.

[16] H. Krim and M. Viberg, "Two decades of array signal processing research. The parametric approach," *IEEE Signal Proc. Mag.*, vol. 13, no. 4, pp. 67–94, July 1996.

[17] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[18] A. J. Barabell, "Improving the resolution performance of eigenstructure based direction-finding algorithms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1983, pp. 336–339.

[19] J. Stoica and K. C. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Trans. Signal Processing*, vol. 38, no. 7, pp. 1132–1143, July 1990.

[20] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. ASSP*, vol. 33, no. 2, pp. 387–392, Feb. 1985.

[21] P. Stoica and A. Nehorai, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Trans. ASSP*, vol. 37, no. 5, pp. 720–741, May 1989.

[22] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Kluwer, 1996.

[23] A. Neumaier, "Solving ill-conditioned and singular linear systems: a tutorial on regularization," *SIAM Review*, vol. 40, no. 3, pp. 636–666, 1998.

[24] P. C. Hansen, "Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems," *Numer. Algorithms*, vol. 6, pp. 1–35, 1994.

[25] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Doklady Akademii Nauk SSSR*, vol. 151, pp. 501–504, 1963.

[26] W. C. Karl, "Regularization in image restoration and reconstruction," in *Handbook of Image and Video Processing*, A. Bovik, Ed. Academic Press, 2000.

[27] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1998.

[28] D. L. Donoho, I. M. Johnstone, J. C. Koch, , and A. S. Stern, "Maximum entropy and the nearly black object," *J. R. Statist. Soc. B*, vol. 54, no. 1, pp. 41–81, 1992.

[29] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[30] W. J. Fu, "Penalized regressions: the bridge versus the LASSO," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, Sept. 1998.

[31] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Processing*, vol. 4, no. 7, pp. 932–946, July 1995.

[32] A. J. Miller, *Subset Selection in Regression*, Chapman and Hall, 2002.

[33] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of Royal Statistical Society, Series B*, vol. 58, pp. 267–288, Nov. 1996.

[34] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, 1975.

[35] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[36] J. S. Sturm, "Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones," Tech. Rep., Tilburg University, Department of Econometrics, Netherlands, 2001, http://fewcal.kub.nl/ sturm.

[37] K. C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3 - a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, no. 12, pp. 545–581, 1999.

[38] R. M. Freund and S. Mizuno, "Interior point methods: Current status and future directions," *OPTIMA*, vol. 51, Oct. 1996.

[39] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Trans. Image Processing*, vol. 7, no. 6, pp. 813–824, Jun 1998.

[40] O. L. Mangasarian and D. R. Musicant, "Robust linear and support vector regression," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 9, pp. 950–955, 2000.

[41] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 3, pp. 367–383, Mar. 1992.

[42] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 3, pp. 55–67, 1970.

[43] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 298–310, Feb. 1997.

[44] A. H. Delaney and Y. Bresler, "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography," *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 204–221, Feb. 1998.

[45] M. D. Zoltowski, G. M. Kautz, and S. D. Silverstein, "Beamspace ROOT-MUSIC," *IEEE Trans. Signal Processing*, vol. 41, no. 1, pp. 344–364, Feb. 1993.

[46] Z. Tian and H. L. Van Trees, "Beamspace MODE," in *Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, 2001, vol. 2, pp. 926–930.

[47] S. Sivanand, J. F. Yang, and M. Kaveh, "Time-domain coherent signal-subspace wideband direction-of-arrival estimation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1989, vol. 4, pp. 2772–2775.

[48] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1996.

[49] V. A. Morozov, "On the solution of functional equations by the method of regularization," *Soviet Math. Dokl.*, vol. 7, pp. 414–417, 1966.

[50] J. T. Lo and S. L. Marple, "Observability conditions for multiple signal direction finding and array sensor localization," *IEEE Trans. Signal Processing*, vol. 40, no. 11, pp. 2641–2650, Nov. 1992.

[51] H. S. M. Coxeter, *Regular Polytopes*, MacMillan, 1963.

[52] P. Shor, "Optimality of the regular simplex for $M(\mathbf{A})$, where $\mathbf{A} \in \mathbb{R}^{K\times(K+1)}$," Private Communication, 2003.

[53] R. Blume Kohout, "Optimality of the regular simplex for $M(\mathbf{A})$, where $\mathbf{A} \in \mathbb{R}^{K\times(K+1)}$," Private Communication, 2003.

[54] W. Sun, "Optimality of the regular simplex for $M(\mathbf{A})$, where $\mathbf{A} \in \mathbb{R}^{K\times(K+1)}$," Private Communication, 2003.

[55] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in grassmannian spaces," *Experimental Mathematics*, vol. 5, pp. 139–159, 1996.

[56] A. Feuer and A. Nemirovski, "On sparse representation in pairs of bases," *IEEE Trans. Information Theory*, vol. 49, no. 6, pp. 1579–1581, 2003.

[57] T. Ericson and V. Zinoviev, *Codes on Euclidean Spheres*, Elsevier Publishers, 2001.

[58] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[59] A. J. Weiss and B. Friedlander, "Effects of modeling errors on the resolution threshold of the MUSIC algorithm," *IEEE Trans. Signal Processing*, vol. 42, no. 6, pp. 1519–1526, June 1994.

[60] A. Swindlehurst and T. Kailath, "A performance analysis of subspace-based methods in the presence of model errors - part I: The MUSIC algorithm," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1758–1774, July 1992.

[61] C. Vaidyanathan and K. M. Buckley, "Comparative studies of MUSIC and MVDR location estimators for model perturbations," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995, vol. 3, pp. 9–12.

[62] S. A. Vorobyov, A. B. Gershman, and Z. Q. Luo, "Robust adaptive beamforming using worst-case performance optimization via second-order cone programming," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 3, pp. 2901–2904.

[63] J. Li, P. Stoica, and Z. Wang, "On robust capon beamforming and diagonal loading," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1702–1715, July 2003.

[64] M. Pesavento, A. B. Gershman, and K. M. Wong, "Direction finding in partly calibrated sensor arrays composed of multiple subarrays," *IEEE Trans. Signal Processing*, vol. 50, no. 9, pp. 2103–2115, Sept. 2002.

[65] Y. Rockah and P. M. Schultheiss, "Array shape calibration using sources in unknown locations: Part I, far-field sources," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. 35, no. 3, pp. 286–299, Mar. 1987.

[66] A. J. Weiss and B. F. Friedlander, "Array shape calibration using sources in unknown locations - a maximum likelihood approach," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. 37, no. 12, pp. 1958–1966, Dec. 1989.

[67] A. Manikas and C. Proukakis, "Modeling and estimation of ambiguities in linear arrays," *IEEE Trans. Signal Processing*, vol. 46, no. 8, pp. 2166–2179, Aug. 1998.

[68] A. Manikas, A. Sleiman, and I. Dacos, "Manifold studies of nonlinear antenna array geometries," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 497–506, Mar. 2001.

[69] A. J. Weiss and B .F. Friedlander, "Array shape calibration using eigenstructure methods," *Signal Processing*, vol. 22, no. 3, pp. 251–258, 1991.

[70] R. Moses, R. Patterson, D. Krishnamurthy, N. Srour, and T. Pham, "Self-calibration of unattended ground sensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 3, pp. 2941–2944.

[71] M. X. Goemans, "Semidefinite programming in combinatorial optimization," *Mathematical Programming*, vol. 79, pp. 143–161, 1997.

[72] A. S. Willsky, G. W. Wornell, and J. H. Shapiro, *Stochastic Processes, Detection and Estimation. 6.432 Course notes*, MIT, 2001.

[73] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, John Wiley and Sons, 1968.

[74] A. Azzalini, *Statistical Inference based on the likelihood*, Chapman and Hall, 1996.

[75] E. W. Barankin, "Locally best unbiased estimates," *Ann. Math. Stat.*, vol. 20, pp. 477–501, 1949.

[76] A. Bhattacharyya, "On some analogues of the amount of information and their use in statistical estimation," *Sankhya*, vol. 8, pp. 1–14, 1946.

[77] E. Weinstein and A. J. Weiss, "A general class of lower bounds in parameter estimation," *IEEE Trans. Information Theory*, vol. 34, no. 2, pp. 338–342, 1988.

[78] L. T. McWhorter and L. L. Scharf, "Properties of quadratic covariance bounds," in *Twenty-Seventh Asilomar Conf. on Signals, Systems and Computers*, Nov. 1993, vol. 2, pp. 1176–1180.

[79] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior methods for nonlinear optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, Oct. 2002.

[80] M. M. Makela and P. Neittaanmaki, *Nonsmooth Optimization. Analysis and Algorithms with Applications to Optimal Control*, World Scientific, 1992.

[81] G. Wahba, *Spline Models for Observational Data*, SIAM, 1990.

[82] J. Shao, "Linear model selection by cross-validation," *Journal of the American Statistical Association. Theory and methods*, vol. 88, no. 422, pp. 486–494, June 1993.

[83] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, Aug. 1994.

[84] I. M. Johnstone, "On mininax estimation of a sparse normal mean vector," *Annals of Statistics*, vol. 22, no. 1, pp. 271–289, Mar. 1994.

[85] S. S. Chen, *Basis Pursuit*, Ph.D. thesis, Stanford Univ., Department of Statistics, 1995.

[86] H. Krim, D. Tucker, S. Mallat, and D. Donoho, "On denoising and best signal representation," *IEEE Trans. Information Theory*, vol. 45, no. 7, pp. 2225–2238, Nov. 1999.