Sparsity-Driven Statistical Inference for Inverse Problems

Ulugbek S. Kamilov

Thèse N° 6545 (mars 2015)

Thèse présentée à la faculté des sciences et techniques de l'ingénieur
pour l'obtention du grade de docteur ès sciences
et acceptée sur proposition du jury

Prof. Jean-Philippe Thiran, *président*
Prof. Michael Unser, *directeur de thèse*
Prof. Mario Figueiredo, *rapporteur*
Prof. Rémi Gribonval, *rapporteur*
Prof. Martin Vetterli, *rapporteur*

École polytechnique fédérale de Lausanne—2015

Cover design by Annette Unser
Printing and binding by Repro-EPFL
Typeset with LaTeX
Copyright © 2015 by Ulugbek S. Kamilov
Available at **http://bigwww.epfl.ch/**

Abstract

This thesis addresses statistical inference for the resolution of inverse problems. Our work is motivated by the recent trend whereby classical linear methods are being replaced by nonlinear alternatives that rely on the sparsity of naturally occurring signals. We adopt a statistical perspective and model the signal as a realization of a stochastic process that exhibits sparsity as its central property. Our general strategy for solving inverse problems then lies in the development of novel iterative solutions for performing the statistical estimation.

The thesis is organized in five main parts. In the first part, we provide a general overview of statistical inference in the context of inverse problems. We discuss wavelet–based and gradient–based algorithms for linear and nonlinear forward models. In the second part, we present an in-depth discussion of cycle spinning, which is a technique used to improve the quality of signals recovered with wavelet–based methods. Our main contribution here is its proof of convergence; we also introduce a novel consistent cycle-spinning algorithm for denoising statistical signals. In the third part, we introduce a stochastic signal model based on Lévy processes and investigate popular gradient–based algorithms such as those that deploy total-variation regularization. We develop a novel algorithm based on belief propagation for computing the minimum mean-square error estimator and use it to benchmark several popular methods that recover signals with sparse derivatives. In the fourth part, we propose and analyze a novel adaptive generalized approximate message passing (adaptive GAMP) algorithm that reconstructs signals with independent wavelet-coefficients from generalized linear measurements. Our algorithm is an extension of the standard GAMP algorithm and allows for the joint learning of unknown statistical parameters. We prove that, when the measurement matrix is independent and identically distributed Gaussian, our algorithm is asymptotically consistent. This means that it performs as well as the oracle algorithm, which knows the parameters exactly. In the fifth and final part, we apply our methodology to an inverse problem in optical tomographic microscopy. In particular, we propose a novel nonlinear forward model and a corresponding algorithm for the quantitative estimation of the refractive index distribution of an object.

Keywords: Approximate message passing, belief propagation, compressive sensing, cycle spinning, inverse problems, iterative shrinkage, phase microscopy, sparsity, statistical inference, tomographic microscopy, total variation regularization, wavelets.

Résumé

Cette thèse s'intéresse à l'inférence statistique pour la résolution de problèmes inverses. Notre travail est motivé par les récentes avancées dans ce domaine où les méthodes linéaires classiques sont de plus en plus souvent remplacées par des alternatives non linéaires qui exploitent la parcimonie des signaux naturels. Nous adoptons une approche statistique et modélisons le signal comme réalisation d'un processus stochastique possédant des propriétés essentielles de parcimonie. Notre stratégie générale pour la résolution des problèmes inverses repose alors sur le développement de nouveaux algorithmes itératifs d'estimation statistique.

Cette thèse s'organise en cinq parties principales. Dans la première partie, nous donnons une perspective générale de l'inférence statistique dans le contexte de la résolution de problèmes inverses. Nous présentons des algorithmes basés sur les ondelettes et le gradient dans le cadre de modèles linéaires et non linéaires. Dans la deuxième partie, nous discutons en détail le concept de *cycle spinning*, une technique utilisée pour améliorer la qualité de signaux reconstruits par des méthodes d'ondelettes. Ici, notre contribution principale est la démonstration de sa convergence ; nous introduisons de plus le nouveau concept de *consistent cycle spinning* pour le débruitage de signaux statistiques. Dans la troisième partie, nous introduisons un modèle de signal stochastique basé sur les processus de Lévy et analysons des algorithmes standards basés sur le gradient, comme les méthodes de variation totale. Nous développons un nouvel algorithme basé sur la méthode de *belief propagation* pour le calcul d'estimateur d'erreur quadratique moyenne minimale, et l'utilisons pour comparer plusieurs méthodes classiques de reconstruction de signaux à gradient parcimonieux. Dans la quatrième partie, nous proposons et analysons un nouvel algorithme d'*adaptive GAMP* pour la reconstruction de signaux à composantes indépendantes à partir de mesures linéaires généralisées. Notre algorithme est une extension de l'algorithme *GAMP* et permet l'estimation de paramètres statistiques inconnus. Nous démontrons que lorsque la matrice de mesure est Gaussienne, indépendante et identiquement distribuée, notre algorithme est asymptotiquement consistent. Cela signifie qu'il est aussi performant qu'un algorithme avec oracle, qui connaîtrait les valeurs exactes des paramètres. Dans la cinquième et dernière partie, nous appliquons nos méthodes à un problème inverse de microscopie tomographique par diffraction. Nous proposons en particulier un nouveau modèle non linéaire et un algorithme itératif associé pour la reconstruction quantitative de la distribution d'indices de réfraction.

Mots clés : Microscopie, ondelettes, problèmes inverses, parcimonie, régularisation, reconstruction.

We are all apprentices in a craft where no one ever becomes a master.

ERNEST HEMINGWAY

To my grandmother—
Mukambar Niyazova (1939–2014).

Acknowledgement

This thesis would not have been completed without the help and support of many people. I take this opportunity to express my gratitude to all of them.

First and foremost, I thank my advisor Prof. Michael Unser. It was a great pleasure completing this work under his guidance. He is a fantastic group leader who created a rare environment where one can truly enjoy being a scientist. Besides his generous personality, I highly valued his enthusiasm for research and his drive for knowledge.

I express my sincere thanks to the president of the thesis jury, Prof. Jean-Philippe Thiran, and the official referees, Prof. Mario Figueiredo, Prof. Rémi Gribonval, and Prof. Martin Vetterli, for reviewing the thesis. I additionally take this opportunity to express my deep gratitude to Prof. Martin Vetterli for personifying top science and magnanimity.

I thank the fellow present and past lab members of the Biomedical Imaging Group (BIG), Prof. Arash Amini, Anaïs Badoual, Dr. Jean-Charles Baritaux, Ayush Bhandari, Emrah Bostan, Dr. Nicolas Chenouard, Dr. Ning Chu, Dr. Ricard Delgado Gonzalo, Prof. Adrien Depeursinge, Julien Fageot, Dr. Denis Fortun, Dr. Matthieu Guerquin-Kern, Dr. Hagai Kirshner, Dr. Florian Luisier, Junhong Min, Masih Nilchian, Pedram Pad, Zsuzsanna Püspöki, Dr. Sathish Ramani, Dr. Daniel Sage, Daniel Schmitter, Prof. Chandra Sekhar Seelamantula, Dr. Tomáš Škovránek, Dr. Martin Storath, Dr. Pouya D. Tafti, Raquel Terrés Cristofani, Dr. Philippe Thévenaz, Virginie Uhlmann, Dr. Cédric Vonesch, and Dr. John Paul Ward. And more particularly, I thank my present and past office mates Dr. Aurélien Bourquard, Emmanuel Froustey, Dr. Stamatis Lefkimmiatis, and Dr. Ha Nguyen. I would also like to thank some of the current and past members of the Medical Image Processing Lab, Dr. Ivana Balic-Jovanovic, Zafer Doğan, Dr. Fikret Işik

Karahanoğlu, Jeffrey Kasten, Dr. Nora Leonardi, and Prof. Dimitri Van De Ville. I really appreciate all the good moments that we shared together. I am grateful to Manuelle Mary for helping me out with various administrative matters inside and outside EPFL.

I must say it was a real pleasure interacting with my numerous colleagues at EPFL. A very special thanks to Aurélien for having supervised me during my semester project in the lab, Emrah for our numerous collaborations on cycle spinning, Arash for showing me what problem solving really means, John Paul for our extensive discussions on the state of the world, Ricard for consistently challenging my fitness levels, Nora for teaching me how to live, Zsuzsanna for the chair that seemlessly integrated into my flat, Francisco for the constant inspiration, Işik for making my start in Boston as smooth as possible, and Emmanuel for introducing me to the world of French comedy.

I thank all my collaborators, Prof. Alyson Fletcher, Dr. Alexandre Goy, Prof. Vivek Goyal, Dr. Ioannis Papadopoulos, Prof. Demetri Psaltis, Prof. Sundeep Rangan, Prof. Philip Schniter, Morteza Shoreh. I also take this opportunity to express gratitude to all the students whose projects I have supervised, Bugra Tekin, Ipek Baz, Abbas Kazerouni, Julien Schwab, Mamoun Benkirane, and Sander Kromwijk. I wish them all the best with their careers. I reserve a special thank you for the deputy head of IC section Sylviane Dal Mas, whose advise and friendship made my study at EPFL a marvelous experience.

I salute all my friends from all around the world. A special thanks to *the calm achiever* Selma Chouaki, *the ultimate scientist* David Kammer, *the contemporary gentleman* Jean-Eloi Lombard, *the cool dude* Javier Meier, *the unrivaled connoisseur* Paulo Refinetti, *the great adventurer* Greg Trottet, and *the absolute kindness* Angélique Umuhire. I would also like to give a special attention to the ones that have been giving me their support and friendship since childhood Sanjar Abidov, Dilmurod Akbarov, Andrey Tsoy, and Abdulla Usmanov.

Finally, I wish to thank my mother Guli Kittel, my sister Sevara Kamilova, and my wife Parizod Kamilova, and also the rest of my large and happy family. I am certain that this thesis would not have been possible without their patience and support.

Contents

Abstract i

Résumé iii

Acknowledgement ix

1 Introduction 1

1.1 Main Contributions . . . 2

1.2 Organization of the Thesis . . . 3

2 A Practical Guide to Inverse Problems 5

2.1 Introduction . . . 5

2.2 Forward Model . . . 6

2.2.1 Continuous-Domain Formulation . . . 6

2.2.2 Discrete Representation of the Signal . . . 9

2.2.3 Discrete Measurement Model . . . 10

2.2.4 Wavelet Discretization . . . 11

2.3 Statistical Inference . . . 12

2.4 Least Squares 14
  2.4.1 Nonlinear Model 15
  2.4.2 Linear Model 16
  2.4.3 Signal Denoising 17
  2.4.4 Accelerated Gradient descent 18
2.5 Wavelet-Domain Reconstruction 20
  2.5.1 Signal Denoising 21
  2.5.2 Linear Model 25
  2.5.3 Nonlinear Model 28
2.6 Gradient–Based Regularization 29
  2.6.1 Linear Model 30
2.7 Experimental Evaluation 33
2.8 Summary 35

3 Cycle Spinning: Boosting Wavelet-Domain Reconstruction 37

3.1 Overview 37
3.2 Introduction 38
3.3 Cycle Spinning 38
3.4 Analysis of Cycle Spinning 40
3.5 Consistent Cycle Spinning 41
3.6 Numerical Evaluation 43
  3.6.1 Convergence of Cycle Spinning 44
  3.6.2 Statistical Estimation with CCS 46
3.7 Summary 50
3.8 Appendix 51
  3.8.1 Useful Facts from Convex Analysis 51

3.8.2 General Model 52
3.8.3 Main Technical Lemmas 52
3.8.4 Proof of Theorem 3.1 55

4 Optimal Estimators for Denoising Lévy Processes 57

4.1 Overview 57
4.2 Introduction 58
4.3 Signal and Measurement Model 59
4.3.1 Lévy Processes 60
4.3.2 Examples 62
4.3.3 Innovation Modeling 64
4.3.4 Measurement Model 65
4.4 Bayesian Formulation 67
4.4.1 MAP Estimation 67
4.5 Message-Passing Estimation 69
4.5.1 Exact Formulation 69
4.5.2 Fourier-Domain Alternative 71
4.5.3 Implementation 72
4.6 Experimental Evaluation 72
4.6.1 AWGN Denoising 73
4.6.2 Signal Interpolation 77
4.6.3 Estimation from Quantized Samples 77
4.7 Summary 78

5 Efficient Approximations to MMSE with Adaptive GAMP 83

5.1 Overview 83
5.2 Introduction 84

5.2.1 Related Literature 86
5.3 Review of GAMP 87
5.3.1 Sum-Product GAMP 88
5.3.2 General GAMP 90
5.3.3 State Evolution Analysis 92
5.3.4 State Evolution Analysis for Sum-Product GAMP 96
5.4 Adaptive GAMP 96
5.4.1 ML Parameter Estimation 97
5.4.2 Relation to EM-GAMP 98
5.5 Convergence and Asymptotic Consistency with Gaussian Transforms . . 100
5.5.1 General State Evolution Analysis 100
5.5.2 Asymptotic Consistency with ML Adaptation 101
5.5.3 Computational Issues 104
5.6 Identifiability and Parameter Set Selection 105
5.6.1 Gaussian Mixtures 106
5.6.2 AWGN output 107
5.6.3 Initialization 107
5.7 Experimental Evaluation 108
5.7.1 Estimation of a Bernoulli-Gaussian input 108
5.7.2 Estimation of a Nonlinear Output Classification Function 110
5.8 Summary 112
5.9 Appendix 113
5.9.1 Convergence of Empirical Distributions 113
5.9.2 Proof of Theorem 5.2 114
5.9.3 Proof of Theorem 5.3 121
5.9.4 Proof of Lemma 5.1 123

6 Numerical Evaluation 127
6.1 Introduction 127
6.2 Compressive Sensing 129
6.3 Image Deconvolution 132
6.4 Discussion 134

7 A Novel Nonlinear Framework for Optical Phase Tomography 143
7.1 Introduction 143
7.2 Inhomogeneous Wave Equation 144
7.3 Fourier Beam Propagation 146
7.3.1 Derivation 146
7.3.2 Implementation 147
7.4 Iterative Reconstruction 148
7.4.1 Derivation of the Gradient 150
7.4.2 Recursive Computation of Gradient 153
7.4.3 Iterative Estimation Algorithm 153
7.5 Numerical Evaluation 153
7.6 Summary 155

8 Conclusions 159
8.1 Summary of Results 159
8.2 Outlook 161

Bibliography 163

Curriculum Vitæ 175

Chapter 1

Introduction

The term *inverse problem* refers to a general framework used to convert observed measurements into information about a physical object. For example, given tomographic measurements of a cell, we might wish to learn about its internal composition and structures. Having the capability to solve inverse problems is useful because it tells us something about physical quantities that we are unable to observe directly. Accordingly, inverse problems are some of the most important and well-studied mathematical problems in science and engineering. They have found many applications in different areas of science, including medical imaging, geophysics, astronomy, etc.

Two main reasons that make inverse problems of practical relevance difficult are as follows:

- they are sometimes ill-posed, which means that many different solutions may be consistent with the measured data;
- they are high-dimensional in the sense that the measurements and the corresponding solution might contain millions or even billions of data entries.

These challenges force algorithmic solutions to strike a specific balance between the quality of the reconstructed signal and the computational complexity for obtaining it. In this thesis, we attempt to refine the theory and develop new algorithms for addressing these challenges in a principled way. In particular, the ill-posed nature of the problem is offset by introducing a statistical framework where observed measurements are complemented with prior information on the statistics of the object. On the other hand, the high-dimensionality of the data is addressed by developing iterative algorithms that are restricted to performing the most basic operations with minimal memory and computational load. Nonetheless, as we shall see, our simple algorithms are still capable of producing satisfactory results for numerous practical applications in biomedical imaging.

1.1 Main Contributions

This thesis brings four main contributions to the field of inverse problems.

1. We provide a theoretical justification for the popular technique called cycle spinning in the context of general linear inverse problems. Cycle spinning has been extensively used for improving the visual quality of images reconstructed with wavelet-domain methods. We also refine traditional cycle spinning by introducing the concept of consistent cycle spinning that can be used to perform wavelet-domain statistical estimation. In particular, we empirically show that consistent cycle spinning achieves the minimum mean-squared error (MMSE) solution for denoising stochastic signals with sparse derivatives.

2. We introduce a continuous-domain stochastic framework for modeling signals with sparse derivatives. Our framework is based on Lévy processes and provides us with a large collection of structured statistical signals for benchmarking various standard algorithms used for solving inverse problems. We also develop a novel MMSE estimation algorithm for our signal model. This algorithm—based on a message-passing methodology—allows us to evaluate the optimality of other commonly used algorithms. For example, it is well known that the popular total variation (TV) method corresponds to performing maximum-a-posteriori (MAP) probability estimation of signals with Laplace distributed gradients. One of our findings is that TV is in general a poor estimator for such Laplace processes; however, it reaches the performance of MMSE estimation for piecewise-constant compound Poisson processes.

3. Generalized approximate message passing (GAMP) is an iterative algorithm for performing statistical inference under generalized linear models. The algorithm can be derived by simplifying the equations of a more general belief propagation (BP) algorithm, which is intractable in the setting of general inverse problems. Our contribution in the context of GAMP is twofold: (a) we extend the traditional GAMP with a novel algorithm called adaptive GAMP that can learn unknown statistical parameters present in the inverse problem during the reconstruction; (b) we prove that adaptive GAMP is asymptotically consistent for certain measurement models when learning is performed via the maximum-likelihood (ML) estimator. This means that adaptive GAMP can perform as well as the oracle algorithm that knows the parameters exactly.

4. We introduce a novel inverse problem formulation suitable for optical tomographic microscopy. The latter is an advanced digital imaging technique that combines the recording of multiple holograms with the use of inversion procedures to retrieve the quantitative information on the object. Here, our contribution is a nonlinear forward model that simulates the physics of the wave propagating through the sample. Compared to existing linear alternatives, our forward model provides an improved description of the measured data due to its ability to properly emulate the diffraction and propagation effects of the wave field. We finally develop a novel iterative algorithm, which uses the structure of our nonlinear forward operator, for quantitatively estimating the object.

1.2 Organization of the Thesis

This thesis is organized as follows: In Chapter 2, we expose the principles behind the formulation and resolution of inverse problems. We also present classical and state-of-the-art reconstruction techniques within a general statistical framework. In Chapter 3, we discuss an algorithmic strategy to perform competitive reconstruction using wavelet-regularization. The key concept that yields the improvements is cycle spinning, which we shall study in great detail. In Chapter 4, we study Lévy processes and the corresponding statistical estimators. The Lévy model will allow us to revisit several state-of-the-art reconstruction algorithms and compare them against the optimal MMSE estimator that we develop. In Chapter 5, we present the adaptive GAMP algorithm that allows us to apply the message-passing philosophy to more general inverse problems.

The algorithm has the capability to learn the unknown statistical parameters during the reconstruction. In Chapter 6, we present several numerical comparisons for algorithms discussed in the thesis. In Chapter 7, we present our algorithm for the optical tomographic microscope, which is a promising technique for quantitative three-dimensional (3D) mapping of the refractive index in biological cells and tissues.

Chapter 2

A Practical Guide to Inverse Problems

2.1 Introduction

In this chapter, we introduce the inverse problem formalism that will be used extensively in the sequel. We start by discussing several categories of forward models that can be used to model various practical acquisition systems. In the process, we also address the issues related to the discretization of continuous-domain inverse problems[1]. This is important since practical methods for solving inverse problems often rely on digital processing of the data in a computer. We finally introduce the statistical framework and several standard approaches that are currently used for solving inverse problems. Experimental evaluations throughout this chapter illustrate the capabilities of the described methods while simultaneously highlighting the necessity for the contributions presented in the rest of this thesis.

---

[1] The subsection on the discretization of inverse problems is based on our paper [1].

2.2 Forward Model

2.2.1 Continuous-Domain Formulation

The usual starting point in the formulation of inverse problems is the formal understanding of the acquisition process relating the physical signal x to the measured data of the form

$$\mathbf{y} = \mathrm{S}\{x\}. \tag{2.1}$$

The continuous-domain signal x may be a function of space and/or time, while the measurements are stored in an M-dimensional vector $\mathbf{y}$. For notational convenience, we assume that the signal and measurements are both real valued; nonetheless, all the algorithms considered in this thesis can be easily extended to work with complex valued signals. We also assume that the signal is of finite energy, and thus it belongs to the normed space $L_2(\mathbb{R}^d)$. In full generality, the scalar value of x at every continuous coordinate $\boldsymbol{r} = (r_1, \dots, r_d)$ inside $\mathbb{R}^d$ is denoted by $x(\boldsymbol{r}) \in \mathbb{R}$. We refer to S as the *forward operator* and typically assume that it models accurately the physics behind the acquisition. Depending on the acquisition modality, computation of S might involve various linear or nonlinear operations, including projection, orthogonal transformation, discretization, etc. In particular, we shall distinguish among the following four categories: nonlinear, generalized linear, linear, and signal denoising.

Nonlinear Models

The most general nonlinear forward model allows us to handle sophisticated acquisition modalities. Such models must still be sufficiently structured to be implemented and computed digitally on a computer. The computational complexity for numerical evaluation of such models should be weighted against the benefits of having a more accurate representation of the acquisition. As we shall see later, nonlinear models also complicate the recovery of the signal by leading to nonconvex costs in variational reconstruction methodology. In such difficult cases, it is beneficial to derive a linearizations of the model. Then, one can use the latter to obtain a good initial guess for the solution that can subsequently be refined by using the nonlinear model. We will illustrate one concrete example of a nonlinear forward model in Chapter 7, where we consider the quantitative recovery of the refractive-index in tomographic microscopy [2].

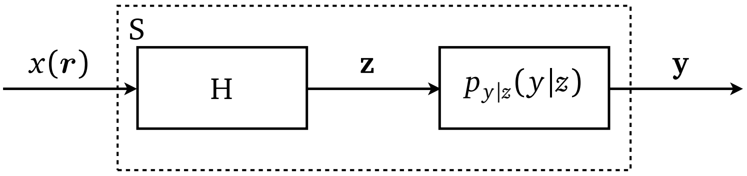


Figure 2.1: Generalized linear model.

Generalized Linear Models

The *generalized linear model* (GLM), illustrated in Figure 2.1, is a special case of nonlinear model that consists of a known linear mixing operator H followed by a probabilistic componentwise nonlinearity $p_{y|z}$. Such models have been extensively studied in statistics [3]. The linear part of the model has the general form

$$z_m = [\mathrm{H}\{x\}]_m = \int_{\mathbb{R}^d} x(\boldsymbol{r})\psi_m(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}, \quad (m = 1, \ldots, M) \tag{2.2}$$

where the measurement function ψ_m represents the spatial response of the mth detector in the acquisition system. The conditional probability distribution $p_{y|z}$ that is subsequently involved can model noise and various other types of deformations that are intrinsic to the physics of the acquisition.

One relevant practical example arises in analog-to-digital conversion (ADC), where one needs to estimate a signal from quantized measurements $\mathbf{y} = \mathrm{Q}(\mathrm{H}\{x\})$. This is a challenging problem because the quantization function Q is nonlinear and the operator H mixes x, thus necessitating joint estimation. Although reconstruction from quantized measurements is typically linear, more sophisticated, nonlinear techniques can offer significant improvements. In the case of ADC the improvement from replacing conventional linear estimation with nonlinear estimation increases with the oversampling factor [4–9].

From the algorithmic side, reconstruction methods based on GLM formulation might also suffer from nonconvexity of the cost function during optimization. The approaches based on message-passing algorithms seem to perform the best for GLM based inverse problems. In Chapter 5, we will introduce adaptive generalized approximate message passing (adaptive GAMP) algorithm that is specifically tailored for statistical estimation under GLM [8–10].

Linear Models

Further simplification of GLM is the linear forward model, which can be obtained by assuming that the distortion, characterized by the output nonlinearity $p_{y|z}$, is additive and signal independent. This allows us to re-express (2.2) as

$$\mathbf{y} = \mathbf{z} + \mathbf{e} \quad \text{with} \quad \mathbf{z} = \mathrm{H}\{x\}, \tag{2.3}$$

where H is given in (2.2) and **e** represents the noise, which is very often assumed to be independent and identically distributed (i.i.d.) Gaussian. Linear inverse problems are central to most modern imaging systems with applications in diverse areas such as biomicroscopy [11], magnetic resonance imaging [12], x-ray tomography [13], etc. The major advantage of linearity is that it allows us to borrow some intuition and rely on standard theoretical results from linear algebra. Typically, when confronted with a more general inverse problem, it is sensible that one reflects if there is a possibility to obtain a suitable linearization of the problem.

Most algorithms we develop in this thesis are particularly well-suited to solve linear inverse problems under Gaussian noise assumption. It is possible to extend many of them to more general types of noise, but this might result in nonconvex formulation of the reconstruction. In such scenarios, we can use the solution obtained under the Gaussian assumption to initialize the more general algorithm.

Signal Denoising

Even further simplification is possible by assuming that our acquisition system simply samples the signal x. In the context of the general form (2.2), this assumes that ψ_m corresponds to a shifted Dirac delta function, which allows us to write

$$\mathbf{y} = \mathbf{z} + \mathbf{e} \quad \text{with} \quad z_m = x(\boldsymbol{r}_m), \quad (m = 1, \dots, M) \tag{2.4}$$

where $\{\boldsymbol{r}_m\}_{m=1,\dots,M}$ are locations where the signal samples are taken and **e** represents the noise. A more general denoising model with a signal dependent noise can be obtained by sampling directly in the GLM formulation.

Signal denoising is considered as the most basic form of signal reconstruction. The sources of noise are typically application dependent, but the two most common types are Gaussian and Poisson noises [14]. Algorithmically, signal denoising is useful as a key sub-routine in methods for solving more general inverse problems. As we shall see in detail, it is referred to as proximal operator corresponding to a particular statistical distribution of the signal [15]. Also, denoising provides a practical scenario for testing various prior distributions since it disregards the effects due to the linear mixing operator H. We shall consider the signal denoising problem in Chapter 4, where we study the optimal estimation of signals with sparse derivatives.

2.2.2 Discrete Representation of the Signal

It is most often impossible to solve an inverse problem analytically, and we must resort to a computer program to recover x. In order to perform computations digitally, we need to discretize x into a finite number of parameters N that can be represented as a vector $\mathbf{x} \in \mathbb{R}^N$. To obtain a clean analytical discretization of the problem, we consider the generalized sampling approach using *shift-invariant* reconstruction spaces [16]. The advantage of such a representation is that it offers the same type of error control as finite-element methods, i.e., the approximation error between the original signal and its representation in the reconstruction space can be made arbitrarily small by choosing a sufficiently fine reconstruction grid.

The idea is to represent the signal x by projecting it onto a reconstruction space. We define our reconstruction space at resolution Δ as

$$V_\Delta(\varphi_{\text{int}}) \triangleq \left\{ x_\Delta(\boldsymbol{r}) = \sum_{\boldsymbol{k}\in\mathbb{Z}^d} x[\boldsymbol{k}]\varphi_{\text{int}}\left(\frac{\boldsymbol{r}}{\Delta} - \boldsymbol{k}\right) \ : \ x[\boldsymbol{k}] \in \ell_\infty(\mathbb{Z}^d) \right\} \tag{2.5}$$

where $x[\boldsymbol{k}] \triangleq x(\boldsymbol{r})|_{\boldsymbol{r}=\boldsymbol{k}}$, and φ_{int} is an interpolating basis function positioned on the reconstruction grid $\Delta\mathbb{Z}^d$. The interpolation property is $\varphi_{\text{int}}(\boldsymbol{k}) = \delta[\boldsymbol{k}]$. For the representation of x in terms of its samples $x[\boldsymbol{k}]$ to be stable and unambiguous, φ_{int} has to be a valid Riesz basis for $V_\Delta(\varphi_{\text{int}})$ [16]. Moreover, to guarantee that the approximation error decays as a function of Δ, the basis function should satisfy the partition of unity property

$$\sum_{\boldsymbol{k}\in\mathbb{Z}^d} \varphi_{\text{int}}(\boldsymbol{r} - \boldsymbol{k}) = 1, \tag{2.6}$$

for all $\boldsymbol{r} \in \mathbb{R}^d$. The projection of the signal onto the reconstruction space $V_\Delta(\varphi_{\text{int}})$ is given by

$$P_{V_\Delta} x(\boldsymbol{r}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} x(\Delta \boldsymbol{k}) \varphi_{\text{int}} \left(\frac{\boldsymbol{r}}{\Delta} - \boldsymbol{k} \right), \tag{2.7}$$

with the property that $P_{V_\Delta} P_{V_\Delta} x = P_{V_\Delta} x$ (i.e. $P_{V_\Delta} x$ is a projection operator). To simplify the notation, we shall use a unit sampling $\Delta = 1$ with the implicit assumption that the sampling error is negligible[2]. Thus, the resulting discretization is

$$x_1(\boldsymbol{r}) = P_{V_1} x(\boldsymbol{r}) = \sum_{\boldsymbol{k} \in \mathbb{Z}^d} x[\boldsymbol{k}] \varphi_{\text{int}}(\boldsymbol{r} - \boldsymbol{k}). \tag{2.8}$$

To summarize, x_1 is the discretized version of the original signal x and it is uniquely described by the samples $x[\boldsymbol{k}] = x(\boldsymbol{r})|_{\boldsymbol{r}=\boldsymbol{k}}$ for $\boldsymbol{k} \in \mathbb{Z}^d$. The main point is that the reconstructed signal is represented in terms of samples even though the problem is still formulated in the continuous-domain.

Although the signal representation (2.8) contains an infinite sum, in practice, we restrict ourselves to a subset of N basis functions with $\boldsymbol{k} \in \Omega$, where Ω is a discrete set of integer coordinates in a region of interest (ROI). Hence, we rewrite (2.8) as

$$x_1(\boldsymbol{r}) = \sum_{\boldsymbol{k} \in \Omega} x[\boldsymbol{k}] \varphi_{\boldsymbol{k}}(\boldsymbol{r}), \tag{2.9}$$

where $\varphi_{\boldsymbol{k}}$ corresponds to $\varphi_{\text{int}}(\cdot - \boldsymbol{k})$ up to modifications at the boundaries (periodization or Neumann boundary condition).

2.2.3 Discrete Measurement Model

The discretization that we just presented allows us to obtain elegant discrete representation for the generalized linear, linear, and signal denoising forward models. For more general nonlinear models the discrete representation might lack a closed form matrix-vector representation.

---

[2] If the sampling error is large, one can use a finer sampling and rescale the reconstruction grid appropriately.

Generalized Linear Models

By using the discretization scheme in (2.9), we are now ready to formally link the continuous model in Figure 2.1 to a corresponding discrete forward model. We substitute the signal representation (2.9) into (2.2) and obtain the discretized measurement model written in a matrix-vector form as

$$\mathbf{y} \sim p_{y|z}(\mathbf{y}|\mathbf{z}) \quad \text{with} \quad \mathbf{z} = \mathbf{H}\mathbf{x}, \tag{2.10}$$

where $\mathbf{y}$ is the M-dimensional measurement vector, $\mathbf{x} \triangleq \{x[\boldsymbol{k}]\}_{\boldsymbol{k}\in\Omega}$ is the N-dimensional signal vector, and $\mathbf{H}$ is the $M \times N$ measurement matrix whose entry $(m, \boldsymbol{k})$ is given by

$$[\mathbf{H}]_{m,\boldsymbol{k}} \triangleq \langle \psi_m, \varphi_{\boldsymbol{k}} \rangle = \int_{\mathbb{R}^d} \psi_m(\boldsymbol{r})\varphi_{\boldsymbol{k}}(\boldsymbol{r})\, \mathrm{d}\boldsymbol{r}. \tag{2.11}$$

This allows us to specify the discrete linear forward model that is compatible with the continuous-domain formulation. The solution of this problem yields the representation $x_1(\boldsymbol{r})$ of $x(\boldsymbol{r})$ which is parameterized in terms of the signal samples $\mathbf{x}$.

Nonlinear Models

Unfortunately, in the case of nonlinear forward models, an elegant model similar to (2.10) is generally not possible. However, it is still necessary to represent the signal as in (2.9) and replace the forward operator S by a discrete operator $\mathbf{S}$ that can be computed by substituting continuous-domain operators with their discrete counterparts acting directly on the samples $\mathbf{x}$

$$\mathbf{y} = \mathbf{S}(\mathbf{x}) + \mathbf{e}, \tag{2.12}$$

where $\mathbf{e}$ models the noise and potential discrepancies due to discretization. An example of such implementation is given in our microscopy application in Chapter 7.

2.2.4 Wavelet Discretization

Occasionally, we might prefer to represent the signal x in terms of its wavelet coefficients [17]. In that case, we constrain the basis function φ_{int} to be a scaling function that satisfies the property of multiresolution [18]. To obtain an equivalent characterization of the object with its orthonormal wavelet coefficients, we simply define the wavelets as a linear combination of $\varphi_{\boldsymbol{k}}$. Then, there exists a discrete wavelet transform (DWT) represented with the matrix $\mathbf{W}$ that bijectively maps $\mathbf{x}$ to the wavelet coefficients $\mathbf{w}$ as

$$\mathbf{w} = \mathbf{W}\mathbf{x} \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{W}^T \mathbf{w}, \tag{2.13}$$

and that represents the signal x in a continuous wavelet basis. Note that the matrix-vector multiplications above have efficient filterbank implementations [19].

2.3 Statistical Inference

As we have mentioned in Chapter 1, inverse problems are often ill-posed. This means that measurements $\mathbf{y}$ cannot explain the signal $\mathbf{x}$ uniquely, and in order to separate meaningful solutions from the noise, we are obliged to introduce supplementary information describing $\mathbf{x}$. Statistical theory provides a unified approach for imposing additional constraints on the solution. The basic idea is to introduce a prior probability distribution p_{x} favoring solutions that we consider to be correct and penalizing those that we consider to be wrong. Then, given the prior p_{x}, one can express the posterior distribution of the signal given the measurements

$$p_{\mathrm{x}|\mathrm{y}}(\mathbf{x}|\mathbf{y}) \propto p_{\mathrm{y}|\mathrm{x}}(\mathbf{y}|\mathbf{x})\, p_{\mathrm{x}}(\mathbf{x}), \tag{2.14}$$

where $\propto$ denotes equality after normalization, and $p_{\mathrm{y}|\mathrm{x}}$ is the conditional distribution of the data given the signal. The posterior (2.14) provides a complete statistical characterization to the problem. In particular, the *maximum-a-posteriority* (MAP) estimator is given by

$$\begin{aligned} \widehat{\mathbf{x}}_{\mathrm{MAP}} &\triangleq \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\max}\left\{p_{\mathrm{x}|\mathrm{y}}(\mathbf{x}|\mathbf{y})\right\} && \text{(2.15a)} \\ &= \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\left\{-\log\left(p_{\mathrm{y}|\mathrm{x}}(\mathbf{y}|\mathbf{x})\right) - \log\left(p_{\mathrm{x}}(\mathbf{x})\right)\right\} && \text{(2.15b)} \\ &= \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\left\{\mathscr{D}(\mathbf{x}) + \phi(\mathbf{x})\right\} && \text{(2.15c)} \end{aligned}$$

In the context of inverse problems, the first term $\mathscr{D}$ in (2.15) is called the *data term* due to its direct dependance on the measurements, while the second term ϕ is called the *regularizer* due to its capability to impose more meaningful solutions. Similarly, we might be interested in finding the *minimal mean squared error* (MMSE) estimator, which is usually expressed as the computationally intractable N-dimensional integral

$$\begin{aligned}\widehat{\mathbf{x}}_{\text{MMSE}} &\triangleq \operatorname*{arg\,min}_{\widehat{\mathbf{x}}\in\mathbb{R}^N} \left\{\mathbb{E}[\|\mathbf{x}-\widehat{\mathbf{x}}\|^2 \mid \mathbf{y}]\right\} && \text{(2.16a)}\\ &= \mathbb{E}[\mathbf{x} \mid \mathbf{y}] && \text{(2.16b)}\\ &= \int_{\mathbb{R}^N} \mathbf{x}\, p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})\, \mathrm{d}\mathbf{x}. && \text{(2.16c)}\end{aligned}$$

Once we accept the statistical perspective, we have several degrees of freedom to develop practical methods for solving inverse problems.

- The first degree of freedom lies in the specification of the prior distribution $p_{\mathbf{x}}$. In some cases such as for MMSE estimation, we would want the prior to match the true empirical distribution of possible signals $\mathbf{x}$ as closely as possible. At the same time, we want the prior to be as simple as possible in order to result in low-computation reconstruction algorithms. Prior distributions promoting *sparse* solutions in some transform domains are currently state-of-the-art in this regard [20].

- Specification of the conditional distribution $p_{\mathbf{y}|\mathbf{x}}$ is the second issue to address. For the generalized linear model (2.10), this task becomes trivial and yields

  $$p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) = p_{y|z}(\mathbf{y}|\mathbf{H}\mathbf{x}) = \prod_{m=1}^{M} p_{y|z}(y_m|[\mathbf{H}\mathbf{x}]_m). \tag{2.17}$$

  In more generic scenarios, one might either find an accurate statistical distribution characterizing the distortion or make a simplification by assuming a Gaussian noise model. The latter results in the popular *least-squares* data term penalizing the quadratic distance between $\mathbf{y}$ and $\mathbf{z}$.

- Once the prior $p_{\mathbf{x}}$ and the data distribution $p_{\mathbf{y}|\mathbf{x}}$ are specified, we must develop an algorithm for computing the MAP, MMSE, or other estimators efficiently. We will cover several state-of-the-art approaches based on two distinct philosophies: (a) methods that explicitly minimize some predetermined cost-function, (b) methods based on passing messages on graphical models. In the former category we have methods such as fast iterative shrinkage/thresholding algorithm (FISTA) [21] and alternating direction method of multipliers (ADMM) [22], while in the latter we have belief propagation (BP) algorithm [23] and generalized approximate message passing (GAMP) [24].

- Sometimes, we know the prior and the data distribution up to a finite number of unknown parameters

$$p_{\mathrm{x}}(\mathbf{x}|\boldsymbol{\theta}_x), \quad p_{\mathrm{y}|\mathrm{z}}(\mathbf{y}|\mathbf{z}, \boldsymbol{\theta}_z), \tag{2.18}$$

  where $\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x$ and $\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z$ represent parameters of the densities and $\boldsymbol{\Theta}_x \subseteq \mathbb{R}^{d_x}$ and $\boldsymbol{\Theta}_z \subseteq \mathbb{R}^{d_z}$ denote the corresponding parameter sets. For example, such scenario is realistic when we know the family of the prior distribution, but have no direct way of obtaining the parameters of the prior. Then, we must also find a method to choose the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ in a suitable way. The adaptive GAMP algorithm that we present in Chapter 5 is an effective method of doing this under generalized linear models.

In the rest of this section, we will discuss some of the standard approaches for solving inverse problems.

2.4 Least Squares

The simplest prior is actually a flat prior. Thus, one basic approach for solving a general inverse problem is the *least-squares* (LS) method that assumes a uniform prior p_{x} and an additive white Gaussian noise (AWGN) [25]. Although, it is particularly well suited for linear forward models with $M \geq N$, nothing prevents it from being applied more generally.

Algorithm 2.1: Gradient-descent to minimize: $\mathscr{C}_{\text{LS}}(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{S}(\mathbf{x})\|_2^2$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        step-size $\gamma \in (0, 1/L]$, where $L$ is the Lipschitz constant of $\nabla \mathscr{C}_{\text{LS}}$,
        and efficient implementation of $\nabla \mathscr{C}_{\text{LS}}$.
set:  $t \leftarrow 1$
repeat
    $\widehat{\mathbf{x}}^t \leftarrow \widehat{\mathbf{x}}^{t-1} - \gamma \nabla \mathscr{C}_{\text{LS}}(\widehat{\mathbf{x}}^{t-1})$          (gradient step)
    $t \leftarrow t + 1$
until stopping criterion
return  $\widehat{\mathbf{x}}^t$
```

2.4.1 Nonlinear Model

In nonlinear form of LS, one seeks the solution

$$\widehat{\mathbf{x}}_{\text{LS}} \triangleq \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \{\mathscr{C}_{\text{LS}}(\mathbf{x})\} \tag{2.19a}$$

$$= \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{S}(\mathbf{x})\|_2^2 \right\}, \tag{2.19b}$$

which can be interpreted as a search for a signal $\mathbf{x}$ that—by means of $\mathbf{S}$—matches $\mathbf{y}$ as closely as possible. The Euclidean norm $\|\cdot\|_2$ in (2.19b) corresponds to the assumed Gaussianity of the noise.

In the most general scenario, the LS function $\mathscr{C}_{\text{LS}}$ is non-convex, which implies that there might be many global and local solutions to the problem (2.19). In such cases, the computation of the global solution may be intractable, and we simply expect our optimization algorithms to find any one of the local minimizers of $\mathscr{C}_{\text{LS}}$.

For differentiable forward models $\mathbf{S}$, the gradient of $\mathscr{C}_{\text{LS}}$ is given by

$$\nabla \mathscr{C}_{\text{LS}}(\mathbf{x}) = \left[\frac{\partial}{\partial \mathbf{x}} \mathbf{S}(\mathbf{x}) \right]^T (\mathbf{S}(\mathbf{x}) - \mathbf{y}), \tag{2.20}$$

where $(\partial / \partial \mathbf{x})\mathbf{S}(\mathbf{x})$ is the Jacobian matrix of $\mathbf{S}$. When the gradient (2.20) can be computed efficiently for all $\mathbf{x}$, we can use the gradient descent method summarized in Algorithm 2.1 for finding a local solution of (2.19). The step-size $\gamma > 0$ must be sufficiently small to guarantee the convergence of the algorithm. More specifically, for a Lipschitz continuous gradient $\nabla \mathscr{C}_{\text{LS}}$

$$\|\nabla \mathscr{C}_{\text{LS}}(\mathbf{x}) - \nabla \mathscr{C}_{\text{LS}}(\mathbf{z})\|_2 \leq L\, \|\mathbf{x} - \mathbf{z}\|_2\,, \quad (\text{for all } \mathbf{x}, \mathbf{z} \in \mathbb{R}^N) \tag{2.21}$$

where $L > 0$ is the Lipschitz constant of $\nabla \mathscr{C}_{\text{LS}}$, the convergence is guaranteed for any $\gamma \in (0, 1/L]$ [21, 26]. In practice, it might be difficult to determine L analytically, and we might need to hand-pick a suitable step-size γ.

2.4.2 Linear Model

In the linear acquisition scenario, the LS reconstruction reduces to

$$\widehat{\mathbf{x}}_{\text{LS}} \triangleq \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\, \{\mathscr{C}_{\text{LS}}(\mathbf{x})\} \tag{2.22a}$$

$$= \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2}\, \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \right\}, \tag{2.22b}$$

where $\mathbf{H}$ is the measurement matrix given by (2.11). In this case, the gradient takes a simple form

$$\nabla \mathscr{C}_{\text{LS}}(\mathbf{x}) = \mathbf{H}^T(\mathbf{H}\mathbf{x} - \mathbf{y})\,. \tag{2.23}$$

In imaging, due to high-dimensionality of the problem the matrices $\mathbf{H}$ and $\mathbf{H}^T$ cannot be stored explicitly in memory, and we require an efficient implementation of two basic operations:

1. `multiply`$(\mathbf{x}) \triangleq \mathbf{H}\mathbf{x}$, (for all $\mathbf{x} \in \mathbb{R}^N$)
2. `multiplyTranspose`$(\mathbf{z}) \triangleq \mathbf{H}^T\mathbf{z}$. (for all $\mathbf{z} \in \mathbb{R}^M$)

Finding Lipschitz constant of the gradient reduces to computing the largest eigenvalue of $\mathbf{H}^T\mathbf{H}$

$$L = \lambda_{\text{max}}(\mathbf{H}^T\mathbf{H}), \tag{2.24}$$

which can be precomputed once by using an iterative method such as power iteration. The iterative LS approach for linear models is summarized in Algorithm 2.2.

Algorithm 2.2: Gradient-descent to minimize: $\mathscr{C}_{\text{LS}}(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{Hx}\|_2^2$

```
**input:** data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
           Lipschitz constant $L = \lambda_{\max}(\mathbf{H}^T\mathbf{H})$,
           and system matrix $\mathbf{H}$.
**set:** $t \leftarrow 1$
**repeat**
    $\widehat{\mathbf{x}}^t \leftarrow \widehat{\mathbf{x}}^{t-1} - (1/L)\mathbf{H}^T(\mathbf{H}\widehat{\mathbf{x}}^{t-1} - \mathbf{y})$        (gradient step)
    $t \leftarrow t + 1$
**until** stopping criterion
**return** $\widehat{\mathbf{x}}^t$
```

When $M \geq N$ and $\mathbf{H}$ is nonsingular, the LS solution is given by

$$\widehat{\mathbf{x}}_{\text{LS}} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{y} \tag{2.25a}$$

$$= \mathbf{x} + \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{e} \tag{2.25b}$$

where $\mathbf{H}^{-1}$ denotes the inverse of the matrix $\mathbf{H}$. In many practical applications, it is the case that $\mathbf{H}$ is ill-conditioned, which implies a presence of small singular values. As a result, LS yields a solution of poor quality due to the amplification of noise through the inverse of $\mathbf{H}^T\mathbf{H}$ in (2.25b).

When $M < N$, the situation is worse, because any $\mathbf{x} \in \mathbb{R}^N$ that satisfies

$$\mathbf{H}^T\mathbf{H}\mathbf{x} = \mathbf{H}^T\mathbf{y}, \tag{2.26}$$

corresponds to a valid LS solution. Since in such scenario $\mathbf{H}$ is singular, the system (2.26) has an infinity of possible solutions. Therefore, depending on the initial guess $\widehat{\mathbf{x}}^0$, Algorithm 2.2 can converge to any of these solutions.

2.4.3 Signal Denoising

In signal denoising, LS returns the trivial solution

$$\widehat{\mathbf{x}}_{\text{LS}} = \mathbf{y}. \tag{2.27}$$

This implies that to denoise a signal, we must go beyond the uniform prior.

Algorithm 2.3: Fast gradient-descent for: $\mathscr{C}_{\text{LS}}(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{S}(\mathbf{x})\|_2^2$

```
input:  the data $\mathbf{y}$, an initial guess $\widehat{\mathbf{x}}^0$,
        a step-size $\gamma \in (0, 1/L]$, where $L$ is the Lipschitz constant of $\boldsymbol{\nabla}\mathscr{C}_{\text{LS}}$,
        and an efficient implementation of $\boldsymbol{\nabla}\mathscr{C}_{\text{LS}}$.
set:  $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
  $\widehat{\mathbf{x}}^t \leftarrow \mathbf{s}^{t-1} - \gamma \boldsymbol{\nabla}\mathscr{C}_{\text{LS}}(\mathbf{s}^{t-1})$        (gradient step)
  $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
  $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
  $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

2.4.4 Accelerated Gradient descent

It is well-known that the sequence of solutions $\{\widehat{\mathbf{x}}^t\}_{t\in\mathbb{N}}$ obtained via the standard gradient descent converges quite slowly to the local minimizer of the cost $\mathscr{C}_{\text{LS}}$. In fact, it has been shown in [21] that its rate of convergence is given by

$$\mathscr{C}_{\text{LS}}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{\text{LS}}(\mathbf{x}^*) \leq \frac{L}{2t}\|\widehat{\mathbf{x}}^t - \mathbf{x}^*\|_2^2, \quad (t > 1) \tag{2.28}$$

where L is the Lipschitz constant of the gradient, and $\mathbf{x}^* \in \mathbb{R}^N$ is any of the local minimizers of $\mathscr{C}_{\text{LS}}$.

It turns out, it is possible to obtain an algorithm that has exactly the same per-iteration complexity as the standard gradient-descent, but has significantly better rate of convergence. The idea was originally developed for differentiable functions by Nesterov in [27]. The improved rate of convergence comes from a controlled over-relaxation that utilizes the previous iterates to produce a better guess for the next update. Possible implementations of the scheme for LS are shown in Algorithms 2.3 and 2.4. The convergence rate of the accelerated gradient-descent is given by

$$\mathscr{C}_{\text{LS}}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{\text{LS}}(\mathbf{x}^*) \leq \frac{2L}{(t+1)^2}\|\widehat{\mathbf{x}}^t - \mathbf{x}^*\|_2^2. \quad (t > 1) \tag{2.29}$$

Algorithm 2.4: Fast gradient-descent for: $\mathscr{C}_{\text{LS}}(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        Lipschitz constant $L = \lambda_{\max}(\mathbf{H}^T\mathbf{H})$,
        and system matrix $\mathbf{H}$.
set:  $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
    $\widehat{\mathbf{x}}^t \leftarrow \mathbf{s}^{t-1} - (1/L)\mathbf{H}^T(\mathbf{H}\mathbf{s}^{t-1} - \mathbf{y})$        (gradient step)
    $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
    $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
    $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

In practice, switching from a linear to a quadratic convergence rate translates into impressive improvements over the standard gradient descent. Extensions of such accelerated techniques to non-smooth functions typically deliver state-of-the-art performance for high-dimensional inverse problems and will therefore constitute our methods of choice [21].

Figure 2.2 provides a concrete example where a linear inverse problem is solved with LS. A *Lung* image of size 256 × 256 in Figure 2.2(a) is measured via 256 equally-spaced Radon projections in the range $[-\pi/2, \pi/2]$ in Figure 2.2(b). The measurements have additionally been corrupted by AWGN with the variance corresponding to $10\log_{10}(\|\mathbf{H}\mathbf{x}\|^2/\|\mathbf{e}\|^2) = 40$ dB. Figures 2.2 (c) and (d) illustrate the solutions of Algorithms 2.2 and 2.4, respectively, after 200 iterations. The algorithms were initialized with the image $\widehat{\mathbf{x}}^0 = \mathbf{0}$. Figures 2.2(e) and (f) illustrate the evolution of the LS cost and the signal-to-noise-ratio (SNR) of the reconstruction, respectively. This example illustrates the convergence gain from using the accelerated gradient descent algorithm for LS. Moreover, it shows a typical behaviour of LS: as the algorithm progresses the quality of the recovered signal drops. The drop can be more or less significant depending on the amount of noise and the condition number of the measurement matrix. One standard approach to circumvent such behaviour at the reconstruction is to regularize the solution by replacing the uniform prior in LS with some other $p_{\mathbf{x}}$ that imposes useful restrictions to the solution.

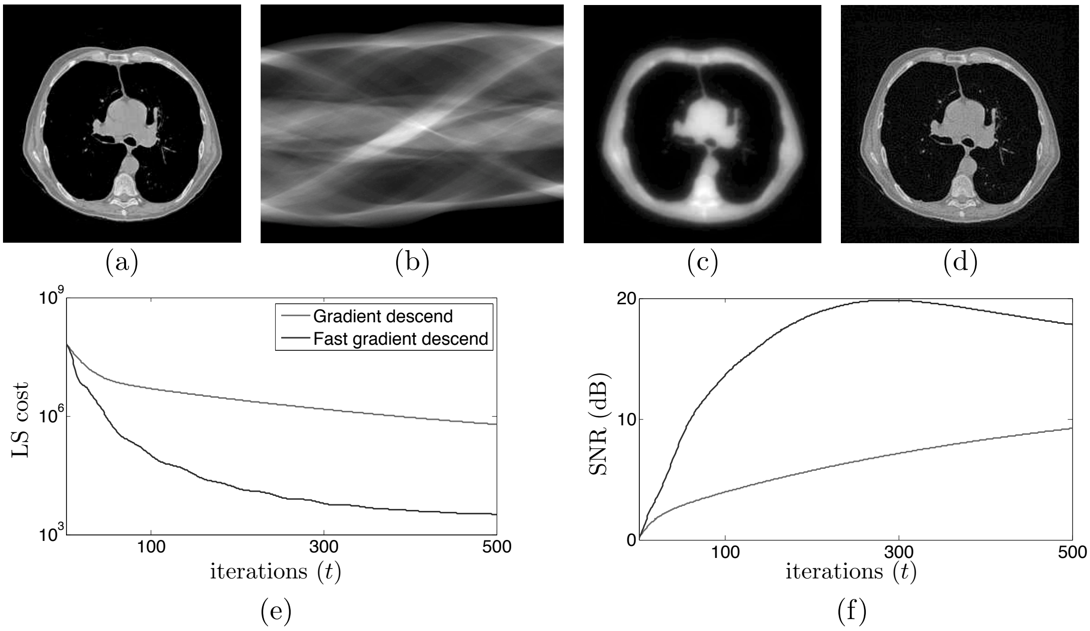


Figure 2.2: Least-squares reconstruction of a 256 × 256 *Lung* image from 256 Radon measurements: (a) original; (b) measurements; (c) gradient-descent solution at $t = 200$; (d) fast gradient-descent solution at $t = 200$; (e) evolution of the cost; (f) evolution of SNR

2.5 Wavelet-Domain Reconstruction

We now present wavelet-based methods for signal reconstruction. The idea is to define the prior distribution of the signal in the orthogonal wavelet domain. This is achieved by first expanding the signal as in (2.13), and introducing a simple prior for the wavelet coefficients. The prior distributions that we will consider have the general separable form

$$p_{\mathrm{x}}(\mathbf{x}) \propto p_{\mathrm{w}}(\mathbf{W}\mathbf{x}) = \prod_{n=1}^{N} p_{w_n}([\mathbf{W}\mathbf{x}]_n), \tag{2.30}$$

where $\propto$ denotes identity after normalization, and p_{w_n} is the distribution of the nth wavelet-coefficient. We define the potential function $\phi_{\mathbf{w}}$ as

$$\phi_{\mathbf{w}}(\mathbf{w}) \triangleq -\log(p_{\mathbf{w}}(\mathbf{w})) \tag{2.31a}$$

$$= -\sum_{n=1}^{N} \log\left(p_{w_n}(w_n)\right) = \sum_{n=1}^{N} \phi_{w_n}(w_n). \tag{2.31b}$$

We start the section with the review of signal denoising, where we revisit the classical wavelet soft-thresholding algorithm [28]. The algorithm is based on two empirical observations: (a) the energy of natural signals concentrates on very few wavelet coefficients; (b) the energy of i.i.d. noise is spread out uniformly in the wavelet domain. Accordingly, the noise can be suppressed by simply discarding small wavelet-coefficients. We then extend the wavelet denoising algorithm to more general inverse problems by introducing the concept of proximal operators. Initially, we will limit our discussion to the linear forward models with AWGN. At the end of the section, we present a more general algorithm for computing wavelet-based solutions under nonlinear forward models.

2.5.1 Signal Denoising

In the simplest case of AWGN denoising, the posterior distribution of the signal becomes

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \propto \prod_{n=1}^{N} \left(\mathcal{G}(x_n - y_n; \sigma^2) p_{w_n}([\mathbf{W}\mathbf{x}]_n)\right), \tag{2.32}$$

where $\mathcal{G}$ denotes the zero-mean Gaussian distribution

$$\mathcal{G}(x; \sigma^2) \triangleq \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2\sigma^2}}. \tag{2.33}$$

The orthonormality of the wavelet-basis implies the norm identity $||\mathbf{x}||_2^2 = ||\mathbf{W}\mathbf{x}||_2^2$ that allows for an equivalent wavelet-domain characterization of the signal statistics

$$p_{\mathbf{w}|\mathbf{u}}(\mathbf{w}|\mathbf{u}) \propto \prod_{n=1}^{N} \left(\mathcal{G}(w_n - u_n; \sigma^2) p_{w_n}(w_n)\right), \tag{2.34}$$

where $\mathbf{w} = \mathbf{W}\mathbf{x}$ and $\mathbf{u} = \mathbf{W}\mathbf{y}$. Such characterization effectively reduces the vector estimation problem into N-scalar estimation subproblems.

MAP Denoising

The wavelet–based MAP estimator is given by

$$\begin{aligned}
\widehat{\mathbf{x}}_{\text{MAP}} &= \arg\min_{\mathbf{x}\in\mathbb{R}^N} \{\mathscr{C}_{\text{MAP}}(\mathbf{x})\} && \text{(2.35a)}\\
&= \arg\min_{\mathbf{x}\in\mathbb{R}^N} \left\{\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x})\right\} && \text{(2.35b)}\\
&\overset{\text{(a)}}{=} \mathbf{W}^T \arg\min_{\mathbf{w}\in\mathbb{R}^N} \left\{\frac{1}{2}\|\mathbf{w}-\mathbf{W}\mathbf{y}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{w})\right\}, && \text{(2.35c)}
\end{aligned}$$

where in (a) we use the orthonormality of the wavelet-basis. Computationally, the estimator (2.35) reduces to

- wavelet-transformation $\mathbf{u} = \mathbf{W}\mathbf{y}$,
- scalar MAP estimation
$$\widehat{w}_n = \arg\min_{w\in\mathbb{R}} \left\{\frac{1}{2}(w-u_n)^2 + \sigma^2\phi_{w_n}(w)\right\}, \quad (n = 1,\dots,N) \tag{2.36}$$
- inverse wavelet-transformation $\widehat{\mathbf{x}}_{\text{MAP}} = \mathbf{W}^T\widehat{\mathbf{w}}$.

Since, the scalar estimator (2.36) can be precomputed and stored in a lookup table, the overall denoising procedure is very efficient. The optimization in (2.35c) can be represented more compactly by defining the *proximal operator*

$$\begin{aligned}
\widehat{\mathbf{w}} &= \text{prox}_{\phi_{\mathbf{w}}}\left(\mathbf{u};\, \sigma^2\right) && \text{(2.37a)}\\
&= \begin{pmatrix} \text{prox}_{\phi_{w_1}}(u_1;\sigma^2) \\ \vdots \\ \text{prox}_{\phi_{w_N}}(u_N;\sigma^2) \end{pmatrix} && \text{(2.37b)}\\
&\triangleq \arg\min_{\mathbf{w}\in\mathbb{R}^N} \left\{\frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{w})\right\}. && \text{(2.37c)}
\end{aligned}$$

Then wavelet-domain MAP estimation can be represented with a simple and efficient formula

$$\widehat{\mathbf{x}}_{\text{MAP}} = \mathbf{W}^T\text{prox}_{\phi_{\mathbf{w}}}(\mathbf{u};\sigma^2) \quad \text{with} \quad \mathbf{u} = \mathbf{W}\mathbf{y}. \tag{2.38}$$

MMSE Denoising

The wavelet-based MMSE estimator can be computed in a similar way by performing the following operations

- wavelet-transformation $\mathbf{u} = \mathbf{W}\mathbf{y}$,
- scalar MMSE estimation

$$\widehat{w}_n = \int_{\mathbb{R}} w_n \, p_{w_n|u_n}(w_n|u_n) \, \mathrm{d}w_n, \quad (n = 1, \dots, N) \tag{2.39a}$$

$$= \frac{\int_{\mathbb{R}} w_n \, \mathcal{G}(w_n - u_n; \sigma^2) p_{w_n}(w_n) \, \mathrm{d}w_n}{\int_{\mathbb{R}} \mathcal{G}(w_n - u_n; \sigma^2) p_{w_n}(w_n) \, \mathrm{d}w_n}, \tag{2.39b}$$

- inverse wavelet-transformation $\widehat{\mathbf{x}}_{\text{MMSE}} = \mathbf{W}^T \widehat{\mathbf{w}}$.

This can be represented more compactly as

$$\widehat{\mathbf{x}}_{\text{MMSE}} = \mathbf{W}^T \mathbb{E}_{\mathbf{w}|\mathbf{u}}[\mathbf{w}|\mathbf{u}] \quad \text{with} \quad \mathbf{u} = \mathbf{W}\mathbf{y}. \tag{2.40}$$

A quick glance at (2.40) and (2.38) reveals major similarities between MMSE and MAP estimators. The only observable difference is the shape of the scalar estimation functions: in MMSE they are obtained via integration (2.39), while in MAP they are obtained via optimization (2.36). Unfortunately, this similarity is restricted to signal denoising with orthogonal wavelets, and breaks for more general forward models and priors. Orthonormality of the wavelet transform and Parseval's norm identity provide the necessary elements for making MMSE computationally equivalent to MAP. One of our contributions in Chapter 3 will be the extension of this idea to more general wavelet-transforms by relying on the concept of consistent cycle spinning.

Illustrative Examples

As the first example, we assume that the wavelet-coefficients of the signal are zero-mean i.i.d. Gaussian with variance σ_x^2. Since the wavelet-expansion of an i.i.d. Gaussian

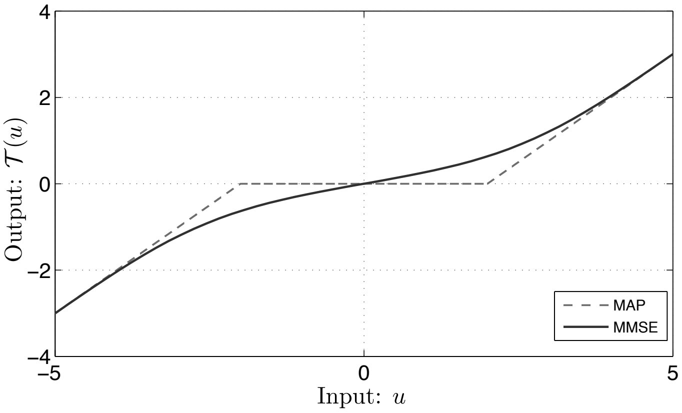


Figure 2.3: Illustration of $\mathscr{T}_{\text{MAP}}$ (dashed) and $\mathscr{T}_{\text{MMSE}}$ (solid) shrinkage functions for a Laplace prior with parameter $\lambda = 2$ and AWGN of variance $\sigma^2 = 1$.

vector has exactly the same distribution, estimation can be performed in the spatial- or the wavelet-domains equivalently. By introducing the prior distribution

$$p_{\mathbf{w}}(\mathbf{w};\sigma_x^2) = \frac{1}{\sqrt{(2\pi\sigma^2)^N}} \mathrm{e}^{-\frac{\|\mathbf{w}\|^2}{2\sigma_x^2}}, \tag{2.41}$$

into (2.36) and (2.39), we obtain

$$\widehat{\mathbf{w}}_{\text{MAP}} = \widehat{\mathbf{w}}_{\text{MMSE}} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma^2}\mathbf{u}. \tag{2.42}$$

The estimation is thus reduced to linearly *shrinking* noisy wavelet-coefficients $\mathbf{u}$ in a way that is inversely proportional to signal power σ_x^2 and proportional to σ^2, i.e., higher noise implies more shrinking.

When the wavelet-coefficients of the signal are assumed to be i.i.d. Laplace random variables with parameter λ, the prior is given by

$$p_{\mathbf{w}}(\mathbf{w};\lambda) = \frac{\lambda}{2}\mathrm{e}^{-\lambda\|\mathbf{w}\|_1}. \tag{2.43}$$

Then, MAP estimator reduces to

$$\widehat{\mathbf{x}}_{\text{MAP}} = \mathbf{W}^T \mathscr{T}_{\text{MAP}}\left(\mathbf{W}\mathbf{y};\lambda\sigma^2\right), \tag{2.44}$$

where the pointwise soft-thresholding function $\mathscr{T}$ is given by

$$\mathscr{T}_{\text{MAP}}(w;\lambda) \triangleq (|w|-\lambda)_{+}\,\text{sgn}(w). \tag{2.45}$$

The resulting denoising method corresponds to the popular wavelet-domain soft-thresholding algorithm that yields sparse solutions [28, 29]. The scalar MMSE estimator $\mathscr{T}_{\text{MMSE}}$ for the Laplace prior has also an analytical expression, albeit a more complicated one, that can easily be found in the literature [15]. The final MMSE estimator is then computed as

$$\widehat{\mathbf{x}}_{\text{MMSE}} = \mathbf{W}^{T}\mathscr{T}_{\text{MMSE}}\left(\mathbf{W}\mathbf{y};\lambda,\sigma^{2}\right), \tag{2.46}$$

Both $\mathscr{T}_{\text{MAP}}$ and $\mathscr{T}_{\text{MMSE}}$ are illustrated in Figure 2.3 for $\lambda = 2$ and $\sigma^2 = 1$.

2.5.2 Linear Model

We now consider the linear inverse problem

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{2.47}$$

where the vector $\mathbf{e}$ represents i.i.d. Gaussian measurement noise of variance σ^2. Given the wavelet-domain prior (2.30), one can express the posterior distribution

$$\begin{aligned} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) &\propto p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})\,p_{\mathbf{x}}(\mathbf{x}) && \text{(2.48a)}\\ &\propto \prod_{m=1}^{M}\mathscr{G}(y_m - [\mathbf{H}\mathbf{x}]_m;\sigma^2)\prod_{n=1}^{N}p_{w_n}([\mathbf{W}\mathbf{x}]_n), && \text{(2.48b)} \end{aligned}$$

where $\mathscr{G}$ is the Gaussian distribution. MAP estimator is given by

$$\begin{aligned} \widehat{\mathbf{x}}_{\text{MAP}} &= \arg\min_{\mathbf{x}\in\mathbb{R}^N}\{\mathscr{C}_{\text{MAP}}(\mathbf{x})\} && \text{(2.49a)}\\ &= \arg\min_{\mathbf{x}\in\mathbb{R}^N}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x})\right\}, && \text{(2.49b)} \end{aligned}$$

where $\phi_{\mathbf{w}}$ is the potential function in (2.31). Although, we might also be interested in finding the MMSE solution to general linear forward models, this becomes computationally intractable. We will thus concentrate in obtaining a method for computing the MAP estimator (2.49). In Chapter 5, we will return to the problem of MMSE estimation and present an adaptive GAMP method that can be used for approximating MMSE.

Algorithm 2.5: ISTA to minimize: $\mathscr{C}_{\text{MAP}}(\mathbf{x}) = (1/2)\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sigma^2 \phi_{\mathrm{w}}(\mathbf{W}\mathbf{x})$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        Lipschitz constant $L = \lambda_{\max}(\mathbf{H}^T\mathbf{H})$, noise variance $\sigma^2$,
        system matrix $\mathbf{H}$, and operator $\mathrm{prox}_{\phi_{\mathrm{w}}}$.
set:  $t \leftarrow 1$
repeat
  $\mathbf{z}^t \leftarrow \widehat{\mathbf{x}}^{t-1} - (1/L)\mathbf{H}^T\left(\mathbf{H}\widehat{\mathbf{x}}^{t-1} - \mathbf{y}\right)$        (gradient step)
  $\widehat{\mathbf{x}}^t \leftarrow \mathbf{W}^T \mathrm{prox}_{\phi_{\mathrm{w}}}(\mathbf{W}\mathbf{z}^t; \sigma^2/L)$        (shrinkage step)
  $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

ISTA Algorithm

An elegant and nonparametric method for computing the estimator (2.49) is the so-called *iterative shrinkage/thresholding algorithm* (ISTA) [30–32]. The latter relies on the separable proximal operator

$$\widehat{\mathbf{w}} = \mathrm{prox}_{\phi_{\mathrm{w}}}(\mathbf{u};\, \lambda) \triangleq \arg\min_{\mathbf{w} \in \mathbb{R}^N} \left\{ \frac{1}{2}\|\mathbf{w} - \mathbf{u}\|_2^2 + \lambda \phi_{\mathrm{w}}(\mathbf{w}) \right\}. \tag{2.50a}$$

The proximal operator corresponds to the wavelet-based MAP solution of the denoising problem (2.37). For our wavelet-domain priors, it reduces to a collection of scalar nonlinear maps that can be precomputed and stored in a lookup table.

Based on the definition of our forward model, ISTA can be expressed as in Algorithm 2.5. Iterations of ISTA combine gradient-descent steps with pointwise proximal operators.

To understand why the algorithm actually minimizes the cost, we bound $\mathscr{C}_{\text{MAP}}$ as follows

$$
\begin{aligned}
\mathscr{C}_{\text{MAP}}(\mathbf{x}) &= \frac{1}{2}\|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x}) && \text{(2.51a)}\\
&\overset{\text{(a)}}{=} \frac{1}{2}\|\mathbf{H}\widehat{\mathbf{x}}^{t-1}-\mathbf{y}\|_2^2 + (\mathbf{x}-\widehat{\mathbf{x}}^{t-1})^T\left[\mathbf{H}^T(\mathbf{H}\widehat{\mathbf{x}}^{t-1}-\mathbf{y})\right] && \text{(2.51b)}\\
&\quad + (\mathbf{x}-\widehat{\mathbf{x}}^{t-1})^T\mathbf{H}^T\mathbf{H}(\mathbf{x}-\widehat{\mathbf{x}}^{t-1}) + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x})\\
&\overset{\text{(b)}}{\leq} \frac{1}{2}\|\mathbf{H}\widehat{\mathbf{x}}^{t-1}-\mathbf{y}\|_2^2 + (\mathbf{x}-\widehat{\mathbf{x}}^{t-1})^T\left[\mathbf{H}^T(\mathbf{H}\widehat{\mathbf{x}}^{t-1}-\mathbf{y})\right] && \text{(2.51c)}\\
&\quad + \frac{1}{2\gamma}\|\mathbf{x}-\widehat{\mathbf{x}}^{t-1}\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x})\\
&= \frac{1}{2\gamma}\left\|\mathbf{x}-\left[\widehat{\mathbf{x}}^{t-1}-\gamma\mathbf{H}^T\left(\mathbf{H}\widehat{\mathbf{x}}^{t-1}-\mathbf{y}\right)\right]\right\|_2^2 + \sigma^2\phi_{\mathbf{w}}(\mathbf{W}\mathbf{x}) + \text{const} && \text{(2.51d)}\\
&= \mathscr{Q}(\mathbf{x},\widehat{\mathbf{x}}^{t-1}), && \text{(2.51e)}
\end{aligned}
$$

where "const" denotes terms that are constant with respect to $\mathbf{x}$. In (a) we performed a quadratic Taylor expansion of the cost around $\mathbf{x}=\widehat{\mathbf{x}}^{t-1}$, in (b) we selected γ such that

$$\frac{1}{2\gamma}\|\mathbf{x}\|_2^2 \geq \|\mathbf{H}\mathbf{x}\|_2^2 \tag{2.52}$$

for all $\mathbf{x}\in\mathbb{R}^N$. To guarantee monotone convergence, equation (2.52) restricts the choice of γ to the interval $(0, 1/L]$, where $L=\lambda_{\text{max}}(\mathbf{H}^T\mathbf{H})$. Note that the auxiliary cost function $\mathscr{Q}(\mathbf{x},\widehat{\mathbf{x}}^{t-1})$ that depends on the previous estimate $\widehat{\mathbf{x}}^{t-1}$ is much simpler than the original cost $\mathscr{C}_{\text{MAP}}$. Our formulation imposes $\mathscr{Q}(\mathbf{x},\widehat{\mathbf{x}}^{t-1}) \geq \mathscr{C}_{\text{MAP}}(\widehat{\mathbf{x}}^{t-1})$ with equality when $\mathbf{x}=\widehat{\mathbf{x}}^{t-1}$. Thus, one can recover the ISTA iteration by simply minimizing the cost $\mathscr{Q}(\mathbf{x},\widehat{\mathbf{x}}^{t-1})$ over $\mathbf{x}$ and setting the solution to be $\widehat{\mathbf{x}}^{t}$.

FISTA

Similar to the gradient descent algorithm for LS estimation, the main weakness of ISTA is in its slow convergence. FISTA, summarized in Algorithm 2.6, is an improvement of the standard ISTA that results in a rate of convergence that is equivalent to accelerated gradient descent of Section 2.4.4. The convergence to the global minimizer of $\mathscr{C}_{\text{MAP}}$ is only achieved when the potential function $\phi_{\mathbf{w}}$ is convex [21]. Note that a simpler,

Algorithm 2.6: FISTA for: $\mathscr{C}_{\text{MAP}}(\mathbf{x}) = (1/2)\|\mathbf{y}-\mathbf{Hx}\|_2^2 + \sigma^2\phi_{\text{w}}(\mathbf{Wx})$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        Lipschitz constant $L = \lambda_{\max}(\mathbf{H}^T\mathbf{H})$, noise variance $\sigma^2$,
        system matrix $\mathbf{H}$, and operator $\text{prox}_{\phi_{\text{w}}}$.
set:  $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
    $\mathbf{z}^t \leftarrow \mathbf{s}^{t-1} - (1/L)\mathbf{H}^T(\mathbf{Hs}^{t-1} - \mathbf{y})$          (gradient step)
    $\widehat{\mathbf{x}}^t \leftarrow \mathbf{W}^T\text{prox}_{\phi_{\text{w}}}(\mathbf{Wz}^t; \sigma^2/L)$          (shrinkage step)
    $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
    $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
    $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

and in some cases more effective way of speeding up ISTA, was proposed by Wright *et al.* [33], where the acceleration is obtained by using larger step-sizes.

2.5.3 Nonlinear Model

In the most general case, we consider the solution $\mathbf{x}$ accurate if it is consistent with the available measurements $\mathbf{y}$. While adopting this principle, we nevertheless want to accomodate for noise and model imperfections. Therefore, it is practical to propose a soft constraint, demanding that $\widehat{\mathbf{x}}$ reintroduced into our forward model results in measurements $\widehat{\mathbf{y}}$ that are close to $\mathbf{y}$. Accordingly, we define the general data discrepancy measure $\mathscr{D}(\mathbf{x})$, which depends explicitly on the signal $\mathbf{x}$, and implicitly on the measurements $\mathbf{y}$ and the forward model $\mathbf{S}$. The data term might also depend on some unknown parameters inherent to the forward model. A popular data term that we have already discussed is the LS cost

$$\mathscr{D}(\mathbf{x}) = \mathscr{C}_{\text{LS}}(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{S}(\mathbf{x})\|_2^2. \tag{2.53}$$

Other possible $\mathscr{D}$ can be obtained by considering different noise distributions as in (2.15c). To obtain a wavelet-based estimator for general forward models, we assume that $\mathscr{D}$ is

Algorithm 2.7: FISTA for: $\mathscr{C}_{\text{MAP}}(\mathbf{x}) = \mathscr{D}(\mathbf{x}) + \phi_{\mathbf{w}}(\mathbf{W}\mathbf{x})$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        step-size $\gamma \in (0, 1/L]$ where $L$ is the Lipschitz constant of $\nabla\mathscr{D}$,
        and efficient implementation of $\nabla\mathscr{D}$ and $\text{prox}_{\phi_{\text{w}}}$.
set:  $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
  $\mathbf{z}^t \leftarrow \mathbf{s}^{t-1} - \gamma\nabla\mathscr{D}(\mathbf{s}^{t-1})$          (gradient step)
  $\widehat{\mathbf{x}}^t \leftarrow \mathbf{W}^T \text{prox}_{\phi_{\text{w}}}(\mathbf{W}\mathbf{z}^t; \gamma)$          (shrinkage step)
  $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
  $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
  $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

differentiable with respect to $\mathbf{x}$ and that its gradient $\nabla\mathscr{D}$ can be efficiently computed. Algorithm 2.7 summarizes the resulting method for computing the MAP estimator for general forward models.

2.6 Gradient–Based Regularization

We now present an alternative to wavelet–based approach. The prior distribution of the signal will be defined in the gradient domain. Although, the idea can be generalized beyond gradient [34], the state-of-the-art performance of gradient–based regularization such as total variation (TV) [35] makes the latter particularly interesting [36]. With the new priors, our algorithms favor signals whose gradient follows certain statistical distributions. For computational reasons, we will restrict ourselves to separable prior distributions of the form

$$p_{\mathbf{x}}(\mathbf{x}) \propto p_{\mathbf{u}}(\mathbf{D}\mathbf{x}) = \prod_{n=1}^{N} p_{\mathbf{u}_n}([\mathbf{D}\mathbf{x}]_n), \tag{2.54}$$

where $\mathbf{D} \in \mathbb{R}^{dN \times N}$ is the discrete-counterpart of the gradient. Note that for each signal pixel n, there are d gradient coefficients

$$\mathbf{u}_n \triangleq [\mathbf{D}\mathbf{x}]_n = \begin{pmatrix} [\mathbf{D}_1\mathbf{x}]_n \\ \vdots \\ [\mathbf{D}_d\mathbf{x}]_n \end{pmatrix}, \tag{2.55}$$

where $\mathbf{D}_k$ denotes the derivative along the dimension k. This implies that $p_{\mathbf{u}_n}$ is a multivariate probability distribution. It is also important to note that sometimes the prior of form (2.54) may be improper, which means that it cannot be normalized.

We define the potential function $\phi_{\mathbf{u}}$ as

$$\begin{aligned} \phi_{\mathbf{u}}(\mathbf{u}) &\triangleq -\log(p_{\mathbf{u}}(\mathbf{u})) && \text{(2.56a)} \\ &= -\sum_{n=1}^{N} \log\left(p_{\mathbf{u}_n}(\mathbf{u}_n)\right) = \sum_{n=1}^{N} \phi_{\mathbf{u}_n}(\mathbf{u}_n). && \text{(2.56b)} \end{aligned}$$

2.6.1 Linear Model

Given our prior (2.54), we can express the posterior distribution

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \propto \prod_{m=1}^{M} \mathscr{G}(y_m - [\mathbf{H}\mathbf{x}]_m; \sigma^2) \prod_{n=1}^{N} p_{\mathbf{u}_n}([\mathbf{D}\mathbf{x}]_n), \tag{2.57a}$$

where $\mathscr{G}$ is the Gaussian distribution. Similar to the wavelet-domain estimation, we will restrict our discussion to the MAP estimation, due to intractability of the MMSE estimator. We will extensively address MMSE estimation for signals with sparse derivatives in Chapter 4. The MAP estimator is specified by

$$\widehat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sigma^2 \phi_{\mathbf{u}}(\mathbf{D}\mathbf{x}) \right\}. \tag{2.58}$$

We will define the proximal operator for $\phi_{\mathbf{u}}$ in (2.56) as

$$\widehat{\mathbf{x}} = \operatorname{prox}_{\phi_{\mathbf{u}}}(\mathbf{z}; \lambda) \triangleq \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda \phi_{\mathbf{u}}(\mathbf{D}\mathbf{x}) \right\}. \tag{2.59}$$

Algorithm 2.8: FISTA for: $\mathscr{C}_{\text{MAP}}(\mathbf{x}) = (1/2)\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \sigma^2\phi_{\mathbf{u}}(\mathbf{D}\mathbf{x})$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        Lipschitz constant $L = \lambda_{\max}(\mathbf{H}^T\mathbf{H})$, noise variance $\sigma^2$,
        system matrix $\mathbf{H}$, and operator $\text{prox}_{\phi_{\mathbf{u}}}$.
set:  $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
  $\mathbf{z}^t \leftarrow \mathbf{s}^{t-1} - (1/L)\mathbf{H}^T(\mathbf{H}\mathbf{s}^{t-1} - \mathbf{y})$    (gradient step)
  $\widehat{\mathbf{x}}^t \leftarrow \text{prox}_{\phi_{\mathbf{u}}}(\mathbf{z}^t; \sigma^2/L)$    (prox step)
  $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
  $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
  $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

The proximal operator corresponds to the gradient–based MAP solution of the denoising problem $\mathbf{H} = \mathbf{I}$. However, unlike the wavelet–based approach, here the proximal operators are not scalar mappings that can be precomputed easily. One needs to develop additional algorithms for computing them [36, 37]. Algorithm 2.8 summarizes the FISTA approach to gradient-based MAP estimation. Note that Algorithm 2.8 and Algorithm 2.6 are nearly identical with only difference being in the computation of the respective proximal operators.

Total Variation prior

The anisotropic version of the popular TV method is obtained by assuming an i.i.d. Laplace prior on the components of $\mathbf{u} = \mathbf{D}\mathbf{x}$. This results in the following minimization problem

$$\widehat{\mathbf{x}}_{\text{TV}} \triangleq \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{D}\mathbf{x}\|_1\right\} \tag{2.60a}$$

$$= \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min}\left\{\frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 + \lambda\sum_{k=1}^{d}\|\mathbf{D}_k\mathbf{x}\|_1\right\}, \tag{2.60b}$$

Algorithm 2.9: FISTA for: $\mathscr{C}_{\text{MAP}}(\mathbf{x}) = \mathscr{D}(\mathbf{x}) + \phi_{\mathbf{u}}(\mathbf{D}\mathbf{x})$

```
input: data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
       step-size $\gamma \in (0, 1/L]$ where $L$ is the Lipschitz constant of $\boldsymbol{\nabla}\mathscr{D}$,
       and efficient implementation of $\boldsymbol{\nabla}\mathscr{D}$ and $\text{prox}_{\phi_{\mathbf{u}}}$.
set: $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
  $\mathbf{z}^t \leftarrow \mathbf{s}^{t-1} - \gamma \boldsymbol{\nabla}\mathscr{D}(\mathbf{s}^{t-1})$        (gradient step)
  $\widehat{\mathbf{x}}^t \leftarrow \text{prox}_{\phi_{\mathbf{u}}}(\mathbf{z}^t; \gamma)$        (prox step)
  $k_t \leftarrow \frac{1}{2}\left(1 + \sqrt{1 + 4k_{t-1}^2}\right)$
  $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1} - 1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
  $t \leftarrow t + 1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

where, without loss of generality, we assumed that λ incorporates both the noise variance σ^2 and the scale parameter of Laplace distribution. The problem with the *anisotropic* formulation in (2.60) is that it gives preference to some orientations of the signal. Consequently, there is an alternative *isotropic* formulation of TV method that results in the minimization of form

$$\widehat{\mathbf{x}}_{\text{TV}} \triangleq \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \sum_{n=1}^{N} \|[\mathbf{D}\mathbf{x}]_n\|_2 \right\} \tag{2.61a}$$

$$= \underset{\mathbf{x} \in \mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda \sum_{n=1}^{N} \sqrt{[\mathbf{D}_1\mathbf{x}]_n^2 + \cdots + [\mathbf{D}_d\mathbf{x}]_n^2} \right\}, \tag{2.61b}$$

with some appropriate boundary conditions for the signal. Although, alternative gradient-priors have been extensively discussed in literature [1,38], classical TV is probably the most popular regularizer for solving inverse problems in imaging.

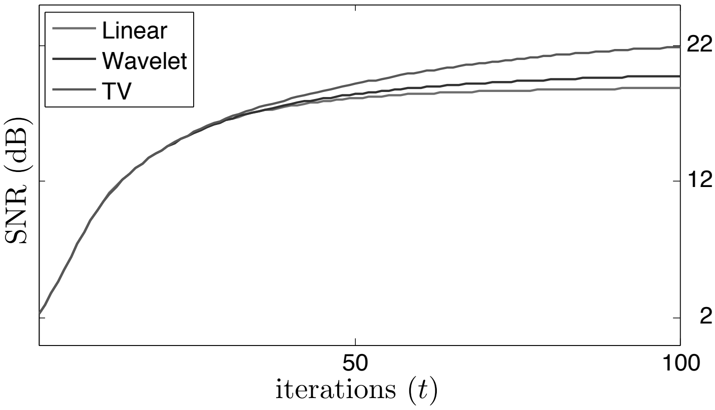


Figure 2.4: Evolution of the reconstruction SNR of a 256 × 256 *Lung* image from 64 Radon measurements with noise variance corresponding to the input SNR of 50 dB.

2.7 Experimental Evaluation

We revisit the problem of reconstructing an image from its Radon measurements. This time, however, the *Lung* image in Figure 2.5(a) is only projected 64 times. The noiseless measurements $\mathbf{z} = \mathbf{H}\mathbf{x}$ have then been corrupted with AWGN corresponding to input SNR of 50 dB. We compare three reconstruction algorithms: linear, wavelet-domain, and TV. The linear solution is obtained by minimizing

$$\mathscr{C}(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2,$$

where $\lambda > 0$ is the parameter, which is hand picked for the lowest MSE. The linear solution is illustrated in Figure 2.5(b) and has the final SNR of 18.86 dB. In wavelet-domain algorithm, we penalize the ℓ_1-norm of wavelet-coefficients to promote sparse solutions to the problem. We use the Haar transform with 5 decomposition levels and the penalization is only performed on the detail (high-pass) coefficients. The parameter λ is again tuned for the optimal MSE performance. The corresponding solution with SNR 19.8 dB is illustrated in Figure 2.5(c). The TV solution is obtained by minimizing the cost (2.61) with MSE optimal λ. The final TV solution llustrated in Figure 2.5(d), has SNR of 21.89 dB.

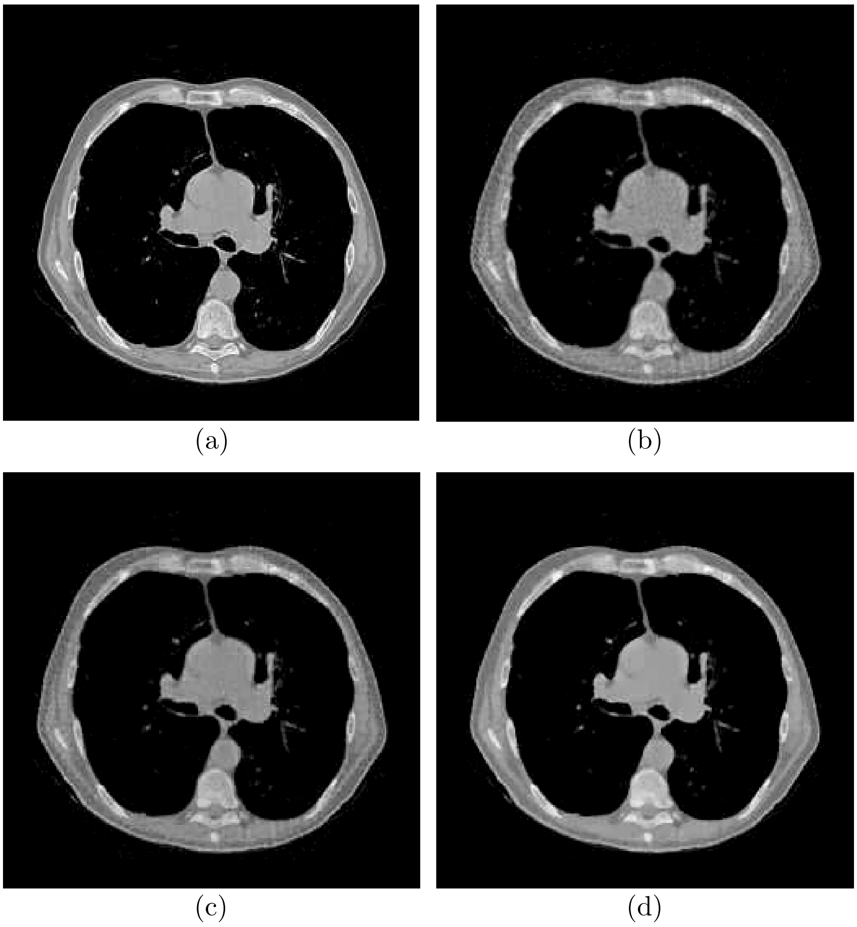

(a) (b) (c) (d)

Figure 2.5: Reconstruction of a 256 × 256 *Lung* image from 64 Radon measurements with noise variance corresponding to the input SNR of 50 dB: (a) original; (b) linear (SNR = 18.86 dB); (c) wavelet-domain ℓ_1 (SNR = 19.8 dB); (d) TV (SNR = 21.89 dB).

The first thing our experimental evaluation illustrates is that it is possible to significantly improve the reconstruction performance by using nonlinear reconstruction. By now, it is well-known that sparsity-driven nonlinear algorithms systematically outperform their classical linear counterparts. The second observation is related to suboptimal performance of orthogonal wavelet-domain methods compared to TV. The superior performance of TV makes it the method of choice for solving general inverse problems. However, it is known that it is possible to improve the performance of wavelet-domain methods by making them shift-invariant. We will discuss this idea in detail in the next chapter. The final observation is that due to oracle-aided tuning of λ, the performances presented here are not achievable in practice and one needs to develop more practical ways to set the unknown parameters in the problem.

2.8 Summary

We reviewed and addressed a range of issues related to the resolution of inverse problems. In particular, we presented wavelet– and gradient–based approaches to compute regularized solutions to inverse problems. Computational simplicity of wavelet-based approaches makes them ideal for dealing with high-dimensional data sets encountered in practice. However, our experimental evaluations have illustrated that the gradient–based methods yield solutions of superior quality. This suboptimal performance of wavelet-based methods is mainly attributed to the lack of translation-invariance in orthogonal wavelet-bases [39]. In Chapter 3, we address this issue by showing that it is possible to make wavelet–based methods equivalent to their gradient–based counterparts.

Due to unavailability of tractable methods for performing MMSE estimation, our algorithms were mostly limited to computing MAP solutions. One delicate aspect of applying our statistical estimators to imaging is that the true statistical distribution of the data is rarely known. Moreover, it was shown by Gribonval [40] that a MAP solution $\widehat{\mathbf{x}}_{\text{MAP}}$ for an assumed prior $p_{\mathbf{x}}$ might, in fact, correspond to an MMSE solution $\widehat{\mathbf{x}}_{\text{MMSE}}$ for another prior $q_{\mathbf{x}}$. Thus, if our data is actually distributed as $q_{\mathbf{x}}$, one might actually use the MAP solution to approximate MMSE. From this perspective, practically successful methods commonly interpreted as MAP estimators (e.g. TV regularization) could actually be computing an MMSE solution of the problem. In Chapter 4, we investigate this issue thoroughly by introducing a generative stochastic model for signals that have sparse-derivatives. Our signals based on Lévy processes are ideally suited for TV-like algorithms. The key to our investigation will be a novel message-passing algorithm for computing the MMSE solution for Lévy processes.

We take the issue of MMSE estimation further in Chapter 5. In particular, we propose and analyze a novel algorithm called adaptive GAMP. This message-passing algorithm is ideally suited for statistical estimation under generalized linear models when the signal prior p_{x} can be made separable in some basis. Besides approximating the MMSE solution, the algorithm is also capable of learning the unknown statistical parameters in the problem. In Chapter 6, we also present several simulations comparing the performance of MMSE estimator computed with GAMP against wavelet– and gradient–based approaches discussed in this chapter.

Finally, in Chapter 7, we apply the non-linear forward model–based algorithm to 3D imaging of objects using optical tomographic microscope, which is a very challenging modality due to the extremely large dimensionality of the data and ill-posed nature of the inverse problem. We shall propose a novel characterization of the forward model as well as TV–based iterative estimation algorithm.

Chapter 3

Cycle Spinning: Boosting Wavelet-Domain Reconstruction

3.1 Overview

In this chapter, we discuss *cycle spinning*, which is a widely used technique for improving the performance of wavelet–based methods for solving linear inverse problems. Extensive numerical experiments have shown that it significantly improves the quality of the recovered signal without increasing the computational cost. Our first contribution is theoretical, where we provide the first convergence result for cycle spinning. We prove that the sequence of reconstructed signals is guaranteed to converge to the minimizer of some global cost function that incorporates all wavelet shifts. Our second contribution is conceptual, where we develop the *consistent cycle spinning* technique that allows to compute MAP and MMSE solutions to the signal denoising problem in synthesis formulation. We empirically validate the MMSE performance of the technique through statistical estimation of Lévy processes[1].

---

[1] This chapter is based on our papers [41–43]

3.2 Introduction

We revisit the problem of estimating an unknown signal from noisy linear observations

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}, \tag{3.1}$$

where the matrix $\mathbf{H} \in \mathbb{R}^{M \times N}$ models the response of the acquisition device; the vector $\mathbf{e}$ represents the measurement noise, which is assumed to be independent and identically distributed (i.i.d.) Gaussian. As mentioned in Chapter 2, when problem (3.1) is ill-posed, the standard approach is to introduce some prior distribution $p_{\mathbf{x}}$ that promotes solutions with desirable properties. In the wavelet–based framework, regularization is achieved by favoring solutions that have *sparse* wavelet expansions. One popular approach is to use the non-smooth convex function $\phi_{\mathrm{w}}(\mathbf{W}\mathbf{x}) = \|\mathbf{W}\mathbf{x}\|_1$, where $\mathbf{W} \in \mathbb{R}^{N \times N}$ represents a wavelet transform [30–32]. Although (2.49) generally does not admit a closed-form solution with non-quadratic regularizers, it can still be computed efficiently using iterative algorithms such as ISTA [30–32]. Based on the definition of our statistical model, ISTA can be expressed as in Algorithm 2.5 with the pointwise shrinkages of the form

$$\mathrm{prox}_{\phi_{\mathrm{w}}}(w; \lambda) = \mathcal{T}(w; \lambda) \triangleq (|w| - \lambda)_{+}\mathrm{sgn}(w), \tag{3.2}$$

Because of its simplicity, ISTA has become the method of choice for finding sparse solutions.

3.3 Cycle Spinning

The theory of wavelet–regularized reconstruction is often formulated with orthogonal wavelet transforms. However, in order to make regularized wavelet–based methods truly competitive, one needs to make the transform shift-invariant. The concept was first introduced by Coifman and Donoho for wavelet–based denoising under the name of *cycle spinning* [39].

Let the matrix $\mathbf{W}_k$ denote an orthogonal wavelet transform with k-th shift applied to all the basis functions[2] in $\mathbf{W}$. We consider the K different shifts $\mathbf{W}_1, \dots, \mathbf{W}_K$ that are

---

[2] Alternatively, this effect can be achieved by shifting and unshifting the signal.

Algorithm 3.1: ISTA with Cycle Spinning

```
input: data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
       step sizes $\{\gamma_t\}_{t\in\mathbb{N}}$, noise variance $\sigma^2$,
       family of $K$ shifted wavelet transforms $\{\mathbf{W}_k\}_{k=1,\ldots,K}$,
       system matrix $\mathbf{H}$, and operator $\mathrm{prox}_{\phi_{\mathrm{w}}}$.
set: $t \leftarrow 1$
repeat
  $k_t \leftarrow 1 + (t-1 \mod K)$                                            (choice of basis)
  $\mathbf{z}^t \leftarrow \widehat{\mathbf{x}}^{t-1} - \gamma_t \mathbf{H}^T \left(\mathbf{H}\widehat{\mathbf{x}}^{t-1} - \mathbf{y}\right)$    (gradient step)
  $\widehat{\mathbf{x}}^t \leftarrow \mathbf{W}_{k_t}^T \mathrm{prox}_{\phi_{\mathrm{w}}}(\mathbf{W}_{k_t}\mathbf{z}^t; \gamma_t\sigma^2)$              (shrinkage step)
  $t \leftarrow t+1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

required to get a shift-invariant version of $\mathbf{W}$. The original idea developed in [39] was to perform a one-time shrinkage of form

$$\widehat{\mathbf{x}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{W}_k^T \widehat{\mathbf{w}}_k \quad \text{with} \quad \widehat{\mathbf{w}}_k = \mathrm{prox}_{\phi_{\mathrm{w}}}(\mathbf{W}_k\mathbf{y}; \sigma^2), \tag{3.3}$$

Practically, the method performs K basic wavelet-denoising operations in parallel and averages the results. A remarkable property of cycle spinning is that it is guaranteed to improve upon the non-redundant wavelet denoising [15].

An alternative technique called *recursive cycle spinning* was introduced by Fletcher *et al.* [44, 45]. They proposed to improve the simple averaging solution in (3.3) with a scheme that iteratively updates the initial guess $\widehat{\mathbf{x}}^0 = \mathbf{y}$ as

$$\widehat{\mathbf{x}}^t = \mathbf{W}_{k_t}^T \mathscr{T}_{\mathrm{RCS}}(\mathbf{W}_{k_t}\widehat{\mathbf{x}}^{t-1}; \lambda), \quad (t \in \mathbb{N}) \tag{3.4}$$

where $k_t = 1 + (t-1 \mod K)$ is an iteration-dependent shift, $\mathscr{T}_{\mathrm{RCS}}$ is some admissible thresholding function, and the parameter $\lambda > 0$ controls the amount of shrinkage. The authors have proved the convergence of this iterative denoising algorithm for thresholding functions $\mathscr{T}_{\mathrm{RCS}}$ that perform projections[3].

---

[3] Note that the soft-thresholding operation (3.2) is not a projection and thus is not usable within this scheme.

The idea of cycle spinning was generalized to more-general linear inverse problems by Figueiredo and Nowak [30]. They incorporated both parallel and recursive variations of cycle spinning into an iterative reconstruction scheme, and reported significant improvements in reconstruction quality. Currently, cycle spinning is used in the majority of wavelet-based reconstruction algorithms to obtain higher-quality solutions with less-blocky artifacts [11,12,46,47]. However, it is rarely accounted for in the accompanying theory.

A simple way to implement cycle spinning for solving general linear inverse problems, without increasing the memory usage, is to consider Algorithm 3.1. Even though Algorithm 3.1 has nearly the same computational cost as Algorithm 2.5, it yields results of significantly higher quality (see Figure 3.2). The algorithm is different from the original formulation of cycle spinning in [39], where the thresholded wavelet-coefficients corresponding to different shifts are simply averaged. It rather corresponds to the recursive cycle spinning approaches used in [11,12,30,44,46,47].

3.4 Analysis of Cycle Spinning

The apparent limitation of the cycle-spinning method in Algorithm 3.1 lies in its greedy nature. At each iteration, the algorithm simply makes a locally optimal step towards the minimization of

$$\mathscr{C}_{k_t}(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sigma^2 \phi_{\mathbf{w}}(\mathbf{W}_{k_t}\mathbf{x}), \tag{3.5}$$

instead of using the information available from all possible shifts. We are not aware of any prior analysis of the convergence properties of such a scheme. In Theorem 3.1 below, we establish the first convergence result for cycle spinning applied for solving linear inverse problems.

Theorem 3.1. *Define*

$$\mathscr{C}_{CS}(\mathbf{x}) \triangleq \frac{1}{K}\sum_{k=1}^{K} \mathscr{C}_k(\mathbf{x}), \tag{3.6}$$

where

$$\mathscr{C}_k(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 + \sigma^2 \phi_{\mathbf{w}}(\mathbf{W}_k\mathbf{x}). \tag{3.7}$$

Assume also a nonempty, convex, bounded, and closed subset $\mathcal{X} \subseteq \mathbb{R}^N$*. Set* $\gamma_t = 1/(L\sqrt{t})$*, where* L *is any constant such that* $L > \lambda_{\max}(\mathbf{H}^T\mathbf{H})$*. Let* $\widehat{\mathbf{x}}^0$ *be an arbitrary vector in* $\mathcal{X}$*, and* $\{\widehat{\mathbf{x}}^t\}_{t\in\mathbb{N}}$ *be the sequence generated by Algorithm 3.1 with an additional projection step onto* $\mathcal{X}$*. Then,*

$$\lim_{t\to\infty} \mathscr{C}_{CS}(\widehat{\mathbf{x}}^t) = \mathscr{C}^*_{CS} \quad \textit{where} \quad \mathscr{C}^*_{CS} = \min_{\mathbf{x}\in\mathcal{X}} \mathscr{C}_{CS}(\mathbf{x}). \tag{3.8}$$

Proof: See Section 3.8.

When $K = 1$, cycle spinning reduces to the standard ISTA, which is known to converge [21]. What our theorem proves is that, by iteratively cycling through K orthogonal wavelets, we are minimizing a cost function that regularizes the solution over all the shifts simultaneously, with the advantage that the underlying regularizer is truly shift-invariant. Note that the set $\mathcal{X}$ is a set of possible values for the signals $\mathbf{x}$. It can be selected as an arbitrary subset of $\mathbb{R}^N$ that is large enough to contain the signals of interest, and that is nonempty, convex, bounded, and closed.

3.5 Consistent Cycle Spinning

We now focus on a simpler problem of signal denoising with the objective of further boosting the performance of wavelet–domain MAP estimation. In particular, we introduce the concept of *consistent cycle spinning* (CCS), which can be used to cast the estimation with redundant transforms into the synthesis form that still preserves the Parseval's identity. Then, it becomes possible to denoise the signal with the MMSE estimator in (2.40). In order to validate MMSE performance of CCS experimentally, we present in Section 3.6.2 numerical evaluations on statistical signals with sparse derivatives.

We start by defining the transform matrix

$$\mathbf{T} \triangleq \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_K \end{bmatrix} \tag{3.9}$$

and its pseudo inverse

$$\mathbf{T}^{\dagger} \triangleq \frac{1}{K}\left[\mathbf{W}_1^T \dots \mathbf{W}_K^T\right] \tag{3.10}$$

that satisfy the following two properties of Parseval frames

$$\arg\min_{\mathbf{x}\in\mathbb{R}^N}\left\{\frac{1}{2}\|\mathbf{z}-\mathbf{T}\mathbf{x}\|_2^2\right\} = \mathbf{T}^{\dagger}\mathbf{z}, \quad (\text{for all } \mathbf{z}\in\mathbb{R}^{KN}) \tag{3.11}$$

and $\mathbf{T}^{\dagger}\mathbf{T} = \mathbf{I}$ [15]. Then, the original (noniterative) cycle spinning algorithm (3.3) can be written as

$$\widehat{\mathbf{x}} = \mathbf{T}^{\dagger}\widehat{\mathbf{w}} \quad \text{with} \quad \widehat{\mathbf{w}} = \text{prox}_{\phi_{\text{w}}}(\mathbf{T}\mathbf{y};\sigma^2). \tag{3.12}$$

In the context of signal reconstruction, (3.12) is often referred to as *synthesis* formulation, while the iterative version analysed in Theorem 3.1 is called the *analysis* formulation [48]. Synthesis and analysis formulations are equivalent when $\mathbf{T}$ is orthogonal (i.e. $K = 1$), but yield different solutions when $\mathbf{T}$ is overcomplete. The underlying reason for this dissagreement is in the Parseval's identity, which is lost for redundant transform matrices. The major advantage of the synthesis formulation is that in the orthogonal case the computation of the MMSE solution becomes simple (See Section 2.5.1). CCS refers to a technique that performs synthesis–type redundant wavelet-domain reconstruction by enforcing the norm equivalence as follows

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^{KN}}\left\{\frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + K\sigma^2\phi_{\mathbf{w}}(\mathbf{w})\right\} \text{ s.t. } \mathbf{T}\mathbf{T}^{\dagger}\mathbf{w} = \mathbf{w}, \tag{3.13}$$

where $\mathbf{u} = \mathbf{T}\mathbf{y}$. In the context of general overcomplete transforms, such formulation was studied by Elad *et al.* [49]. The factor K that appears in front of σ^2 is due to the relation

$$K\|\mathbf{x}\|_2^2 = \|\mathbf{T}\mathbf{x}\|_2^2, \tag{3.14}$$

which is true for any $\mathbf{x}\in\mathbb{R}^N$. Thus, CCS combines better approximation capabilities of redundant representations with wavelet–domain solutions that behave as if the transform were truly orthogonal. Note that under the consistency constraint in (3.13), the estimation reduces to a simple pointwise shrinkage of wavelet-coefficients. This observation hints at the use of shrinkage functions other than MAP. For example, one can use the shrinkage that computes MMSE solution to the denoising problem. In view of results by Gribonval in [40], usage of MMSE shrinkages corresponds to implicit penalization with the corresponding MMSE potential function $\phi_{\mathbf{w}}^{\text{MMSE}}$.

A practical optimization scheme for the constrained minimization problem (3.13) with possibly non-convex potential functions can be obtained by using an augmented-Lagrangian

Algorithm 3.2: Consistent Cycle Spinning (CCS) for Signal Denoising

```
input:  data $\mathbf{u} = \mathbf{T}\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        quadratic penalty $\tau > 0$, noise variance $\sigma^2$,
        transform matrix $\mathbf{T}$,
        and scalar shrinkage $\mathscr{T}$.
set:  $t \leftarrow 1$ and $\mu^0 \leftarrow \mathbf{0}$
repeat
    $\widehat{\mathbf{w}}^t = \mathscr{T}\left((\mathbf{u} + \tau\mathbf{T}\widehat{\mathbf{x}}^{t-1} + \mu^{t-1})/(1+\tau); K\sigma^2/(1+\tau)\right)$
    $\widehat{\mathbf{x}}^t = \mathbf{T}^\dagger\left(\widehat{\mathbf{w}}^t - \mu^{t-1}/\tau\right)$
    $\mu^t = \mu^{t-1} - \tau(\widehat{\mathbf{w}}^t - \mathbf{T}\widehat{\mathbf{x}}^t)$
    $t \leftarrow t + 1$
until stopping criterion
return  $\widehat{\mathbf{x}}^t$
```

approach [22]. The idea is to introduce the augmented-Lagrangian cost function

$$\mathscr{L}(\mathbf{w},\mathbf{x}) = \frac{1}{2}\|\mathbf{w}-\mathbf{u}\|_2^2 + K\sigma^2\phi_{\mathbf{w}}(\mathbf{w}) + \frac{\tau}{2}\|\mathbf{w}-\mathbf{T}\mathbf{x}\|_2^2 - \mu^T(\mathbf{w}-\mathbf{T}\mathbf{x}), \tag{3.15}$$

where $\tau > 0$ is the penalty parameter and $\mu \in \mathbb{R}^{KN}$ is the vector of Lagrange multipliers. The condition $\mathbf{w} = \mathbf{T}\mathbf{x}$ asserted by the penalty function constrains $\mathbf{w}$ to the column space of $\mathbf{T}$, which is equivalent to the consistency condition $\mathbf{w} = \mathbf{T}\mathbf{T}^\dagger\mathbf{w}$. Although the quadratic penalty term in $\mathscr{L}$ does not influence the final solution, it typically improves the convergence behaviour of the iterative optimization [50]. To solve the minimization, we alternate between solving the problem for $\mathbf{w}$ for $\mathbf{x}$ fixed and *vice versa*, which corresponds to the alternating direction method of multipliers (ADMM) algorithm [22]. Implementation of CCS is summarized in Algorithm 3.2.

3.6 Numerical Evaluation

We now present some numerical results illustrating the concepts discussed in this section. We divide experiments into two main parts. In the first, we illustrate the results of Theorem 3.1 by considering general linear inverse problems. In the second part,

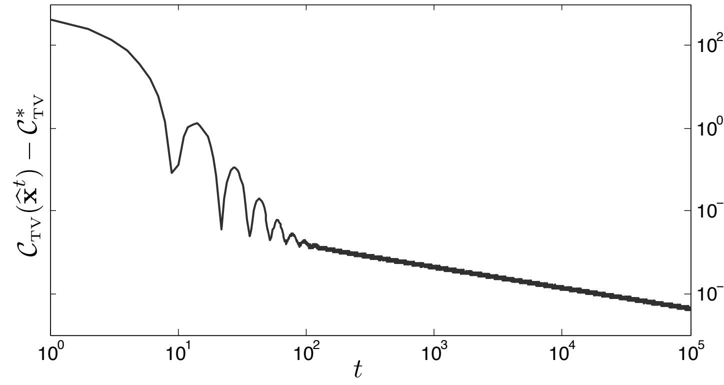


Figure 3.1: Estimation of a sparse signal from noisy measurements. We plot the gap $\left(\mathscr{C}_{\mathrm{TV}}(\widehat{\mathbf{x}}^t) - \mathscr{C}^*_{\mathrm{TV}}\right)$ against the iteration number t. The plot illustrates the convergence of cycle spinning to the minimizer of the cost function $\mathscr{C}_{\mathrm{TV}}$.

we highlight the MMSE denoising capability of CCS by considering a statistical signal model based on Lévy processes. Advantage of using the Lévy signal model is that we can then compare the results to the MMSE estimator computed by the message-passing algorithm developed in Chapter 4.

3.6.1 Convergence of Cycle Spinning

We illustrate Theorem 3.1 with two simple examples. In the first example, we consider the estimation of a piecewise constant signal of length $N = 128$ corrupted by AWGN corresponding to an input signal-to-noise ratio (SNR) of 30 dB. An interesting property of such signals is that they can be sparsified with the finite-difference operator, which justifies the use of TV regularization [51]. Since the TV regularizer corresponds to an ℓ_1-penalty applied to the finite differences of the signal, our theorem indicates that it can also be minimized with cycle spinning when **W** corresponds to the Haar-wavelet basis with one level of decomposition and a zero weight in the lowpass. In Figure 3.1, we plot the per-iteration gap $\left(\mathscr{C}_{\mathrm{TV}}(\widehat{\mathbf{x}}^t) - \mathscr{C}^*_{\mathrm{TV}}\right)$, where $\widehat{\mathbf{x}}^t$ is computed with cycle spinning and $\mathscr{C}_{\mathrm{TV}}$ is the TV-regularized least-squares cost. We set $\lambda = 0.05$ and, following our analysis, we set the step-size to $\gamma_t = 1/(4\sqrt{t})$. As expected, we observe that, as $t \rightarrow \infty$,

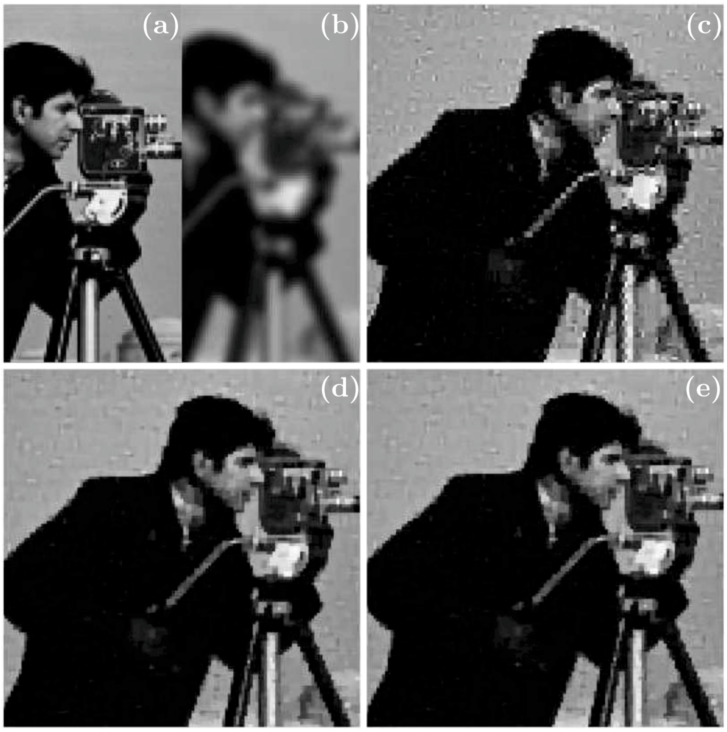


Figure 3.2: Reconstruction of *Cameraman* from blurry and noisy measurements. (a) original, (b) blurry, (c) standard wavelet-based ISTA (SNR = 19.10 dB), (d) reconstruction with TV (SNR = 21.58 dB), (e) wavelet-based reconstruction with cycle spinning (SNR = 21.58 dB).

we have that $\left(\mathscr{C}_{\mathrm{TV}}(\widehat{\mathbf{x}}^t) - \mathscr{C}^*_{\mathrm{TV}}\right) \rightarrow 0$. Moreover, we note that, for large t, the slope of $\left(\mathscr{C}_{\mathrm{TV}}(\widehat{\mathbf{x}}^t) - \mathscr{C}^*_{\mathrm{TV}}\right)$ in log-log domain tends to $-1/2$, which indicates the asymptotic rate of convergence $\mathcal{O}(1/\sqrt{t})$.

In the second example, we consider an image-deblurring problem where the *Cameraman* image of size 256 × 256 is blurred with a 7 × 7 Gaussian kernel of standard deviation 2 with the addition of AWGN of variance $\sigma^2 = 10^{-5}$. In Figure 3.2, we present the result of the reconstruction with three different methods: standard Haar-domain ℓ_1-regularization, anisotropic TV [52], and cycle spinning with 1D Haar-basis functions applied horizontally and vertically to imitate TV. The regularization parameters for the

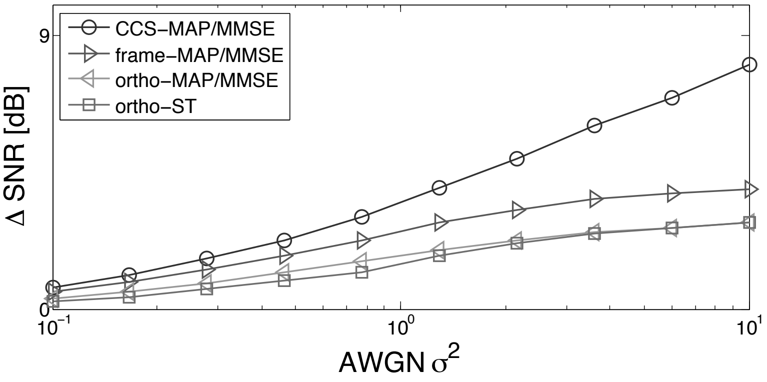


Figure 3.3: SNR improvement as a function of the level of noise for Brownian motion. The wavelet-denoising methods by reverse order of performance are: standard soft-thresholding (ortho-ST), optimal shrinkage in a wavelet basis (ortho-MAP/MMSE), shrinkage in a redundant system (frame-MAP/MMSE), and optimal shrinkage with consistent cycle spinning (CCS-MAP/MMSE).

standard wavelet approach and TV were optimized for the least-error performance. The regularization parameter of cycle spinning was set by rescaling the regularization parameter of TV according to $\lambda_{\mathrm{CS}} = \sqrt{2}K\lambda_{\mathrm{TV}}$, with $K = 4^4$. Therefore, we expect cycle spinning to again match the TV solution. It is clear from Figure 3.2 that cycle spinning outperforms the standard wavelet regularization (improvement of at least 2 dB). As expected, the solution obtained by cycle spinning exactly matches that of TV both visually and in terms of SNR.

3.6.2 Statistical Estimation with CCS

We experimentally evaluate CCS by considering two versions of the algorithm: CCS-MAP and CCS-MMSE that use MAP and MMSE shrinkage functions, respectively. We test the estimation performance of the algorithms on Lévy processes that will be discussed with much greater detail in Chapter 4. One can view Lévy processes as continuous-

---

[4] The goal is for cycle spinning to be minimizing exactly the same cost function as TV. Thus, the factor K is due to the number of shifts, while the factor $\sqrt{2}$ is due to the normalization of the Haar wavelets.

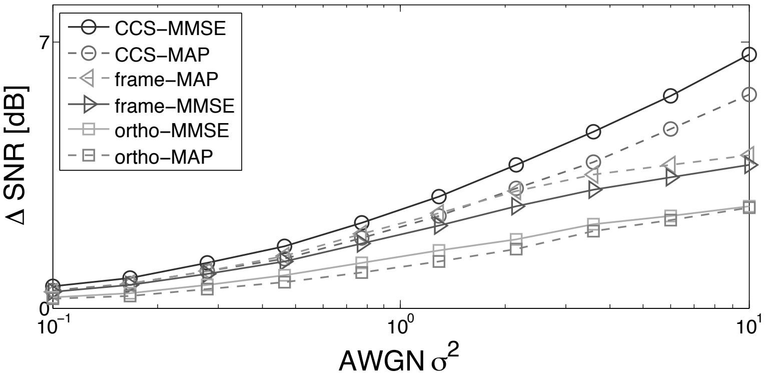


Figure 3.4: SNR improvement as a function of the level of noise for a Lévy process with Laplace-distributed increments. The wavelet-denoising methods by reverse order of performance are: ortho-MAP (equivalent to soft-thresholding with fixed λ), ortho-MMSE, frame-MMSE, frame-MAP, CCS-MAP, and CCS-MMSE. The results of CCS-MMSE are compatible with the ones of the reference MMSE estimator in Chapter 4.

time analogs of random walks. Their defining property is that they have independent and stationary increments [53, 54], which implies that the applications of finite-difference operator on samples of a Lévy process decouples it into a sequence of independent random variables. In the experiments, we consider the denoising of four types of Lévy processes of lengths $N = 2048$ in the Haar-wavelet domain. All the results were obtained by averaging 10 random realizations of the problem for various noise levels. The metric used for comparisons is the SNR improvement

$$\Delta\text{SNR} \triangleq 10\log_{10}\left(\frac{\|\mathbf{x}-\mathbf{y}\|_2^2}{\|\mathbf{x}-\widehat{\mathbf{x}}\|_2^2}\right) \tag{3.16}$$

The four statistical distributions for our signals are (see Figure 4.2 for an illustration of their realizations)

- **Gaussian:** This is the best-known example of a Lévy process. We set the distribution of increments to $\mathcal{N}(0,1)$.
- **Laplace:** We set the scale parameter to one.

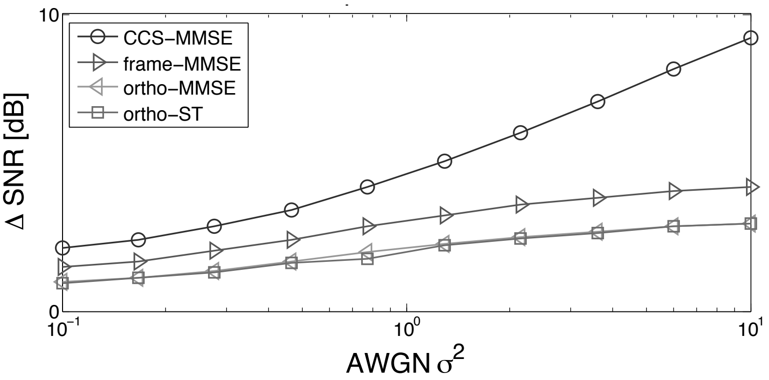


Figure 3.5: SNR improvement as a function of the level of noise for a compound Poisson process (piecewise-constant signal). The wavelet-denoising methods by reverse order of performance are: ortho-ST, ortho-MMSE, frame-MMSE, and CCS-MMSE. The results of CCS-MMSE are compatible with the ones of the reference MMSE estimator obtained using message passing in Chapter 4.

- **Compound-Poisson:** We consider the sparse scenarios and set the mass probability at 0 to 0.6. The size of the jumps follow the standard Gaussian distribution.
- **Cauchy:** We set the distribution of the increments to be Cauchy ($\alpha = 1$) with scale parameter $\rho = 1$.

The wavelet denoising methods compared are the standard orthogonal wavelet soft-thresholding (ortho-ST), statistically optimal shrinkages in the orthogonal wavelet domain (ortho-MAP/MMSE), traditional cycle spinning (3.12) with statistically optimal shrinkages (frame-MAP/MMSE), and statistically optimal shrinkages with CCS (CCS-MAP/MMSE). Note that for the Gaussian signals in Figure 3.3 MAP and MMSE estimators coincide resulting in the same shrinkage operation that is linear.

In Figure 3.4 we compare 6 statistical estimators for denoising Lévy processes with Laplace increments: ortho-MAP, ortho-MMSE, frame-MAP, frame-MMSE, CCS-MAP, and CCS-MMSE. We remark that for Haar basis and the Laplace prior, CCS-MAP reduces to the popular TV-denoising algorithm [41]. Figure 3.4 exposes several possible ways for improving the quality of the reconstructed signal. First, as expected, when the

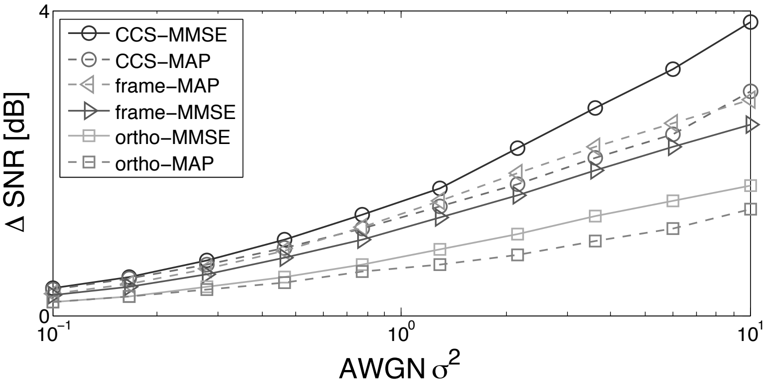


Figure 3.6: SNR improvement as a function of the level of noise for a Lévy flight with Cauchy-distributed increments. The wavelet-denoising methods by reverse order of performance are: ortho-MAP, ortho-MMSE, frame-MMSE, frame-MAP, CCS-MAP, and CCS-MMSE. The results of CCS-MMSE are compatible with the ones of the reference MMSE estimator obtained using message passing in Chapter 4.

transform is orthogonal MMSE outperforms MAP. We also see the boost in performance due to cycle spinning with both frame-MAP and frame-MMSE giving better results that ortho-MAP and ortho-MMSE. Due to redundancy, wavelet-domain MMSE estimator loses its optimality in space domain, which is corroborated by the relative performance of frame-MAP against frame-MMSE. The correct MMSE/MAP order, however, is re-established for CCS, where the strict norm equivalence between domains is enforced.

In Figure 3.5, we denoise piecewise-constant compound Poisson signals. In this case, we omit MAP estimators which correspond to trivial solutions due to the mass at 0. As before, we observe that model based ortho-MMSE improves over the standard wavelet soft-thresholding. Another significant boost is obtained by passing from orthogonal wavelets to cycle spinning. Finally, the best performance is obtained by enforce the norm equivalence via CCS.

In Figure 3.5, we consider the highly compressible Lévy signal with Cauchy increments [55]. We again see that CCS-MMSE yields the best SNR performance, while ortho-MAP yields the worse one.

Finally, it is important to note that the results of CCS-MMSE presented here are fully compatible with the message-passing MMSE estimator developed in Chapter 4. This implies that CCS-MMSE does indeed perform MMSE estimation for Lévy processes in AWGN.

3.7 Summary

We have considered cycle spinning as a simple tool for boosting the performance of wavelet-domain statistical estimators. We have established the convergence result of the iterative cycle-spinning technique for solving linear inverse problems. We have proved that the algorithm converges to the minimizer of a regularized least-squares cost function where the regularizer penalizes translation-invariant wavelet coefficients in the analysis form [48].

We have then introduced the CCS technique that performs estimation in the synthesis form. We empirically illustrated its application for performing MAP and MMSE denoising with redundant wavelets. Although, the current formulation of CCS is restricted to the problem of signal denoising, the obtained results are encouraging and show the potential of MMSE type estimators. We will see in Chapter 5 an alternative way of obtaining similar estimators for other types of linear inverse problems.

One can imagine numerous possible extensions of our results. The analysis of cycle spinning presented in this chapter was deterministic; an interesting avenue for future research would be to see if it also holds in the stochastic setting, where k_t would be generated randomly at each iteration. Our analysis indicates that the rate of convergence of cycle spinning is no worse than $\mathcal{O}(1/\sqrt{t})$. A possible direction of research is to search for a faster convergence rate by using various standard acceleration techniques for ISTA algorithms [21]. On the other hand, the empirical MMSE performance of CCS requires an in-depth investigation, which is also a topic of future research.

3.8 Appendix

3.8.1 Useful Facts from Convex Analysis

Before embarking on the actual proof of Theorem 3.1, it is convenient to summarize a few facts that will be used next.

A subgradient of a convex function f at $\mathbf{x}$ is any vector $\mathbf{g}$ that satisfies the inequality $f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{g}, \mathbf{y} - \mathbf{x} \rangle$, for all $\mathbf{y}$. When f is differentiable, the only possible choice for $\mathbf{g}$ is $\nabla f(\mathbf{x})$. The set of subgradients of f at $\mathbf{x}$ is the subdifferential of f at $\mathbf{x}$, denoted $\partial f(\mathbf{x})$. The condition that $\mathbf{g}$ be a subgradient of f at $\mathbf{x}$ can then be written $\mathbf{g} \in \partial f(\mathbf{x})$.

The proximal operator is defined as

$$\begin{aligned} \mathbf{x} &= \operatorname{prox}_{\phi}(\mathbf{z}; \gamma) \qquad (3.17) \\ &= \arg\min_{\mathbf{x} \in \mathscr{X}} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \gamma \phi(\mathbf{x}) \right\}, \end{aligned}$$

where $\mathscr{X} \subseteq \mathbb{R}^N$, $\gamma > 0$ and ϕ is a convex continuous function. The proximal operator is characterized by the following inclusion, for all $\mathbf{x}, \mathbf{z} \in \mathscr{X}$:

$$\mathbf{x} = \operatorname{prox}_{\phi}(\mathbf{z}; \gamma) \iff \mathbf{z} - \mathbf{x} \in \gamma \partial \phi(\mathbf{x}). \qquad (3.18)$$

We say that an operator $\mathrm{T} : \mathscr{X} \to \mathscr{X}$ is nonexpanding if, for all $\mathbf{x}, \mathbf{z} \in \mathscr{X}$, it satisfies $\|\mathrm{T}\mathbf{x} - \mathrm{T}\mathbf{z}\|_2 \leq \|\mathbf{x} - \mathbf{z}\|_2$. Note that the proximal operator is nonexpansive.

Next, we present the Browder-Göhde-Kirk's fixed-point theorem. It is a standard theorem in convex analysis (see [56, Theorem 4.19])

Theorem 3.2. *Let $\mathscr{X}$ be a nonempty bounded closed convex subset of $\mathbb{R}^N$ and let $\mathrm{T} : \mathscr{X} \to \mathscr{X}$ be a nonexpansive operator. Then, the operator T has a nonempty set of fixed points, i.e., $\operatorname{Fix}\mathrm{T} \neq \emptyset$.*

The Krasnoselskii–Mann theorem will also be of use (see [56, Theorem 5.14]).

Theorem 3.3. *Let $\mathscr{X}$ be a nonempty closed convex subset of $\mathbb{R}^N$, let $\mathrm{T} : \mathscr{X} \to \mathscr{X}$ be a nonexpansive operator such that $\operatorname{Fix}\mathrm{T} \neq \emptyset$, let $(\theta_t)_{t \in \mathbb{N}}$ be a sequence in $(0, 1)$ such that $\sum_{t \in \mathbb{N}} \theta_t(1 - \theta_t) = \infty$, and let $\mathbf{x}^0 \in \mathscr{X}$. Set $\widehat{\mathbf{x}}^t = \widehat{\mathbf{x}}^{t-1} + \theta_t(\mathrm{T}\widehat{\mathbf{x}}^{t-1} - \widehat{\mathbf{x}}^{t-1})$. Then, the sequence $\{\widehat{\mathbf{x}}^t\}_{t \in \mathbb{N}}$ converges to a point in $\operatorname{Fix}\mathrm{T}$.*

3.8.2 General Model

For the purpose of our analysis, we consider the cost function

$$\mathscr{C}_{\mathrm{cs}}(\mathbf{x}) = \frac{1}{K}\sum_{k=1}^{K}\mathscr{C}_k(\mathbf{x}) = \mathscr{D}(\mathbf{x}) + \frac{1}{K}\sum_{k=1}^{K}\phi_k(\mathbf{x}). \tag{3.19}$$

It is more general that the cost function of Theorem 3.1, which it includes as a special case. We shall also make the following assumptions:

- The feasible set $\mathscr{X} \subseteq \mathbb{R}^N$ is nonempty, convex, bounded, and closed. There exists $D > 0$ such that for all $\mathbf{x}, \mathbf{z} \in \mathscr{X}$, $\|\mathbf{x} - \mathbf{z}\|_2 \leq D$.
- $\mathscr{D} : \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and differentiable, with a Lipschitz-continuous gradient. There exists $L > 0$ such that, for all $\mathbf{x}, \mathbf{z} \in \mathscr{X}$, $\|\boldsymbol{\nabla}\mathscr{D}(\mathbf{x}) - \boldsymbol{\nabla}\mathscr{D}(\mathbf{z})\|_2 \leq L\|\mathbf{x} - \mathbf{z}\|_2$.
- For each k, ϕ_k is a continuous, convex function that is possibly nondifferentiable.
- The gradient of $\mathscr{D}$ and the subgradients of ϕ_k are bounded. There exists $G > 0$ such that, for all k and all $\mathbf{x} \in \mathscr{X}$, $\|\boldsymbol{\nabla}\mathscr{D}(\mathbf{x})\|_2 \leq G$ and $\|\partial_x\phi_k(\mathbf{x})\|_2 \leq G$.

Note that for $\phi_k(\mathbf{x}) = \|\mathbf{W}_k\mathbf{x}\|_1$, we have $\partial_x\|\mathbf{W}_k\mathbf{x}\|_1 = \{\mathbf{W}_k^T\mathbf{g} \mid \|\mathbf{g}\|_\infty \leq 1, \langle\mathbf{W}_k^T\mathbf{g}, \mathbf{x}\rangle = \|\mathbf{W}_k\mathbf{x}\|_1\}$, which indeed implies that the subgradients are bounded [57]. Then, the Algorithm 3.1 is a special case of the proximal-gradient algorithm summarized in Algorithm 3.3.

3.8.3 Main Technical Lemmas

The proof of Theorem 3.1 relies on two lemmas that we shall prove now.

Lemma 3.1. *For all* $t = 1, 2, \ldots$, *and for any* $\mathbf{x}^* \in \mathscr{X}$, *we have that*

$$\begin{aligned}&\mathscr{C}_{k_t}\left(\widehat{\mathbf{x}}^t\right) - \mathscr{C}_{k_t}\left(\mathbf{x}^*\right)\\ &\quad\leq \frac{1}{2\gamma_t}\left(\|\widehat{\mathbf{x}}^{t-1} - \mathbf{x}^*\|_2^2 - \|\widehat{\mathbf{x}}^t - \mathbf{x}^*\|_2^2\right) + 6\gamma_t G^2.\end{aligned} \tag{3.20}$$

Algorithm 3.3: Minimizes: $\mathscr{C}_{\text{cs}}(\mathbf{x}) = \mathscr{D}(\mathbf{x}) + (1/K)\sum_{k=1}^{K} \phi_k(\mathbf{x})$

```
input:  data $\mathbf{y}$, initial guess $\widehat{\mathbf{x}}^0$,
        steps $\{\gamma_t\}_{t\in\mathbb{N}}$ in $(0, 1/L]$ where $L$ is the Lipschitz constant of $\nabla\mathscr{D}$,
        and an efficient implementation of $\nabla\mathscr{D}$ and $\text{prox}_{\phi_k}$.
set:  $t \leftarrow 1$
repeat
    $k_t \leftarrow 1 + (t-1 \mod K)$                                  (choice of regularizer)
    $\mathbf{z}^t \leftarrow \widehat{\mathbf{x}}^{t-1} - \gamma_t \nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1})$                      (gradient step)
    $\widehat{\mathbf{x}}^t \leftarrow \text{prox}_{\phi_k}(\mathbf{z}^t; \gamma_t)$                              (shrinkage step)
    $t \leftarrow t + 1$
until stopping criterion
return  $\widehat{\mathbf{x}}^t$
```

Proof. The optimality condition of (3.18) implies that there must exist a vector $\mathbf{g}^t \in \partial_x \phi_{k_t}(\widehat{\mathbf{x}}^t)$ such that

$$\widehat{\mathbf{x}}^t = \widehat{\mathbf{x}}^{t-1} - \gamma_t \left(\nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t\right). \tag{3.21}$$

Then, we can write

$$\begin{aligned} \|\widehat{\mathbf{x}}^t - \mathbf{x}^*\|_2^2 &= \|\widehat{\mathbf{x}}^{t-1} - \gamma_t \left(\nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t\right) - \mathbf{x}^*\|_2^2 \\ &= \|\widehat{\mathbf{x}}^{t-1} - \mathbf{x}^*\|_2^2 - 2\gamma_t \langle \nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t, \widehat{\mathbf{x}}^{t-1} - \mathbf{x}^*\rangle \\ &\quad + \gamma_t^2 \|\nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t\|_2^2. \end{aligned} \tag{3.22}$$

By using the triangle inequality, we can bound the last term as

$$\|\nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t\|_2^2 \leq 4G^2. \tag{3.23}$$

To bound the second term we proceed in two steps. We first write that

$$\begin{aligned} \langle \nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}), \widehat{\mathbf{x}}^{t-1} - \mathbf{x}^*\rangle &\overset{\text{(a)}}{\geq} \mathscr{D}(\widehat{\mathbf{x}}^{t-1}) - \mathscr{D}(\mathbf{x}^*) \\ &\overset{\text{(b)}}{\geq} \mathscr{D}(\widehat{\mathbf{x}}^t) - \langle \nabla\mathscr{D}(\widehat{\mathbf{x}}^t), \widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1}\rangle - \mathscr{D}(\mathbf{x}^*) \\ &= \mathscr{D}(\widehat{\mathbf{x}}^t) - \mathscr{D}(\mathbf{x}^*) - \langle \nabla\mathscr{D}(\widehat{\mathbf{x}}^t), -\gamma_t \left(\nabla\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t\right)\rangle \\ &\overset{\text{(c)}}{\geq} \mathscr{D}(\widehat{\mathbf{x}}^t) - \mathscr{D}(\mathbf{x}^*) - 2\gamma_t G^2, \end{aligned} \tag{3.24}$$

where in (a) and (b) we used the convexity of $\mathscr{D}$, in (c) we used the Cauchy-Schwarz inequality and the bound on the gradient. Then, in a similar fashion, we can also write that

$$\begin{aligned} \langle \mathbf{g}^t, \widehat{\mathbf{x}}^{t-1} - \mathbf{x}^* \rangle &= \langle \mathbf{g}^t, \widehat{\mathbf{x}}^t - \mathbf{x}^* \rangle - \langle \mathbf{g}^t, \widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1} \rangle \\ &\overset{(a)}{\geq} \phi_{k_t}(\widehat{\mathbf{x}}^t) - \phi_{k_t}(\mathbf{x}^*) - \langle \mathbf{g}^t, -\gamma_t \left(\boldsymbol{\nabla}\mathscr{D}(\widehat{\mathbf{x}}^{t-1}) + \mathbf{g}^t \right) \rangle \\ &\overset{(b)}{\geq} \phi_{k_t}(\widehat{\mathbf{x}}^t) - \phi_{k_t}(\mathbf{x}^*) - 2\gamma_t G^2, \end{aligned} \tag{3.25}$$

where in (a) we used the convexity of ϕ_{k_t}, in (b) we used the Cauchy-Schwarz inequality followed with a bound on the gradient.

By plugging (3.23), (3.24), and (3.25) into (3.22), by reorganizing the terms, and by using the definition $\mathscr{C}_{k_t}(\mathbf{x}) = \mathscr{D}(\mathbf{x}) + \phi_{k_t}(\mathbf{x})$, we obtain the claim. ☐

Lemma 3.2. *With $\{\widehat{\mathbf{x}}^t\}_{t\in\mathbb{N}}$ in $\mathscr{X}$ and $\bar{\mathbf{x}} \in \mathscr{X}$, let $\widehat{\mathbf{x}}^t \to \bar{\mathbf{x}}$. Then,*

$$\lim_{n\to\infty} \left\{ \frac{1}{nK} \sum_{t=1}^{nK} \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) \right\} = \mathscr{C}_{\mathrm{CS}}(\bar{\mathbf{x}}). \tag{3.26}$$

Proof. We introduce the shorthand notation $\delta_t = \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{k_t}(\bar{\mathbf{x}})$. The convergence of $\widehat{\mathbf{x}}^t$ and the continuity of $\mathscr{C}_k$ imply that, for a given $\epsilon > 0$, there exists an m such that, for all $t > mK$, $|\delta_t| = |\mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{k_t}(\bar{\mathbf{x}})| < \epsilon/2$. Then,

$$\begin{aligned} \left| \frac{1}{nK} \sum_{t=1}^{nK} \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{\mathrm{CS}}(\bar{\mathbf{x}}) \right| &= \left| \frac{1}{nK} \sum_{t=1}^{nK} \left(\mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{k_t}(\bar{\mathbf{x}}) \right) \right| \\ &\leq \frac{1}{nK} \left| \sum_{t=1}^{mK} \delta_t \right| + \frac{1}{nK} \left| \sum_{t=mK+1}^{nK} \delta_t \right| \\ &\leq \frac{1}{nK} \sum_{t=1}^{mK} |\delta_t| + \frac{1}{nK} \sum_{t=mK+1}^{nK} |\delta_t| \\ &\leq \frac{m}{n} \left(\max_{t\in[1\ldots mK]} |\delta_t| \right) + \left(1 - \frac{m}{n}\right) \left(\max_{t\in[mK+1\ldots nK]} |\delta_t| \right). \end{aligned} \tag{3.27}$$

Now, considering $\tilde{n} \geq (2m/\epsilon)\big(\max_{t\in[1\ldots mK]}|\delta_t|\big)$ and realizing that the second term is bounded by $\epsilon/2$, we conclude that, for a given $\epsilon > 0$, there exists $\tilde{n}$ such that, for all $n > \tilde{n}$,

$$\left| \frac{1}{nK} \sum_{t=1}^{nK} \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{\mathrm{CS}}(\bar{\mathbf{x}}) \right| < \epsilon. \tag{3.28}$$

□

3.8.4 Proof of Theorem 3.1

Let $\mathbf{x}^*$ denote a minimizer of $\mathscr{C}_{\mathrm{CS}}$. We introduce the shorthand notation $\Delta_t = \|\widehat{\mathbf{x}}^t - \mathbf{x}^*\|_2^2$. By following an approach similar to [58], we sum the bound in Lemma 3.1

$$\begin{aligned}
\sum_{t=1}^{nK} \big(\mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{k_t}(\mathbf{x}^*)\big) &= \sum_{t=1}^{nK} \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - nK\mathscr{C}_{\mathrm{CS}}(\mathbf{x}^*) \\
&\leq \frac{1}{2}\sum_{t=1}^{nK} \frac{1}{\gamma_t}(\Delta_{t-1} - \Delta_t) + 6G^2 \sum_{t=1}^{nK} \gamma_t \\
&\leq \frac{1}{2\gamma_1}\Delta_0 + \frac{1}{2}\sum_{t=1}^{nK-1} \left(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t}\right)\Delta_t + 6G^2 \sum_{t=1}^{nK} \gamma_t \\
&\overset{\text{(a)}}{\leq} \frac{D^2}{2\gamma_1} + \frac{D^2}{2}\sum_{t=1}^{nK-1} \left(\frac{1}{\gamma_{t+1}} - \frac{1}{\gamma_t}\right) + 6G^2 \sum_{t=1}^{nK} \gamma_t \\
&\leq \frac{D^2}{2\gamma_{nK}} + 6G^2 \sum_{t=1}^{nK} \gamma_t,
\end{aligned} \tag{3.29}$$

where in (a) we used the boundedness of $\mathscr{X}$. Then, by choosing $\gamma_t = 1/(L\sqrt{t})$, by using the bound $\sum_{t=1}^{T} 1/\sqrt{t} \leq 2\sqrt{T}$, and by dividing the two sides of inequality by nK, we obtain

$$\frac{1}{nK}\sum_{t=1}^{nK} \mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t) - \mathscr{C}_{\mathrm{CS}}(\mathbf{x}^*) \leq \frac{C}{\sqrt{n}}, \tag{3.30}$$

where the constant C is given by

$$C = \frac{LD^2}{2\sqrt{K}} + \frac{12G^2}{L\sqrt{K}}. \tag{3.31}$$

To complete the proof, we argue that the sequence $\{\widehat{\mathbf{x}}^t\}_{t\in\mathbb{N}}$ converges to a fixed point in $\mathscr{X}$. On one hand, note that, due to the nonexpansiveness of $\mathrm{prox}_{\phi_{k_t}}(\cdot;\gamma_t)$ and the Lipschitz property of $\mathscr{D}$, the operator $\mathrm{T}_k : \mathscr{X} \to \mathscr{X}$,

$$\mathrm{T}_k\mathbf{x} = \mathrm{prox}_{\phi_k}(\mathbf{x} - \gamma\boldsymbol{\nabla}\mathscr{D}(\mathbf{x});\gamma) \tag{3.32}$$

is nonexpanding for any $\gamma \in (0, 1/L]$. Therefore, the composition $\mathrm{T} = \mathrm{T}_K \cdots \mathrm{T}_1$ is also nonexpanding. Then, from Theorem 3.2, we know that there exists at least one fixed-point of T in $\mathscr{X}$. On the other hand, for our choice $\gamma_t = 1/(L\sqrt{t})$, Theorem 3.3 implies that the subsequence generated via $\widehat{\mathbf{x}}^{Kt} = \mathrm{T}\widehat{\mathbf{x}}^{K(t-1)}$ converges to the fixed-point $\bar{\mathbf{x}} \in \mathrm{Fix}\,\mathrm{T}$. Since, $\gamma_t \to 0$, we have that $\left(\widehat{\mathbf{x}}^{t-1} - \mathrm{T}_{k_t}\widehat{\mathbf{x}}^{t-1}\right) \to 0$ and conclude that $\widehat{\mathbf{x}}^t \to \bar{\mathbf{x}}$. Finally, this allows us to show that

$$\begin{aligned} 0 &\overset{(a)}{\leq} \mathscr{C}_{\mathrm{CS}}(\bar{\mathbf{x}}) - \mathscr{C}_{\mathrm{CS}}(\mathbf{x}^*) \\ &\overset{(b)}{=} \lim_{n\to\infty}\left\{\frac{1}{nK}\sum_{t=1}^{nK}\mathscr{C}_{k_t}(\widehat{\mathbf{x}}^t)\right\} - \mathscr{C}^*_{\mathrm{CS}} \overset{(c)}{\leq} 0, \end{aligned} \tag{3.33}$$

where (a) comes from the optimality of $\mathbf{x}^*$, (b) from Lemma 3.2, and (c) from the upper bound (3.30).

Chapter 4

Optimal Estimators for Denoising Lévy Processes

4.1 Overview

In this chapter, we investigate a stochastic signal-processing framework for signals with sparse derivatives. The signal model is based on *Lévy processes* that include the well-known Brownian motion and piecewise-constant Poisson process; moreover, the Lévy family also contains other interesting members exhibiting heavy-tail statistics that fulfill the requirements of compressibility. We use our simple generative model to develop and test various standard reconstruction algorithms. For example, we will see that the MAP estimator for the Lévy process with Laplace increments coincides with total-variation (TV) regularization.

In order to concentrate on the important issue of the dependancy between various reconstruction algorithms and our prior probability model, we restrict ourselves to the problem of signal denoising. There are two main advantages of working in this simplified regime: (i) the estimated solutions do not depend on the type of measurement model $\mathbf{H}$; (ii) it becomes possible to develop a tractable computational method for evaluating the MMSE estimator. In fact, one of our major contributions here is a novel implementation of the MMSE estimator based on the belief propagation algorithm performed in the Fourier domain. Our algorithm takes advantage of the fact that the joint statistics of general Lévy processes are much easier to describe by their characteristic function, as the probability densities do not always admit closed form expressions. We then use our new estimator as a benchmark to compare the performance of existing algorithms for optimal recovery of signals with sparse gradients[1].

4.2 Introduction

We restrict our attention to the problem of estimating a signal **x** from a noisy vector $\mathbf{y} = \mathbf{x} + \mathbf{e}$ where the components of **e** are independent and distributed with a known probability distribution. It is clear that if we suppose that the components of the vector **x** are also independent, then the estimation problem becomes separable and reduces to N scalar estimation problems. In practice, however, due to correlations between the components of **x**, simple pointwise techniques are suboptimal and more refined methods often perform significantly better. Here, we consider the problem of estimating signals that have sparse derivatives. We take a continuous-domain perspective and propose Lévy processes [15, 53, 54, 59] as a natural approach to model such signals.

The fundamental defining property of a Lévy process is that it has independent and stationary increments. Therefore, the application of a finite-difference operator on samples of a Lévy process decouples it into a sequence of independent random variables. Interestingly, the class of Lévy processes is in one-to-one correspondence with the class of infinitely divisible distributions. Such distributions typically exhibit a heavy-tail behavior that has recently been proven to fulfill the requirements of compressibility [55, 60]. Therefore, Lévy processes can be considered as the archetype of sparse stochastic signals [15].

Many recent algorithms for the recovery of sparse signals can be interpreted as MAP estimators relying on some specific priors. From this Bayesian perspective, state-of-the-art methods based on gradient regularizers, such as TV [35] minimization, implicitly assume the signals to be sampled instances of Lévy processes [61]. Our aim is to investigate MMSE estimator for Lévy processes. The estimator provides a lower-bound on the mean-squared error (MSE) for the problem of recovery of signals with sparse derivatives. Unfortunately, due to high-dimensional integration, MMSE estimators are computationally intractable for general signals. By considering the Lévy signal model, we propose a novel method for computing the MMSE estimator based on the belief-propagation (BP) algorithm on cycle-free factor graphs [62–64].

---

[1] This chapter is based on our paper [51]

The main contributions in this chapter are as follows:

- Bayesian formulation of the signal-recovery problem under the Lévy hypothesis for a general "signal+noise" measurement model. With this formulation, we are able to derive an equivalence between MAP estimators for Lévy processes and some existing algorithms for the recovery of sparse signals.

- Characterization of the MSE optimal solution and the determination of performance bounds. We show that the MMSE estimator can be computed directly with the BP algorithm. The algorithm also obtains the marginals of the posterior distribution, which allows us to estimate the MSE of the reconstruction and to provide confidence intervals.

- Development of a novel frequency-domain message-passing algorithm specifically tailored to the MMSE estimation of Lévy processes. Some of the sparsest priors considered here do not have closed-form probability density functions. Indeed, they are represented in terms of their characteristic function obtained by the Lévy-Khintchine theorem [53,54]. The frequency-domain algorithm allows us to use the characteristic function directly without any numerical inversion.

- Experimental evaluation and comparison with standard solutions such as LMMSE, ℓ_1–minimization, and ℓ_p–relaxation [65]. In particular, the availability of MMSE allows us to benchmark these estimators on signals with desired properties such as sparsity.

4.3 Signal and Measurement Model

In this section, we describe the signal model summarized in Figure 4.1. We first give a powerful, yet simple continuous-domain stochastic formulation of the signal. The one-to-one mapping between our model and the extended family of infinitely divisible

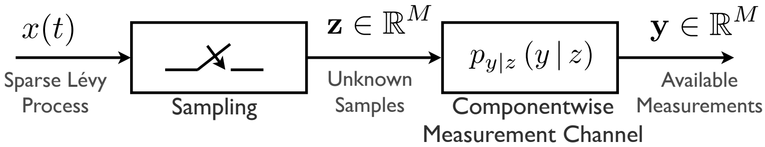

$x(t)$
Sparse Lévy Process
Sampling
$\mathbf{z} \in \mathbb{R}^M$
Unknown Samples
$p_{y|z}(y \mid z)$
Componentwise Measurement Channel
$\mathbf{y} \in \mathbb{R}^M$
Available Measurements

Figure 4.1: Signal model considered in this work. The continuous-domain Lévy process $x(t)$ is sampled, and the resulting vector $\mathbf{z} \in \mathbb{R}^M$ is passed through a separable measurement channel $p_{y|\mathbf{z}}$ to yield $\mathbf{y} \in \mathbb{R}^M$. We investigate the estimation of interpolated vectors $\mathbf{x} \in \mathbb{R}^N, N \geq M$, from the noisy measurements $\mathbf{y}$.

distributions is discussed. We finally describe the measurement model and provide examples of typical measurement channels.

4.3.1 Lévy Processes

Stochastic processes are often used to model random signals, the Brownian motion and the Poisson process being the two most common examples. Lévy processes—often seen as analogues of random walks in continuous time—extend those two processes to a larger family of distributions. They represent a fundamental and well-studied class of stochastic processes [53, 54]. Let $\{x(t) : t \geq 0\}$ be a continuous-time stochastic process. It is called a Lévy process if

1. $x(0) = 0$ almost surely;
2. for each $N \in \mathbb{N}$ and $0 \leq t_1 < t_2 < \cdots < t_N < \infty$ the random variables $\{x(t_{n+1}) - x(t_n) : 1 \leq n \leq N-1\}$ are independent;
3. for each $0 \leq t_1 < t_2 < \infty$, the random variable $(x(t_2) - x(t_1))$ is equal in distribution to $x(t_2 - t_1)$;
4. for all $\epsilon > 0$ and for all $t_1 \geq 0$

$$\lim_{t_2 \to t_1} \mathrm{Prob}(|x(t_2) - x(t_1)| > \epsilon) = 0.$$

Together, Properties 2 and 3 are commonly referred to as the *stationary-independent-increments* property, while Property 4 is called the *stochastic continuity*.

One of the most powerful results concerning Lévy processes is that they are in one-to-one correspondence with the class of *infinitely divisible* probability distributions. The random variable x is said to be infinitely divisible if, for any positive $n \in \mathbb{N}$, there exist i.i.d. random variables $y^{(1)}, \ldots, y^{(n)}$ such that

$$x \overset{d}{=} y^{(1)} + \cdots + y^{(n)}.$$

In other words, it must be possible to express the pdf p_x as the n-th convolution power of p_y. In fact, it is easy to show that the pdf of the increment $u_t = x(t+s) - x(s)$ of length t of any Lévy process is infinitely divisible

$$u_t \overset{d}{=} x(t) \overset{d}{=} u_{t/n}^{(1)} + \cdots + u_{t/n}^{(n)},$$

where each

$$u_{t/n}^{(k)} = x\left(\frac{kt}{n}\right) - x\left(\frac{(k-1)t}{n}\right).$$

The increments $u_{t/n}^{(k)}$ are of length t/n and are i.i.d. by the stationary-independent-increments property. Conversely, it has also been proved that there is a Lévy process for each infinitely divisible probability distribution [53].

The fundamental *Lévy-Khintchine formula* provides the characteristic function of *all* infinitely divisible distributions: p_u is an infinitely divisible probability distribution if and only if its characteristic function can be written as

$$\hat{p}_u(\omega) = \mathbb{E}\left[\mathrm{e}^{\mathrm{j}\omega u}\right] = \exp\left(\mathrm{j}a\omega - \frac{1}{2}b\omega^2 + \int_{\mathbb{R}\setminus\{0\}} \left(\mathrm{e}^{\mathrm{j}\omega z} - 1 - \mathrm{j}z\omega\mathbb{I}_{|z|<1}(z)\right) v(z)\,\mathrm{d}z\right),$$

where $a \in \mathbb{R}$, $b \geq 0$, and where $\mathbb{I}_{|z|<1}$ is an indicator function. The function $v \geq 0$ is the Lévy density satisfying

$$\int_{\mathbb{R}\setminus\{0\}} \min\left(1, z^2\right) v(z)\,\mathrm{d}z < \infty.$$

The representation of $\hat{p}_u$ by a triplet $(a, b, v(\cdot))$ is unique. Here, we limit our attention to even-symmetric pdfs $p_u(x) = p_u(-x)$ which results in the simplified Lévy-Khintchine

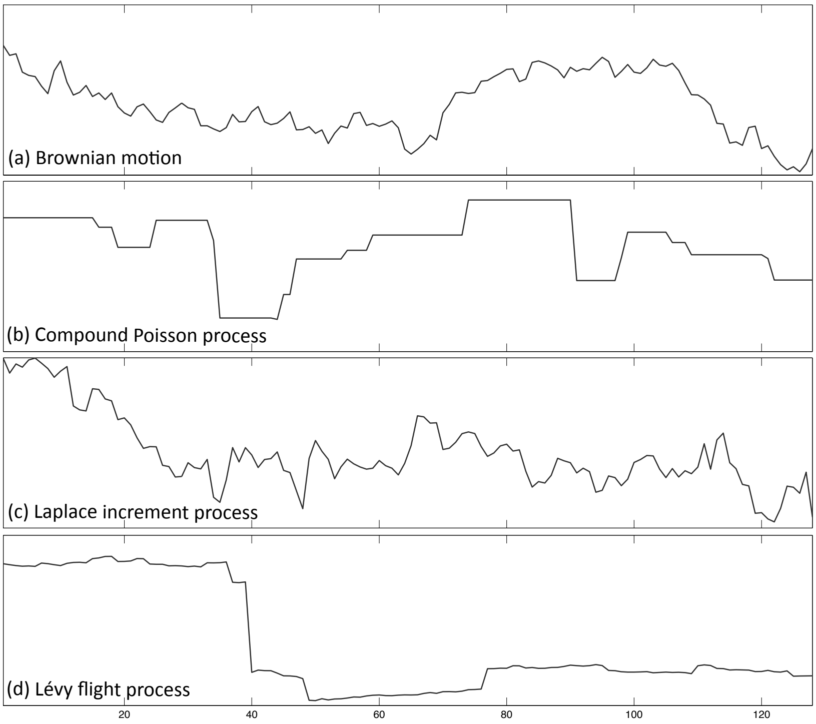


Figure 4.2: Sample paths of Lévy processes discussed in this chapter.

formula

$$\hat{p}_u(\omega) = \exp\Big(-\tfrac{1}{2}b\omega^2 - \int_{\mathbb{R}\setminus\{0\}} (1-\cos(\omega z))\,v(z)\,\mathrm{d}z\Big). \tag{4.1}$$

4.3.2 Examples

We now give examples of a few Lévy processes that are particularly interesting for us. Sample paths of these processes are shown in Figure 4.2. Without loss of generality, we assume an increment $u = x(s) - x(s-1)$ for some fixed $s \geq 1$.

1. **Brownian Motion:** By setting $a = 0$ and choosing the Lévy density $v(z) = 0$, we obtain the familiar Brownian motion that has stationary independent increments

characterized by

$$\hat{p}_u(\omega) = \mathrm{e}^{-\frac{1}{2}b\omega^2}, \tag{4.2}$$

with $b \geq 0$. This implies that the increments of the resulting Lévy process are Gaussian random variables with mean 0 and variance b, which corresponds to $u \sim \mathcal{N}(0, b)$. We illustrate in Figure 4.2(a) a single realization of a Brownian motion.

2. **Compound Poisson Process:** Let $\{z_k : k \in \mathbb{Z}_+\}$ be a sequence of i.i.d. random variables with distribution p_z and let $n(t) \sim \text{Poisson}(\lambda)$ be a Poisson process of intensity $\lambda > 0$ that does not depend on any z_k. The compound Poisson process y is then defined as

$$y(t) = \sum_{k=1}^{n(t)} z_k,$$

for each $t \geq 0$. This is a Lévy process obtained by setting the parameter triplet to $(0, 0, v(z) = \lambda p_z(z))$, which results in the characterization of increments

$$\hat{p}_u(\omega) = \mathrm{e}^{\lambda(\hat{p}_z(\omega)-1)}, \tag{4.3}$$

where $\hat{p}_z$ is the Fourier transform of p_z. On finite intervals, the sample paths of the process are piecewise-constant (Figure 4.2(b)), while the size of the jumps is determined by p_z [54]. Compound Poisson processes are piecewise-constant signals for which TV-like estimation algorithms are well suited [66]. The parameter λ controls the sparsity of the signal; it represents the rate of discontinuities.

3. **Laplace Increment Process:** The Lévy process with Laplace-distributed increment u is obtained by setting the parameter triplet to $\left(0, 0, v(z) = \mathrm{e}^{-\lambda|z|}/|z|\right)$, which results in

$$\hat{p}_u(\omega) = \frac{\lambda^2}{\lambda^2 + \omega^2}, \tag{4.4}$$

where $\lambda > 0$ is the scale parameter of the Laplace distribution. To obtain the characteristic function (4.4), we remark that

$$\begin{aligned} \log(\hat{p}_u(\omega)) &= \int_{\mathbb{R}\setminus\{0\}} \left(\mathrm{e}^{\mathrm{j}\omega z} - 1\right) \frac{\mathrm{e}^{-\lambda|z|}}{|z|}\, \mathrm{d}z \\ &= 2\int_0^{\infty} (\cos(\omega z) - 1)\frac{\mathrm{e}^{-\lambda z}}{z}\, \mathrm{d}z. \end{aligned}$$

Then, by differentiation with respect to ω and integrating back using the condition $\hat{p}_u(0) = 1$, we obtain (4.4). The corresponding pdf is

$$p_u(u) = \frac{\lambda}{2} \mathrm{e}^{-\lambda|u|}. \tag{4.5}$$

An interesting observation is that the Bayesian MAP interpretation of the TV regularization method with a first-order finite-differences operator inherently assumes the underlying signal to be a Lévy process with Laplace increments. We give in Figure 4.2(c) an illustration of such a process.

4. **Lévy-Flight Process:** Stable random variables are such that a linear combination of two independent ones results in a third stable random variable [53]. In the symmetric case, they are often referred to as symmetric α-stable random variables and written as $u \sim S\alpha S$, where $0 < \alpha < 2$ is the stability parameter. It is possible to generate a Lévy process with α-stable increments by setting $\big(0, 0, v(z) = c_\alpha/|z|^{1+\alpha}\big)$, which results in

$$\hat{p}_u(\omega) = \mathrm{e}^{-\rho|\omega|^\alpha}, \tag{4.6}$$

with $\rho > 0$ and $0 < \alpha < 2$. Such distributions are heavy-tailed and are known to result in highly compressible sequences [55]. A sample signal generated from a Cauchy-increment Lévy flight, which corresponds to the α-stable process with $\alpha = 1$, is illustrated in Figure 4.2(d).

4.3.3 Innovation Modeling

Recently, an alternative system-theoretic formulation of Lévy processes was proposed in the context of the general theory of sparse stochastic processes [15]. The authors specify the Lévy process $\{x(t) : t \geq 0\}$ as the solution of the stochastic differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t} x(t) = w(t), \tag{4.7}$$

where the differentiation is interpreted in the weak sense of distributions. The process w is a non-Gaussian white noise referred to as a *continuous-time innovation process*. According to the formalism developed in [15], the Lévy process is then generated by integrating the white noise according to

$$x(t) = \int_0^t w(t')\,\mathrm{d}t', \tag{4.8}$$

which provides a convenient linear-system interpretation. The only delicate aspect of this interpretation is that the white noise must be considered as a tempered distribution, since it is too rough to admit a classical interpretation as a function of t. The result confirms that, for all positive $k \in \mathbb{Z}_+$, the quantities

$$\begin{aligned} u_k &= x(k) - x(k-1) = \mathrm{D_d} x(k) \\ &= \int_{k-1}^{k} w(t)\,\mathrm{d}t = \langle \mathrm{rect}\left(\cdot - k + \tfrac{1}{2}\right), w(\cdot)\rangle \end{aligned} \tag{4.9}$$

are i.i.d. random variables that can be seen as discrete innovations [67]. The symbol $\langle \cdot, \cdot \rangle$ denotes an inner product between two functions, $\mathrm{D_d}$ is the finite-difference operator, and rect is the rectangular function, which is 1 inside the interval $[-1/2, 1/2]$ and zero outside. The fundamental observation is that the increment is obtained by applying the discrete version of the derivative to x, in an attempt to emulate (4.7) using discrete means only.

4.3.4 Measurement Model

Consider the measurement model illustrated in Figure 4.1. The vector $\mathbf{z} \in \mathbb{R}^M$ contains the uniformly sampled values of x

$$z_m = x(m\Delta_s), \quad (m = 1, \ldots, M) \tag{4.10}$$

where $\Delta_s > 0$ is the sampling interval. The components of $\mathbf{y}$ are generated by a separable measurement channel given by the conditional probability distribution

$$p_{\mathbf{y}|\mathbf{z}}(\mathbf{y} \,|\, \mathbf{z}) = \prod_{m=1}^{M} p_{y|z}(y_m \,|\, z_m). \tag{4.11}$$

The measurement channel models distortions affecting the signal during the acquisition process. This chapter addresses the computation of the estimator $\widehat{\mathbf{x}}$ of the vector $\mathbf{x} \in \mathbb{R}^N$

on some uniform grid

$$x_n = x(n\Delta_e), \quad (n = 1, \dots, N) \tag{4.12}$$

where $\Delta_e > 0$ is the interpolation interval. We wish to minimize the squared-error of the reconstruction in the situations when $\Delta_s = m_s \Delta_e$ for some positive $m_s \in \mathbb{N}$. This implies that in general $N \geq M$. In other words, we seek more estimates than there are samples. The special case $N = M$ reduces the problem to signal denoising. From now on, we assume $\Delta_s(M-1) = \Delta_e(N-1)$ and set $\Delta_e = 1$ to simplify the expressions. In particular, this implies that for any $m_s = \Delta_s / \Delta_e$ we have $z_m = x_{m_s(m-1)+1}$ for all $m \in [1 \dots m]$.

The generality of the measurement channel allows us to handle both signal–dependent and –independent distortions. Some common noise models encountered in practice are

1. **Additive White Gaussian Noise:** The measurements in the popular AWGN noise model are given by $\mathbf{y} = \mathbf{z} + \mathbf{e}$, where $\mathbf{e} \in \mathbb{R}^N$ is a signal-independent Gaussian vector with i.i.d components $e_n = y_n - z_n \sim \mathcal{N}(0, \sigma^2)$. The transitional probability distribution then reduces to

$$p_{y|z}(y \,|\, z) = \mathcal{G}(y - z; \sigma^2). \tag{4.13}$$

2. **Scalar Quantization:** Another common source of signal distortion is the analog-to-digital converter (ADC). When the conversion corresponds to a simple mapping of the analog voltage input to some uncoded digital output, it can be modeled as standard AWGN followed by a lossy mapping $\mathrm{Q} : \mathbb{R} \to \mathcal{C}$. The nonlinear function Q is often called a K-level scalar quantizer [68]. It maps the K-partitions of the real line $\left\{\mathrm{Q}^{-1}(c_i) : i = 1, \dots, K\right\} \subseteq \mathbb{R}$ into the set of discrete output levels $\mathcal{C} = \{c_i : i = 1, \dots, K\}$. This channel is signal-dependent. It is described in terms of the transitional probability distribution

$$p_{y|z}(y \,|\, z) = \int_{\mathrm{Q}^{-1}(y)} \mathcal{G}(z' - z; \sigma^2)\, \mathrm{d}z', \tag{4.14}$$

where $\mathrm{Q}^{-1}(y) = \{z \in \mathbb{R} : \mathrm{Q}(z) = y\}$ denotes a single partition.

4.4 Bayesian Formulation

We now specify explicitly the class of problems we wish to solve and identify corresponding statistical estimators. Consider the vector $\mathbf{u} \in \mathbb{R}^N$ obtained by applying the finite-difference matrix $\mathbf{D}$ to $\mathbf{x}$, whose components are given in (4.12). Then, from the stationary independent increments property of Lévy processes the components

$$u_n = [\mathbf{D}\mathbf{x}]_n = x_n - x_{n-1}, \tag{4.15}$$

of the vector $\mathbf{u}$ are realizations of i.i.d. random variables characterized by the simplified Lévy-Khintchine formula (4.1). Note that, from the definition of the Lévy process we have $x_0 = 0$. We construct the conditional probability distribution for the signal $\mathbf{x}$ given the measurements $\mathbf{y}$ as

$$\begin{aligned} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}\,|\,\mathbf{y}) &\propto p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}\,|\,\mathbf{x})\, p_{\mathbf{x}}(\mathbf{x}) \\ &\propto \prod_{m=1}^{M} p_{y|z}(y_m\,|\,z_m) \prod_{n=1}^{N} p_u([\mathbf{D}\mathbf{x}]_n), \end{aligned} \tag{4.16}$$

where we use $\propto$ to denote identity after normalization to unity. The distribution of the whitened elements p_u is, in principle, obtained by taking the inverse Fourier transform $p_u(u) = \mathscr{F}^{-1}\{\hat{p}_u\}(u)$; however, it does not necessarily admit a closed-form formula. The posterior distribution (4.16) of the signal provides a complete statistical characterization of the problem. In particular, the MAP and MMSE estimators of $\mathbf{x}$ are specified by

$$\widehat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\max}\left\{p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}\,|\,\mathbf{y})\right\} \tag{4.17a}$$

$$\widehat{\mathbf{x}}_{\text{MMSE}} = \mathbb{E}[\mathbf{x}\,|\,\mathbf{y}]. \tag{4.17b}$$

4.4.1 MAP Estimation

An estimation based on the minimization of some cost functional is a popular way of obtaining the MAP estimator $\widehat{\mathbf{x}}_{\text{MAP}}$. The availability of efficient numerical methods for convex and nonconvex optimization partially explain the success of such methods [36, 38, 66, 69]. The MAP estimator in (4.17a) can be reformulated as

$$\begin{aligned}\widehat{\mathbf{x}}_{\text{MAP}} &= \arg\max_{\mathbf{x}\in\mathbb{R}^N} \left\{p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}\,|\,\mathbf{y})\right\} \\ &= \arg\min_{\mathbf{x}\in\mathbb{R}^N} \{\mathscr{D}(\mathbf{x}) + \phi_{\mathbf{u}}(\mathbf{D}\mathbf{x})\},\end{aligned} \tag{4.18a}$$

where

$$\mathscr{D}(\mathbf{x}) = -\sum_{m=1}^{M} \log\left(p_{y|z}(y_m\,|\,z_m)\right), \tag{4.18b}$$

$$\phi_{\mathbf{u}}(\mathbf{D}\mathbf{x}) = -\sum_{n=1}^{N} \log\left(p_u([\mathbf{D}\mathbf{x}]_n)\right). \tag{4.18c}$$

The term $\mathscr{D}$ is the data term and $\phi_{\mathbf{u}}$ the regularization term.

In the AWGN, model the MAP estimation reduces to the popular regularized least-squares minimization problem

$$\widehat{\mathbf{x}}_{\text{MAP}} = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \left\{\frac{1}{2}\,\|\mathbf{y}-\mathbf{z}\|_2^2 + \sigma^2 \sum_{n=1}^{N} \phi_{\mathbf{u}}([\mathbf{D}\mathbf{x}]_n)\right\}, \tag{4.19}$$

where $\mathbf{z} \in \mathbb{R}^m$ is given in (4.10).

The estimator in (4.19) clearly illustrates the connections between the standard variational methods and our stochastic model. In particular, in the framework of the Lévy process, the Brownian motion yields the classical Tikhonov regularizer. The Lévy process with Laplace increments provides the ℓ_1–based TV regularizer. Finally, the Lévy-flight process results in a log–based regularizer that is linked to the limit case of the ℓ_p relaxation as p tends to zero [65, 70]. Such regularizers have been shown to be effective in several problems of the recovery of sparse signals [38, 66]. In [41] the authors have proposed an efficient method for solving the regularized-least-squares–based MAP denoising of Lévy processes. We also point out that the MAP estimation of compound Poisson processes yields a trivial solution due to a point mass at zero.

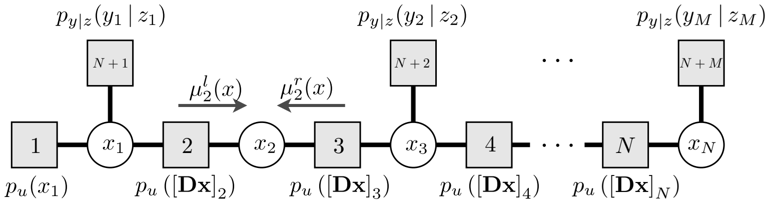


Figure 4.3: Factor-graph representation of the posterior distribution (4.16) with $m_s = 2$. In the graph, square factor nodes represent the probability densities and circled variable nodes represent the unknowns. The functions μ_2^l and μ_2^r represent beliefs at the variable node 2.

4.5 Message-Passing Estimation

4.5.1 Exact Formulation

In this section, we specify the MMSE estimator $\widehat{\mathbf{x}}_{\text{MMSE}}$ in (4.17b) for the signals under the Lévy-process model. Unfortunately, due to the high-dimensionality of the integral, this estimation is intractable in the direct form. We propose to use the sum-product belief-propagation (BP) [62] method to efficiently approximate the marginalization of the posterior (4.16), whose direct computation is intractable otherwise. The BP–based message-passing methods have successfully been used in numerous inference problems in statistical physics, computer vision, channel coding, and signal processing [24, 62–64, 71–75].

In order to apply the BP, we construct the bipartite factor-graph $G = (V, F, E)$, structured according to the posterior distribution in (4.16). We illustrate in Figure 4.3 an example of a factor-graph for $m_s = 2$. The graph consists of two sets of nodes, the variable nodes $V = \{1, \ldots, N\}$ (circles), the factor nodes $F = \{1, \ldots, N + M\}$ (squares), and a set of edges E linking variables to the factors they participate in. To introduce the BP algorithm, we define the functions μ_n^l and μ_n^r, which denote the messages exchanged along the edges of the graph. These messages—often referred to as *beliefs*—are in fact pdfs representing the desirable state of the variable node n. We also define for all

$n \in (1 \dots N)$ and $j = 1 + (n-1)/m_s$ the function

$$\eta_n(x) = \begin{cases} p_{y|z}(y_j \,|\, x), & \text{when } j \in \mathbb{N} \\ 1, & \text{otherwise.} \end{cases} \tag{4.20}$$

Whenever the component x_n has a corresponding measurement, the function η_n is equivalent to the probability distribution of the noise. Otherwise, η_n is equivalent to the constant function.

Given the measurements $\mathbf{y} \in \mathbb{R}^M$ and the functions η_n and p_u, the steps of the BP estimation are

1. **Initialization:** Set

$$\mu_1^l(x) = p_u(x), \tag{4.21a}$$

$$\mu_N^r(x) = 1. \tag{4.21b}$$

2. **Message Updates:** For $n = 1, \dots, N-1$, compute

$$\mu_{n+1}^l(x) \propto \int_{\mathbb{R}} p_u(x-z)\eta_n(z)\mu_n^l(z)\,\mathrm{d}z, \tag{4.22a}$$

$$\mu_{N-n}^r(x) \propto \int_{\mathbb{R}} p_u(z-x)\eta_j(z)\mu_j^r(z)\,\mathrm{d}z, \tag{4.22b}$$

   where $j = N - n + 1$. As in (4.16), the symbol $\propto$ denotes identity after normalization to unity. Since the pdf p_u is symmetric, the expressions can be rewritten in terms of the convolutions $\mu_{n+1}^l \propto p_u * \eta_n \mu_n^l$ and $\mu_{N-n}^r \propto p_u * \eta_j \mu_j^r$.

3. **Result:** For $n = 1, \dots, N$, compute

$$[\widehat{\mathbf{x}}_{\text{MMSE}}]_n = \int_{\mathbb{R}} x p_{x_n|\mathbf{y}}(x \,|\, \mathbf{y})\,\mathrm{d}x, \tag{4.23a}$$

   where the marginal pdf is obtained by

$$p_{x_n|\mathbf{y}}(x \,|\, \mathbf{y}) \propto \mu_n^l(x)\mu_n^r(x)\eta_n(x). \tag{4.23b}$$

The proposed update rules recursively marginalize the posterior distribution, reducing intractable high-dimensional integration into $2N$ convolutions. It is well-known that BP gives exact marginal probabilities for all the nodes in any singly connected graph. Consequently, for our problem the solution of the algorithm coincides with $\widehat{\mathbf{x}}_{\text{MMSE}}$.

4.5.2 Fourier-Domain Alternative

The BP algorithm presented in Section 4.5.1 assumes availability of a closed-form expression for the pdf p_u. Unfortunately this form is often unavailable, since the distribution is defined by its characteristic function $\hat{p}_u$ obtained by the Lévy-Khintchine formula (4.1). When the general shape of the pdf is unknown, a naïve numerical evaluation of the inverse Fourier-transform of the characteristic function can lead to unexpected results. As an example, consider the compound Poisson process. The characteristic function (4.3) describes the distribution of the increments, but does not generally admit a closed-form expression of its inverse Fourier transform. Moreover, it results in a pdf containing a probability mass (a Dirac delta function) at zero, which needs to be taken into account explicitly for a correct numerical inversion.

Fortunately, the BP algorithm presented above can readily be performed in the frequency domain. The message-update equations are obtained by the convolution property of the Fourier transform, which amounts to switching the role of multiplications and convolutions in (4.22) and (4.23b). The final estimation step is also simplified by applying the moment property

$$\int_{\mathbb{R}} x^n f(x)\,\mathrm{d}x = \mathrm{j}^n \left.\frac{\mathrm{d}^n}{\mathrm{d}\omega^n}\hat{f}(\omega)\right|_{\omega=0}, \tag{4.24}$$

where $\hat{f}(\omega) = \int_{\mathbb{R}} f(x)\mathrm{e}^{-\mathrm{j}\omega x}\,\mathrm{d}x$ is the Fourier transform of f.

1. **Initialization:** Set

$$\hat{\mu}_1^l(\omega) = \hat{p}_u(\omega), \tag{4.25a}$$

$$\hat{\mu}_N^r(\omega) = \delta(\omega), \tag{4.25b}$$

   where δ is the Dirac delta function.

2. **Message updates:** For $n = 1, \dots, N-1$, compute

$$\hat{\mu}_{n+1}^l(\omega) \propto \hat{p}_u(\omega)\cdot(\hat{\eta}_n * \hat{\mu}_n^l)(\omega), \tag{4.26a}$$

$$\hat{\mu}_{N-n}^r(\omega) \propto \hat{p}_u(\omega)\cdot(\hat{\eta}_j * \hat{\mu}_j^r)(\omega), \tag{4.26b}$$

   where $j = N-n+1$. The symbol $\propto$ denotes identity after normalization by the zero frequency component. The functions $\hat{\eta}_n$ represent the Fourier transform of (4.20).

3. **Result:** For $n = 1, \dots, N$, compute

$$[\widehat{\mathbf{x}}_{\text{MMSE}}]_n = \mathrm{j}\, \frac{\mathrm{d}}{\mathrm{d}\omega} \hat{p}_{x_n|\mathbf{y}}(\omega \,|\, \mathbf{y}) \Big|_{\omega=0}, \tag{4.27a}$$

   where the characteristic function $\hat{p}_{x_n|\mathbf{y}}(\omega \,|\, \mathbf{y})$ of the marginalized posterior is obtained by

$$\hat{p}_{x_n|\mathbf{y}}(\omega \,|\, \mathbf{y}) \propto \left(\hat{\mu}_n^l * \hat{\mu}_n^r * \hat{\eta}_n\right)(\omega). \tag{4.27b}$$

Note that (4.27a) and (4.27b) can be evaluated with a single integral. This is achieved by reusing convolutions in (4.26) and evaluating the derivative only at zero.

4.5.3 Implementation

In principle, the BP equations presented above yield the exact MMSE estimator for our problem. However, due to the existence of continuous-time integrals in the updates, they cannot be implemented in the given form. To obtain a realizable solution, we need to choose some practical discrete parameterization for the messages exchanged in the algorithm. The simplest and the most generic approach is to sample the functions and represent them on a uniform grid with finitely many samples. In our implementation, we fix the support set of the functions to $\left[-K_{\Omega,\epsilon}\Omega, K_{\Omega,\epsilon}\Omega\right]_{\mathbb{Z}}$. We retain only these samples for which $f(x) \geq \epsilon$. Thus, the total number of samples for representing the function depends on both the truncation parameter $\epsilon > 0$ and on the sampling step $\Omega > 0$. It is given by $L_{\Omega,\epsilon} = 2K_{\Omega,\epsilon} + 1$. The proper parameter values depend on the distribution to represent and on the measurements $\mathbf{y}$. Then, both time- and frequency-domain versions can be obtained by approximating continuous integrals by standard quadrature rules. In our implementation, we use Riemann sums to approximate the integrals.

4.6 Experimental Evaluation

We now present several experiments with the goal of comparing various signal-estimation methods. The performance of the estimator is judged based on the MSE given by

$$\text{MSE} \triangleq 10 \log_{10}\left(\frac{1}{N}\|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2\right), \tag{4.28}$$

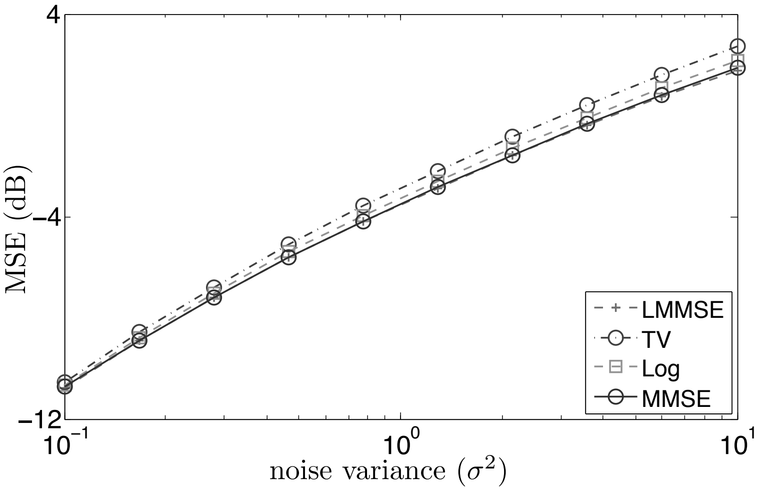


Figure 4.4: Denoising of Brownian motion.

where $\mathbf{x}, \widehat{\mathbf{x}} \in \mathbb{R}^N$. We concentrate on the four Lévy processes discussed in Section 4.3.2 and set the parameters of these processes as

- **Brownian Motion:** The increments are generated from a standard Gaussian distribution with $u_n = [\mathbf{Dx}]_n \sim \mathcal{N}(0, 1)$.
- **Compound Poisson Process:** We concentrate on sparse signals and set the mass probability to $P(u_n = 0) = \mathrm{e}^{-\lambda} = 0.9$. The size of the jumps follow the standard Gaussian distribution.
- **Laplace Increment Process:** The increments are generated from the Laplace distribution of scale $\lambda = 1$.
- **Lévy Flight:** We set the distribution of the increments to be Cauchy ($\alpha = 1$) with scale parameter $\rho = 1$.

4.6.1 AWGN Denoising

In the first set of experiments, we consider the denoising of Lévy processes in AWGN. We compare the performance of several popular estimation methods over a range of

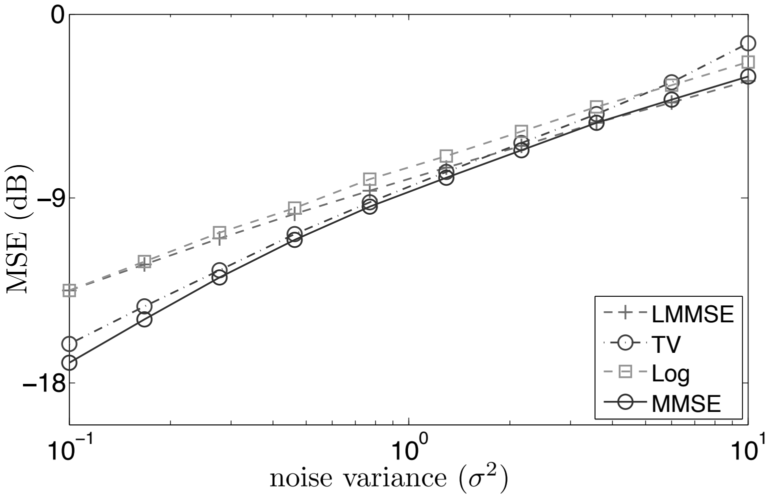


Figure 4.5: Denoising of compund Poisson process.

noise levels σ^2. We perform 1000 random realization of the denoising problem for each value of σ^2 and plot the average MSE in Figures 4.4–4.7. The signal length is set to $N = M = 200$. The proposed message-passing estimator is compared with the regularized least-squares estimators

$$\widehat{\mathbf{x}} = \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{y}-\mathbf{x}\|_2^2 + \tau \sum_{n=1}^{N} \phi_u([\mathbf{D}\mathbf{x}]_k) \right\}, \tag{4.29}$$

where $\mathbf{D}$ is a finite-difference matrix and $\tau > 0$ is the regularization parameter optimized for the best MSE performance.

The curve labeled LMMSE corresponds to the MSE optimal linear estimator, which can be obtained by setting the potential function $\phi_u(x) = x^2$ [76]. The TV method corresponds to the potential function $\phi_u(x) = |x|$ and can be efficiently implemented by the FISTA [36]. The Log estimator corresponds to the potential function $\phi_u(x) = \log\left(x^2 + \epsilon\right)$, where the parameter $\epsilon > 0$ controls the sparsity of the signal. Log-based regularizers have been shown to outperform traditional ℓ_1-based regularizers in various applications [38, 66]. In our experiments, we fix $\epsilon = 1$, which corresponds to the MAP estimator for the Lévy-flight process with Cauchy increments. The Log-based denoising was implemented efficiently by the algorithm described in [41].

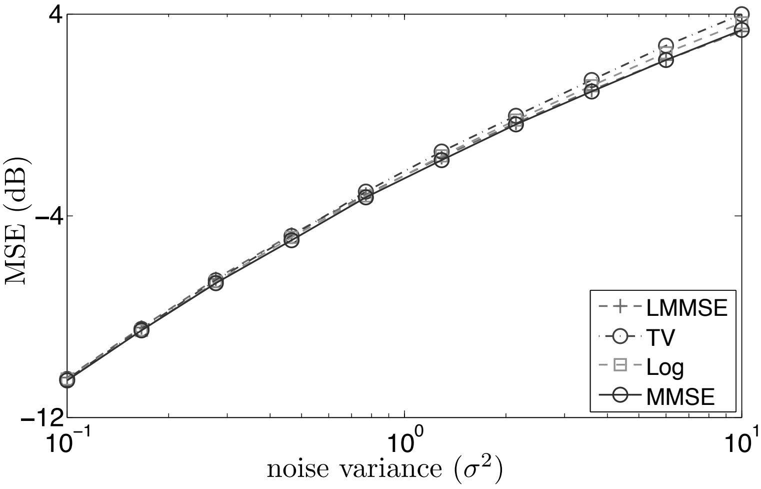


Figure 4.6: Denoising of a Lévy process with Laplace increments.

It is well known that the LMMSE estimator is optimal for Brownian motion. In Figure 4.4, it is precisely matched by the message-passing MMSE estimator. The worst performance is observed for TV regularization, which yields piecewise-constant solutions by removing small variations of the signal. The performance of the Log-based method is significantly better; it preserves important details by allowing small variations of the signal.

In Figure 4.5, we observe excellent MSE performance of TV for compound Poisson processes over many noise levels. This happens because the piecewise-constant solution of TV estimator is well matched to such signals. In this experiment, we have also measured the average running times for all the algorithms. For example, for $\sigma^2 = 1$ the average estimation times for LMMSE, TV, Log, and MMSE were 0.03, 0.05, 0.01, and 0.29 seconds, respectively. The theoretical relevance of the compound Poisson process is extensively discussed in [61].

In Figure 4.6, we observe a surprisingly poor performance of TV, which corresponds to the MAP estimator for Lévy processes with Laplace increments. This highlights the fact that, in some situations, a MAP estimator can result in suboptimal MSE performance.

We observe that LMMSE performs poorly for the Lévy-flight process in Figure 4.7. It fails to preserve signal edges, which results in a suboptimal MSE performance for all

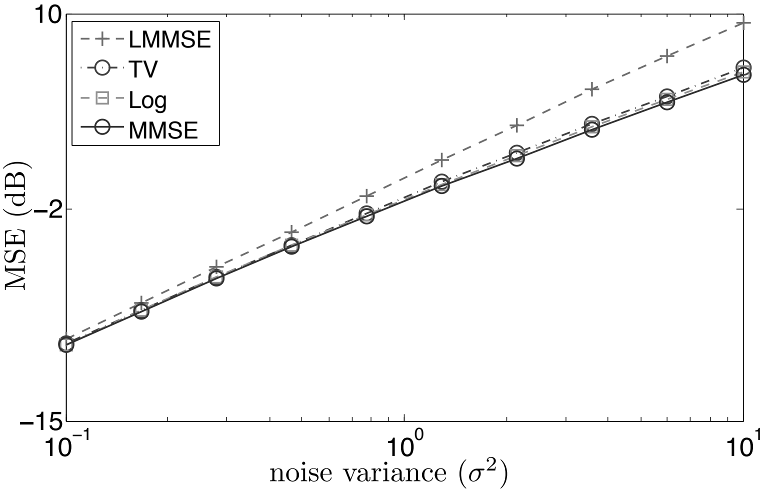


Figure 4.7: Denoising of Lévy-flight process.

noise levels. Both TV and Log methods, which are known to be edge-preserving, yield results close to the MMSE estimator (within 0.2 dB for Log). For such signals, the Log-based regularizers implement the MAP estimator.

The message-passing algorithm considered in this chapter computes the marginals of the posterior distribution. The algorithm yields the MMSE estimator by finding the mean of the marginalized distribution. But the posterior distribution actually provides much more information. For example, the algorithm can predict the MSE of the reconstruction by computing the variance of the posterior

$$\mathrm{var}\left[x_n \,|\, \mathbf{y}\right] = \mathbb{E}\left[x_n^2 \,|\, \mathbf{y}\right] - ([\widehat{\mathbf{x}}_{\text{MMSE}}]_n)^2,$$

where $[\widehat{\mathbf{x}}_{\text{MMSE}}]_n$ is given in (4.27). The second moment can be evaluated by using the moment property (4.24). The capability to predict the MSE of the reconstruction is useful to complement the solution of the estimator with a confidence interval. In Table 4.1, the MSE predicted by the algorithm is presented for Gaussian and Cauchy increment processes. For comparison, we also provide the oracle MSE obtained by comparing the true signal $\mathbf{x}$ with $\widehat{\mathbf{x}}$. The average predicted MSE is obtained from 1000 random realizations of each process. Table 4.1 also provides the standard deviation of the predicted MSE values around the mean. This illustrates the accuracy of the predicted MSE values across noise levels.

Table 4.1: MMSE Prediction.

Prior	Noise (σ^2)	Oracle MSE (dB)	Predicted MSE (dB)
Gaussian	0.1	−10.74	$-10.73 \pm 5.4 \times 10^{-5}$
	1	−3.54	$-3.49 \pm 5.9 \times 10^{-5}$
	10	1.85	$1.95 \pm 6.5 \times 10^{-5}$
Cauchy	0.1	−10.37	-10.34 ± 0.03
	1	−1.54	-1.53 ± 0.11
	10	6.15	6.22 ± 0.21

4.6.2 Signal Interpolation

In Figure 4.8, we illustrate the interpolation of four types of Lévy processes from noisy measurements. We assume AWGN of variance $\sigma^2 = 1$ and set the interpolation rate to $m_s = \Delta_s/\Delta_e = 10$. Given 10 noisy measurements, this results in 91 estimated values. An interesting observation is that the MSE optimal interpolator seems to yield piecewise-linear results, independently of the process considered. In fact, it is known that, for Brownian motion, piecewise-linear interpolation is optimal [77]. Note that this does not imply that the estimator is itself linear, in the sense of being homogeneous and additive—in general, it is not.

In Table 4.2, we compare the MSE performance of message-passing estimators with linear estimators for the interpolation problem with $m_s = 2$. Each value in the table is obtained by averaging over 1000 problem instances. For the interpolation problem, the average estimation MSE for the Lévy-flight process is not defined and can only be characterized conditioned on a given $\mathbf{y}$. Therefore, this process was omitted from the table

4.6.3 Estimation from Quantized Samples

We next consider the highly nonlinear problem of estimating Lévy processes from quantized measurements (4.14). To do so, we generate a compound Poisson process of length $N = 200$. An AWGN of variance 0.1 is added to the signal prior to quantization.

Table 4.2: Interpolation of Lévy processes: MSE for different noise levels.

Prior	Noise (σ^2)	LMMSE (dB)	MMSE (dB)
Gaussian	0.1	−4.9315	−4.9315
	1	−1.3866	−1.3866
	10	3.4221	3.4221
Compound Poisson	0.1	−11.3233	−12.7016
	1	−6.3651	−6.8164
	10	−1.5267	−1.6012
Laplace	0.1	−2.4691	−2.4724
	1	0.2644	0.2279
	10	4.9509	4.9406

The quantizer is uniform with granular region of length $2\|\mathbf{y}\|_\infty$. It is centered at the origin.

In Figure 4.9, we compare the MSE performance of the message-passing estimator with the standard LMMSE estimator. The parameter τ of the linear estimator was optimized for the best MSE performance. In this figure, we plot the mean of the MSE from 1000 problem instances for several quantization levels K. For such nonlinear measurement channels, the message-passing estimator yields significant improvements in the reconstruction performance over the standard linear estimator.

4.7 Summary

We have presented an in-depth investigation of the Lévy-process framework for modeling signals with sparse derivatives. We have also characterized the corresponding statistical estimators. Lévy processes are fundamental members of a recently proposed family of stochastic processes for the continuous-domain modeling of sparse signals. Our key contribution is a simple message-passing algorithm for the MMSE estimation of Lévy processes from noisy measurements. The proposed algorithm can handle a large class of priors, including those that do not have closed-form pdfs. Moreover, it can incorporate a large class of noise distributions, provided that the noise components are independent among themselves. The algorithm has also the ability to handle signal-dependent noise. Due to the tree-like structure of the underlying factor graph, when the messages are continuous-time functions, the message-passing algorithm obtains the MMSE estimator of the signal. This motivates its application as a benchmark to judge the optimality of various existing gradient–based estimators including TV- and Log-regularization algorithms.

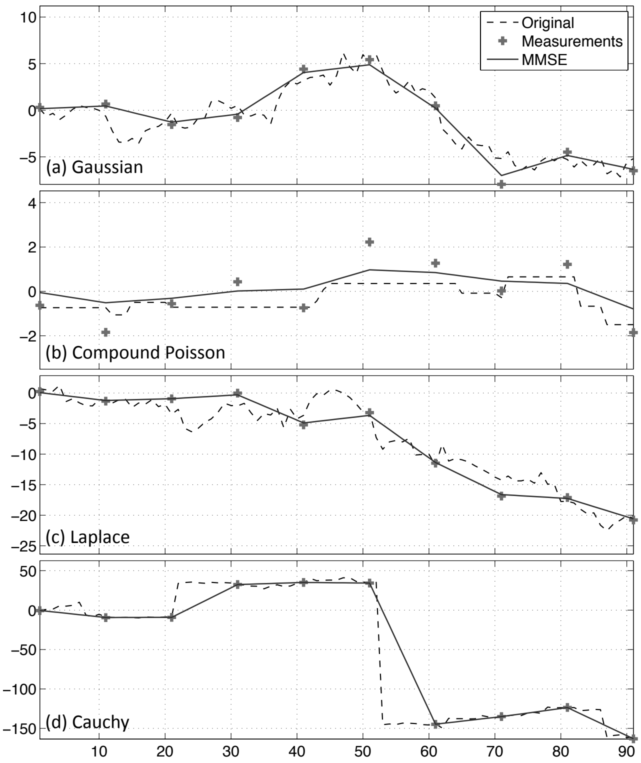


Figure 4.8: Ten-fold interpolation of Lévy processes from AWGN measurements. From top to bottom: (a) Brownian motion; (b) compound Poisson process; (c) Lévy process with Laplace increments; (d) Lévy flight process. Surprisingly, for all priors the optimal estimator appears to be a piecewise linear function.

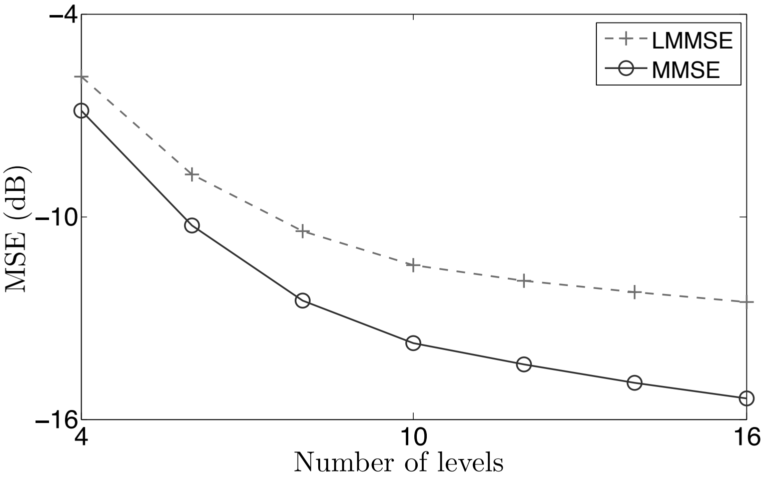


Figure 4.9: Estimation of the compound Poisson process from quantized measurements. We compare the standard LMMSE against MMSE, thereby illustrating the suboptimality of standard linear reconstructions.

Chapter 5

Efficient Approximations to MMSE with Adaptive GAMP

5.1 Overview

In this chapter, we take a step beyond denoising by considering a more general forward model that involves the measurement matrix $\mathbf{H}$. We will, however, make an assumption that our signal $\mathbf{x}$ can be perfectly decoupled with some orthogonal wavelet transform $\mathbf{W}$. This simplifying assumption allows us to concentrate on estimating the wavelet coefficiets $\mathbf{w} = \mathbf{W}\mathbf{x}$ under the combined forward operator $\mathbf{H}_{\text{wav}} = \mathbf{H}\mathbf{W}^T$. Nonetheless, in order to keep notations simple, we will overload the matrix notation $\mathbf{H}$ and the vector notation $\mathbf{x}$ to denote $\mathbf{H}_{\text{wav}}$ and $\mathbf{w}$, respectively. Consequently, in the sequel, our aim will be on estimating a vector $\mathbf{x}$ with independent components from measurements $\mathbf{y}$.

We introduce a novel method called *adaptive generalized approximate message passing* (adaptive GAMP), which enables the joint learning of the unknown statistical parameters of the problem along with the estimation of the signal $\mathbf{x}$. The method can be used to approximate both MAP and MMSE solutions under generalized linear forward models consisting of a known linear transform followed by a probabilistic measurement channel. Our first theoretical contribution proves that for large independent and identically distributed (i.i.d.) Gaussian measurement matrices[1], the asymptotic componentwise behaviour of adaptive GAMP is predicted by a simple set of scalar equations called *state evolution* (SE) equations. We then show that adaptive GAMP yields asymptotically consistent parameter estimates, when maximum-likelihood estimation can be performed in each step. This implies that the algorithm achieves a reconstruction quality equivalent to the oracle algorithm that knows the correct parameter values. Remarkably, this result applies to essentially arbitrary parametrizations of the unknown distributions, including nonlinear and non-Gaussian ones. Our algorithm thus provides a systematic, general and computationally efficient method applicable to a large range of generalized linear models[2].

5.2 Introduction

Consider the estimation of a random vector $\mathbf{x} \in \mathbb{R}^N$ from the measurement model illustrated in Figure 5.1. The random vector $\mathbf{x}$, which is assumed to have i.i.d. components $x_n \sim p_x$, is passed through a known linear transform that outputs $\mathbf{z} = \mathbf{H}\mathbf{x}$. The components of $\mathbf{y} \in \mathbb{R}^M$ are generated by the componentwise transfer function $p_{y|z}$. We address the problem of the estimation of $\mathbf{x}$ when the distributions p_x and $p_{y|z}$ have finite number of unknown parameters, $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$, that must be learned during the process.

Such joint-estimation and learning problems with linear transforms and componentwise nonlinearities arise in a range of applications, including empirical Bayesian approaches to inverse problems in signal processing, linear regression, and classification. It is equally relevant for Bayesian compressed sensing for the estimation of sparse vectors $\mathbf{x}$ from underdetermined measurements [60]. Also, since the parameters in the output transfer function $p_{y|z}$ can model unknown nonlinearities, this problem formulation can be applied to the identification of linear-nonlinear cascade models of dynamical systems, in particular for neural spike responses [79–81].

When the distributions p_x and $p_{y|z}$ are known, there are a number of estimation methods available. In recent years, there has been significant interest in *approximate mes-*

---

[1] Note that for orthogonal $\mathbf{W}$ and i.i.d. Gaussian $\mathbf{H}$, the combined matrix $\mathbf{H}_{\text{wav}} = \mathbf{H}\mathbf{W}^T$ also has i.i.d. Gaussian elements.

[2] This chapter is based on our paper [78]

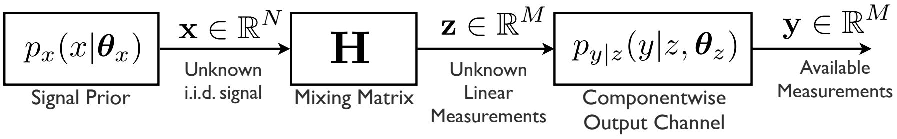

$p_x(x|\boldsymbol{\theta}_x)$
Signal Prior
$\mathbf{x} \in \mathbb{R}^N$
Unknown
i.i.d. signal
H
Mixing Matrix
$\mathbf{z} \in \mathbb{R}^M$
Unknown
Linear
Measurements
$p_{y|z}(y|z,\boldsymbol{\theta}_z)$
Componentwise
Output Channel
$\mathbf{y} \in \mathbb{R}^M$
Available
Measurements

Figure 5.1: Measurement model considered.

sage passing (AMP) and related methods based on Gaussian approximations of belief propagation (BP) [24, 72, 74, 82–85]. These methods originate from CDMA multiuser detection problems and have received considerable recent attention in the context of compressed sensing [9, 24, 72, 74, 83–86]. The Gaussian approximations used in AMP are also closely related to expectation propagation techniques [87, 88], but with additional simplifications that exploit the linear coupling between the variables **x** and **z**. The key benefits of AMP methods are their computational simplicity, their broad range of application, and, for certain large random **H**, their exact asymptotic performance characterizations with testable conditions for optimality [74, 82, 85]. We consider the generalized AMP (GAMP) method [24] that extends the algorithm in [72] to arbitrary output distributions $p_{y|z}$.

Although the current formulation of AMP and GAMP methods is attractive, in practice, one rarely knows the prior and noise distributions exactly. The expectation-maximization–based (EM) approach [89, 90] overcomes this limitation by jointly learning the parameters $(\boldsymbol{\theta}_x, \boldsymbol{\theta}_z)$ along with the estimation of the vector **x**. EM-GAMP inspired our work in [10]. While simulations indicate excellent performance, no analysis of these methods is available in the literature. This chapter provides a unifying analytic framework for such AMP–based joint estimation and learning methods. Our main contributions here are as follows:

- The generalization of the GAMP method of [24] to a class of algorithms we call *adaptive GAMP* that enable the joint estimation of the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ along with vector **x**. The methods are computationally fast and general. In addition, adaptive GAMP includes the EM-GAMP algorithms of [89–92] as special cases.
- The exact characterization of the asymptotic behavior of adaptive GAMP. We show that, similar to the analysis of the AMP and GAMP algorithms in [24, 74, 82, 85], the component-wise asymptotic behavior of adaptive GAMP can be described by a

simple set of scalar *state-evolution* (SE) equations.

- The demonstration of the asymptotic consistency of adaptive GAMP with maximum-likelihood (ML) parameter estimation. We show that, when the ML parameter estimation is computed exactly, the estimated parameters converge to the true values and the performance of adaptive GAMP asymptotically coincides with the performance of the oracle GAMP algorithm that knows the correct parameter values. Remarkably, this result applies to essentially arbitrary parameterizations of the unknown distributions p_x and $p_{y|z}$, thus enabling provably consistent estimation on non-convex and nonlinear problems.

- The experimental evaluation of the algorithm for the problems of learning sparse priors in compressed sensing and of identification of linear-nonlinear cascade models in neural spiking processes. Our simulations illustrate the performance gain of adaptive GAMP and its asymptotic consistency.

5.2.1 Related Literature

The adaptive GAMP method proposed here can be seen as a generalization of the EM methods in [89–92]. In [89, 90], the prior p_x is described by a generic L-term Gaussian mixture (GM) whose parameters are identified by an EM procedure [93]. The *expectation* step or E-step is performed by GAMP, which can approximately determine the marginal posterior distributions of the components x_n given the observations $\mathbf{y}$ and the current parameter estimates of the GM distribution p_x. A related EM-GAMP algorithm has also appeared in [91, 92] for the case of certain sparse priors and AWGN outputs. Simulations in [89, 90] show remarkably good performance and computational speed for EM-GAMP over a wide class of distributions, particularly in the context of compressed sensing. Also, using arguments from statistical physics, Krzakala *et al.* [91, 92] present SE equations for the joint evolution of the parameters and vector estimates and confirm them numerically.

As discussed in Section 5.4.2, EM-GAMP is a special case of adaptive GAMP with a particular choice of the adaptation functions. Therefore, one contribution of our work is to provide a theoretical basis for the EM-GAMP methodology. In particular, Theorem 5.2 provides a rigorous justification of the SE analysis in [91, 92] along with extensions to a broader class of input and output channels and adaptation methods.

An alternate method for joint learning and estimation has been presented in [94], which assumes that the distributions on the source and output channels are themselves described by graphical models with the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ appearing as unknown variables. The method in [94], called hybrid-GAMP, iteratively combines standard loopy BP with AMP methods. One avenue of future work is to see if our methodology here can be applied to analyze the hybrid-GAMP methods as well.

Finally, it should be pointed out that, while the simultaneous recovery of unknown parameters is appealing conceptually, it is not a strict requirement. An alternate solution to the problem is to assume that the signal belongs to a known class of distributions and to minimize the maximal mean-squared error (MSE) for the class. This *minimax* approach [95] was proposed for the AMP recovery of sparse signals in [72]. Although minimax yields estimators that are uniformly good over the entire class of distributions, there may be a significant gap between the MSE achieved by the minimax approach and the oracle algorithm that knows the distribution exactly. Indeed, reducing this gap was the main justification of the EM-GAMP methods in [89, 90]. Due to its asymptotic consistency with ML parameter estimation, adaptive GAMP provably achieves the performance of the oracle algorithm.

5.3 Review of GAMP

Before describing the adaptive GAMP algorithm, it is useful to review the basic (non-adaptive) GAMP. There are two important variants of GAMP:

- **Sum-product GAMP:** This method is used for approximately computing the posterior marginals
  $$p_{x_n|\mathbf{y}}(x_n|\mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z), \tag{5.1}$$
  with respect to the joint density $p_{\mathbf{x}|\mathbf{y}}$. From these posterior marginals, one can compute the MMSE estimator via the posterior means and variances,
  $$\widehat{x}_n = \mathbb{E}[x_n \mid \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z] \tag{5.2a}$$
  $$\tau_{x_n} = \mathrm{var}[x_n \mid \mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z]. \tag{5.2b}$$

  The GAMP algorithm in this case is based on a Gaussian approximation of sum-product loopy belief propagation.

- **Max-sum GAMP:** This variant is used to approximately compute the MAP estimate

  $$\widehat{\mathbf{x}} \triangleq \underset{\mathbf{x}\in N}{\arg\max} \left\{ p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z) \right\}, \tag{5.3}$$

  and is based on a quadratic approximation of the max-sum loopy belief propagation.

5.3.1 Sum-Product GAMP

We begin with a description of the most basic—and perhaps most important—variant of GAMP, namely *sum-product GAMP*. Consider the observation model in Fig. 5.1 where the componentwise probability density functions on the inputs and outputs have some parametric form,

$$p_x(x|\boldsymbol{\theta}_x), \quad p_{y|z}(y|z, \boldsymbol{\theta}_z), \tag{5.4}$$

where $\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x$ and $\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z$ represent parameters of the densities and $\boldsymbol{\Theta}_x \subseteq \mathbb{R}^{d_x}$ and $\boldsymbol{\Theta}_z \subseteq \mathbb{R}^{d_z}$ denote the corresponding parameter sets that are of finite dimensions. Now, suppose that the components of $\mathbf{x}$ are i.i.d. with $x_n \sim p_x(x_n|\boldsymbol{\theta}_x)$ and, conditional on the transform output $\mathbf{z} = \mathbf{H}\mathbf{x}$, the components of the observations $\mathbf{y}$ have a likelihood $y_m \sim p_{y|z}(y_m|z_m, \boldsymbol{\theta}_z)$. Then, the posterior joint probability density of $\mathbf{x}$ is given by

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z) \propto \prod_{m=1}^{M} p_{y|z}(y_m|[\mathbf{H}\mathbf{x}]_m, \boldsymbol{\theta}_z) \prod_{n=1}^{N} p_x(x_n|\boldsymbol{\theta}_x), \tag{5.5}$$

where $\propto$ denotes identity after normalization. The GAMP algorithm of [24] can be seen as a class of methods for approximately estimating the vector $\mathbf{x}$ under this joint distribution in the case when the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ are *known*.

The basic steps of the sum-product GAMP algorithm are shown in Algorithm 5.1. The algorithm is an iterative procedure generating a sequence of estimates $\widehat{\mathbf{x}}^t$, τ_x^t representing estimates of the posterior means and variances in (5.2).

Exact computation of the means and variance of the components x_n and z_m of the posterior joint density (5.5) is generally intractable, since it involves a marginalization over N variables. The main concept in the GAMP algorithm is to approximately reduce this vector-valued estimation problem to a sequence of scalar mean and variance computations. Specifically, the expectations and variances in Algorithm 5.1 are to be taken with

Algorithm 5.1: Sum-Product GAMP

```
input: the data $\mathbf{y}$, the densities $p_x$ and $p_{y|z}$ with known $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$,
       a constant $||\mathbf{H}||_F^2$,
       and an efficient implementation of $\mathbf{H}$ and $\mathbf{H}^T$
set: $t \leftarrow 0$, $\mathbf{s}^{t-1} \leftarrow 0$
$\widehat{\mathbf{x}}^0 \leftarrow \mathbb{E}[x|\boldsymbol{\theta}_x]$, $\tau_x^0 \leftarrow \mathrm{var}[x|\boldsymbol{\theta}_x]$.
repeat
  {Output node update}
  $\tau_p^t \leftarrow ||\mathbf{H}||_F^2 \tau_x^t / M$
  $\mathbf{p}^t \leftarrow \mathbf{H}\widehat{\mathbf{x}}^t - \mathbf{s}^{t-1}\tau_p^t$
  $\widehat{z}_m^t \leftarrow \mathbb{E}[z_m|y_m, p_m^t, \tau_p^t, \boldsymbol{\theta}_z]$,   $(m = 1, \ldots, M)$
  $\tau_{z_m}^t \leftarrow \mathrm{var}[z_m|y_m, p_m^t, \tau_p^t, \boldsymbol{\theta}_z]$,   $(m = 1, \ldots, M)$
  $s_m^t \leftarrow (z_m^t - p_m^t)/\tau_p^t$,   $(m = 1, \ldots, M)$
  $\tau_s^t \leftarrow (1/M)\sum_m (1 - \tau_{z_m}^t/\tau_p^t)/\tau_p^t$

  {Input node update}
  $1/\tau_r^t \leftarrow ||\mathbf{H}||_F^2 \tau_s^t / N$
  $\boldsymbol{r}^t \leftarrow \widehat{\mathbf{x}}^t + \tau_r^t \mathbf{H}^T \mathbf{s}^t$
  $\widehat{x}_n^{t+1} \leftarrow \mathbb{E}[x_n|r_n^t, \tau_r^t, \boldsymbol{\theta}_x]$,   $(n = 1, \ldots, N)$
  $\tau_x^{t+1} \leftarrow (\tau_r^t/N)\sum_n \mathrm{var}[x_n|r_n^t, \tau_r^t, \boldsymbol{\theta}_x]$
until stopping criterion
return $\widehat{\mathbf{x}}^{t+1}$
```

respect to the probability density

$$p(x_n|r_n^t, \tau_r^t, \boldsymbol{\theta}_x) \propto p_x(x_n|\boldsymbol{\theta}_x)\exp\left(-\frac{1}{2\tau_r^t}|x_n - r_n^t|^2\right), \tag{5.6a}$$

$$p(z_m|y_m, p_m^t, \tau_p^t, \boldsymbol{\theta}_z) \propto p_{y|z}(y_m|z_m, \boldsymbol{\theta}_z)\exp\left(-\frac{1}{2\tau_p^t}|z_m - p_m^t|^2\right). \tag{5.6b}$$

The densities (5.6a) and (5.6b) are the GAMP approximation of the posterior marginal densities $p(x_n|\mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z)$ and $p_{z_m|\mathbf{y}}(z_m|\mathbf{y}, \boldsymbol{\theta}_x, \boldsymbol{\theta}_z)$, respectively. Since these densities are over one-dimensional random variables, even if their means and variances do not have closed-form expressions, they can be computed via numerical integration. In addition, the densities can be interpreted as posterior distributions on scalar random variables x_n and z_m with respect to observations r_n^t and (y_m, p_m^t) of the form

$$r_n^t = x_n + \mathcal{N}(0, \tau_r^t), \quad x_n \sim p_x(x_n|\boldsymbol{\theta}_x) \tag{5.7a}$$

$$y_m \sim p_{y|z}(y_m|z_m), \quad z_m \sim \mathcal{N}(p_m^t, \tau_p^t). \tag{5.7b}$$

Hence, computing the posterior mean and variance of x_n is equivalent to a set of scalar AWGN estimation problems. In this way, the sum-product GAMP algorithm reduces the inherently vector-valued inference problem to a sequence of scalar AWGN estimation problems at the input and output, along with transform by $\mathbf{H}$ and $\mathbf{H}^T$. This is computationally attractive since the algorithm involves no vector-valued estimation steps or matrix inverses.

Of course, the GAMP algorithm is only an approximation of the true inference problem. The convergence results of the algorithm are reviewed in Section 5.3.3; they can also be found in references mentioned above.

5.3.2 General GAMP

As mentioned above, the sum-product GAMP algorithm is a particular instance of a more general class of algorithms that includes the max-sum GAMP algorithm for MAP estimation. To provide the most general results for the adaptive GAMP, we briefly review the general (non-adaptive) GAMP algorithm. Full details of the general GAMP algorithm can be found in [24]. For completeness, we restate the steps of the general GAMP algorithm in Algorithm 5.2.

Comparing Algorithms 5.1 and 5.2, we see that there are two generalizations in the general GAMP algorithm. First, the mean and variance computations of the sum-product GAMP algorithm, Algorithm 5.1, are replaced with general *estimation functions* G_x^t, G_s^t and G_z^t. These estimation functions take the outputs $\boldsymbol{r}^t$ and $\mathbf{p}^t$ and generate the estimates $\mathbf{x}^t$, $\mathbf{s}^t$ and $\mathbf{z}^t$. Their derivatives results in the variance terms τ_x^t and τ_s^t. It is shown in [24] that with appropriate selection of these estimation functions, one can incorporate both the sum-product and max-sum variants of the GAMP algorithm.

For the case of the sum-product GAMP, we can recover Algorithm 5.1 with the estimation

Algorithm 5.2: General GAMP

```
input: the data $\mathbf{y}$, estimation functions $G_x^t$, $G_s^t$ and $G_z^t$,
       parameter estimates $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$,
       a constant $\|\mathbf{H}\|_F^2$,
       and an efficient implementation of $\mathbf{H}$ and $\mathbf{H}^T$
set: $t \leftarrow 0$, $\mathbf{s}^{t-1} \leftarrow 0$ and select some initial values for $\widehat{\mathbf{x}}^0$ and $\tau_x^0$.
repeat
  {Output node update}
  $\tau_p^t \leftarrow \|\mathbf{H}\|_F^2 \tau_x^t / M$
  $\mathbf{p}^t \leftarrow \mathbf{H}\widehat{\mathbf{x}}^t - \mathbf{s}^{t-1}\tau_p^t$
  $\widehat{z}_m^t \leftarrow G_z^t(p_m^t, y_m, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t), \quad (m = 1, \ldots, M)$
  $s_m^t \leftarrow G_s^t(p_m^t, y_m, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t), \quad (m = 1, \ldots, M)$
  $\tau_s^t \leftarrow -(1/M) \sum_m \partial G_s^t(p_m^t, y_m, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t) / \partial p_m^t$

  {Input node update}
  $1/\tau_r^t \leftarrow \|\mathbf{H}\|_F^2 \tau_s^t / N$
  $\boldsymbol{r}^t \leftarrow \widehat{\mathbf{x}}^t + \tau_r^t \mathbf{H}^T \mathbf{s}^t$
  $\widehat{x}_n^{t+1} \leftarrow G_x^t(r_n^t, \tau_r^t, \bar{\boldsymbol{\theta}}_x^t), \quad (n = 1, \ldots, N)$
  $\tau_x^{t+1} \leftarrow (\tau_r^t / N) \sum_n \partial G_x^t(r_n^t, \tau_r^t, \bar{\boldsymbol{\theta}}_x^t) / \partial r_n$
until stopping criterion
return $\widehat{\mathbf{x}}^{t+1}$
```

functions

$$G_x^t(r, \tau_r, \boldsymbol{\theta}_x) \triangleq \mathbb{E}[x \mid r, \tau_r, \boldsymbol{\theta}_x], \tag{5.8a}$$

$$G_z^t(p, y, \tau_p, \boldsymbol{\theta}_z) \triangleq \mathbb{E}[z \mid p, y, \tau_p, \boldsymbol{\theta}_z], \tag{5.8b}$$

$$G_s^t(p, y, \tau_p, \boldsymbol{\theta}_z) \triangleq \frac{1}{\tau_p}\left(G_z^t(p, y, \tau_p, \boldsymbol{\theta}_z) - p\right), \tag{5.8c}$$

where the expectations are with respect to the distributions in (5.6a) and (5.6b). It is shown in [24] that the derivatives of these estimation functions agree with the variance computations of Algorithm 5.1.

The second difference between the the sum-product GAMP algorithm in Algorithm 5.1 and the more general Algorithm 5.2 is that the fixed parameter values $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ are replaced by a deterministic sequence of parameter values $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$. Of course, if the parameters are known, there is no reason to change the parameter estimates on each iteration. However, we need to consider this generalization to enable the study of the adaptive GAMP algorithm below.

Illustrative Example: AWGN Outputs with Sparse Priors

As discussed, much of the current interest in the AMP and GAMP methods have been in the context of compressed sensing [24, 72, 74, 83–85]. Thus, it is useful to briefly describe this particular application in more detail. The original AMP formulations in [72, 83, 84] consider the special case of AWGN output

$$y_m = z_m + e_m, \quad e_m \sim \mathcal{N}(0, \sigma^2), \tag{5.9}$$

where the additive noise e_m is i.i.d. and independent of $\mathbf{z}$. In this case, as shown in [24], the output updates in line (12) and 13 reduce to

$$s_m^t = \frac{y_m - p_m^t}{\tau_p^t + \sigma^2}, \quad \tau_s^t = \frac{1}{\tau_p^t + \sigma^2}. \tag{5.10}$$

For Bayesian forms of compressed sensing, one then takes a sparse prior for the density p_x. A common density is the Laplacian prior,

$$p_x(x_n|\lambda) = \frac{\lambda}{2} e^{-\lambda |x_n|}.$$

In this case, the MAP estimate (5.3) corresponds to the classic LASSO estimate [96]. Then, the equations for the estimation function G_x^t of Algorithm 5.2 reduce to the classic soft-thresholding operator. In this way, the max-sum GAMP with a Laplacian prior reduces to a variant of an iterative soft-thresholding algorithm discussed in Section 2.5.

5.3.3 State Evolution Analysis

In addition to its computational simplicity and generality, a key motivation of the GAMP algorithm is that its asymptotic behavior can be precisely characterized when $\mathbf{H}$ is a large i.i.d. Gaussian transform. The asymptotic behavior is described by what is known as a *state evolution* (SE) analysis. By now, there are a large number of SE results for AMP-related algorithms [24, 72, 74, 82–85]. Here, we review the particular SE analysis from [24, 97] which is based on the framework in [85].

Assumption 5.1. *Consider a sequence of random realizations of the general GAMP algorithm, Algorithm 5.2, indexed by the dimension N, satisfying the following assumptions:*

(a) *For each N, the matrix $\mathbf{H} \in \mathbb{R}^{M\times N}$ has i.i.d. components with $H_{mn} \sim \mathcal{N}(0, 1/M)$ and the dimension $M = M(N)$ is a deterministic function of N satisfying $N/M \to \beta$ for some $\beta > 0$ as $N \to \infty$.*

(b) *The input vectors $\mathbf{x}$ and initial condition $\widehat{\mathbf{x}}^0$ are deterministic sequences whose components converge empirically with bounded moments of order $s = 2k - 2$ as*

$$\lim_{N\to\infty} (\mathbf{x}, \widehat{\mathbf{x}}^0) \overset{\mathrm{PL}(s)}{=} (x, \widehat{x}^0), \tag{5.11}$$

to some random vector $(x, \widehat{x}^0)$ for some $k \geq 2$. Loosely, this convergence implies that the empirical distribution of the components of $(\mathbf{x}, \widehat{\mathbf{x}}^0)$ converge to the distribution of $(x, \widehat{x}^0)$. A precise definition is given in Section 5.9.1.

(c) *The output vectors $\mathbf{z}$ and $\mathbf{y} \in \mathbb{R}^M$ are generated by*

$$\mathbf{z} = \mathbf{H}\mathbf{x} \text{ and } y_m = h(z_m, e_m), \quad (m = 1, \ldots, M), \tag{5.12}$$

for some function $h(z, e)$ and vector $\mathbf{e} \in \mathbb{R}^M$ representing an output distortion. It is assumed that the output distortion vector $\mathbf{e}$ is deterministic, but empirically converges as

$$\lim_{N\to\infty} \mathbf{e} \overset{\mathrm{PL}(s)}{=} e, \tag{5.13}$$

where $s = 2k - 2$ is as in Assumption 5.1 (b) and e is some random variable. We let $p_{y|z}$ denote the conditional distribution of the random variable $y = h(z, e)$.

(d) *The estimation function G^t_x, and its derivative with respect to r, are Lipschitz continuous in r at $(\tau_r, \boldsymbol{\theta}_x) = (\bar{\tau}^t_r, \bar{\boldsymbol{\theta}}^t_x)$, where $\bar{\tau}^t_r$ is a deterministic parameter from the SE equations below. A similar assumptions holds for G^t_z.*

Assumption 5.1(a) simply states that we are considering large, Gaussian i.i.d. matrices **H**. Assumptions (b) and (c) state that the input vector **x** and output disturbance **e** are modeled as deterministic, but whose empirical distributions asymptotically appear as i.i.d. This deterministic model is a feature of Bayati and Montanari's analysis in [85]. Assumption (d) is a mild continuity condition.

Note that, for now, there is no assumption that the true distribution of x or the true conditional distribution of y given z must belong to the class of distributions (5.4) for any parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$. The analysis can thus model the effects of model mismatch.

Next, we define the sets of two vectors

$$\vartheta_x^t \triangleq \{(x_n, r_n^t, \widehat{x}_n^{t+1}) : n = 1, \ldots, N\}, \tag{5.14a}$$

$$\vartheta_z^t \triangleq \{(z_m, \widehat{z}_m^t, y_m, p_m^t) : m = 1, \ldots, M\}. \tag{5.14b}$$

The first set ϑ_x^t represents the components of the true, but unknown, input vector **x**, its GAMP estimate $\widehat{\mathbf{x}}^t$ as well as $\boldsymbol{r}^t$. The second set ϑ_z^t contains the components of the true, but unknown, output vector **z**, its GAMP estimate $\widehat{\mathbf{z}}^t$, as well as $\mathbf{p}^t$ and the observed output **y**. The sets ϑ_x^t and ϑ_z^t are implicitly functions of the dimension N.

The main result of [24] shows that if we fix the iteration t, and let $N \to \infty$, the asymptotic joint empirical distribution of the components of ϑ_x^t and ϑ_z^t converges to random vectors of the form

$$\bar{\vartheta}_x^t \triangleq (x, r^t, \widehat{x}^{t+1}), \qquad \bar{\vartheta}_z^t \triangleq (z, \widehat{z}^t, y, p^t). \tag{5.15}$$

We precisely state the nature of convergence in Theorem 5.1 below. In (5.15), x is the random variable in Assumption 5.1(b), while r^t and $\widehat{x}^{t+1}$ are given by

$$r^t = \alpha_r^t x + v^t, \qquad v^t \sim \mathcal{N}(0, \xi_r^t), \tag{5.16a}$$

$$\widehat{x}^{t+1} = G_x^t(r^t, \bar{\tau}_r^t, \bar{\boldsymbol{\theta}}_x^t) \tag{5.16b}$$

for some deterministic constants α_r^t, ξ_r^t, and $\bar{\tau}_r^t$ that are defined below. Similarly, $(z, p^t) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_p^t)$ for some covariance matrix $\mathbf{K}_p^t$, and

$$y = h(z, e), \qquad \widehat{z}^t = G_z^t(p^t, y, \bar{\tau}_p^t, \bar{\boldsymbol{\theta}}_z^t), \tag{5.17}$$

where e is the random variable in (5.13) and $\mathbf{K}_p^t$ contains deterministic constants.

The deterministic constants α_r^t, ξ_r^t, $\bar{\tau}_r^t$ and $\mathbf{K}_p^t$ represent parameters of the distributions of $\bar{\vartheta}_x^t$ and $\bar{\vartheta}_z^t$ and depend on both the distributions of the input and outputs as well as the choice of the estimation and adaptation functions. The SE equations provide a simple method for recursively computing these parameters. The equations are best described algorithmically as shown in Algorithm 5.4. In order not to repeat ourselves, in Algorithm 5.4, we have written the SE equations for adaptive GAMP: For non-adaptive GAMP, the updates (5.32b) and (5.33a) can be ignored as the values of $\bar{\boldsymbol{\theta}}_z^t$ and $\bar{\boldsymbol{\theta}}_x^t$ are pre-computed.

With these definitions, we can state the main result from [24].

Theorem 5.1. *Consider the random vectors ϑ_x^t and ϑ_z^t generated by the outputs of GAMP under Assumption 5.1. Let $\bar{\vartheta}_x^t$ and $\bar{\vartheta}_z^t$ be the random vectors in* (5.15) *with the parameters determined by the SE equations in Algorithm 5.4. Then, for any fixed t, the elements of the sets ϑ_x^t and ϑ_z^t converge empirically with bounded moments of order k as*

$$\lim_{N\to\infty} \vartheta_x^t \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}_x^t, \qquad \lim_{N\to\infty} \vartheta_z^t \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}_z^t. \tag{5.18}$$

where $\bar{\vartheta}_x^t$ and $\bar{\vartheta}_z^t$ are given in (5.15)*. In addition, for any t, the limits*

$$\lim_{N\to\infty} \tau_r^t = \bar{\tau}_r^t, \qquad \lim_{N\to\infty} \tau_p^t = \bar{\tau}_p^t, \tag{5.19}$$

also hold almost surely.

The theorem shows that the components of the vectors $\mathbf{x}$ and $\mathbf{z}$, and their GAMP estimates $\widehat{\mathbf{x}}^t$ and $\widehat{\mathbf{z}}^t$ have the same statistical distribution as random variables x, z, $\widehat{x}^t$ and $\widehat{z}^t$ in a simple scalar equivalent system. This scalar equivalent model appears in several analyses and can be thought of as a *single-letter characterization* [73] of the system. Remarkably, this limiting property holds for essentially arbitrary distributions and estimation functions, even the ones that arise from problems that are highly nonlinear or noncovex. From the single-letter characterization, one can compute the asymptotic value of essentially any component-wise performance metric such as mean-squared error.

5.3.4 State Evolution Analysis for Sum-Product GAMP

For the special case of the sum-product GAMP algorithm in Algorithm 5.1, the SE equations in Algorithm 5.4 reduce to a particularly simple form. The variance terms $\bar{\tau}_r^t$ and ξ_r^t in (5.32) are given by

$$\bar{\tau}_r^t = \xi_r^t = \left(\mathbb{E}\left[\frac{\partial^2}{\partial p^2}\log\left(p_{y|p}(y|p^t)\right)\right]\right)^{-1}, \tag{5.20a}$$

where the expectations are over the random variables $(z, p^t) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_p^t)$ and y is given in (5.17). The covariance matrix $\mathbf{K}_p^t$ has the form

$$\mathbf{K}_p^t = \begin{bmatrix} \beta\tau_{x0} & \beta\tau_{x0} - \bar{\tau}_p^t \\ \beta\tau_{x0} - \bar{\tau}_p^t & \beta\tau_{x0} - \bar{\tau}_p^t \end{bmatrix}, \tag{5.20b}$$

where τ_{x0} is the variance of x, and $\beta > 0$ is the asymptotic measurement ratio (For details see Assumption 5.1). The scaling constant (5.32e) becomes $\alpha_r^t = 1$. The update rule for $\bar{\tau}_x^{t+1}$ also simplifies to

$$\bar{\tau}_x^{t+1} = \mathbb{E}\left[\mathrm{var}\left(x|r^t\right)\right], \tag{5.20c}$$

where the expectation is over the random variables in (5.16). The SE equations for the sum-product GAMP will be initialized with

$$\bar{\tau}_p^0 = \beta\tau_{x0} \tag{5.21}$$

so that the initial value of the covariance matrix in (5.20b) is

$$\mathbf{K}_p^0 = \begin{bmatrix} \beta\tau_{x0} & 0 \\ 0 & 0 \end{bmatrix}. \tag{5.22}$$

5.4 Adaptive GAMP

The above review of the standard GAMP algorithms in Algorithms 5.1 and 5.2 shows that the methods apply to the case when the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ in the distributions

(5.4) are known. The adaptive GAMP method proposed here, and shown in Algorithm 5.3, is an extension of Algorithm 5.2 that enables simultaneous identification of finite dimensional $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ along with estimation of $\mathbf{x}$.

The key modification is the introduction of the two *adaptation functions*: H_z^t and H_x^t. In each iteration, these functions output estimates, $\widehat{\boldsymbol{\theta}}_z^t$ and $\widehat{\boldsymbol{\theta}}_x^t$ of the parameters based on the data $\mathbf{p}^t$, $\mathbf{y}$, $\boldsymbol{r}^t$, τ_p^t and τ_r^t.

The basic GAMP algorithm in Algorithm 5.2 is a special case when the estimation functions H_x^t and H_z^t output fixed values

$$H_z^t(\mathbf{p}^t, \mathbf{y}, \tau_p^t) = \bar{\boldsymbol{\theta}}_z^t, \qquad H_x^t(\boldsymbol{r}^t, \tau_r^t) = \bar{\boldsymbol{\theta}}_x^t, \tag{5.23}$$

for the *pre-computed* sequence of parameters $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$. We call these values precomputed since, in the case of the non-adaptive GAMP algorithm, the parameter estimates $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$ do not depend on the data through the vectors $\mathbf{p}^t$, $\mathbf{y}^t$, and $\boldsymbol{r}^t$. A particular case of the non-adaptive algorithm would be the oracle scenario, where $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$ are set to the true values of the parameters and do not change with the iteration number t.

However, the adaptive GAMP algorithm in Algorithm 5.3 is significantly more general and enables a large class of methods for estimating the parameters based on the data. One particular adaptation method is based on on *maximum likelihood* (ML) as described next.

5.4.1 ML Parameter Estimation

As one possible method to estimate the parameters, recall from Theorem 5.1 that the empirical distribution of the components of $\boldsymbol{r}^t$ converges weakly to the distribution of r^t in (5.16). Now, the distribution of r^t only depends on three parameters – α_r^t, ξ_r^t and $\boldsymbol{\theta}_x$. It is thus natural to attempt to estimate these parameters from the empirical distribution of the components of $\boldsymbol{r}^t$ and thereby recover the parameter $\boldsymbol{\theta}_x$.

To this end, let $\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r)$ be the log likelihood

$$\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r) \triangleq \log(p_r(r|\boldsymbol{\theta}_x, \alpha_r, \xi_r)), \tag{5.24}$$

where the right-hand side is the probability density of a random variable r with distribution

$$r = \alpha_r x + v, \quad x \sim p_x(\cdot|\boldsymbol{\theta}_x), \quad v \sim \mathcal{N}(0, \xi_r). \tag{5.25}$$

Then, at any iteration t, we can attempt to obtain a ML estimate

$$\widehat{\boldsymbol{\theta}}_x^t = H_x^t(\boldsymbol{r}^t, \tau_r^t) \triangleq \underset{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x}{\arg\max} \left\{ \max_{(\alpha_r, \xi_r) \in S_x(\tau_r^t)} \left\{ \frac{1}{N} \sum_{n=1}^{N} \phi_x(r_n^t, \boldsymbol{\theta}_x, \alpha_r, \xi_r) \right\} \right\}. \tag{5.26}$$

Here, the set $S_x(\tau_r^t)$ is a set of possible values for the parameters α_r, ξ_r. This set may depend on the measured variance τ_r^t and we will see its precise role below. The selection of the sets is critical and discussed in detail in Section 5.6.

Similarly, the individual components of $\mathbf{p}^t$ and $\mathbf{y}$ have the same distribution as (p^t, y) which depend only on the parameters $\mathbf{K}_p$ and $\boldsymbol{\theta}_z$. Thus, we can define the likelihood

$$\phi_z(p, y, \boldsymbol{\theta}_z, \mathbf{K}_p) \triangleq \log\big(p_{y,p}(y, p|\boldsymbol{\theta}_z, \mathbf{K}_p)\big), \tag{5.27}$$

where the right-hand side is the joint probability density of (p, y) with distribution

$$y \sim p_{y|z}(\cdot|z, \boldsymbol{\theta}_z), \quad (z, p) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_p). \tag{5.28}$$

Then, we estimate $\boldsymbol{\theta}_z$ via the ML estimate

$$\widehat{\boldsymbol{\theta}}_z^t = H_z^t(\mathbf{p}^t, \mathbf{y}, \tau_p^t) \triangleq \underset{\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z}{\arg\max} \left\{ \max_{\mathbf{K}_p \in S_z(\tau_p^t)} \left\{ \frac{1}{M} \sum_{m=1}^{M} \phi_z(p_m^t, y_m, \boldsymbol{\theta}_z, \mathbf{K}_p) \right\} \right\}. \tag{5.29}$$

Again, the set $S_z(\tau_p^t)$ is a set of possible covariance matrices $\mathbf{K}_p$.

5.4.2 Relation to EM-GAMP

Before discussing the convergence of the adaptive GAMP algorithm with ML parameter estimation, it is useful to briefly compare the ML parameter estimation with the EM-GAMP method proposed by Vila and Schniter [89,90] and Krzakala *et. al.* [91,92]. Both EM-GAMP methods combine the Bayesian AMP or GAMP algorithms with a standard EM procedure [93] as follows. First, the algorithms use the sum-product version of

GAMP, so that the outputs provide an estimate of the posterior distributions on the components of $\mathbf{x}$ given the current parameter values. From the discussion in Section 5.3.1, we know that (5.6a) and (5.6b) can be taken as an approximations of the true posteriors of x_n and z_m for a *given* set of parameter values $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$. Using the approximation, we approximately implement the EM procedure to update the parameter estimate via a maximization

$$\widehat{\boldsymbol{\theta}}_x^t = H_x^t(\boldsymbol{r}^t, \tau_r^t) \triangleq \underset{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x}{\arg\max} \left\{ \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}\left[\log p_x(x_n | \boldsymbol{\theta}_x) | r_n^t, \tau_r^t, \widehat{\boldsymbol{\theta}}_x^{t-1} \right] \right\}, \tag{5.30}$$

where the expectation is with respect to the distribution in (5.6a). In [89, 90], the parameter update (5.30) is performed only once every few iterations to allow $\widehat{p}^t$ to converge to the approximation of the posterior distribution of x_n given the current parameter estimates. In [91,92], the parameter estimate is updated at every iteration. A similar procedure is performed for the estimation of $\boldsymbol{\theta}_z$.

We thus see that the EM-GAMP procedures in [89,90] and in [91,92] are both special cases of the adaptive GAMP algorithm in Algorithm 5.3 with particular choices of the adaptation functions H_x^t and H_z^t. As a result, our analysis in Theorem 5.2 below applies to these algorithms as well and provides rigorous asymptotic characterizations of the EM-GAMP performance. However, at the current time, we can only prove the asymptotic consistency result for the ML adaptation functions (5.26) and (5.29) described above.

That being said, it should be pointed out that the EM-GAMP update (5.30) is generally computationally much simpler than the ML updates in (5.26) and (5.29). For example, when p_x is an exponential family, the optimization in (5.30) is convex. Also, the optimizations in (5.26) and (5.29) require searches over additional parameters such as α_r and ξ_r. Thus, an interesting avenue of future work is to apply the analysis result of Theorem 5.3 below, to see if the EM-GAMP method or some similarly computationally simple technique can be developed which also provides asymptotic consistency.

5.5 Convergence and Asymptotic Consistency with Gaussian Transforms

5.5.1 General State Evolution Analysis

Before proving the asymptotic consistency of adaptive GAMP with ML adaptation, we first prove a more general convergence result.

Assumption 5.2. *Consider the adaptive GAMP algorithm running on a sequence of problems indexed by the dimension N, satisfying the following assumptions:*

(a) Same as Assumption 5.1(a) to (c) with $k=2$.

(b) For every t, the adaptation function H_x^t is a functional over $\mathbf{r}$ satisfying the following weak pseudo-Lipschitz continuity property: Consider any sequence of vectors $\mathbf{r}=\mathbf{r}^{(N)}$ and sequence of scalars $\tau_r=\tau_r^{(N)}$, indexed by N satisfying

$$\lim_{N\to\infty} \mathbf{r}^{(N)} \overset{\mathrm{PL}(k)}{=} r^t, \qquad \lim_{N\to\infty} \tau_r^{(N)} = \bar{\tau}_r^t,$$

where r^t and $\bar{\tau}_r^t$ are the outputs of the state evolution. Then,

$$\lim_{N\to\infty} H_x^t(\mathbf{r}^{(N)}, \tau_r^{(N)}) = H_x^t(r^t, \bar{\tau}_r^t).$$

Similarly, H_z^t satisfies analogous continuity conditions in τ_p and $(\mathbf{y},\mathbf{p})$. See Section 5.9.1 for a general definition of weakly pseudo-Lipschitz continuous functionals.

(c) The scalar-valued function G_x^t and its derivative G'^t_x with respect to r are continuous in $\boldsymbol{\theta}_x$ uniformly over r in the following sense: For every $\epsilon>0$, t, τ_r^* and $\boldsymbol{\theta}_x^* \in \boldsymbol{\Theta}_x$, there exists an open neighborhood U of $(\tau_r^*, \boldsymbol{\theta}_x^*)$ such that for all $(\tau_r, \boldsymbol{\theta}_x) \in U$ and r,

$$|G_x^t(r,\tau_r,\boldsymbol{\theta}_x) - G_x^t(r,\tau_r^*,\boldsymbol{\theta}_x^*)| < \epsilon,$$
$$|G'^t_x(r,\tau_r,\boldsymbol{\theta}_x) - G'^t_x(r,\tau_r^*,\boldsymbol{\theta}_x^*)| < \epsilon.$$

In addition, the functions G_x^t and G'^t_x must be Lipschitz continuous in r with a Lipschitz constant that can be selected continuously in τ_r and $\boldsymbol{\theta}_x$. The functions G_s^t, G_z^t and their derivatives G'^t_s G'^t_z satisfy analogous continuity assumptions with respect to p, y, τ_p and $\boldsymbol{\theta}_z$.

Although technical, assumptions (b) and (c) are mild continuity conditions that can be satisfied by a large class of adaptation functionals and estimation functions. For example, from the definitions in Section 5.9.1, the continuity assumption (b) will be satisfied for any functional given by an empirical average

$$H_x^t(\boldsymbol{r}, \tau_r) = \frac{1}{N} \sum_{n=1}^{N} \phi_x^t(r_n, \tau_r),$$

where, for each t, $\phi_x^t(r_n, \tau_r)$ is pseudo-Lipschitz continuous in r of order p and continuous in τ_r uniformly over r. A similar functional can be used for H_z^t. As we will see in Section 5.5.2, the ML functionals (5.26) and (5.29) also satisfy the required conditions.

Theorem 5.2. *Consider the random vectors ϑ_x^t and ϑ_z^t generated by the outputs of the adaptive GAMP under Assumption 5.2. Let $\bar{\vartheta}_x^t$ and $\bar{\vartheta}_z^t$ be the random vectors in* (5.15) *with the parameters determined by the SE equations in Algorithm 5.4. Then, for any fixed t, the components of ϑ_x^t and ϑ_z^t converge empirically with bounded moments of order $k = 2$ as*

$$\lim_{N \to \infty} \vartheta_x^t \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}_x^t, \qquad \lim_{N \to \infty} \vartheta_z^t \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}_z^t, \tag{5.34}$$

where $\bar{\vartheta}_x^t$ and $\bar{\vartheta}_z^t$ are given in (5.15)*. In addition, for any t, the limits*

$$\lim_{N \to \infty} \boldsymbol{\theta}_x^t = \bar{\boldsymbol{\theta}}_x^t, \qquad \lim_{N \to \infty} \boldsymbol{\theta}_z^t = \bar{\boldsymbol{\theta}}_z^t, \tag{5.35a}$$

$$\lim_{N \to \infty} \tau_r^t = \bar{\tau}_r^t, \qquad \lim_{N \to \infty} \tau_p^t = \bar{\tau}_p^t, \tag{5.35b}$$

also hold almost surely.

The result is a natural generalization of Theorem 5.1 and provides a simple extension of the SE analysis to incorporate the adaptation. The SE analysis applies to essentially arbitrary adaptation functions. It particular, it can be used to analyze both the behavior of the adaptive GAMP algorithm with either ML and EM-GAMP adaptation functions in the previous section.

5.5.2 Asymptotic Consistency with ML Adaptation

We now use Theorem 5.2 to prove the asymptotic consistency of adaptive GAMP with the ML parameter estimation described in Section 5.4.1. To guarantee consistency of the adaptive GAMP algorithm, we need to impose certain *identifiability* conditions. To understand the conditions, given parameters $(\boldsymbol{\theta}_x, \alpha_r, \xi_r)$ and $(\boldsymbol{\theta}_z, \mathbf{K}_p)$, let

$$p_r(\cdot|\boldsymbol{\theta}_x, \alpha_r, \xi_r), \quad p_{y,p}(\cdot|\boldsymbol{\theta}_z, \mathbf{K}_p) \tag{5.36}$$

be the distributions of the random variables r and (y, p) in (5.25) and (5.28), respectively.

Definition 1. *Consider a family of distributions,* $\{p_x(x|\boldsymbol{\theta}_x) : \boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x\}$*, a set* S_x *of parameters* (α_r, ξ_r) *of a Gaussian channel, and the function* $\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r)$*. We say that* $p_x(x|\boldsymbol{\theta}_x)$ *is* identifiable with Gaussian outputs *with parameter set* S_x *and function* ϕ_x *if:*

(a) The sets S_x *and* $\boldsymbol{\Theta}_x$ *are compact.*

(b) For any true parameters $\boldsymbol{\theta}_x^* \in \boldsymbol{\Theta}_x$*, and* $(\alpha_r^*, \xi_r^*) \in S_x$*, the maximization*

$$\widehat{\boldsymbol{\theta}}_x = \underset{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x}{\arg\max} \left\{ \max_{(\alpha_r, \xi_r) \in S_x} \left\{ \mathbb{E}\left[\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r) | \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^* \right] \right\} \right\}, \tag{5.37}$$

is well-defined, unique and returns the true value, $\widehat{\boldsymbol{\theta}}_x = \boldsymbol{\theta}_x^*$*. The expectation in* (5.37) *is with respect to the distribution* $r \sim p_r(\cdot|\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*)$ *in* (5.36).

(c) For every $\boldsymbol{\theta}_x$ *and* α_r, ξ_r*, the function* $\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r)$ *is pseudo-Lipschitz continuous of order* $k = 2$ *in* r*. In addition, it is continuous in* $\boldsymbol{\theta}_x, \alpha_r, \xi_r$ *uniformly over* r *in the following sense: For every* $\epsilon > 0$ *and* $\widehat{\boldsymbol{\theta}}_x, \widehat{\alpha}_r, \widehat{\xi}_r$*, there exists an open neighborhood* U *of* $\widehat{\boldsymbol{\theta}}_x, \widehat{\alpha}_r, \widehat{\xi}_r$*, such that for all* $(\boldsymbol{\theta}_x, \alpha_r, \xi_r) \in U$ *and all* r*,*

$$|\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r) - \phi_x(r, \widehat{\boldsymbol{\theta}}_x, \widehat{\alpha}_r, \widehat{\xi}_r)| < \epsilon.$$

Definition 2. *Consider a family of conditional distributions,* $\{p_{y|z}(y|z, \boldsymbol{\theta}_z) : \boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z\}$ *generated by the mapping* $y = h(z, e, \boldsymbol{\theta}_z)$ *where* $e \sim p_e$ *is some random variable and* $h(z, w, \boldsymbol{\theta}_z)$ *is a scalar-valued function. Let* S_z *be a set of covariance matrices* $\mathbf{K}_p$ *and let* $\phi_z(y, p, \boldsymbol{\theta}_z, \mathbf{K}_p)$ *be some function. We say that the conditional distribution family* $p_{y|z}(\cdot|\cdot, \boldsymbol{\theta}_z)$ *is* identifiable with Gaussian inputs *with covariance set* S_z *and function* ϕ_z *if:*

(a) The parameter sets S_z *and* $\boldsymbol{\Theta}_z$ *are compact.*

(b) *For any true parameter* $\boldsymbol{\theta}_z^* \in \boldsymbol{\Theta}_z$ *and true covariance* $\mathbf{K}_p^*$*, the maximization*

$$\widehat{\boldsymbol{\theta}}_z = \arg\max_{\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z} \left\{ \max_{\mathbf{K}_p \in S_z} \left\{ \mathbb{E}\left[\phi_z(y, p, \boldsymbol{\theta}_z, \mathbf{K}_p) | \boldsymbol{\theta}_z^*, \mathbf{K}_p^* \right] \right\} \right\}, \tag{5.38}$$

is well-defined, unique and returns the true value, $\widehat{\boldsymbol{\theta}}_z = \boldsymbol{\theta}_z^*$*, The expectation in* (5.38) *is with respect to* $(y, p) \sim p_{y,p}(\cdot|\boldsymbol{\theta}_z^*, \mathbf{K}_p^*)$.

(c) *For every* $\boldsymbol{\theta}_z$ *and* $\mathbf{K}_p$*, the function* $\phi_z(y, p, \boldsymbol{\theta}_z, \mathbf{K}_p)$ *is pseudo-Lipschitz continuous in* (p, y) *of order* $k = 2$*. In addition, it is continuous in* $\boldsymbol{\theta}_z, \mathbf{K}_p$ *uniformly over* p *and* y.

Conditions (a) and (c) in both definitions are mild continuity and boundedness conditions. The main requirements is condition (b). Qualitatively, the definitions state that if r and (y, p) are generated by models of the form (5.25) and (5.28), then the parameters in those models can be estimated through maximization of the functions ϕ_x and ϕ_z. The functions ϕ_x and ϕ_z can be the log likelihood functions (5.24) and (5.27), although we permit other functions as well, since the maximization may be computationally simpler. Such functions are sometimes called *pseudo-likelihoods*. We will discuss these conditions and the role of the sets S_x and S_z in more detail in Section 5.6.

Assumption 5.3. *Let* $p_x(x|\boldsymbol{\theta}_x)$ *and* $p_{y|z}(y|z, \boldsymbol{\theta}_z)$ *be families of distributions and consider the adaptive GAMP algorithm, Algorithm 5.3, run on a sequence of problems, indexed by the dimension* N *satisfying the following assumptions:*

(a) *Same as Assumption 5.1(a) to (c) with* $k = 2$*. In addition, the distributions for the vector* x *is given by* $p_x(\cdot|\boldsymbol{\theta}_x^*)$ *for some true parameter* $\boldsymbol{\theta}_x^* \in \boldsymbol{\Theta}_x$ *and the conditional distribution of* y *given* z *is given by* $p_{y|z}(y|z, \boldsymbol{\theta}_z^*)$ *for some true parameter* $\boldsymbol{\theta}_z^* \in \boldsymbol{\Theta}_z$.

(b) *Same as Assumption 5.2(c).*

(c) *The adaptation functions are set to* (5.26) *and* (5.29).

Theorem 5.3. *Consider the outputs of the adaptive GAMP algorithm with ML adaptation as described in Assumption 5.3. Then, for any fixed* t,

(a) *The components of* ϑ_x^t *and* ϑ_z^t *in* (5.14) *converge empirically with bounded moments of order* $k = 2$ *as in* (5.34) *and the limits* (5.35) *hold almost surely.*

(b) *In addition, if* $(\alpha_r^t, \xi_r^t) \in S_x(\tau_r^t)$, *and the family of distributions* $\{p_x(\cdot|\boldsymbol{\theta}_x) : \boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x\}$ *is identifiable in Gaussian noise with parameter set* $S_x(\tau_r^t)$ *and pseudo-likelihood* ϕ_x, *then*

$$\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}_x^t = \bar{\boldsymbol{\theta}}_x^t = \boldsymbol{\theta}_x^* \tag{5.39}$$

almost surely.

(c) *Similarly, if* $\mathbf{K}_p^t \in S_z(\tau_p^t)$ *for some* t, *and the family of distributions* $p_{y|z}(\cdot|\boldsymbol{\theta}_z)$, $\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z$ *is identifiable with Gaussian inputs with parameter set* $S_z(\tau_p^t)$ *and pseudo-likelihood* ϕ_z *then*

$$\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}_z^t = \bar{\boldsymbol{\theta}}_z^t = \boldsymbol{\theta}_z^* \tag{5.40}$$

almost surely.

Proof: See Section 5.9.3.

Remarkably, the theorem shows that for a very large class of the parameterized distributions, adaptive GAMP with ML adaptation is able to asymptotically estimate the correct parameters. Moreover, there is asymptotically no performance loss between adaptive GAMP and a corresponding oracle GAMP algorithm that knows the correct parameters in the sense that the empirical distributions of the algorithm outputs are described by the same SE equations.

5.5.3 Computational Issues

While Theorem 5.3 shows that adaptive GAMP with ML adaptation can recover consistent parameter estimates, the ML optimizations in (5.26) and (5.29) theoretically need to be computed exactly. In general, these optimizations will be non-convex. This requirement can be seen as the main disadvantage of the ML adaptation proposed here relative to the EM-GAMP methods in [89–92]: while the proposed ML adaptation may have guaranteed consistency, the optimizations in each iteration may be non-convex. The EM iterations, in general are simpler.

Indeed, in the simulations in Section 5.7, we will need to approximate the optimization either through gradient ascent or other nonlinear optimization methods. Thus, the theory will not hold exactly. However, we will still observe a close match between the adaptive GAMP with an oracle GAMP with the correct parameters. Moreover, the ML adaptation is a non-convex optimization only over a number of variables only equal to the number of unknown parameters in $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$, not the vectors $\mathbf{x}$ and $\mathbf{z}$. Thus, for many practical problem, the overall optimization can be significantly simpler than the original non-convex problem.

5.6 Identifiability and Parameter Set Selection

In addition to the numerical optimization issues, Theorem 5.3 also imposes certain restrictions on the sets S_x and S_z over which the ML optimization must be performed. On the one hand, Theorem 5.3 requires that, to guarantee consistency, the sets must be sufficiently large to ensure that, for some iteration t, either $(\alpha_r^t, \xi_r^t) \in S_x(\tau_r^t)$ or $\mathbf{K}_p^t \in S_z(\tau_p^t)$. On the other hand, as we will see now, the sets may need to be constrained in order to satisfy the identifiability conditions in Definitions 1 and 2. In this section, we briefly provide some examples to illustrate under what cases these conditions can be met.

As discussed in the previous section, the main challenge in meeting the identifiability requirements in both Definitions 1 and 2 is condition (b). To understand this condition, we begin with the following simple lemma.

Lemma 5.1. *Consider the distributions p_r and $p_{y,p}$ in* (5.36).

(a) When ϕ_x is the log-likelihood function in (5.24)*, then condition (b) of Definition 1 is satisfied if the mapping*

$$(\boldsymbol{\theta}_x, \alpha_r, \xi_r) \mapsto p_r(\cdot|\boldsymbol{\theta}_x, \alpha_r, \xi_r) \tag{5.41}$$

is one-to-one in the set $\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x$ and $(\alpha_r, \xi_r) \in S_x$.

(b) Similarly, when ϕ_z is the log-likelihood function in (5.27)*, then condition (b) of Definition 2 is satisfied if the mapping*

$$(\boldsymbol{\theta}_z, \mathbf{K}_p) \mapsto p_{y,p}(\cdot|\boldsymbol{\theta}_z, \mathbf{K}_p) \tag{5.42}$$

is one-to-one in the set $\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z$ and $\mathbf{K}_z \in S_z$.

Proof: See Section 5.9.4.

Lemma 5.1 essentially states that if the true likelihood functions are used, then identifiability is equivalent to the parametrizations of the distributions r and (y, p) in (5.25) and (5.28) being unique. That is, with sufficient observations of these variables, we should be able to uniquely recover the parameter values. To understand this in this context of the adaptive GAMP algorithm, recall from the state evolution analysis, that the components of the vectors $\boldsymbol{r}^t$ and $(\mathbf{y}, \mathbf{p}^t)$ are asymptotically distributed as r or (y, p) in (5.25) and (5.28), respectively. Thus, if the parametrizations in (5.41) or (5.42) are not one-to-one, two different parameters values may give rise to the same asymptotic distributions on $\boldsymbol{r}^t$ and $(\mathbf{y}, \mathbf{p}^t)$. In this case, the adaptation functions in (5.26) and (5.29) that base the parameter estimates on $\boldsymbol{r}^t$ and $(\mathbf{y}, \mathbf{p}^t)$, cannot hope to distinguish between two such parameter values. On the other hand, if the parametrizations are one-to-one, the lemma shows that the ML parameter estimation will be able to correctly identify the parameter values. We now provide some examples.

5.6.1 Gaussian Mixtures

Suppose that x is a K-term Gaussian mixture with distribution,

$$x \sim \mathcal{N}(\mu_k, \tau_k) \text{ with probability } p_k,$$

with the unknown parameters being $\boldsymbol{\theta}_x = \{(\mu_k, \tau_k, p_k) : k = 1, \ldots, K\}$. Then, the variable r in (5.25) will also be a Gaussian mixture, but with different components

$$r \sim \mathcal{N}(\alpha_r \mu_k, \alpha_r^2 \tau_k + \xi_r) \text{ with probability } p_k.$$

It is easy to check that two parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}'_x$ will generically result in the same distribution on r if and only if

$$p'_k = p_k, \quad \alpha'_r \mu'_k = \alpha_r \mu_k, \tag{5.43a}$$

$$(\alpha'_r)^2 \tau'_k + \xi_r = \alpha_r^2 \tau_k + \xi_r, \tag{5.43b}$$

for $k = 1, \ldots, K$. That is, the component means, variances and probabilities must match.

Now, $\boldsymbol{\theta}_x$ has $3K$ parameters, so $(\boldsymbol{\theta}_x, \alpha_r, \xi_r)$ has a total of $3K + 2$ parameters. Since (5.43) has $3K$ constraints, the mapping (5.41) would in general need two additional constraints to be one-to-one to meet condition (b) of Definition 1. As one example for such constraints, we could know *a priori* that x has a known mean and variance, thereby providing two constraints. Alternatively, we could know that one of the mixtures, say $k = 1$, is strictly zero so that $\mu_1 = \tau_1 = 0$. This requirement would also provide two additional constraints. In either of these two examples, we need no additional constraints on the set S_x to meet the conditions of Lemma 5.1. Alternatively, if S_x can be restricted in some manner, then we could relax those assumptions.

5.6.2 AWGN output

Now consider an AWGN output channel where $p_{y|z}$ is given by

$$y = z + e, \quad e \sim \mathcal{N}(0, \sigma^2), \tag{5.44}$$

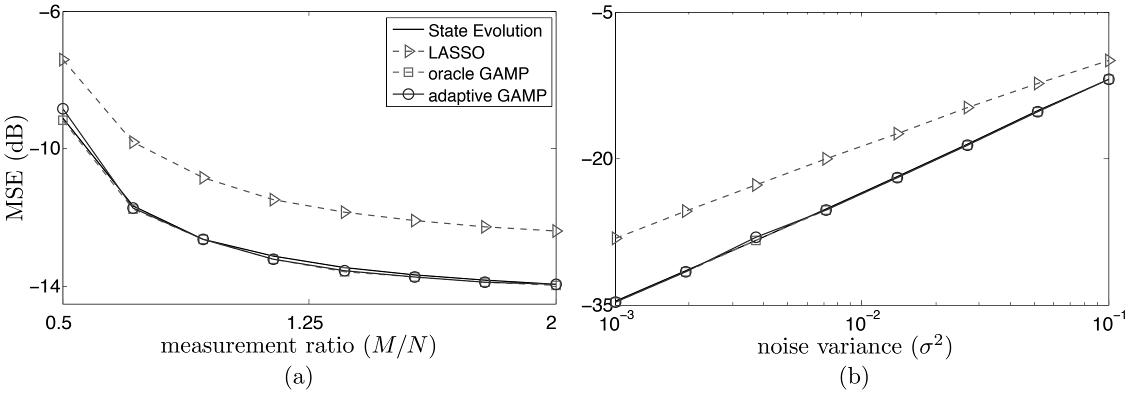
where e is independent of z. Here, the unknown parameter is $\boldsymbol{\theta}_z = \sigma^2$. Then, given a covariance matrix $\mathbf{K}_p$, the distribution $p_{y,p}$ in (5.36) is given by

$$(y, p) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad \mathbf{Q} = \mathbf{K}_p + \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix},$$

which is uniquely specified by the covariance matrix $\mathbf{Q}$. In this case, if we know the $(1,1)$-element of $\mathbf{K}_p$, we can determine σ^2 from $(1,1)$ element of $\mathbf{Q}$.

5.6.3 Initialization

One case where the covariance matrix $\mathbf{K}_p^t$ could be known is in the initial step of the algorithm. Suppose, for example, that we know the mean and variance of x, the random variable describing the components of $\mathbf{x}$. That is, the mean and variance of the input distribution $p_x(\cdot|\boldsymbol{\theta}_x)$ is the same for all values of $\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x$. In this case, even though we do not know the value of the parameter, we can perform the initialization in line 5 for the sum-product GAMP algorithm in Algorithm 5.1. Then, from the state evolution equations in Section 5.3.4, we would then know the initial covariance matrix $\mathbf{K}_p^t$ for $t = 0$ as given in (5.22).



(a) (b)

Figure 5.2: Reconstruction of a Bernoulli-Gaussian signal from noisy measurements. The average reconstruction MSE is plotted against (a) measurement ratio M/N and (b) AWGN variance σ^2. The plots illustrate that adaptive GAMP yields considerable improvement over ℓ_1–based LASSO estimator. Moreover, it matches the performance of oracle GAMP that knows the prior parameters.

5.7 Experimental Evaluation

5.7.1 Estimation of a Bernoulli-Gaussian input

Recent findings [98] suggest that there is considerable value in learning of priors p_x in the context of compressed sensing, which considers the estimation of sparse vectors $\mathbf{x}$ from underdetermined measurements $(M < N)$. It is known that estimators such as LASSO offer certain optimal min-max performance over a large class of sparse distributions [99]. However, for many particular distributions, there is a potentially large performance gap between LASSO and MMSE estimator with the correct prior. This gap was the main motivation for the work of Vila and Schniter [89,90] which showed large gains of the EM-GAMP method due to its ability to learn the prior.

Here, we illustrate the performance and asymptotic consistency of adaptive GAMP in a simple compressed sensing example. Specifically, we consider the estimation of a sparse vector $\mathbf{x} \in \mathbb{R}^N$ from M noisy measurements

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e} = \mathbf{z} + \mathbf{e},$$

where the additive noise $\mathbf{e}$ is random with i.i.d. entries $e_m \sim \mathcal{N}(0, \sigma^2)$. Here, the "output" channel is determined by the statistics of $\mathbf{e}$, which are assumed to be known to the estimator. So, there are no unknown parameters $\boldsymbol{\theta}_z$.

As a model for the sparse input vector $\mathbf{x}$, we assumed that the components are i.i.d. with the Bernoulli-Gaussian distribution,

$$x_n \sim \begin{cases} 0 & \text{prob } = 1 - \rho, \\ \mathcal{N}(0, \sigma_x^2) & \text{prob } = \rho \end{cases} \tag{5.45}$$

where ρ represents the probability that the component is non-zero (i.e. the vector's sparsity ratio) and σ_x^2 is the variance of the non-zero components. The parameters $\boldsymbol{\theta}_x = (\rho, \sigma_x^2)$ are treated as unknown.

Now, the Gaussian mixture in (5.45) has only two unknown parameters: ρ and σ_x^2. As described in Section 5.6.1, this mixture is sufficiently constrained so that if we apply the full ML estimation in (5.26) with no restrictions in the set S_x, we can identify the parameters correctly. We thus use this ML adaption in the first iteration and the above theory suggests that the algorithm should recover the correct parameters right away.

However, in our implementation, we continue to update the parameters at all iterations since there may be parameter errors on finite sample sizes. However, to simplify the ML adaptation, we can restrict the set selection S_x for iterations $t > 1$ as follows. Assuming the parameters were selected correctly up to some iteration $t - 1$, the adaptive GAMP algorithm should behave the same as an oracle sum-product GAMP algorithm with the correct parameters. Now, as described in Section 5.3.4, for the sum-product GAMP, the SE equations simplify so that $\alpha_r^t = 1$ and $\xi_r^t = \bar{\tau}_r^t$. Thus, the parameters α_r^t and ξ_r^t do not need to be estimated, and (5.26) conveniently simplifies to

$$H_x(\boldsymbol{r}, \tau_r) = \underset{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x}{\arg\max} \left\{ \frac{1}{N} \sum_{n=1}^{N} \log\left(p_r(r_n | \boldsymbol{\theta}_x, \tau_r)\right) \right\}, \tag{5.46}$$

where $\boldsymbol{\Theta}_x = [0, 1] \times [0, +\infty)$. In our implementation, we approximate the ML adaptation (5.46) with the EM update (5.30), which is run for several iterations. At each iteration of adaptive GAMP, we run iteratively the EM updates either until $\|\widehat{\boldsymbol{\theta}}_x^t - \widehat{\boldsymbol{\theta}}_x^{t-1}\|_2^2 / \|\widehat{\boldsymbol{\theta}}_x^{t-1}\|_2^2 \leq 10^{-4}$ for 3 consecutive iterations, or for a maximum of 200 iterations.

Fig. 5.2 illustrates the performance of adaptive GAMP on signals of length $N = 400$ generated with the parameters $\boldsymbol{\theta}_x = (\rho = 0.2, \sigma_x^2 = 5)$. The performance of adaptive

GAMP is compared to that of LASSO with MSE optimal regularization parameter, and oracle GAMP that knows the parameters of the prior exactly. For generating the graphs, we performed 1000 random trials by forming the measurement matrix **H** from i.i.d. zero-mean Gaussian random variables of variance $1/M$. In Figure 5.2(a), we keep the variance of the noise fixed to $\sigma^2 = 0.1$ and plot the average MSE of the reconstruction against the measurement ratio M/N. In Figure 5.2(b), we keep the measurement ratio fixed to $M/N = 0.75$ and plot the average MSE of the reconstruction against the noise variance σ^2. For completeness, we also provide the asymptotic MSE values computed via SE recursion. The results illustrate that GAMP significantly outperforms LASSO over the whole range of M/N and σ^2. Moreover, the results corroborate the consistency of adaptive GAMP which nearly achieves the reconstruction quality of oracle GAMP. Note also that in Figure 5.2 the average reconstruction times—across all realizations and undersampling rates—were 0.35, 0.06, and 0.22 seconds for LASSO, oracle GAMP, and adaptive GAMP, respectively. The results indicate that adaptive GAMP can be an effective method for estimation when the parameters of the problem are difficult to characterize and must be estimated from data.

5.7.2 Estimation of a Nonlinear Output Classification Function

As second example, we consider the estimation of the linear-nonlinear-Poisson (LNP) cascade model [81]. The model has been successfully used to characterize neural spike responses in early sensory pathways of the visual system. In the context of the LNP cascade model, the vector $\mathbf{x} \in \mathbb{R}^N$ represents the linear filter, which models the linear receptive field of the neuron. AMP techniques combined with the parameter estimation have been recently proposed for neural receptive field estimation and connectivity detection in [100].

As in Section 5.7.1, we model $\mathbf{x}$ as a Bernoulli-Gaussian vector with unknown parameters $\boldsymbol{\theta}_x = (\rho, \sigma_x^2)$. To obtain the measurements $\mathbf{y}$, the vector $\mathbf{z} = \mathbf{Hx}$ is passed through a component-wise nonlinearity u specified by

$$u(z) = \frac{1}{1 + \mathrm{e}^{-z}}. \tag{5.47}$$

The final measurement vector $\mathbf{y}$ is generated by a measurement channel with a condi-

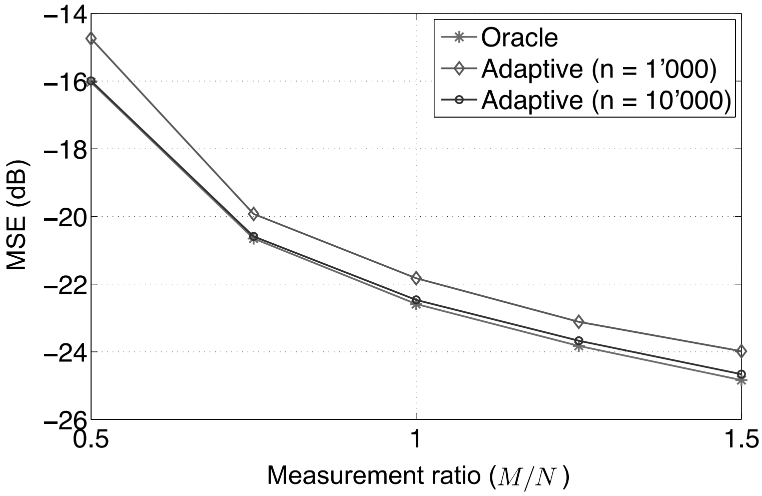


Figure 5.3: Identification of linear-nonlinear-Poisson cascade model. The average reconstruction MSE is plotted against the measurement ratio M/N. This plots illustrates near consistency of adaptive GAMP for large N.

tional density of the form

$$p_{y|z}(y_m|z_m, \boldsymbol{\theta}_z) = \frac{f(z_m)^{y_m}}{y_m!} \mathrm{e}^{-f(z_m)}, \tag{5.48}$$

where f denotes the nonlinearity given by

$$f(z; \boldsymbol{\theta}_z) = \exp\left(\sum_{i=1}^{r} \theta_{z,i} u^{i-1}(z)\right).$$

Adaptive GAMP can now be used to also estimate vector of polynomial coefficients $\boldsymbol{\theta}_z$, which together with $\mathbf{x}$, completely characterizes the LNP system.

The estimation of $\boldsymbol{\theta}_z$ is performed with ML estimator described in Section 5.4.1. We assume that the mean and variance of the vector $\mathbf{x}$ are known at iteration $t = 0$. As discussed in Section 5.6.3, this implies that for sum-product GAMP the covariance $\mathbf{K}_p^0$

is initially known and the optimization (5.29) simplifies to

$$H_z(\mathbf{p}, \mathbf{y}, \tau_p) = \underset{\boldsymbol{\theta}_z \in \boldsymbol{\Theta}_z}{\arg\max} \left\{ \frac{1}{M} \sum_{m=1}^{M} \log\left(p_y(y_m | \boldsymbol{\theta}_z)\right) \right\}, \tag{5.49}$$

where $\boldsymbol{\Theta}_z \subset \mathbb{R}^r$. The estimation of $\boldsymbol{\theta}_x$ is performed as in Section 5.7.1. As before, for iteration $t > 0$, we assume that the maximizations (5.46) and (5.49) yield correct parameter estimates $\widehat{\boldsymbol{\theta}}_x^t = \boldsymbol{\theta}_x$ and $\widehat{\boldsymbol{\theta}}_z^t = \boldsymbol{\theta}_z$, respectively. Thus we can conclude by induction that for $t > 0$ the adaptive GAMP algorithm should continue matching oracle GAMP for large enough N. In our simulations, we implemented (5.49) with a gradient ascend algorithm and run it until convergence.

In Fig. 5.3, we compare the reconstruction performance of adaptive GAMP against the oracle version that knows the true parameters $(\boldsymbol{\theta}_x, \boldsymbol{\theta}_z)$ exactly. We consider the vector $\mathbf{x}$ generated with true parameters $\boldsymbol{\theta}_x = (\rho = 0.1, \sigma_x^2 = 30)$. We consider the case $r = 3$ and set the parameters of the output channel to $\boldsymbol{\theta}_z = [-4.88, 7.41, 2.58]$. To illustrate the asymptotic consistency of the adaptive algorithm, we consider the signals of length $N = 1000$ and $N = 10000$. We perform 10 and 100 random trials for long and short signals, respectively, and plot the average MSE of the reconstruction against M/N. As expected, for large N, the performance of adaptive GAMP is nearly identical (within 0.15) to that of oracle GAMP. For this experiment the average reconstruction times for $N = 1000$ were 120.76 and 1031.5 seconds for oracle and adaptive GAMP, respectively, where the output updates were responsible for the majority of the computation time.

5.8 Summary

We have presented an adaptive GAMP method for the estimation of i.i.d. vectors $\mathbf{x}$ observed through a known linear transforms followed by an arbitrary, component-wise random transform. The procedure, which is a generalization of a popular EM-GAMP methodology, estimates both the vector $\mathbf{x}$ as well as parameters in the source and component-wise output transform. In the case of large i.i.d. Gaussian transforms, it is shown that the adaptive GAMP method with ML parameter estimation is provably asymptotically consistent in that the parameter estimates converge to the true values. This convergence result holds over a large class of models with essentially arbitrarily complex parameterizations. Moreover, the algorithm is computationally efficient since it reduces the vector-valued estimation problem to a sequence of scalar estimation problems in Gaussian noise. We believe that this method is applicable to a large class of linear-nonlinear models with provable guarantees can have applications in a wide range of problems. We have mentioned the use of the method for learning sparse priors in compressed sensing.

There are however several limitations that may be addressed in future work. Most significantly, the SE results are currently limited to large i.i.d. matrices. However, many matrices in practice are not well-modeled as large i.i.d. Recent work of [101] has attempted to understand the behavior of GAMP in non-asymptotic settings and an avenue of future work is to see if these results can be extended to adaptive GAMP.

Also, a critical assumption in our analysis is that the parameters $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_z$ are finite dimensional and whose dimensions do not grow. Another avenue of work would be see if the methods can be extended to non-parametric estimation of the densities in the adaptation steps or estimation with growing numbers of parameters.

Finally, as we discussed in Section 5.5.3, the ML adaptation is generally non-convex and thus must often be approximated. An open question is what tractable, approximate methods can be applied while guaranteeing consistency.

5.9 Appendix

5.9.1 Convergence of Empirical Distributions

Bayati and Montanari's analysis in [85] employs certain deterministic models on the vectors and then proves convergence properties of related empirical distributions. To apply the same analysis here, we need to review some of their definitions. We say a function $\phi : \mathbb{R}^r \to \mathbb{R}^s$ is *pseudo-Lipschitz* of order $k > 1$, if there exists an $L > 0$ such for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^r$,

$$\|\phi(\mathbf{x}) - \phi(\mathbf{y})\| \leq L(1 + \|\mathbf{x}\|^{k-1} + \|\mathbf{y}\|^{k-1})\|\mathbf{x} - \mathbf{y}\|.$$

Now suppose that for each $N = 1, 2, \ldots$, the set $\mathbf{v}^{(N)}$ is a set of vectors

$$\mathbf{v}^{(N)} = \{\mathbf{v}_i(N) : i = 1, \ldots, \ell(N)\}, \tag{5.50}$$

where each element $\mathbf{v}_i(N) \in \mathbb{R}^s$ and $\ell(N)$ is the number of elements in the set. Thus, $\mathbf{v}^{(N)}$ can itself be regarded as a vector with $s\ell(N)$ components. We say that $\mathbf{v}^{(N)}$ *empirically converges with bounded moments of order* k as $n \to \infty$ to a random vector $\mathbf{V}$ on $\mathbb{R}^s$ if: For all pseudo-Lipschitz continuous functions, ϕ, of order k,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{v}_n(N)) = \mathbb{E}[\phi(\mathbf{V})] < \infty.$$

When the nature of convergence is clear, we may write (with some abuse of notation)

$$\mathbf{v}^{(N)} \overset{\mathrm{PL}(k)}{\to} \mathbf{V} \text{ as } n \to \infty \quad \text{ or } \quad \lim_{N\to\infty} \mathbf{v}^{(N)} \overset{\mathrm{PL}(k)}{=} \mathbf{V}.$$

Finally, let $\mathscr{P}_k^s$ be the set of probability distributions on $\mathbb{R}^s$ with bounded kth moments, and suppose that $H : \mathscr{P}_k^s \to \mathbf{\Theta}$ is a functional $\mathscr{P}_k^s$ to some topological space $\mathbf{\Theta}$. Given a set $\mathbf{v}^{(N)}$ as in (5.50), write $H(\mathbf{v})$ for $H(p_{\mathbf{v}})$ where $p_{\mathbf{v}}$ is the empirical distribution on the components of $\mathbf{v}$. Also, given a random vector $\mathbf{V}$ with distribution $p_{\mathbf{V}}$ write $H(\mathbf{V})$ for $H(p_{\mathbf{V}})$. Then, we say that the functional H is *weakly pseudo-Lipschitz continuous* of order k if

$$\lim_{N\to\infty} \mathbf{v}^{(N)} \overset{\mathrm{PL}(k)}{=} \mathbf{V} \Longrightarrow \lim_{N\to\infty} H(\mathbf{v}^{(N)}) = H(\mathbf{V}),$$

where the limit on the right hand side is in the topology of $\mathbf{\Theta}$.

5.9.2 Proof of Theorem 5.2

We use the tilde superscript on quantities such as $\tilde{\mathbf{x}}^t, \tilde{\mathbf{r}}^t, \tilde{\tau}_r^t, \tilde{\mathbf{p}}^t, \tilde{\tau}_p^t, \tilde{\mathbf{s}}^t$, and $\tilde{\mathbf{z}}^t$ to denote values generated via a non-adaptive version of the GAMP. The non-adaptive GAMP algorithm has the same initial conditions as the adaptive algorithm (i.e. $\tilde{\mathbf{x}}^0 = \widehat{\mathbf{x}}^0, \tilde{\tau}_p^0 = \tau_p^0, \tilde{\mathbf{s}}^{-1} = \mathbf{s}^{-1} = 0$), but with $\widehat{\boldsymbol{\theta}}_x^t$ and $\widehat{\boldsymbol{\theta}}_z^t$ replaced by their deterministic limits $\bar{\boldsymbol{\theta}}_x^t$ and $\bar{\boldsymbol{\theta}}_z^t$, respectively. That is, we set

$$\begin{aligned} \tilde{\mathbf{z}}_m^t &= G_z^t(p_m^t, y_m, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t), \qquad \tilde{\mathbf{s}}_m^t = G_s^t(p_m^t, y_m, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t), \\ \tilde{\mathbf{x}}_n^{t+1} &= G_x^t(r_n^t, \tau_r^t, \bar{\boldsymbol{\theta}}_x^t). \end{aligned}$$

This non-adaptive algorithm is precisely the standard GAMP method. The results in that paper show that the outputs of the non-adaptive algorithm satisfy all the required limits from the SE analysis. That is,

$$\lim_{N\to\infty} \tilde{\vartheta}^t_x \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}^t_x, \qquad \lim_{N\to\infty} \tilde{\vartheta}^t_z \overset{\mathrm{PL}(k)}{=} \bar{\vartheta}^t_z,$$

where $\tilde{\vartheta}^t_x$ and $\tilde{\vartheta}^t_z$ are the sets generated by the non-adaptive GAMP algorithm:

$$\begin{aligned}\tilde{\vartheta}^t_x &\triangleq \left\{(x_n, \tilde{r}^t_n, \tilde{x}^{t+1}_n) : n = 1, \ldots, N\right\},\\ \tilde{\vartheta}^t_z &\triangleq \left\{(z_m, \tilde{z}^t_m, y_m, \tilde{p}^t_m) : m = 1, \ldots, M\right\}.\end{aligned}$$

The limits (5.34) are now proven through a continuity argument that shows that the adaptive and non-adaptive quantities must asymptotically agree with one another. Specifically, we will start by proving that the following limits holds almost surely for all $t \geq 0$

$$\lim_{N\to\infty} \Delta^t_x = \lim_{N\to\infty} \frac{1}{N} \|\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t\|^k_k = 0,, \tag{5.51a}$$

$$\lim_{N\to\infty} \Delta^t_{\tau_p} = \lim_{N\to\infty} |\tau^t_p - \tilde{\tau}^t_p| = 0 \tag{5.51b}$$

where $\|\cdot\|_k$ is usual the k-norm. Moreover, in the course of proving (5.51), we will also show that the following limits hold almost surely

$$\lim_{M\to\infty} \Delta^t_p = \lim_{M\to\infty} \frac{1}{M} \|\mathbf{p}^t - \tilde{\mathbf{p}}^t\|^k_k = 0, \tag{5.52a}$$

$$\lim_{N\to\infty} \Delta^t_r = \lim_{N\to\infty} \frac{1}{N} \|\boldsymbol{r}^t - \tilde{\boldsymbol{r}}^t\|^k_k = 0, \tag{5.52b}$$

$$\lim_{M\to\infty} \Delta^t_s = \lim_{M\to\infty} \frac{1}{M} \|\mathbf{s}^t - \tilde{\mathbf{s}}^t\|^k_k = 0, \tag{5.52c}$$

$$\lim_{M\to\infty} \Delta^t_z = \lim_{M\to\infty} \frac{1}{M} \|\widehat{\mathbf{z}}^t - \tilde{\mathbf{z}}^t\|^k_k = 0, \tag{5.52d}$$

$$\lim_{N\to\infty} \Delta^t_{\tau_r} = \lim_{N\to\infty} |\tau^t_r - \tilde{\tau}^t_r| = 0, \tag{5.52e}$$

$$\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}^t_x = \bar{\boldsymbol{\theta}}^t_x, \tag{5.52f}$$

$$\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}^t_z = \bar{\boldsymbol{\theta}}^t_z, \tag{5.52g}$$

The proof of the limits (5.51) and (5.52) is achieved by an induction on t. Although we only need to show the above limits for $k = 2$, most of the arguments hold for arbitrary $k \geq 2$. We thus present the general derivation where possible.

To begin the induction argument, first note that the non-adaptive algorithm has the same initial conditions as the adaptive algorithm. Thus the limits (5.51) and (5.52c) hold for $t = 0$ and $t = -1$, respectively.

We now proceed by induction. Suppose that $t \geq 0$ and the limits (5.51) and (5.52c) hold for some t and $t-1$, respectively. Since $\mathbf{H}$ has i.i.d. components with zero mean and variance $1/M$, it follows from the Marčenko-Pastur Theorem [102] that its 2-norm operator norm is bounded. That is, there exists a constant C_A such that almost surely we have

$$\lim_{N\to\infty} \|\mathbf{H}\|_k \leq C_A, \quad \lim_{N\to\infty} \|\mathbf{H}^T\|_k \leq C_A. \tag{5.53}$$

This bound is the only part of the proof that specifically requires $k = 2$. From (5.53), we obtain

$$\begin{aligned}
\|\mathbf{p}^t - \tilde{\mathbf{p}}^t\|_k &= \|\mathbf{H}\widehat{\mathbf{x}}^t - \tau_p^t \mathbf{s}^{t-1} - \mathbf{H}\tilde{\mathbf{x}}^t + \tilde{\tau}_p^t \tilde{\mathbf{s}}^{t-1}\|_k \\
&= \|\mathbf{H}(\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t) + \tau_p^t(\tilde{\mathbf{s}}^{t-1} - \mathbf{s}^{t-1}) + (\tilde{\tau}_p^t - \tau_p^t)\tilde{\mathbf{s}}^{t-1}\|_k \\
&\leq \|\mathbf{H}(\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t)\|_k + |\tau_p^t|\|\tilde{\mathbf{s}}^{t-1} - \mathbf{s}^{t-1}\|_k + |\tilde{\tau}_p^t - \tau_p^t|\|\tilde{\mathbf{s}}^{t-1}\|_k \\
&\overset{(a)}{\leq} \|\mathbf{H}\|_k\|\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t\|_k + |\tau_p^t|\|\tilde{\mathbf{s}}^{t-1} - \mathbf{s}^{t-1}\|_k + |\tilde{\tau}_p^t - \tau_p^t|\|\tilde{\mathbf{s}}^{t-1}\|_k \\
&\leq C_A\|\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t\|_k + |\tau_p^t|\|\mathbf{s}^{t-1} - \tilde{\mathbf{s}}^{t-1}\|_k + |\tau_p^t - \tilde{\tau}_p^t|\|\tilde{\mathbf{s}}^{t-1}\|_k
\end{aligned} \tag{5.54}$$

almost surely, where (a) is due to the norm inequality $\|\mathbf{H}\mathbf{x}\|_k \leq \|\mathbf{H}\|_k\|\mathbf{x}\|_k$. Since $k \geq 1$, we have that for any positive numbers a and b

$$(a+b)^k \leq 2^k(a^k + b^k). \tag{5.55}$$

Applying the inequality (5.55) into (5.54), we obtain

$$\begin{aligned}
\frac{1}{M}\|\mathbf{p}^t - \tilde{\mathbf{p}}^t\|_k^k &\leq \frac{1}{M}\left(C_A\|\widehat{\mathbf{x}}^t - \tilde{\mathbf{x}}^t\|_k + |\tau_p^t|\|\mathbf{s}^{t-1} - \tilde{\mathbf{s}}^{t-1}\|_k + \Delta_{\tau_p}^t\|\tilde{\mathbf{s}}^{t-1}\|_k\right)^k \\
&\leq 2^k C_A \frac{N}{M}\Delta_x^t + 2^k|\tau_p^t|^k\Delta_s^{t-1} + 2^k(\Delta_{\tau_p}^t)^k\left(\frac{1}{M}\|\tilde{\mathbf{s}}^{t-1}\|_k^k\right).
\end{aligned} \tag{5.56}$$

Now, since $\tilde{\mathbf{s}}^t$ and $\tilde{\tau}_p^t$ are the outputs of the non-adaptive algorithm, they satisfy the limits

$$\lim_{N\to\infty} \frac{1}{M}\|\tilde{\mathbf{s}}^t\|_k^k = \lim_{N\to\infty} \frac{1}{M}\sum_{m=1}^{M} |\tilde{s}_m^t|^k = \mathbb{E}\left[|S^t|^k\right] < \infty, \tag{5.57a}$$

$$\lim_{N\to\infty} \tilde{\tau}_p^t = \bar{\tau}_p^t < \infty. \tag{5.57b}$$

Now, the induction hypotheses state that Δ_x^t, Δ_s^{t-1} and $\Delta_{\tau_p}^t \to 0$. Applying these along the bounds (5.57a), and the fact that $N/M \to \beta$, we obtain (5.52a).

To establish (5.52g), we first prove the empirical convergence of $(\mathbf{p}^t, \mathbf{y})$ to (p^t, y). Towards this end, let $\phi(p, y)$ be any pseudo-Lipschitz continuous function ϕ of order k. Then

$$\begin{aligned}
&\left|\frac{1}{M}\sum_{m=1}^{M}\phi(p_m^t, y_m) - \mathbb{E}\left[\phi(p^t, y)\right]\right| \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\left|\phi(p_m^t, y_m) - \phi(\tilde{p}_m^t, y_m)\right| \\
&+ \left|\frac{1}{M}\sum_{m=1}^{M}\phi(\tilde{p}_m^t, y_m) - \mathbb{E}\left[\phi(p^t, y)\right]\right| \\
&\overset{(a)}{\leq} \frac{L}{M}\sum_{m=1}^{M}\left(1 + |p_m^t|^{k-1} + |\tilde{p}_m^t|^{k-1} + |y_m|^{k-1}\right)|p_m^t - \tilde{p}_m^t| \\
&+ \left|\frac{1}{M}\sum_{m=1}^{M}\phi(\tilde{p}_m^t, y_m) - \mathbb{E}\left[\phi(p^t, y)\right]\right| \\
&\overset{(b)}{\leq} LC\Delta_p^t + \left|\frac{1}{M}\sum_{m=1}^{M}\phi(\tilde{p}_m^t, y_m) - \mathbb{E}\left[\phi(p^t, y)\right]\right|.
\end{aligned} \tag{5.58}$$

In (a) we use the fact that ϕ is pseudo-Lipschitz, and in (b) we use Hölder's inequality

$|\widehat{\mathbf{x}}^T\mathbf{y}| = \|\mathbf{x}\|_k\|\mathbf{y}\|_q$ with $q = p/(p-1)$. The constant is defined as

$$\begin{aligned}
C &\triangleq \left[\frac{1}{M}\sum_{m=1}^{M}\left(1+|p_m^t|^{k-1}+|\tilde{p}_m^t|^{k-1}+|y_m|^{k-1}\right)\right]^{k/(k-1)} \\
&\leq \frac{1}{M}\sum_{m=1}^{M}\left(1+|p_m^t|^{k-1}+|\tilde{p}_m^t|^{k-1}+|y_m|^{k-1}\right)^{k/(k-1)} \\
&\leq \text{const} \times \left[1+\left(\frac{1}{M}\|\mathbf{p}^t\|_k^k\right)^{\frac{k-1}{k}}\right. \\
&\quad \left. +\left(\frac{1}{M}\|\tilde{\mathbf{p}}^t\|_k^k\right)^{\frac{k-1}{k}}+\left(\frac{1}{M}\|\mathbf{y}\|_k^k\right)^{\frac{k-1}{k}}\right],
\end{aligned} \tag{5.59}$$

where the first step is from Jensen's inequality. Since $(\tilde{\mathbf{p}}^t, \mathbf{y})$ satisfy the limits for the non-adaptive algorithm, we have:

$$\lim_{N\to\infty}\frac{1}{M}\|\tilde{\mathbf{p}}^t\|_k^k = \lim_{N\to\infty}\frac{1}{M}\sum_{m=1}^{M}|\tilde{p}_m^t|^k = \mathbb{E}\left[|p^t|^k\right] < \infty \tag{5.60a}$$

$$\lim_{N\to\infty}\frac{1}{M}\|\mathbf{y}\|_k^k = \lim_{N\to\infty}\frac{1}{M}\sum_{m=1}^{M}|y_m|^k = \mathbb{E}\left[|y|^k\right] < \infty \tag{5.60b}$$

Also, from the induction hypothesis (5.52a), it follows that the adaptive output must satisfy the same limit

$$\lim_{N\to\infty}\frac{1}{M}\|\mathbf{p}^t\|_k^k = \lim_{N\to\infty}\frac{1}{M}\sum_{m=1}^{M}|p_m^t|^k = \mathbb{E}\left[|p^t|^k\right] < \infty. \tag{5.61}$$

Combining (5.58), (5.59), (5.60), (5.61), (5.52a) we conclude that for all $t \geq 0$

$$\lim_{N\to\infty}(\mathbf{p}^t, \mathbf{y}) \overset{\mathrm{PL}(k)}{=} (p^t, y). \tag{5.62}$$

The limit (5.62) along with (5.51b) and the continuity condition on H_z^t in Assumption 5.1(d) prove the limit in (5.52g).

The limit (5.52a) together with continuity conditions on G_z^t in Assumptions 5.1 show that (5.52c), (5.52d) and (5.52e) hold for t. For example, to show (5.52d), we consider the limit $M \to \infty$ of the following expression

$$\begin{aligned}\frac{1}{M}\|\widehat{\mathbf{z}}^t - \tilde{\mathbf{z}}^t\|_k^k &= \frac{1}{M}\|G_z^t(\mathbf{p}^t, \mathbf{y}, \tau_p^t, \widehat{\boldsymbol{\theta}}_z^t) - G_z^t(\tilde{\mathbf{p}}^t, \mathbf{y}, \tau_p^t, \bar{\boldsymbol{\theta}}_z^t)\|_k^k \\ &\overset{\text{(a)}}{\leq} \frac{L}{M}\|\mathbf{p}^t - \tilde{\mathbf{p}}^t\|_k^k = L\Delta_p^t,\end{aligned}$$

where at (a) we used the Lipschitz continuity assumption. Similar arguments can be used for (5.52c) and (5.52e).

To prove (5.52b), we proceed exactly as for (5.52a). Due to the continuity assumptions on H_x, this limit in turn shows that (5.52f) holds almost surely. Then, (5.51a) and (5.51b) follow directly from the continuity of G_x in Assumptions 5.1, together with (5.52b) and (5.52f). We have thus shown that if the limits (5.51) and (5.52) hold for some t, they hold for $t + 1$. Thus, by induction they hold for all t.

Finally, to establish (5.34), let ϕ be any pseudo-Lipschitz continuous function $\phi(x, r, \widehat{x})$, and define

$$\epsilon^t \triangleq \left| \frac{1}{N}\sum_{n=1}^{N} \phi(x_n, \tilde{r}_n^t, \tilde{x}_n^{t+1}) - \mathbb{E}\left[\phi(x, r^t, \widehat{x}^{t+1})\right] \right|, \tag{5.63}$$

which, due to convergence of non-adaptive GAMP, can be made arbitrarily small by choosing N large enough. Then, consider

$$
\begin{aligned}
&\left|\frac{1}{N}\sum_{n=1}^{N}\phi(x_n,\hat{r}_n^t,\widehat{x}_n^{t+1})-\mathbb{E}\big[\phi(x,r^t,\widehat{x}^{t+1})\big]\right| \\
&\leq \epsilon_N^t+\frac{1}{N}\sum_{n=1}^{N}\left|\phi(x_n,\hat{r}_n^t,\widehat{x}_n^{t+1})-\phi(x_n,\tilde{r}_n^t,\tilde{x}_n^{t+1})\right| \\
&\overset{(a)}{\leq}\epsilon_N^t+L\|\boldsymbol{r}^t-\tilde{\mathbf{r}}^t\|_1+L\|\hat{\mathbf{x}}^{t+1}-\tilde{\mathbf{x}}^{t+1}\|_1 \\
&+\frac{L'}{N}\sum_{n=1}^{N}\big(|\hat{r}_n^t|^{k-1}+|\tilde{r}_n^t|^{k-1}\big)\big(|\hat{r}_n^t-\tilde{r}_n^t|+|\widehat{x}_n^{t+1}-\tilde{x}_n^{t+1}|\big) \\
&+\frac{L'}{N}\sum_{n=1}^{N}\big(|\widehat{x}_n^{t+1}|^{k-1}+|\tilde{x}_n^{t+1}|^{k-1}\big)\big(|\hat{r}_n^t-\tilde{r}_n^t|+|\widehat{x}_n^{t+1}-\tilde{x}_n^{t+1}|\big) \\
&\overset{(b)}{\leq}\epsilon_N^t+L\left(\Delta_r^t\right)^{\frac{1}{k}}+L\left(\Delta_x^t\right)^{\frac{1}{k}} \\
&+L'\left(\Delta_r^t\right)^{\frac{1}{k}}\left((\tilde{M}_x^{t+1})^{\frac{k-1}{k}}+(\hat{M}_x^{t+1})^{\frac{k-1}{k}}+(\tilde{M}_r^t)^{\frac{k-1}{k}}+(\hat{M}_r^t)^{\frac{k-1}{k}}\right) \\
&+L'\left(\Delta_x^t\right)^{\frac{1}{k}}\left((\tilde{M}_x^{t+1})^{\frac{k-1}{k}}+(\hat{M}_x^{t+1})^{\frac{k-1}{k}}+(\tilde{M}_r^t)^{\frac{k-1}{k}}+(\hat{M}_r^t)^{\frac{k-1}{k}}\right)
\end{aligned}
\tag{5.64}
$$

where $L,\, L'$ are constants independent of N and

$$
\begin{aligned}
\hat{M}_x^{t+1} &\triangleq \frac{1}{N}\left\|\hat{\mathbf{x}}^{t+1}\right\|_k^k, \quad & \hat{M}_r^t &\triangleq \frac{1}{N}\left\|\boldsymbol{r}^t\right\|_k^k, \\
\tilde{M}_x^{t+1} &\triangleq \frac{1}{N}\left\|\tilde{\mathbf{x}}^{t+1}\right\|_k^k, \quad & \tilde{M}_r^t &\triangleq \frac{1}{N}\left\|\tilde{\boldsymbol{r}}^t\right\|_k^k
\end{aligned}
$$

In (a) we use the fact that ϕ is pseudo-Lipshitz, in (b) we use ℓ_p-norm equivalence $\|\mathbf{x}\|_1 \leq N^{1-1/p}\|\mathbf{x}\|_k$ and Hölder's inequality $|\widehat{\mathbf{x}}^T\mathbf{y}| = \|\mathbf{x}\|_k\|\mathbf{y}\|_q$ with $q = p/(p-1)$. By applying of (5.51a), (5.52b) and since, $\hat{M}_x^{t+1}$, $\tilde{M}_x^{t+1}$, $\hat{M}_r^t$, and $\tilde{M}_r^t$ converge to a finite value we can obtain the first equation of (5.34) by taking $N \to \infty$. The second equation in (5.34) can be shown in a similar way. This proves the limits (5.34).

Also, the first two limits in (5.35) are a consequence of (5.52f) and (5.52f). The second two limits follow from continuity assumptions in Assumption 5.1(e) and the convergence of the empirical distributions in (5.34). This completes the proof.

5.9.3 Proof of Theorem 5.3

Part (a) of Theorem 5.3 is an application of Theorem 5.2. To apply this general result, first observe that Assumptions 5.3(a) and (c) immediately imply the corresponding items in Assumptions 5.2. So, we only need to verify the continuity condition in Assumption 5.2(b) for the adaptation functions in (5.26) and (5.29).

We begin by proving the continuity of H_z^t. Fix t, and let $(\mathbf{y}^{(N)}, \mathbf{p}^{(N)})$ be a sequence of vectors and $\tau_p^{(N)}$ be a sequence of scalars such that

$$\lim_{N\to\infty} (\mathbf{y}^{(N)}, \mathbf{p}^{(N)}) \overset{\mathrm{PL}(p)}{=} (y, p^t) \qquad \lim_{N\to\infty} \tau_p^{(N)} = \bar{\tau}_p^t, \tag{5.65}$$

where (y, p^t) and $\bar{\tau}_p^t$ are the outputs of the state evolution equations. For each N, let

$$\widehat{\boldsymbol{\theta}}_z^{(N)} \triangleq H_z^t(\mathbf{y}^{(N)}, \mathbf{p}^{(N)}, \tau_p^{(N)}). \tag{5.66}$$

We wish to show that $\widehat{\boldsymbol{\theta}}_z^{(N)} \to \boldsymbol{\theta}_z^*$, the true parameter. Since $\widehat{\boldsymbol{\theta}}_z^{(N)} \in \boldsymbol{\Theta}_z$ and $\boldsymbol{\Theta}_z$ is compact, it suffices to show that any limit point of any convergent subsequence is equal to $\boldsymbol{\theta}_z^*$. So, suppose that $\widehat{\boldsymbol{\theta}}_z^{(N)} \to \widehat{\boldsymbol{\theta}}_z$ to some limit point $\widehat{\boldsymbol{\theta}}_z$ on some subsequence $\widehat{\boldsymbol{\theta}}_z^{(N)}$.

From $\widehat{\boldsymbol{\theta}}_z^{(N)}$ and the definition (5.29), it follows that

$$\frac{1}{M}\sum_{m=1}^{M} \phi_z(p_m^{(N)}, y_m^{(N)}, \widehat{\boldsymbol{\theta}}_z^{(N)}, \mathbf{K}_p) \geq \frac{1}{M}\sum_{m=1}^{M} \phi_z(p_m^{(N)}, y_m^{(N)}, \boldsymbol{\theta}_z^*, \mathbf{K}_p), \tag{5.67}$$

where $\mathbf{K}_p \in S_z(\tau_p^{(N)})$ is the solution of the first maximization of (5.29). Now, since $\tau_p^{(N)} \to \bar{\tau}_p^t$ and $\widehat{\boldsymbol{\theta}}_z^{(N)} \to \widehat{\boldsymbol{\theta}}_z$, we apply the continuity condition in Definition 2(c) to obtain

$$\liminf_{N\to\infty} \frac{1}{M}\sum_{m=1}^{M} \left[\phi_z(p_m^{(N)}, y_m^{(N)}, \widehat{\boldsymbol{\theta}}_z, \mathbf{K}_p) - \phi_z(p_m^{(N)}, y_m^{(N)}, \boldsymbol{\theta}_z^*, \mathbf{K}_p)\right] \geq 0. \tag{5.68}$$

Also, the limit (5.65) and the fact that ϕ_z is pseuedo-Lipschitz continuous of order k implies that

$$\mathbb{E}[\phi_z(p^t, y, \widehat{\boldsymbol{\theta}}_z, \mathbf{K}_p)] \geq \mathbb{E}[\phi_z(p^t, y, \boldsymbol{\theta}_z^*, \mathbf{K}_p)]. \tag{5.69}$$

But, Property (b) of Definition 2 shows that $\boldsymbol{\theta}_z^*$ is the unique maxima of the right-hand side, so

$$\mathbb{E}[\phi_z(p^t, y, \widehat{\boldsymbol{\theta}}_z, \mathbf{K}_p)] = \mathbb{E}[\phi_z(p^t, y, \boldsymbol{\theta}_z^*, \mathbf{K}_p)], \tag{5.70}$$

with $\widehat{\boldsymbol{\theta}}_z = \boldsymbol{\theta}_z^*$. Since this limit point is the same for all convergent subsequences, we see that $\widehat{\boldsymbol{\theta}}_z^{(N)} \to \boldsymbol{\theta}_z^*$ over the entire sequence. We have thus shown that given limits (5.65), the outputs of the adaptation function converge as

$$H_z^t(\mathbf{y}^{(N)}, \mathbf{p}^{(N)}, \tau_p^{(N)}) = \widehat{\boldsymbol{\theta}}_z^{(N)} \to \boldsymbol{\theta}_z^* = H_z^t(y, p^t, \bar{\tau}_p^t).$$

Thus, the continuity condition on H_z^t in Assumption 5.2(b) is satisfied. The analogous continuity condition on H_x^t can be proven in a similar manner.

Therefore, all the conditions of Assumption 5.2 are satisfied and we can apply Theorem 5.2. Part (a) of Theorem 5.3 immediately follows from Theorem 5.2.

So, it remains to show parts (b) and (c) of Theorem 5.3. We will only prove (b); the proof of (c) is similar. Also, since we have already established (5.35), we only need to show that the output of the SE equations matches the true parameter. That is, we need to show $\bar{\boldsymbol{\theta}}_x^t = \boldsymbol{\theta}_x^*$. This fact follows immediately from the selection of the adaptation functions:

$$\begin{aligned}
\bar{\boldsymbol{\theta}}_x^t &\overset{(a)}{=} H_x^t(r^t, \bar{\tau}_r^t) \\
&\overset{(b)}{=} \operatorname*{arg\,max}_{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x} \left\{ \max_{(\alpha_r, \xi_r) \in S_x(\bar{\tau}_r^t)} \left\{ \mathbb{E}\left[\phi_x(r^t, \boldsymbol{\theta}_x, \alpha_r, \xi_r)\right] \right\} \right\} \\
&\overset{(c)}{=} \operatorname*{arg\,max}_{\boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x} \left\{ \max_{(\alpha_r, \xi_r) \in S_x(\bar{\tau}_r^t)} \left\{ \mathbb{E}\left[\phi_x(\alpha_r^t x + v^t, \boldsymbol{\theta}_x, \alpha_r, \xi_r) | \boldsymbol{\theta}_x^*, \xi_r^t\right] \right\} \right\} && (5.71) \\
&\overset{(d)}{=} \boldsymbol{\theta}_x^* && (5.72)
\end{aligned}$$

where (a) follows from the SE equation (5.33a); (b) is the definition of the ML adaptation function H_x^t when interpreted as a functional on a random variable r^t; (c) is the definition of the random variable r^t in (5.16) where $v^t \sim \mathcal{N}(0, \xi_r^t)$; and (d) follows from Definition 1(b) and the hypothesis that $(\alpha_r^*, \xi_r^*) \in S_x(\bar{\tau}_r^t)$. Thus, we have proven that $\bar{\boldsymbol{\theta}}_x^t = \boldsymbol{\theta}_x^*$, and this completes the proof of part (b) of Theorem 5.3. The proof of part (c) is similar.

5.9.4 Proof of Lemma 5.1

We will just prove part (a). The proof of (b) is similar. Suppose that $r \sim p_r(\cdot|\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*)$ for some true parameter $\boldsymbol{\theta}_x^*$ and (α_r^*, ξ_r^*). Let

$$L(\boldsymbol{\theta}_x, \alpha_r, \xi_r) \triangleq \mathbb{E}\big[\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r) \mid \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*\big],$$

be the expected value of ϕ_x under the true parameters for r. According to Definition 1(b), we need to show that L is maximized uniquely at $(\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*)$. To this end, consider any other parameter set $(\boldsymbol{\theta}_x, \alpha_r, \xi_r)$. Then, if ϕ_x is the log-likelihood function in (5.24),

$$\begin{aligned} L(\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*) - L(\boldsymbol{\theta}_x, \alpha_r, \xi_r) & \\ &\overset{(a)}{=} \mathbb{E}\big[\phi_x(r, \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*) \mid \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*\big] \\ &\quad - \mathbb{E}\big[\phi_x(r, \boldsymbol{\theta}_x, \alpha_r, \xi_r) \mid \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*\big] \\ &\overset{(b)}{=} \mathbb{E}\big[\log p_r(r|\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*) \mid \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*\big] \\ &\quad - \mathbb{E}\big[\log p_r(r|\boldsymbol{\theta}_x, \alpha_r, \xi_r) \mid \boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*\big] \\ &\overset{(c)}{=} D\big(p_r(\cdot|\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*), p_r(\cdot|\boldsymbol{\theta}_x, \alpha_r, \xi_r)\big) \end{aligned} \tag{5.73}$$

where (a) follows from the definition of L; (b) follows from the fact that ϕ_x is the log likelihood in (5.24) and (c) is the Kullback-Liebler divergence. Now, if

$$(\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*) \neq (\boldsymbol{\theta}_x, \alpha_r, \xi_r),$$

the hypothesis that the map (5.41) is one-to-one implies that the two distributions in (5.73) are not equal. Therefore, the Kullback-Liebler divergence will be strictly positive [103] and thus the function L is uniquely maximized at $(\boldsymbol{\theta}_x^*, \alpha_r^*, \xi_r^*)$.

Algorithm 5.3: Adaptive GAMP

```
input:  the data $\mathbf{y}$, estimation functions $G_x^t$, $G_s^t$ and $G_z^t$,
        adaptation functions $H_x^t$ and $H_z^t$,
        a constant $\|\mathbf{H}\|_F^2$, and an efficient implementation of $\mathbf{H}$ and $\mathbf{H}^T$
set:  $t \leftarrow 0$, $\mathbf{s}^{t-1} \leftarrow 0$ and select some initial values for $\widehat{\mathbf{x}}^0$ and $\tau_x^0$.
repeat
    {Output node update}
    $\tau_p^t \leftarrow \|\mathbf{H}\|_F^2 \tau_x^t / M$
    $\mathbf{p}^t \leftarrow \mathbf{H}\widehat{\mathbf{x}}^t - \mathbf{s}^{t-1}\tau_p^t$
    $\widehat{\boldsymbol{\theta}}_z^t \leftarrow H_z^t(\mathbf{p}^t, \mathbf{y}, \tau_p^t)$
    $\widehat{z}_m^t \leftarrow G_z^t(p_m^t, y_m, \tau_p^t, \widehat{\boldsymbol{\theta}}_z^t), \quad (m = 1, \ldots, M)$
    $s_m^t \leftarrow G_s^t(p_m^t, y_m, \tau_p^t, \widehat{\boldsymbol{\theta}}_z^t), \quad (m = 1, \ldots, M)$
    $\tau_s^t \leftarrow -(1/M)\sum_m \partial G_s^t(p_m^t, y_m, \tau_p^t, \widehat{\boldsymbol{\theta}}_z^t)/\partial p_m^t$

    {Input node update}
    $1/\tau_r^t \leftarrow \|\mathbf{H}\|_F^2 \tau_s^t / N$
    $\boldsymbol{r}^t \leftarrow \mathbf{x}^t + \tau_r^t \mathbf{H}^T \mathbf{s}^t$
    $\widehat{\boldsymbol{\theta}}_x^t \leftarrow H_x^t(\boldsymbol{r}^t, \tau_r^t)$
    $\widehat{x}_n^{t+1} \leftarrow G_x^t(r_n^t, \tau_r^t, \widehat{\boldsymbol{\theta}}_x^t) \quad (n = 1, \ldots, N)$
    $\tau_x^{t+1} \leftarrow (\tau_r^t/N)\sum_n \partial G_x^t(r_n^t, \tau_r^t, \widehat{\boldsymbol{\theta}}_x^t)/\partial r_n$
until stopping criterion
return  $\widehat{\mathbf{x}}^{t+1}$
```

Algorithm 5.4: Adaptive GAMP State Evolution

Given the distributions in Assumption 5.1, compute the sequence of parameters as follows:

- **Initialization:** Set $t = 0$ with

$$\mathbf{K}_x^0 = \mathrm{cov}(x, \widehat{x}^0), \qquad \bar{\tau}_x^0 = \tau_x^0, \tag{5.31}$$

  where the expectation is over the random variables $(x, \widehat{x}^0)$ in Assumption 5.1(b) and τ_x^0 is the initial value in the GAMP algorithm.

- **Output node update:** Compute the variables associated with the output nodes Compute the variables

$$\bar{\tau}_p^t = \beta \bar{\tau}_x^t, \qquad \mathbf{K}_p^t = \beta \mathbf{K}_x^t, \tag{5.32a}$$

$$\bar{\boldsymbol{\theta}}_z^t = H_z^t(p^t, y, \bar{\tau}_p^t), \tag{5.32b}$$

$$\bar{\tau}_r^t = -\left(\mathbb{E}\left[\frac{\partial}{\partial p} G_s^t(p^t, y, \bar{\tau}_p^t, \bar{\boldsymbol{\theta}}_z^t)\right]\right)^{-1}, \tag{5.32c}$$

$$\xi_r^t = (\bar{\tau}_r^t)^2 \mathbb{E}\left[G_s^t(p^t, y, \bar{\tau}_p^t, \bar{\boldsymbol{\theta}}_z^t)\right], \tag{5.32d}$$

$$\alpha_r^t = \bar{\tau}_r^t \mathbb{E}\left[\frac{\partial}{\partial z} G_s^t(p^t, h(z, e), \bar{\tau}_p^t, \bar{\boldsymbol{\theta}}_z^t)\right], \tag{5.32e}$$

  where the expectations are over the random variables $(z, p^t) \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_p^t)$ and y is given in (5.17).

- **Input node update:** Compute

$$\bar{\boldsymbol{\theta}}_x^t = H_x^t(r^t, \bar{\tau}_r^t), \tag{5.33a}$$

$$\bar{\tau}_x^{t+1} = \bar{\tau}_r^t \mathbb{E}\left[\frac{\partial}{\partial r} G_x^t(r^t, \bar{\tau}_r^t, \bar{\boldsymbol{\theta}}_x^t)\right], \tag{5.33b}$$

$$\mathbf{K}_x^{t+1} = \mathrm{cov}(x, \widehat{x}^{t+1}), \tag{5.33c}$$

  where the expectations are over the random variables in (5.16).

Chapter 6

Numerical Evaluation

6.1 Introduction

In this chapter, we report several numerical results comparing some of the methods that were discussed in the previous chapters. In particular, we are interested in evaluating the performance of the GAMP algorithm discussed in Chapter 5 against wavelet– and gradient–based MAP approaches. We will focus on linear inverse problems and consider two types of measurement matrices: (a) compressive sensing matrices with i.i.d. Gaussian elements; (b) matrices representing a periodic convolution. The former scenario constitutes an ideal setting for message-passing algorithms and is useful for understanding the performance bounds of GAMP. The latter represents a more realistic setting that illustrates the potential of message-passing for solving practical inverse problems.

Apart from the measurement model, we will also consider two distinct types of signals: statistical signals generated from a known distribution $p_{\mathbf{x}}$, and some standard test images. In the latter case, we face the common difficulty in applying statistical estimation to imaging: the true statistical distribution of an image is rarely known. To circumvent this problem, we first postulate a family of priors $\{p_{\mathbf{x}}(\cdot|\boldsymbol{\theta}_x) \mid \boldsymbol{\theta}_x \in \boldsymbol{\Theta}_x\}$, then pick $\boldsymbol{\theta}_x$ that maximizes the likelihood of a given image $\mathbf{x}$ under $p_{\mathbf{x}}$. In Chapter 5, we showed

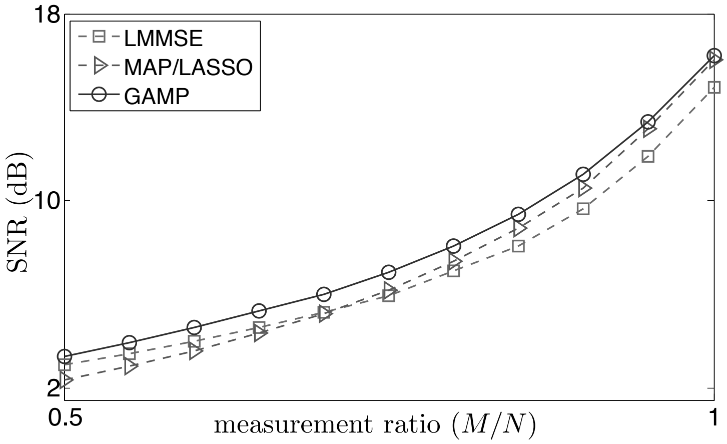


Figure 6.1: Reconstruction of an i.i.d. Laplace signal from noisy linear measurements. The average reconstruction SNR is plotted against the measurement rate M/N. The plot illustrates the superior performance of sum-product GAMP with Gaussian random forward models.

that this type of maximum-likelihood (ML) fitting can be done in a fully automated and consistent fashion for i.i.d. Gaussian measurement matrices when using adaptive GAMP. In this chapter, however, we will opt for an oracle fitting scenario that uses $\mathbf{x}$ to directly learn the parameters $\boldsymbol{\theta}_x$. The advantage of such approach is that it does not depend on the type of the measurement matrix $\mathbf{H}$, and thus allows us to concentrate on the actual performance of GAMP.

We observed that under ML fitting MAP–based estimators consistently yield suboptimal SNR performances. For example, when ML learning of Laplace parameter λ is combined with TV-regularized reconstruction, the result loses most of the visual details, which negatively affects the SNR. Thus, the SNR of MAP–based algorithms on test images is typically boosted by picking the parameter $\boldsymbol{\theta}_x$ that minimizes the MSE.

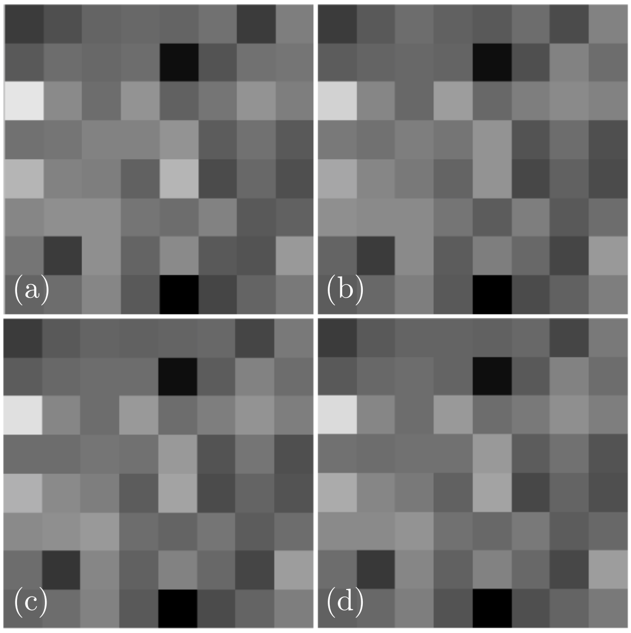


Figure 6.2: Illustration of a reconstructed i.i.d. Laplace signal for a measurement ratio of $M/N = 0.9$, the upper-left 8×8 crop being shown. (a) original, (b) LMMSE: 9.80 dB, (c) MAP: 10.78 dB, (d) GAMP: 11.49 dB.

6.2 Compressive Sensing

We start by evaluating the performance of several reconstruction algorithms for solving the standard compressive sensing problem $\mathbf{y} = \mathbf{Hx} + \mathbf{e}$, where $\mathbf{e}$ is AWGN at input $\mathrm{SNR} \triangleq 10\log_{10}\left(\|\mathbf{Hx}\|^2/\|\mathbf{e}\|\right) = 30$ dB, and measurement matrix $\mathbf{H}$ is drawn with i.i.d. $\mathcal{N}(0, 1/M)$ entries [104, 105].

In the first experiment, illustrated in Figure 6.1, we recover an i.i.d. signal $\mathbf{x}$ generated from the Laplace distribution

$$p_x(x) = \frac{\lambda}{2}\mathrm{e}^{-\lambda|x|},$$

where $\lambda > 0$ is the scale parameter. We fix the signal size to 32×32 and set $\lambda = 1$.

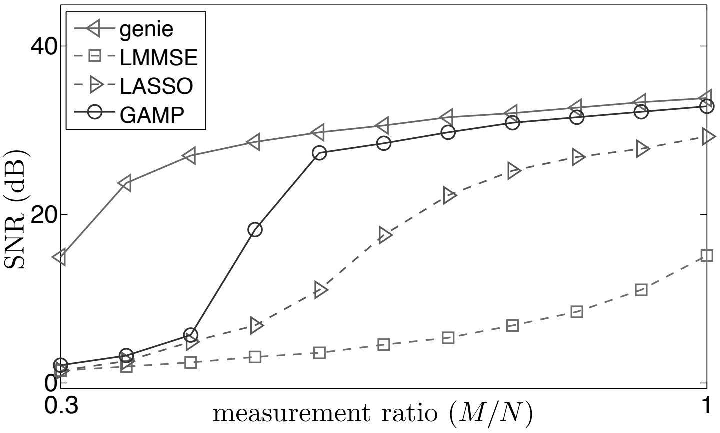


Figure 6.3: Reconstruction of an i.i.d. Bernoulli-Gaussian signal of sparsity $\rho = 0.3$ from noisy linear measurements. The average reconstruction SNR is plotted against the measurement rate M/N. The plot illustrates the superior performance of sum-product GAMP for Gaussian random forward models.

We compare the SNR performance of GAMP against two standard statistical estimators, linear MMSE (LMMSE) and MAP, at different measurement ratios M/N. The LMMSE solution $\widehat{\mathbf{x}}_{\text{LMMSE}}$ is obtained by minimizing ℓ_2-regularized least-squares cost with MSE optimal regularization parameter, while the MAP solution $\widehat{\mathbf{x}}_{\text{MAP}}$ corresponds to the minimization ℓ_1-regularized least-squares cost. The latter is also known as *least absolute shrinkage and selection operator* (LASSO) [96]. Both estimators can be efficiently computed via the FISTA method summarized in Algorithm 2.6. The GAMP algorithm here refers to the sum-product version summarized in Algorithm 5.1, which we use to approximate the MMSE estimator $\widehat{\mathbf{x}}_{\text{MMSE}}$. The curves in Figure 6.1 were obtained by averaging the results of 100 instances of the problem. The results illustrate that MMSE estimator computed via GAMP outperforms both MAP and LMMSE estimators over the whole range of M/N. At lower measurement rates, we observe the superior performance of LMMSE compared to MAP, which corroborates the suboptimality of ℓ_1-norm regularization for the recovery of Laplace signals. A particular instance for $M/N = 0.9$ is illustrated in Figure 6.2, where the first 8×8 pixels from the top-left corner of the image are shown. Visual inspection suggests that Laplace signals are not sparse, which

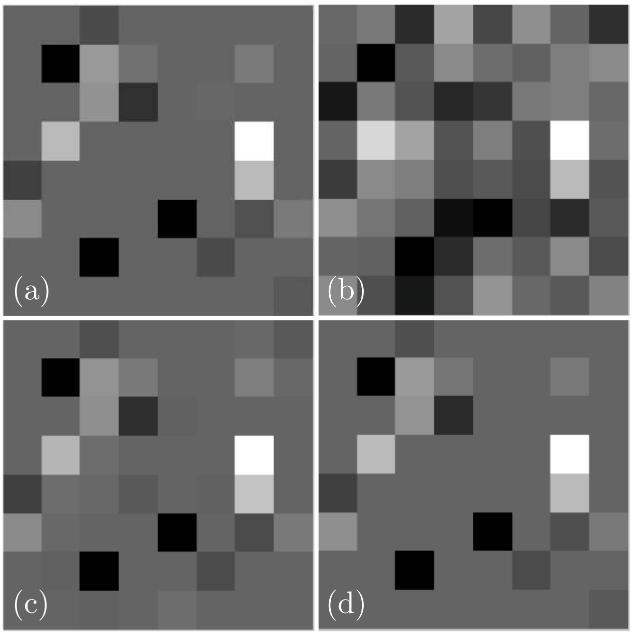


Figure 6.4: Illustration of a reconstructed i.i.d. Bernoulli-Gaussian signal of sparsity $\rho = 0.3$ for a measurement ratio of $M/N = 0.8$, the upper-left 8×8 crop being shown. (a) original, (b) LMMSE: 5.22 dB , (c) LASSO: 20.98 dB, (d) GAMP: 29.97 dB.

explains the difficulty of recovering them when $M < N$.

In the second experiment, illustrated in Figure 6.3, the sparse signal is generated from the i.i.d. Bernoulli-Gaussian distribution

$$p_x(x) = \rho\, \mathscr{G}(x; 1) + (1 - \rho)\delta(x),$$

where $\rho \in [0, 1]$ is the sparsity ratio, $\mathscr{G}$ is the Gaussian probability density function, and δ is the Dirac delta function. We fix the sparsity ratio to $\rho = 0.3$, and compare GAMP against three other methods: LMMSE, LASSO, as well as the support aware MMSE estimator labeled *genie*. The regularization parameter λ of LASSO was optimized for the best SNR performance. The curves in Figure 6.3 were also obtained by averaging the results from 100 instances of the problem. As expected, the results illustrate the suboptimal performance of LMMSE for the recovery of sparse signals. On the other hand, GAMP significantly outperforms LASSO, which is currently considered as the algorithm of choice for the recovery of sparse signals. We can also see that for the high measurement ratios M/N, GAMP approaches the performance of the support aware MMSE estimator. A specific instance at $M/N = 0.8$ is illustrated in Figure 6.2, where the first 8×8 pixels from the top-left corner of the image are shown. This visual result corroborates the SNR values; the reconstruction obtained with GAMP is sharper than the one obtained with LASSO.

In Figures 6.5, 6.6, and 6.7, we consider reconstruction of three test images *Lena*, *Cameramen*, and *Lung*, respectively, for the measurement ratio $M/N = 0.7$. We compare seven reconstruction strategies: orthogonal-wavelet with ℓ_2-regularization (ortho-ℓ_2), orthogonal-wavelet with ℓ_1-regularization (ortho-ℓ_1), orthogonal-wavelet with GAMP (ortho-GAMP), cycle-spinning with ℓ_2-regularization (CS-ℓ_2), cycle-spinning with ℓ_1-regularization (CS-ℓ_1), total-variation regularization (TV), and cycle-spinning with GAMP (CS-GAMP). For the transform, we used Haar wavelet-transform with 4 decomposition levels. When applying cycle-spinning we considered 16 horizontal and 16 vertical adjacent shifts (256 possible shifts in total). We assumed a separable Bernoulli-Gaussian prior for GAMP, where the sparsity ratio and the variance of the Gaussian components were learned from the true signal, separately for each channel, using EM procedure [93]. Regularization parameters of other algorithms were tuned for the best SNR performance.

The results on test images illustrate that within the class of orthogonal wavelet-domain methods GAMP yields the best SNR performance, while ℓ_2–based method yields the worst. We also observe that cycle-spinning significantly boosts the performance of wavelet–domain estimation. For example, CS-ℓ_1 significantly outperforms ortho-GAMP for all images. We finally see the superior performance of TV compared to all wavelet-domain methods, except for CS-GAMP, which yields the solutions with the highest SNR for all considered images.

6.3 Image Deconvolution

In this section, we evaluate the potential of GAMP for solving wavelet-based deconvolution problems, and compare our approach to sparsity-promoting reconstruction based on the ℓ_1-norm. In particular, our forward operator $\mathbf{H}$ corresponds to a convolution with a Laplacian kernel. Such deconvolution is common in phase microscopy, where it is obtained via the *transport-of-intensity equation* (TIE) [106]. Originating from the parabolic wave equation, TIE links the phase image to the variations in the intensity induced by wave propagation. Specifically, TIE grants an experimental access to the Laplacian of the phase by acquiring three defocused images via moving the stage of the microscope [107].

Theorem 5.1 shows that for i.i.d. Gaussian $\mathbf{H}$ the convergence of GAMP is assured via the state evolution equations. Unfortunately, for generic measurement matrices $\mathbf{H}$ the algorithm may diverge. Although, comprehensive convergence analysis of GAMP for general $\mathbf{H}$ is still an open question, some important ideas were presented by Rangan *et al.* in [108]. In particular, they showed that for Gaussian prior and noise the local convergence of both MAP and MMSE variants of GAMP can be guaranteed with a minor modification to the algorithm that damps the updates. Therefore, here, we use iteration dependent damping proportional to $1/\sqrt{t}$, which closely resembles to the strategy used for the convergence proof of cycle spinning in Theorem 3.1.

In the first set of experiments, we consider the restoration of a sparse i.i.d. Bernoulli-Gaussian image of size 64×64 that has a sparsity ratio equal to $\rho = 0.1$. Each nonzero value is drawn from the zero-mean Gaussian distribution of variance 1. Our measurements consist in the circular convolution between the original image and a Laplacian kernel with the addition of AWGN. In Figure 6.8, we compare the deconvolution performance of GAMP with the above Bernoulli-Gaussian prior against LMMSE and LASSO. The regularization parameter λ used in the latter method was optimized for best SNR performance. The results indicate that for the problem considered, GAMP outperforms LASSO over the whole range of input SNRs. A particular instance at input SNR equal to 15 dB is illustrated in Figure 6.9, where the first 8×8 pixels from the top-left corner of the image are shown.

The problem of image restoration is considered next on the standard test images *Cameraman* and *Lena*. We compare GAMP against TV-regularized reconstruction that was considered for TIE microscopy by Bostan *et al.* in [107]. For GAMP, the sparsifying domain $\mathbf{W}$ is set to the Daubechies4-wavelet basis with 5 decomposition levels. As before, for each wavelet coefficient, our approach assumes Bernouilli-Gaussian statistics with a learned sparsity ratio and variance. The performance of GAMP–based wavelet estimation is also boosted with cycle-spinning (we label the algorithm CS-GAMP) that uses 32 horizontal and 32 vertical adjacent shifts (1024 total shifts). The results are provided in Figures 6.10 and 6.11 for an input SNR of 30 dB. The results illustrate the competitiveness of GAMP with TV both visually and in terms of SNR.

6.4 Discussion

In this chapter, we performed several numerical simulations comparing GAMP–based MMSE estimator with other MAP–based approaches. In particular, Section 6.2 considered compressive sensing scenario, where the measurement matrix **H** has i.i.d. Gaussian elements. State evolution analysis in Theorem 5.1 shows that for such measurement matrices, intermediate quantities computed by GAMP admit precise statistical interpretation. In a way, this indicates that such inverse problems represent an ideal scenario, where GAMP is expected to perform at its best. Indeed, our simulations confirm that for compressive sensing GAMP yields state-of-the-art performance by largely outperforming other methods.

In image reconstruction with GAMP, the best performance was obtained when the method was used in conjunction with cycle spinning (CS-GAMP). Nonetheless, we note that the success of CS-GAMP is purely empirical and the method still requires theoretical convergence analysis similar to the one for ISTA in Theorem 3.1. Accordingly, the development of convergence results for CS-GAMP is an important topic for future research.

In Section 6.3, we considered a more practical setting applicable to phase microscopy, where the measurements correspond to periodic convolution with a Laplacian kernel. Unfortunately, for such **H** the convergence of GAMP is still not theoretically understood. Our experimental results, however, indicate that GAMP significantly outperforms LMMSE and LASSO for the recovery of sparse statistical signals. Moreover, the results on test images show that GAMP makes wavelet-domain estimation competitive with TV.

The central conclusion from our simple evaluations here is that GAMP has an immense potential for becoming a method of choice for statistical resolution of inverse problems. There are, however, two crucial aspects of the algorithm that still require extensive theoretical investigation: its convergence and MSE optimality for arbitrary matrices **H**. Additionally, future work might consider extending GAMP beyond i.i.d. priors in some basis, in order to accommodate more powerful priors that are becoming common in imaging [109].

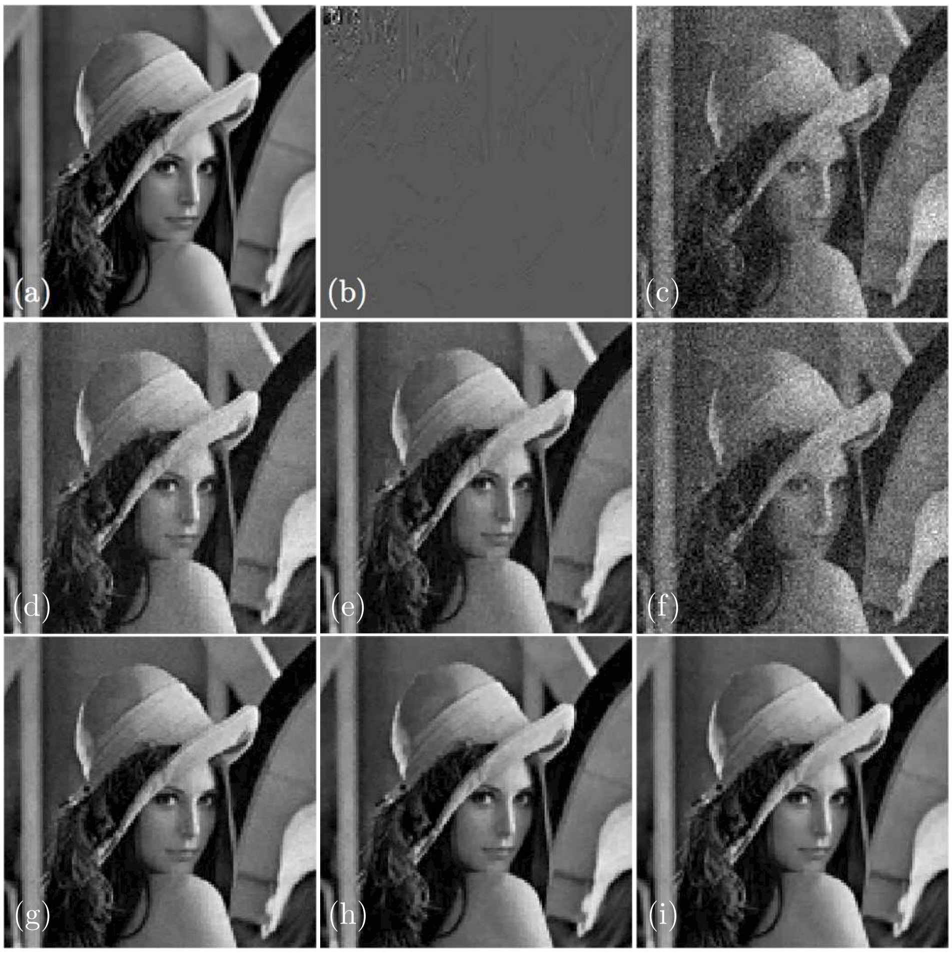


Figure 6.5: Compressive sensing reconstruction of 128 × 128 *Lena* for a measurement ratio of $M/N = 0.7$. (a) original, (b) wavelet decomposition, (c) ortho-ℓ_2: 7.71 dB, (d) ortho-ℓ_1: 17.81 dB, (e) ortho-GAMP: 19.41 dB, (f) CS-ℓ_2: 7.80 dB, (g) CS-ℓ_1: 21.02 dB, (h) TV: 22.46 dB, (i) CS-GAMP: 23.52 dB.

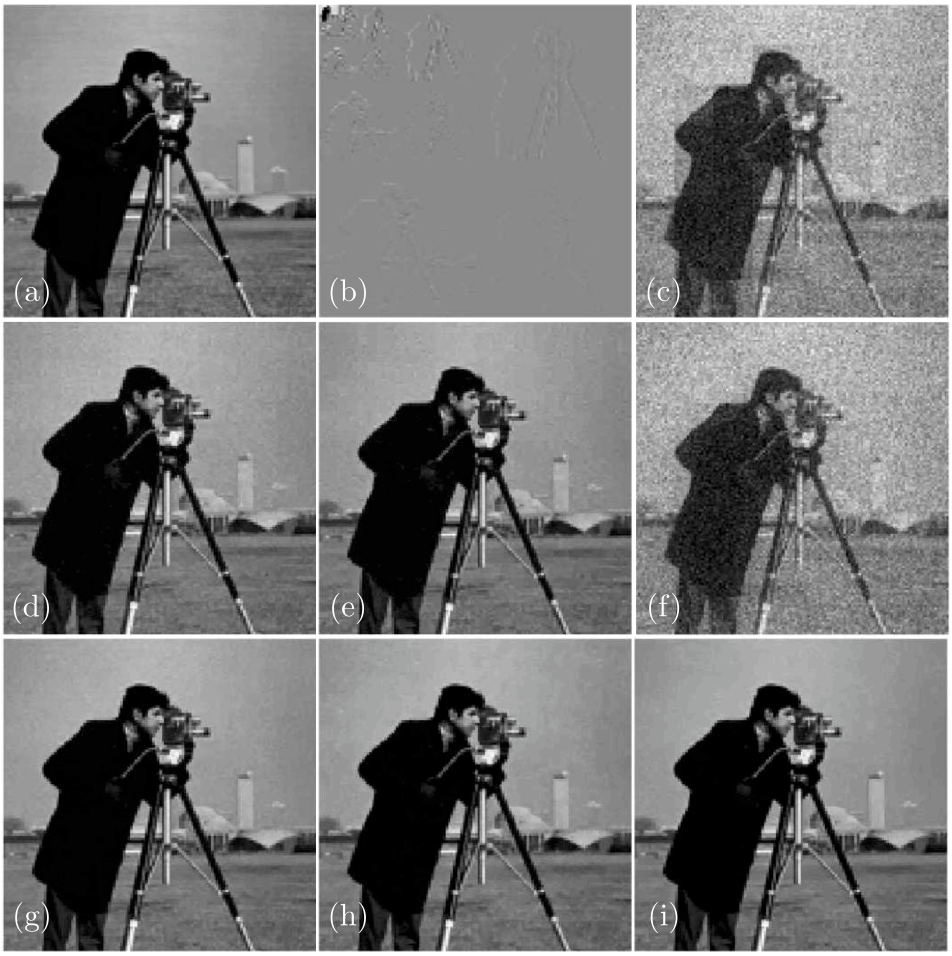


Figure 6.6: Compressive sensing reconstruction of 128 × 128 *Cameraman* for a measurement ratio of $M/N = 0.7$. (a) original, (b) wavelet decomposition, (c) ortho-ℓ_2: 10.00 dB, (d) ortho-ℓ_1: 21.77 dB, (e) ortho-GAMP: 22.54 dB, (f) CS-ℓ_2: 10.12 dB, (g) CS-ℓ_1: 24.25 dB, (h) TV: 25.20 dB, (i) CS-GAMP: 26.62 dB.

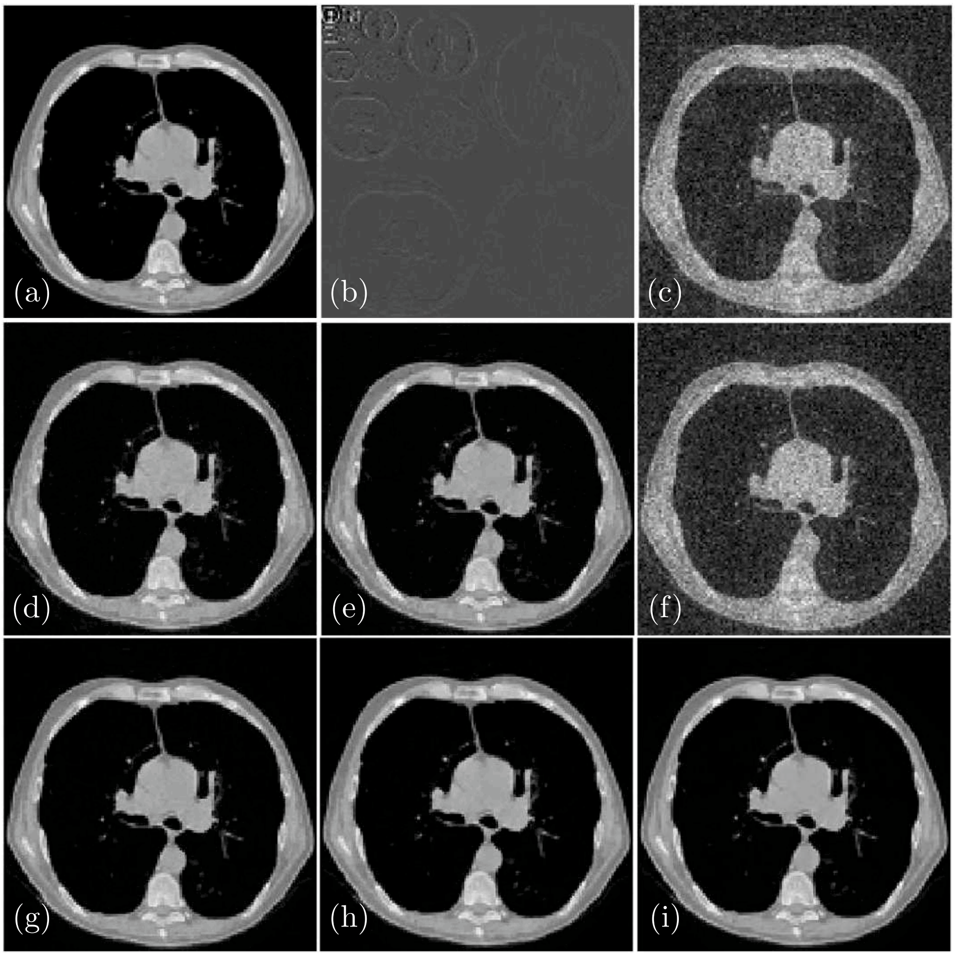


Figure 6.7: Compressive sensing reconstruction of 128 × 128 *Lung* for a measurement ratio of $M/N = 0.7$. (a) original, (b) wavelet decomposition, (c) ortho-ℓ_2: 7.68 dB, (d) ortho-ℓ_1: 22.05 dB, (e) ortho-GAMP: 23.30 dB, (f) CS-ℓ_2: 8.37 dB, (g) CS-ℓ_1: 25.66 dB, (h) TV: 26.37 dB, (i) CS-GAMP: 27.74 dB.

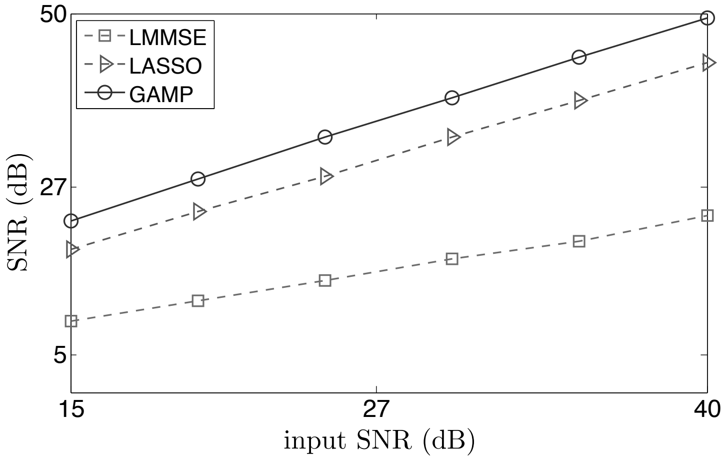


Figure 6.8: Deconvolution of a sparse 64 × 64 Bernoulli-Gaussian image. The average reconstruction SNR is plotted against the input SNR. The plot illustrates the superior MSE performance of GAMP over LASSO and LMMSE.

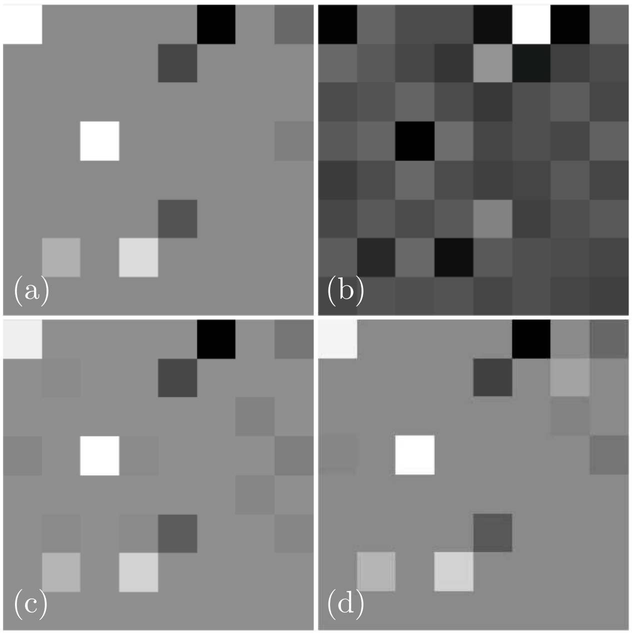


Figure 6.9: Deconvolution of a sparse Gauss-Bernoulli image with $\rho = 0.1$, at input SNR of 15 dB; the upper-left 8×8 crop is shown. (a) original, (b) convolved, (c) LASSO: 18.67 dB, (d) GAMP: 20.04 dB.

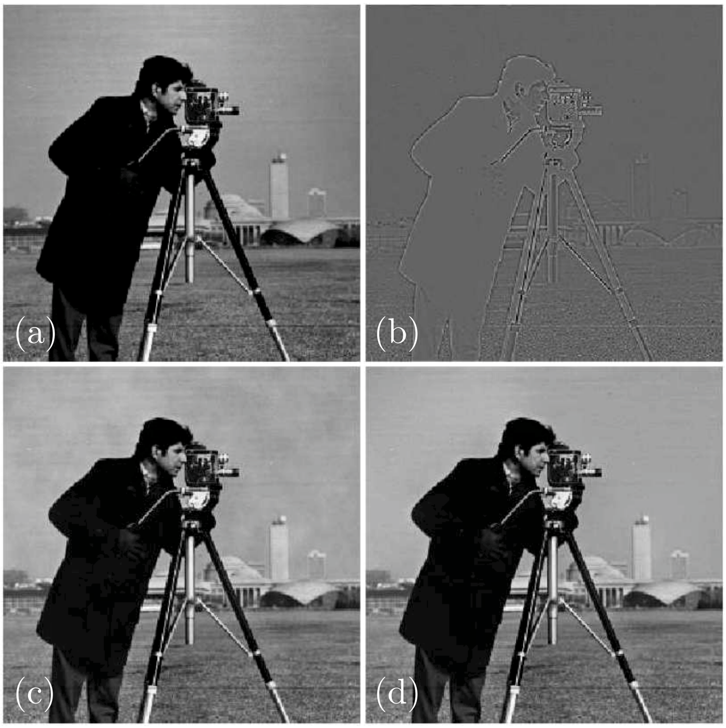


Figure 6.10: Deconvolution of 256 × 256 *Cameraman*. (a) original, (b) convolved, (c) TV: 16.12 dB, (d) CS-GAMP: 16.15 dB.

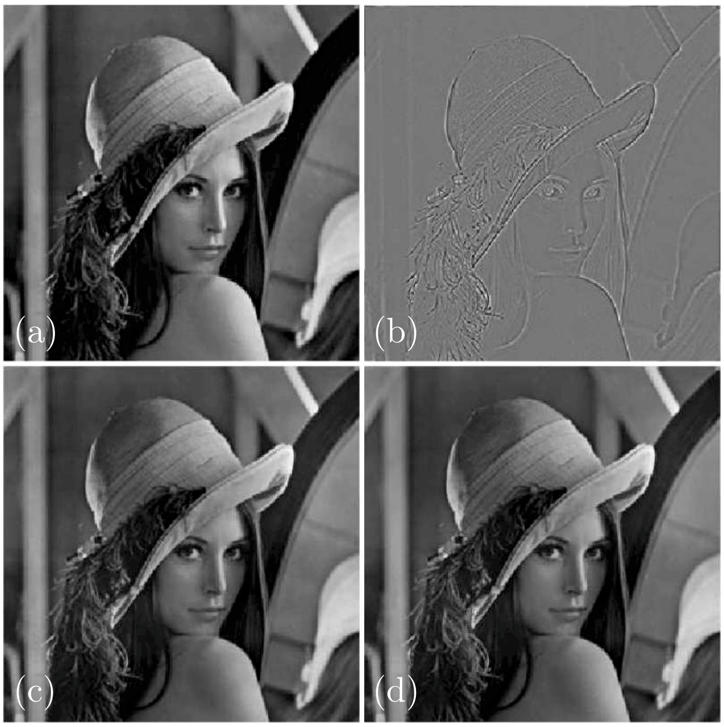


Figure 6.11: Deconvolution of 256 × 256 *Lena*. (a) original, (b) convolved, (c) TV: 10.92 dB, (d) CS-GAMP: 10.93 dB.

Chapter 7

A Novel Nonlinear Framework for Optical Phase Tomography

7.1 Introduction

In this chapter, we leave the world of linear inverse problems by formulating and solving a nonlinear alternative for optical tomographic microscopy. In particular, we present a novel technique for quantitatively estimating the distribution of the *refractive index* in a 3D object from measurements of the transmitted wave-field. Refractive index can be used to study internal structure, as well as physical properties of nearly transparent objects such as cells or tissues. Its accurate estimation is thus an important topic in biomicroscopy.

Most approaches for estimating the refractive index rely on various approximations to linearize the relationship between the refractive index and the measured wave-field [2]. For example, one standard approach interprets the phase of the transmitted wave-field as a line integral of the refractive index along the propagation direction and uses the filtered back-projection algorithm for reconstructing it [110]. Another approach relies on the Fourier diffraction theorem to establish a Fourier transform–based relationship between the measured field and the refractive index [111]. Such linear forward models are typically valid only for objects that are weakly scattering; their application on highly contrasted or large objects often results in estimates of poor resolution.

Contrary to the standard linear approaches, here, we propose a *nonlinear* forward model based on modelling the physics of wave-field propagation. In particular, we rely on a popular technique in optics called *beam propagation method* (BPM), which is extensively used for modelling diffraction and propagation effects of waves [112]. Accordingly, BPM provides a more accurate model than its linear counterparts, when scattering effects cannot be neglected. We thus develop a statistical algorithm that relies on BPM for modelling the measurements and on TV for modelling the signal. The central element of our approach is a novel *time-reversal scheme* for computing the derivative of the transmitted wave-field with respect to the distribution of refractive index. We validate our algorithm on simulations as well as on experimentally measured data. To the best of our knowledge, our approach is the first successful method that can handle this type of nonlinear forward models on such a large scale imaging problem.

This chapter is composed of two main parts. The first part is devoted to BPM in general. Specifically, in Section 7.2, we introduce the inhomogeneous Helmholtz equation that completely characterizes the wave-field at all spatial positions in a time-independent form [113]. We then describe the important paraxial simplification of the Helmholz equation, which is often used for describing the propagation of electromagnetic waves. Section 7.3 introduces the Fourier–based variant of BPM as a computational technique for solving the simplified Helmholz equation. The second part of the chapter focuses on the iterative image reconstruction model based on the nonlinear BPM forward model. The algorithm developed in Section 7.4 estimates the refractive index by minimizing a cost function, where the data-term is based on BPM and the regularizer promotes gradient-sparsity of the estimate. Fundamentally, the algorithm relies on the computation of the derivatives of the forward model with respect to the refractive index, which will be presented in a great detail. We finally close the chapter by presenting some experimental results in Section 7.5.

7.2 Inhomogeneous Wave Equation

Consider the inhomogeneous Helmholtz equation [113]

$$\left(\Delta + k^2(\boldsymbol{r})\mathrm{I}\right)u(\boldsymbol{r}) = 0, \tag{7.1}$$

where $\boldsymbol{r} = (x, y, z)$, u is the total wave-field at $\boldsymbol{r}$, $\Delta = (\partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2)$ is the Laplacian, I is the identity operator, and $k = 2\pi/\lambda$ is the wavenumber of the field. We will write

$$k(\boldsymbol{r}) = k_0 n(\boldsymbol{r}) = k_0(n_0 + \delta n(\boldsymbol{r})), \tag{7.2}$$

where $k_0 = 2\pi/\lambda_0 = \omega/c_0 = \omega\sqrt{\mu_0\epsilon_0}$ is the wavenumber in the free space, $c_0 \approx 3\times10^8$ m/s is the speed of light in free space, μ_0 and ϵ_0 are permeability and permittivity of free space, respectively. The quantity n is the refractive index of the sample, which we have written in terms of the refractive index of the medium n_0 and the perturbation δn due to inhomogeneities in the object. We next develop the paraxial Helmholtz equation for the complex envelope $a(\boldsymbol{r})$ of the paraxial wave[1]

$$u(\boldsymbol{r}) = a(\boldsymbol{r})\mathrm{e}^{\mathrm{j}k_0 n_0 z}. \tag{7.3}$$

One way to regard (7.3) is to say that it corresponds to a plane wave propagating along z in the medium, modulated by the complex amplitude a. Now consider

$$\begin{aligned}\frac{\partial^2}{\partial z^2}u(\boldsymbol{r}) &= \frac{\partial}{\partial z}\left(\left(\frac{\partial}{\partial z}a(\boldsymbol{r})\right)\mathrm{e}^{\mathrm{j}k_0 n_0 z} + \mathrm{j}k_0 n_0 a(\boldsymbol{r})\mathrm{e}^{\mathrm{j}k_0 n_0 z}\right)\\ &= \mathrm{e}^{\mathrm{j}k_0 n_0 z}\left(\left(\frac{\partial^2}{\partial z^2}a(\boldsymbol{r})\right) + 2\mathrm{j}k_0 n_0\left(\frac{\partial}{\partial z}a(\boldsymbol{r})\right) - k_0^2 n_0^2 a(\boldsymbol{r})\right).\end{aligned}$$

By using this expression and by substituting (7.3) into (7.1), we obtain

$$\begin{aligned}&\left(\Delta + k^2(\boldsymbol{r})\mathrm{I}\right)u(\boldsymbol{r})\\ &= \left(\Delta_\perp + \frac{\partial^2}{\partial z^2} + 2\mathrm{j}k_0 n_0\frac{\partial}{\partial z} - k_0^2 n_0^2\mathrm{I} + k_0^2\left(n_0^2 + 2n_0\,\delta n(\boldsymbol{r}) + (\delta n(\boldsymbol{r}))^2\right)\mathrm{I}\right)a(\boldsymbol{r})\mathrm{e}^{\mathrm{j}k_0 n_0 z}\\ &= \left(\Delta_\perp + \frac{\partial^2}{\partial z^2} + 2\mathrm{j}k_0 n_0\frac{\partial}{\partial z} + 2k_0^2 n_0\,\delta n(\boldsymbol{r})\mathrm{I} + k_0^2\,(\delta n(\boldsymbol{r}))^2\,\mathrm{I}\right)a(\boldsymbol{r})\mathrm{e}^{\mathrm{j}k_0 n_0 z} = 0,\end{aligned}$$

where $\Delta_\perp = \partial^2/\partial x^2 + \partial^2/\partial y^2$. We will now introduce two simplifications. The first is the slowly varying envelope approximation (SVEA), which is valid when $|(\partial^2/\partial z^2)a| \ll$

---

[1] A wave is said to be paraxial if its wavefront normals are paraxial rays (i.e. where $\sin(\theta) \approx \theta$ is valid). The variation of a with position must be slow within the distance of a wavelength, so that the wave approximately maintains its underlying plane-wave nature.

$|k_0 n_0 (\partial/\partial z) a|$ and allows us to suppress the second derivative of a in z [113]. In the second simplification, we ignore the term $(\delta n)^2$. We thus obtain

$$\frac{\partial}{\partial z} a(\boldsymbol{r}) = \left(\mathrm{j} \frac{1}{2k_0 n_0} \Delta_\perp + \mathrm{j} k_0 \, \delta n(\boldsymbol{r}) \mathrm{I} \right) a(\boldsymbol{r}). \tag{7.4}$$

7.3 Fourier Beam Propagation

BPM is a class of algorithms designed for calculating the optical field distribution in space or in time given initial conditions[2]. The paraxial Helmholz equation (7.4) is an evolution equation in which the space coordinate z plays the role of the evolution parameter.

7.3.1 Derivation

We start by rewriting (7.4) in operator form

$$\frac{\partial}{\partial z} a(\boldsymbol{r}) = \mathrm{D}\{a\}(\boldsymbol{r}) + \mathrm{N}\{a\}(\boldsymbol{r}), \tag{7.5}$$

where

$$\mathrm{D} = \mathrm{j} \frac{1}{2k_0 n_0} \Delta_\perp \quad \text{and} \quad \mathrm{N} = \mathrm{j} k_0 \, \delta n(\boldsymbol{r}) \mathrm{I}.$$

Note that the operator D is linear and translation-invariant (LTI), while the operator N corresponds to a pointwise multiplication. The formal solution of (7.5) is a complex exponential[3]

$$a(x, y, z) = \mathrm{e}^{(\mathrm{D}+\mathrm{N})z} a(x, y, 0). \tag{7.6}$$

The operators exp(Dz) and exp(Nz) do a priori not commute; however, Baker-Campbell-Hausdorff formula [114] can be applied to show that the error from treating them as if they do will be of order δz^2 if we are taking a small but finite z step δz. This suggests the following approximation

---

[2] We thank Alexandre Goy and Ioannis Papadopoulos, members of the Optics Laboratory at EPFL, for formally initiating us into the world of BPM. We also thank the head of the Optics Laboratory Prof. Demetri Psaltis.

[3] Note that for an operator T, we define a new operator $\mathrm{e}^{\mathrm{T}z}$ in terms of the series expansion $\mathrm{e}^{\mathrm{T}z} = \sum_{n=0}^{\infty} \frac{z^n}{n!} \mathrm{T}^n$. Therefore, for $a(\boldsymbol{r})$, we write $\mathrm{e}^{\mathrm{T}z}\{a\}(\boldsymbol{r}) = \sum_{n=0}^{\infty} \frac{z^n}{n!} \mathrm{T}^n \{a\}(\boldsymbol{r})$

$$a(x, y, z + \delta z) = \mathrm{e}^{\mathrm{N}\delta z} \mathrm{e}^{\mathrm{D}\delta z} a(x, y, z). \tag{7.7}$$

Now, it is possible to get explicit expressions for the diffraction $\exp(\mathrm{D}\delta z)$ and refraction $\exp(\mathrm{N}\delta z)$ operators, since they are independent. Diffraction is handled in the Fourier domain as

$$a(\omega_x, \omega_y, z + \delta z) = \mathrm{e}^{-\mathrm{j}\frac{\omega_x^2 + \omega_y^2}{2k_0 n_0}\delta z} a(\omega_x, \omega_y, z), \tag{7.8}$$

which can also be expressed for a fixed z with a 2D Fourier transform

$$a(x, y, z + \delta z) = \mathscr{F}^{-1}\left\{\mathscr{F}\left\{a(\cdot, \cdot, z)\right\}(\omega_x, \omega_y)\ \mathrm{e}^{-\mathrm{j}\frac{\omega_x^2 + \omega_y^2}{2k_0 n_0}\delta z}\right\}(x, y, z + \delta z). \tag{7.9}$$

For refraction, we get

$$a(x, y, z + \delta z) = \mathrm{e}^{\mathrm{j}k_0(\delta n(x,y,z))\delta z} a(x, y, z), \tag{7.10}$$

which amounts to a simple multiplication with a phase mask in the spatial domain.

The BPM for simulating waves propagating at larger angles was derived by Feit and Fleck in [115]. By relying on their results, we can replace the diffraction step (7.9) by a more refined alternative

$$\begin{aligned} &a(x, y, z + \delta z) \\ &= \mathscr{F}^{-1}\left\{\mathscr{F}\left\{a(\cdot, \cdot, z)\right\}(\omega_x, \omega_y)\ \mathrm{e}^{-\mathrm{j}\left(\frac{\omega_x^2 + \omega_y^2}{k_0 n_0 + \sqrt{k_0^2 n_0^2 - \omega_x^2 - \omega_y^2}}\right)\delta z}\right\}(x, y, z + \delta z). \end{aligned} \tag{7.11}$$

It is important to note that if the solution exists for an arbitrary initial condition $a_0 \triangleq a(x, y, z = 0)$, it implies that a_0 does not depend on $a(\boldsymbol{r})$, which means that BPM that solves this problem ignores reflections.

7.3.2 Implementation

We consider the 3D volume $[-L_x/2, L_x/2] \times [-L_y/2, L_y/2] \times [0, L_z]$ that we refer to as computational domain. The domain is sampled with steps $\delta x, \delta y$, and δz, which will result in N_x, N_y, and K samples, respectively. We will additionally use the following matrix-vector notations

- **x**: samples of the refractive-index distribution δn in the computational domain.
- **y**: samples of the complex wave-field a.
- **S**: non-linear forward operator that implements BPM and maps the refractive index distribution into the complex wave-field $\mathbf{y} = \mathbf{S}(\mathbf{x})$.

We use the index k to refer to the quantities described above at the k-th slice along the optical axis z. For simplicity, we assume that all 2D quantities at the k-th slice are stored in a vector. Then, given the initial input field $\mathbf{y}_0 = \mathbf{S}_0(\mathbf{x})$ and the refractive index distribution $\mathbf{x}$, the total field $\{\mathbf{y}_k\}_{k\in[1\ldots K]}$ can be computed recursively as follows

$$\mathbf{S}_k(\mathbf{x}) = \mathrm{diag}\left(\mathbf{p}_k(\mathbf{x}_k)\right) \mathbf{H}\, \mathbf{S}_{k-1}(\mathbf{x}). \quad (k = 1, \ldots, K) \tag{7.12}$$

Here, **H** denotes the diffraction operator corresponding to (7.11); it is implemented by taking the DFT of the input field, multiplying it by a frequency-domain phase mask, and taking the inverse DFT. The vector $\mathbf{p}_k(\mathbf{x}_k) = \mathrm{e}^{\mathrm{j}k_0\, \delta z\, \mathbf{x}_k}$, which depends on k-th slice of the refractive index $\mathbf{x}_k$, accounts for a phase factor corresponding to the implementation of the refraction step (7.10).

Figure 7.1 illustrates a simulation where a plane-wave of $\lambda = 561$ nm with a Gaussian amplitude is propagated in immersion oil ($n_0 = 1.518$ at $\lambda = 561$ nm) with an angle of $\pi/32$ with respect to optical axis z. The computational domain of dimensions $L_x = L_y = L_z = 36.86$ μm is sampled with steps $\delta x = \delta y = \delta z = 144$ nm. In (a)–(c) we illustrate the propagation of the wave-field in immersion oil, while in (d)–(f) we illustrate the propagation when a spherical bead of diameter 10 μm with refractive index $n = 1.548$ is immersed in the oil. As we can see in (f) even for a relatively week refractive index contrast of $\delta n = 0.03$, one can clearly observe the effects of scattering on the magnitude of the wave-field.

7.4 Iterative Reconstruction

In practice, the input field $\mathbf{y}_0$ is known and the output field $\mathbf{y}_K$ is measured using a holographic technique that gives access to the full complex-valued wave-field. Our goal

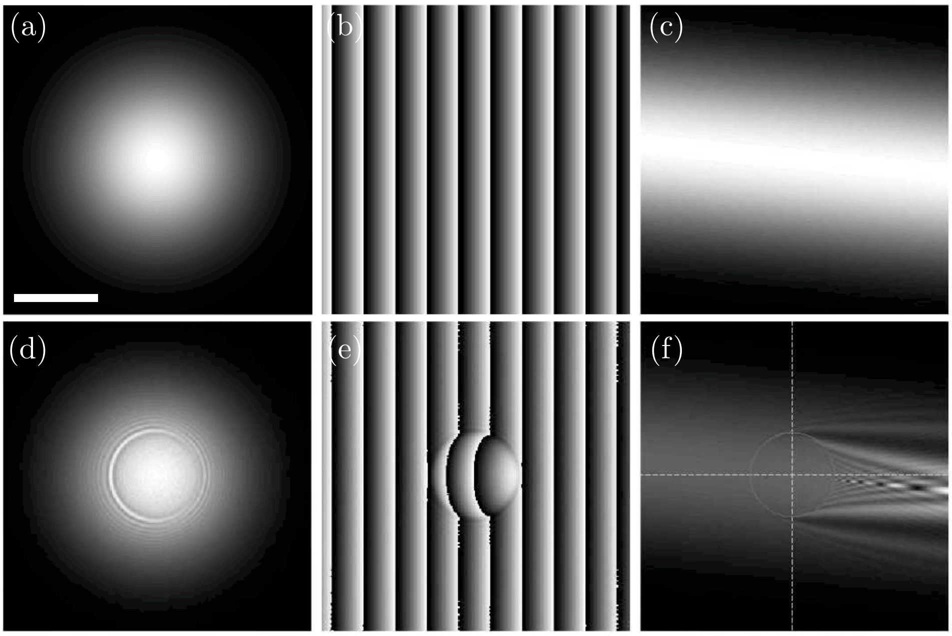


Figure 7.1: Propagation of a plane-wave of $\lambda = 561$ nm in immersion oil with $n_0 = 1.518$ simulated with BPM. (a–c) Propagation in oil. (d–f) Immersion of a 10 μm bead of $n = 1.548$ at $\lambda = 561$ nm. (a, d) x-y slice of the beam magnitude at $z = L_z/2$, (b, e) x-y slice of the beam phase at $z = L_z/2$. (c, f) x-z slice of the beam magnitude at $y = 0$. The circle in (f) illustrates the boundary of the bead at $y = 0$. Scale bar, 10 μm.

is to recover $\mathbf{x}$ from a set of L views $\{\mathbf{y}_K^\ell\}_{\ell\in[1\ldots L]}$ corresponding to input fields $\{\mathbf{y}_0^\ell\}_{\ell\in[1\ldots L]}$.

```
Algorithm 7.1: Time-reversal scheme for computing $\boldsymbol{\nabla}\mathscr{D}^H$
input:  input field $\mathbf{y}_0$, output field $\mathbf{y}_K$,
        and current estimate of the refractive-index distribution $\widehat{\mathbf{x}}$.
output: the gradient $[\boldsymbol{\nabla}\mathscr{D}(\widehat{\mathbf{x}})]^H$.
algorithm:
  1. Compute the total field $\widehat{\mathbf{y}} = \mathbf{S}(\widehat{\mathbf{x}})$ using the Beam propagation recursion (7.12),
     keeping all the intermediate wave-fields $\widehat{\mathbf{y}}_k = \mathbf{S}_k(\widehat{\mathbf{x}})$ in memory.
  2. Compute the residual $\boldsymbol{r}_K = \widehat{\mathbf{y}}_K - \mathbf{y}_K$ and set $\mathbf{s}_K = 0$.
  3. Compute $\mathbf{s}_0 = \left[\frac{\partial}{\partial \mathbf{x}}\mathbf{S}_K(\widehat{\mathbf{x}})\right]^H \boldsymbol{r}_K$ using the following iterative procedure
     (a) $\mathbf{s}_{k-1} = \mathbf{s}_k + \left[\frac{\partial}{\partial \mathbf{x}}\mathbf{p}_k(\widehat{\mathbf{x}}_k)\right]^H \mathrm{diag}\left(\overline{\mathbf{H}\,\widehat{\mathbf{y}}_{k-1}}\right)\boldsymbol{r}_k, \quad (k = K, \dots, 1)$.
     (b) $\boldsymbol{r}_{k-1} = \mathbf{H}^H \mathrm{diag}\left(\overline{\mathbf{p}_k(\widehat{\mathbf{x}}_k)}\right)\boldsymbol{r}_k, \quad (k = K, \dots, 1)$.
  4. Return $[\boldsymbol{\nabla}\mathscr{D}(\widehat{\mathbf{x}})]^H = \mathrm{Re}\{\mathbf{s}_0\}$.
```

To perform estimation, we minimize the following regularized least-squares function

$$
\begin{aligned}
\mathscr{C}(\mathbf{x}) &= \mathscr{D}(\mathbf{x}) + \phi(\mathbf{x}) && \text{(7.13a)}\\
&= \frac{1}{L}\sum_{\ell=1}^{L} \mathscr{D}_\ell(\mathbf{x}) + \phi(\mathbf{x}) && \text{(7.13b)}\\
&= \frac{1}{2L}\sum_{\ell=1}^{L} \left\|\mathbf{y}_K^\ell - \mathbf{S}_K^\ell(\mathbf{x})\right\|_2^2 + \phi(\mathbf{x}), && \text{(7.13c)}
\end{aligned}
$$

where ϕ is one of the sparsity-promoting regularizers discussed in Chapter 2.

7.4.1 Derivation of the Gradient

The gradient of the data term $\mathscr{D}$ with respect to $\mathbf{x}$ is required for MAP–based reconstruction algorithm. Accordingly, we now derive a recursive algorithm for computing

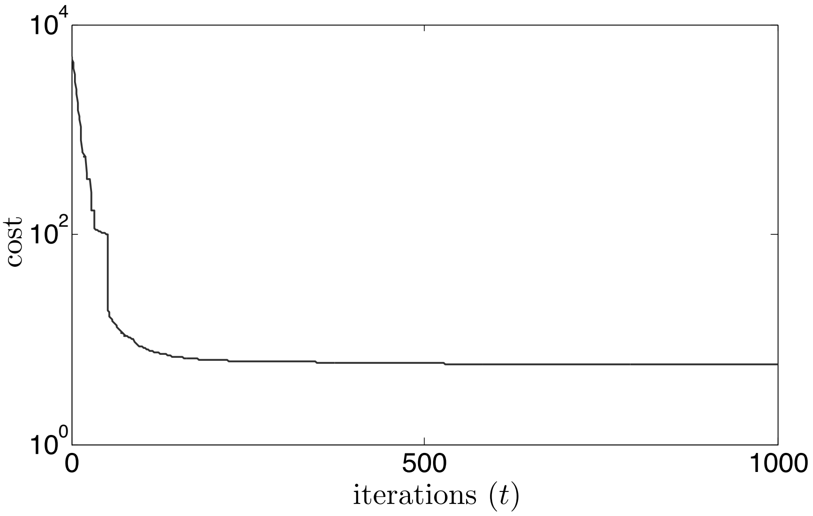


Figure 7.2: Reconstruction of a 10 μm bead of refractive index 1.548 at $\lambda = 561$ nm in immersion oil with $n_0 = 1.518$ from BPM simulated measurements. Evolution of the cost function $\mathscr{C}(\widehat{\mathbf{x}}^t)$ during the reconstruction over 1000 iterations.

this quantity[4]. For simplicity, we consider the scenario of a single view and thus drop indices ℓ from the subsequent derivations. The final formula can be easily generalized for an arbitrary number of views L.

We start by expanding the least-squares function as follows

$$\begin{aligned}\mathscr{D}(\mathbf{x}) &= \frac{1}{2}\,\|\mathbf{y}_K - \mathbf{S}_K(\mathbf{x})\|^2 && (7.14a)\\ &= \frac{1}{2}\,\langle \mathbf{y}_K, \mathbf{y}_K\rangle - \mathrm{Re}\,\{\langle \mathbf{S}_K(\mathbf{x}), \mathbf{y}_K\rangle\} + \frac{1}{2}\,\langle \mathbf{S}_K(\mathbf{x}), \mathbf{S}_K(\mathbf{x})\rangle\,, && (7.14b)\end{aligned}$$

where $\langle \mathbf{x}, \mathbf{z}\rangle = \mathbf{z}^H\mathbf{x}$, where Hermitian transposition ^{H} is due to the complex nature of the quantities. We would like to compute the gradient of this expression. We introduce a convention

---

[4] We thank Cédric Vonesch for establishing the recursive computation formula for the gradient.

$$\frac{\partial}{\partial x_j}\mathbf{S}(\mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_j}[\mathbf{S}(\mathbf{x})]_1 \\ \vdots \\ \frac{\partial}{\partial x_j}[\mathbf{S}(\mathbf{x})]_M \end{bmatrix}. \tag{7.15}$$

Then, the gradient is given by

$$\nabla\mathscr{D}(\mathbf{x}) = \left[\frac{\partial\mathscr{D}(\mathbf{x})}{\partial x_1}\dots\frac{\partial\mathscr{D}(\mathbf{x})}{\partial x_N}\right] = \mathrm{Re}\left\{(\mathbf{S}_K(\mathbf{x})-\mathbf{y}_K)^H\left[\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_K(\mathbf{x})\right]\right\}, \tag{7.16}$$

where

$$\begin{aligned}\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_K(\mathbf{x}) &= \left[\frac{\partial}{\partial x_1}[\mathbf{S}_K(\mathbf{x})]\dots\frac{\partial}{\partial x_N}[\mathbf{S}_K(\mathbf{x})]\right] \\ &= \begin{bmatrix} \frac{\partial}{\partial x_1}[\mathbf{S}_K(\mathbf{x})]_1 & \dots & \frac{\partial}{\partial x_N}[\mathbf{S}_K(\mathbf{x})]_1 \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_1}[\mathbf{S}_K(\mathbf{x})]_M & \dots & \frac{\partial}{\partial x_N}[\mathbf{S}_K(\mathbf{x})]_M \end{bmatrix}.\end{aligned}$$

In practice, we are interested in a column vector

$$[\nabla\mathscr{D}(\mathbf{x})]^H = \mathrm{Re}\left\{\left[\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_K(\mathbf{x})\right]^H(\mathbf{S}_K(\mathbf{x})-\mathbf{y}_K)\right\}. \tag{7.17}$$

Therefore, we need to derive a tractable algorithm to compute (7.17). By looking at (7.12), we see that

$$\begin{aligned}\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_k(\mathbf{x}) &= \frac{\partial}{\partial\mathbf{x}}\left[\mathrm{diag}\left(\mathbf{p}_k(\mathbf{x}_k)\right)\mathbf{H}\,\mathbf{S}_{k-1}(\mathbf{x})\right] \\ &= \mathrm{diag}\left(\mathbf{H}\,\mathbf{S}_{k-1}(\mathbf{x})\right)\left[\frac{\partial}{\partial\mathbf{x}}\mathbf{p}_k(\mathbf{x}_k)\right] + \mathrm{diag}\left(\mathbf{p}_k(\mathbf{x}_k)\right)\mathbf{H}\left[\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_{k-1}(\mathbf{x})\right].\end{aligned}$$

Then, we have that

$$\begin{aligned}&\left[\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_k(\mathbf{x})\right]^H \\ &= \left[\frac{\partial}{\partial\mathbf{x}}\mathbf{S}_{k-1}(\mathbf{x})\right]^H\mathbf{H}^H\mathrm{diag}\left(\overline{\mathbf{p}_k(\mathbf{x}_k)}\right) + \left[\frac{\partial}{\partial\mathbf{x}}\mathbf{p}_k(\mathbf{x}_k)\right]^H\mathrm{diag}\left(\overline{\mathbf{H}\,\mathbf{S}_{k-1}(\mathbf{x})}\right),\end{aligned} \tag{7.18}$$

where the vector $\overline{\mathbf{v}}$ contains complex conjugated elements of vector $\mathbf{v}$. Also, note that for $k=0$, we have

$$\left[\frac{\partial}{\partial \mathbf{x}}\mathbf{S}_0(\mathbf{x})\right]^H = \mathbf{0}. \tag{7.19}$$

7.4.2 Recursive Computation of Gradient

Based on the recursion (7.18), we obtain a practical implementation of (7.17), which is summarized in Algorithm 7.1. Conceptually, our metod is similar to error *backpropagation* algorithm extensively used to train neural networks [116]. Similarly to backpropagation, we compute the gradient by propagating the error in a *time-reversed* fashion.

7.4.3 Iterative Estimation Algorithm

By relying on gradient, we propose a novel stochastic variant of FISTA, summarized in Algorithm 7.2, that iteratively estimates the refractive index distribution $\mathbf{x}$. One notable difference of Algorithm 7.2 with respect to the standard FISTA in Algorithm 2.9 is that the gradient is only computed with respect $\tilde{L} \leq L$ measurements selected uniformly at random in each iteration from the complete set of measurements $\{\mathbf{y}_K^\ell\}_{\ell \in [1 \dots L]}$. This *stochastic gradient* approach [117] reduces the per-iteration cost of reconstruction significantly; moreover, since gradient computation is highly parallelizable the number $\tilde{L}$ can be adapted to match the number of available computing units. We also set the steps of the algorithm proportional to $1/\sqrt{t}$ to assure its local convergence.

7.5 Numerical Evaluation

In the first experiment, we verify the performance of our reconstruction algorithm based on our recursion summarized in Algorithm 7.1. In particular, we consider the reconstruction of a simple 10 μm bead of refractive index $n = 1.548$ at $\lambda = 561$ nm immersed into oil of refractive index $n_0 = 1.518$. We simulate $L = 61$ measurements with equally spaced angles in $[-\pi/8, \pi/8]$ with BPM. Angles are with respect to optical axis z in dimension x. The dimensions of computational domain are set to $L_x = L_y = 36.86$ μm

Algorithm 7.2: Minimizes: $\mathscr{C}(\mathbf{x}) = (1/L)\sum_{\ell=1}^{L} \mathscr{D}_\ell(\mathbf{x}) + \phi(\mathbf{x})$

```
input: data $\{\mathbf{y}^\ell\}_{\ell\in[1\ldots L]}$, initial guess $\widehat{\mathbf{x}}^0$, steps $\{\gamma_t\}_{t\in\mathbb{N}}$, parameter $\tilde{L} \in [1\ldots L]$
       and efficient implementation of $\{\boldsymbol{\nabla}\mathscr{D}_\ell\}_{\ell\in[1\ldots L]}$ and $\mathrm{prox}_\phi$.
set: $t \leftarrow 1$, $\mathbf{s}^0 \leftarrow \widehat{\mathbf{x}}^0$, $k_0 \leftarrow 1$
repeat
    Select uniformly at random $\tilde{L}$ views indexed by $\{\ell_i\}_{i\in[1\ldots\tilde{L}]}$
    $\mathbf{z}^t \leftarrow \mathbf{s}^{t-1} - (\gamma_t/\tilde{L})\sum_{i=1}^{\tilde{L}} \boldsymbol{\nabla}\mathscr{D}_{\ell_i}(\mathbf{s}^{t-1})$      (stochastic gradient step)
    $\widehat{\mathbf{x}}^t \leftarrow \mathrm{prox}_{\phi_\mathrm{u}}(\mathbf{z}^t;\gamma)$                                  (prox step)
    $k_t \leftarrow \frac{1}{2}\left(1+\sqrt{1+4k_{t-1}^2}\right)$
    $\mathbf{s}^t \leftarrow \widehat{\mathbf{x}}^t + ((k_{t-1}-1)/k_t)(\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1})$
    $t \leftarrow t+1$
until stopping criterion
return $\widehat{\mathbf{x}}^t$
```

and $L_z = 18.45$ μm and it is sampled with steps $\delta x = \delta y = \delta z = 144$ nm. Reconstruction is performed by finding a local solution to the following minimization problem

$$\min_{\mathbf{x}} \left\{ \mathscr{C}(\mathbf{x}) = \frac{1}{2L}\sum_{\ell=1}^{L} \left\| \mathbf{y}_K^\ell - \mathbf{S}_K^\ell(\mathbf{x}) \right\|_2^2 + \tau \mathrm{TV}(\mathbf{x}) \right\} \quad \text{subject to} \quad \{0 \le \mathbf{x} \le 0.1\}, \tag{7.20}$$

where $\tau > 0$ is the regularization parameter. For this experiment, the regularization parameter was set to $\tau = 0.01$. For all our experiments, we set $\tilde{L} = 8$. We illustrate the convergence of our minimization algorithm, in Algorithm 7.2, in Figure 7.2, where we plot the cost in (7.20) during 1000 iterations.

In Figure 7.3, we illustrate the true and estimated refractive index distributions. After 1000 iterations the signal-to-noise ratio (SNR) of the estimate is 22.74 dB. The visual quality of the reconstruction is excellent. In particular, we observe the ability of our method to correct for the missing cone due to limited angle of illumination. It yields a much better resolution along z-axis compared to *filtered back projection* (FBP) approach used in [110].

We next validated our BPM–based forward model and reconstruction algorithm with data acquired with an experimental setup of a optical tomographic microscope. The setup splits a laser source of $\lambda = 561$ nm into the reference and sample beams that are combined into a hologram, which is subsequently used to extract the complex wave-field at the measurement plane [112]. The sample is a 10 μm polystyrene bead of refractive index $n = 1.588$ at $\lambda = 561$ nm immersed in oil with a refractive index of $n_0 = 1.518$ so that the refractive index contrast is $\delta n = 0.07$. The data was obtained by obtaining $L = 61$ measurements with angles in the range $[-32.16°, 30.80°]$. The estimate is obtained by solving the problem (7.20), with the regularization parameter set to $\tau = 10$. The result, illustrated in Figures 7.4 (a)–(d), was obtained by initializing our iterative algorithm with the solution of FBP [118] performed on the phase of the measured wave-field. Such warm initialization is useful due to non-convex nature of our optimization problem. In the x-y slice at $z = 21.17$ μm, the bead reconstructed with our method has the diameter of approximately 10.08 μm and average refractive index of 0.067. At exactly same plane, FBP–based solution obtains a bead of slightly smaller diameter, but with refractive index of 0.075. As we can see, our method significantly improves on the FBP–based solution and corrects the missing cone visible in Figures 7.4 (g) and (h).

7.6 Summary

In this chapter, we have demonstrated a novel iterative method for estimating the refractive index distribution of a 3D object from measurements of the transmitted wave-field. The core of our method is a nonlinear forward model, which is based on simulating the physical propagation of electromagnetic waves with BPM. Our estimate is obtained by minimizing the discrepancy between the measured wave-fileds and those computed with BPM. To compensate for the ill-posedness of the problem, we also regularize our solution with TV. We have validated our forward model by reconstructing a polystyrene bead with a known refractive index distribution.

The results presented in this chapter are still very fresh and preliminary. Yet, they show for the first time the feasibility of using a nonlinear forward model in a practical imaging modality such as tomographic microscope. The next step is to replace the polystyrene bead with a cell or a multicellular organism. One also needs to perform more systematic comparisons between our BPM–based method and the linear approaches. Depending on the signals considered, one can imagine using other regularizers discussed in this thesis. Current implementation of our algorithm requires several hours of running time; however, one can imagine significant reductions via the usage of GPU–based techniques that are similar to those used to train neural networks.

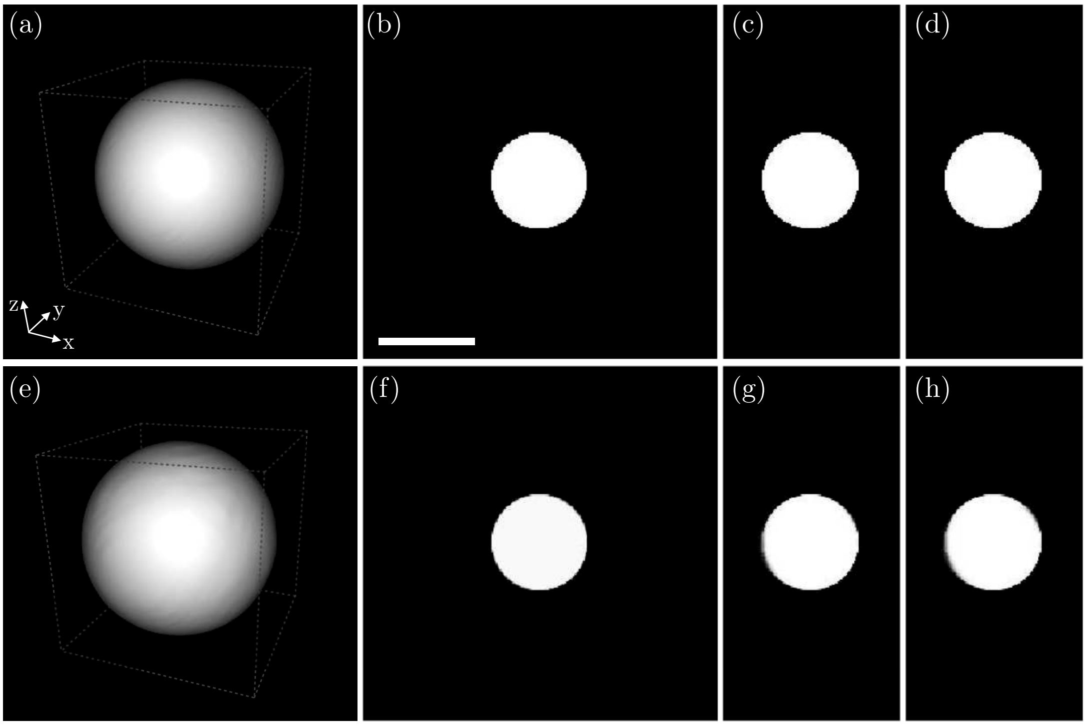


Figure 7.3: Reconstruction of a 10 μm bead of refractive index 1.548 at $\lambda =$ 561 nm in immersion oil with $n_0 = 1.518$ from BPM simulated measurements. (a–d) True refractive index distribution. (e–h) Reconstructed refractive index distribution: SNR = 22.74 dB. (a, e) A 3D rendered image of the bead. (b, f) x-y slice of the bead at $z = L_z/2$. (c, g) z-x slice of the bead at $y = 0$. (d, h) z-y slice of the bead at $x = 0$. Scale bar, 10 μm.

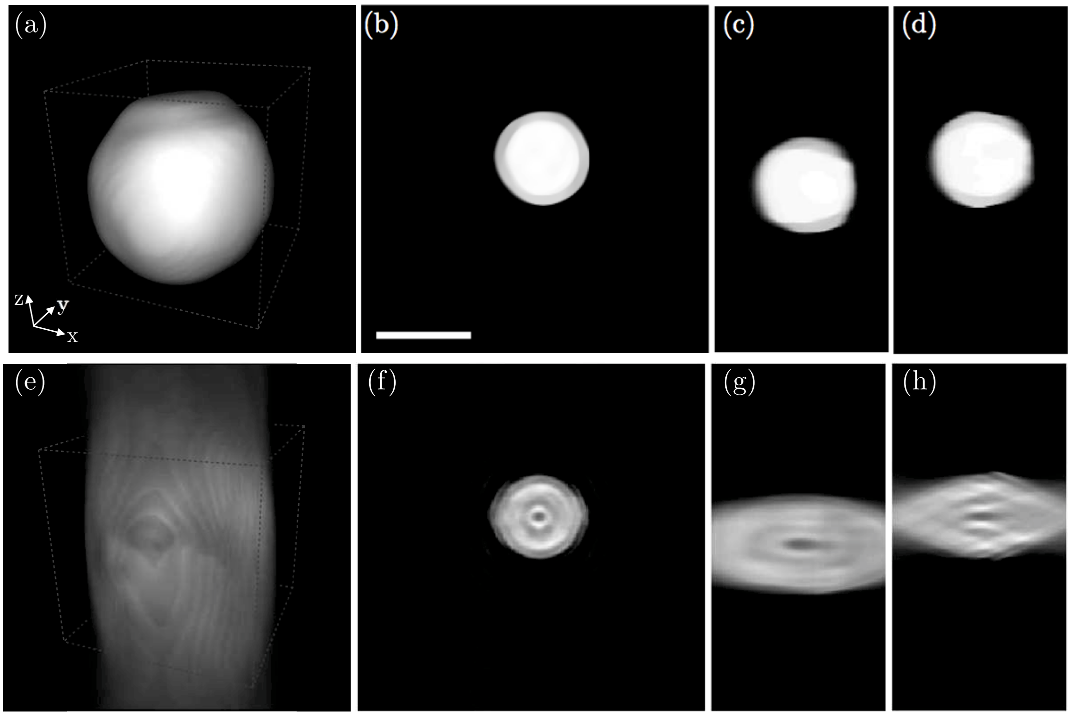


Figure 7.4: Reconstruction of a 10 μm bead of refractive index 1.588 at $\lambda =$ 561 nm in immersion oil with $n_0 = 1.518$ from real measurements obtained with an tomographic diffraction microscope. (a–d) Reconstruction using our algorithm. (e–h) Reconstruction using filtered back projection algorithm. (a, e) A 3D rendered image of the bead. (b, f) x-y slice of the bead at $z = 21.17$ μm. (c, g) z-x slice of the bead at $y = -2.30$ μm. (d, h) z-y slice of the bead at $x = 0.58$ μm. Scale bar, 10 μm.

Chapter 8

Conclusions

In this thesis, we presented a collection of new and competitive ideas for the resolution of inverse problems. In the first section below, we summarize the main contributions of our work. In the second section, we discuss the potential areas of interest for future research related to our work.

8.1 Summary of Results

Justification and improvement of cycle spinning: We provided the first theoretical justification of the popular cycle spinning technique in the context of linear inverse problems. The proof is general and can be used to develop and analyze algorithms relying on compound regularization. We also introduced the idea of consistent cycle spinning, which amounts to imposing certain linear constraints on estimated wavelet coefficients when the transform is redundant. These constraints enforce Parseval's norm relation, which, in turn, allows for statistical interpretation of the solution. We have empirically demonstrated statistical MAP and MMSE estimation under consistent cycle spinning for signals with sparse derivatives.

Lévy processes and optimal recovery of signals with sparse derivatives: Availability of an analog generative signal model is very convenient for assessing the performances of reconstruction methods. To that end, we introduced a continuous-domain stochastic framework for modelling signals with sparse derivatives. Its important aspects include among other things the following: (a) statistical properties of its members are fully characterized by the Lévy-Khintchine formula; (b) several popular algorithms can be interpreted as MAP estimators for our processes; (c) possibility to develop an MSE optimal estimation algorithm. We succeeded in designing MMSE estimation algorithm and used it to benchmark the performances of several reconstruction algorithms. In particular, we demonstrated that the superior estimation performance of TV cannot be solely attributed to its statistical MAP interpretation.

Novel adaptive version of GAMP: We presented the GAMP algorithm that can be used to perform MAP and MMSE estimation of signals that can be decoupled in some basis. Our numerical evaluations illustrated that under suitable conditions that are reminiscent of compressive sensing GAMP can largely outperform other standard approaches. In particular, in imaging, the best performance is achieved when GAMP–based MMSE estimation is combined with cycle spinning. We have also extended the traditional GAMP such that it can learn unknown statistical parameters present in the inverse problem during the reconstruction. We have proved that for i.i.d. Gaussian measurement models when learning is performed via the maximum-likelihood estimator, adaptive GAMP is asymptotically consistent.

A novel nonlinear framework for phase tomography: We developed a novel approach for estimating the 3D refractive index distribution of an object with an optical tomographic microscope. The difficulty of the problem lies in the nonlinear nature of the underlying measurement process and on very high-dimensionality of the data. Our approach is based on simulating the wave-field propagation through the object via the beam propagation method. We provide a corresponding inverse problem formulation and develop a novel iterative inversion algorithm. We demonstrate that our algorithm recovers the distribution of the refractive index distribution on simulated as well as on experimentally measured data. To the best of our knowledge, this is the first successful reconstruction algorithm that can handle this kind of nonlinear model on such a large scale (3D) imaging problem.

8.2 Outlook

More sophisticated priors for imaging: Finding priors that can accurately model various types of natural images is still a major research topic that requires considerable attention. In this thesis, we restricted ourselves to priors that assume independence in some transform domain. One avenue would be to investigate the empirical success of the patch–based methods such as nonlocal means [119] or BM3D [120]. An encouraging step towards this direction was taken in [109], where the authors have constructed overcomplete frames formalizing BM3D image model. The latter can then be used in the context of regularized resolution of linear inverse problems discussed in this thesis.

Extension and analysis of consistent cycle spinning: We have presented consistent cycle spinning as a technique for enforcing norm equivalence between the image and the redundant-transform domains. In particular, our transforms consisted of translations of an orthogonal wavelet basis making the representation shift-invariant. However, the idea can be applied more generally to any redundant representation. For example, Tekin *et al.* [121] have used the concept to boost the performance of image denoising with steerable wavelet-transforms. One interesting direction would be to combine all the ingredients by boosting MMSE consistent cycle spinning with successful redundant transforms such as BM3D frames [109]

GAMP for larger classes of problems: As we saw in Chapter 6, the best empirical results for linear inverse problems were obtained when combining GAMP with cycle spinning. Practical applications of GAMP, however, are still at early stages and require further theoretical analysis. One related direction is the convergence of the algorithm. It was recently shown by Rangan [108] that GAMP diverges for certain problems where the forward operator is especially ill-posed. One strategy that we relied on and which is commonly used to circumvent this issue is damping. Since by damping strongly, one also hampers the convergence of the algorithm, there is a need for some theoretical alternatives to damping.

Another issue is that the current formulations of GAMP require a basis for whitening the signal. For example, our experiments relied on wavelet-transforms to decouple the signal components. Such approaches preclude the usage of other types of linear or nonlinear operators such as TV. Two things that need to be done are as follows: (a) develop GAMP–based algorithms relying on priors based on gradient-like operators; (b) theoretically justify the union between GAMP and cycle spinning in a way that is similar to our analysis in Chapter 3.

Speedup and improvements for nonlinear forward models: Substantial speedups in the BPM–based reconstruction is possible using parallelization on dedicated GPU units. In fact, such parallelization is extensively used in backprojection–based deep learning algorithms for neural networks. There is no apparent obstacle that would prevent the same strategy with our method to further speedup the reconstruction. We validated our BPM–based reconstruction on a simple bead. However, the algorithm can and should be applied for other types of data such as biological cells.

Bibliography

[1] E. Bostan, U. S. Kamilov, M. Nilchian, and M. Unser, “Sparse stochastic processes and discretization of linear inverse problems,” *IEEE Trans. Image Process.*, vol. 22, pp. 2699–2710, July 2013.

[2] O. Haeberlé, K. Belkebir, H. Giovaninni, and A. Sentenac, “Tomographic diffractive microscopy: basic, techniques, and perspectives,” *J. Mod. Opt.*, vol. 57, pp. 686–699, May 2010.

[3] J. A. Nelder and R. W. M. Wedderburn, “Generalized linear models,” *J. R. Statist. Soc. A*, vol. 135, no. 3, pp. 370–384, 1972.

[4] N. T. Thao and M. Vetterli, “Reduction of the MSE in R-times oversampled A/D conversion from $O(1/R)$ to $O(1/R^2)$,” *IEEE Trans. Signal Process.*, vol. 42, pp. 200–203, 1994.

[5] N. T. Thao and M. Vetterli, “Deterministic analysis of oversampled A/D conversion and decoding improvement based on consistent estimates,” *IEEE Trans. Signal Process.*, vol. 42, no. 3, pp. 519–531, 1994.

[6] V. K. Goyal, N. T. Thao, and M. Vetterli, “Quantized overcomplete expansions in $\mathbb{R}^n$: Analysis, synthesis, and algorithms,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 16–31, January 1998.

[7] S. Rangan and V. K. Goyal, “Recursive consistent estimation with bounded noise,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 457–464, January 2001.

[8] U. S. Kamilov, A. Bourquard, A. Amini, and M. Unser, "One-bit measurements with adaptive thresholds," *IEEE Signal Process. Lett.*, vol. 19, pp. 607–610, October 2012.

[9] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing de-quantization with applications to compressed sensing," *IEEE Trans. Signal Process.*, vol. 60, pp. 6270–6281, December 2012.

[10] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," in *Adv. Neural Information Processing Systems (NIPS)*, (Lake Tahoe, NV, USA), pp. 2447–2455, December 3-6, 2012.

[11] C. Vonesch and M. Unser, "A fast multilevel algorithm for wavelet-regularized image restoration," *IEEE Trans. Image Process.*, vol. 18, pp. 509–523, March 2009.

[12] M. Guerquin-Kern, M. Häberlin, K. P. Prüssmann, and M. Unser, "A fast wavelet-based reconstruction method for magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 30, pp. 1649–1660, September 2011.

[13] M. Nilchian and M. Unser, "Differential phase-contrast X-ray computed tomography: From model discretization to image reconstruction," in *Proc. Int. Symp. Biomedical Imaging*, (Barcelona, Spain), May 2012.

[14] F. Luisier, *The SURE-LET Approach to Image Denoising*. PhD thesis, École polytechnique fédérale de Lausanne, January 2010. Thesis number 4566.

[15] M. Unser and P. Tafti, *An Introduction to Sparse Stochastic Processes*. Cambridge Univ. Press, 2014.

[16] M. Unser, "Sampling - 50 Years After Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.

[17] S. Mallat, *A Wavelet Tool of Signal Processing: The Sparse Way*. San Diego: Academic Press, 3rd ed., 2009.

[18] M. Unser and T. Blu, "Wavelet theory demystified," *IEEE Trans. Signal Process.*, vol. 51, pp. 470–483, February 2003.

[19] M. Vetterli, J. Kovačević, and V. K. Goyal, *Fourier and Wavelet Signal Processing*. Cambridge Univ. Press, 2014. `http://fourierandwavelets.org/`.

[20] E. Bostan, U. S. Kamilov, and M. Unser, "Reconstruction of biomedical images and sparse stochastic modelling," in *Proc. Int. Symp. Biomedical Imaging*, (Barcelona, Spain), May 2012.

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[23] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, pp. 2282–2312, July 2005.

[24] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Information Theory*, (St. Petersburg, Russia), pp. 2168–2172, July 31-August 5, 2011.

[25] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Philadelphia: SIAM, 1995.

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[27] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983. (in Russian).

[28] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[29] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, September 1994.

[30] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, pp. 906–916, August 2003.

[31] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, pp. 1413–1457, November 2004.

[32] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. ECCV* (Springer, ed.), vol. 3024, (New York), pp. 1–13, 2004.

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

[34] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-based norm regularization for image restoration with biomedical applications," *IEEE Trans. Image Process.*, vol. 21, pp. 983–995, March 2012.

[35] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, November 1992.

[36] A. Beck and M. Teboulle, "Fast gradient-based algorithm for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, pp. 2419–2434, November 2009.

[37] U. S. Kamilov, E. Bostan, and M. Unser, "Generalized total variation denoising via augmented Lagrangian cycle spinning with Haar wavelets," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, (Kyoto, Japan), pp. 909–912, March 25–30, 2012.

[38] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. of Fourier Anal. Appl.*, vol. 14, pp. 877–905, October 2008.

[39] R. R. Coifman and D. L. Donoho, *Springer Lecture Notes in Statistics*, ch. Translation-invariant de-noising, pp. 125–150. Springer-Verlag, 1995.

[40] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?," *IEEE Trans. Signal Process.*, vol. 59, pp. 2405–2410, May 2011.

[41] U. S. Kamilov, E. Bostan, and M. Unser, "Wavelet shrinkage with consistent cycle spinning generalizes total variation denoising," *IEEE Signal Process. Lett.*, vol. 19, pp. 187–190, April 2012.

[42] A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser, "Bayesian denoising: From MAP to MMSE using consistent cycle spinning," *IEEE Signal Process. Lett.*, vol. 20, pp. 249–252, March 2013.

[43] U. S. Kamilov, E. Bostan, and M. Unser, "Variational justification of cycle spinning for wavelet-based solutions of inverse problems," *IEEE Signal Process. Lett.*, vol. 21, pp. 1326–1330, November 2014.

[44] A. K. Fletcher, K. Ramchandran, and V. K. Goyal, "Wavelet denoising by recursive cycle spinning," in *Proc. IEEE Int. Conf. Image Proc. (ICIP'02)*, (Rochester, NY, USA), pp. II.873–II.876, September 22–25, 2002.

[45] A. K. Fletcher, "Denoising via recursive wavelet thresholding," Master's thesis, University of California, Berkeley, 2002.

[46] C. Vonesch and M. Unser, "A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution," *IEEE Trans. Image Process.*, vol. 17, pp. 539–549, April 2008.

[47] S. Ramani and J. A. Fessler, "A hybrid regularizer combining orthonormal wavelets and finite differences for statistical reconstruction in 3-D CT," in *Proc. 2nd Intl. Mtg. on image formation in X-ray CT*, (Salt Lake City, UT, USA), pp. 348–351, 2012.

[48] I. W. Selesnick and M. A. T. Figueiredo, "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors," in *Proc. SPIE 7446, 74460D*, 2009.

[49] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[50] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2 ed., 2006.

[51] U. S. Kamilov, P. Pad, A. Amini, and M. Unser, "MMSE estimation of sparse Lévy processes," *IEEE Trans. Signal Process.*, vol. 61, pp. 137–147, January 2013.

[52] A. Beck and M. Teboulle, *Convex Optimization in Signal Processing and Communications,* ch. Gradient-Based Algorithms with Applications to Signal Recovery Problems, pp. 42–88. Cambridge, 2009.

[53] K. Sato, *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, 1999.

[54] D. Applebaum, *Lévy Processes and Stochastic Calculus*. Cambridge University Press, 2009.

[55] A. Amini, M. Unser, and F. Marvasti, "Compressibility of deterministic and random infinite sequences," *IEEE Trans. Signal Process.*, vol. 59, pp. 5193–5201, November 2011.

[56] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2010.

[57] S. Boyd and L. Vandenberghe, "Subgradients," April 2008. Class notes for Convex Optimization II. `http://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf`.

[58] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Machine Learning (ICML)*, (Washington DC, USA), August 21-24, 2003.

[59] M. Unser, P. Tafti, and Q. Sun, "A unified formulation of Gaussian versus sparse stochastic processes—Part I: Continuous-domain theory," *IEEE Trans. Inf. Theory*, vol. 60, pp. 1945–1962, March 2014.

[60] V. Cevher, "Learning with compressible priors," in *Adv. Neural Information Processing Systems (NIPS)*, (Vancouver, BC, Canada), December 2009.

[61] M. Unser and P. D. Tafti, "Stochastic models for sparse and piecewise-smooth signals," *IEEE Trans. Signal Process.*, vol. 59, pp. 989–1006, March 2011.

[62] J. S. Yedidia, W. T. Freeman, and Y. Weiss, *Understanding Belief Propagation and Its Generalizations,* ch. Exploring Artificial Intelligence in the New Millennium, pp. 239–269. San Francisco, CA: Morgan Kaufmann Publishers, 2003.

[63] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, “The factor graph approach to model-based signal processing,” *Proc. IEEE*, vol. 95, pp. 1295–1322, June 2007.

[64] M. J. Wainwright and M. I. Jordan, “Graphical models, exponential families, and variational inference,” *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[65] D. Wipf and S. Nagarajan, “Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions,” *IEEE J. Sel. Topics Signal Process.*, vol. 4, pp. 317–329, April 2010.

[66] M. Nikolova, “Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares,” *SIAM Multiscale Model. Simul.*, vol. 4, pp. 960–991, September 2005.

[67] A. Amini, U. S. Kamilov, E. Bostan, and M. Unser, “Bayesian estimation for continuous-time sparse stochastic processes,” *IEEE Trans. Sig*, vol. 61, pp. 907–920, February 2013.

[68] R. M. Gray and D. L. Neuhoff, “Quantization,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 2325–2383, October 1998.

[69] M. Lustig, D. L. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magn. Reson. Med.*, vol. 58, pp. 1182–1195, December 2007.

[70] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “Adaptive total variation image deblurring: A majorizationâĂŞminimization approach,” *Signal Process.*, vol. 89, pp. 1683–1693, September 2009.

[71] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 599–618, February 2001.

[72] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci.*, vol. 106, pp. 18914–18919, November 2009.

[73] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, pp. 269–280, January 2010.

[74] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," in *Proc. Conf. on Inform. Sci. & Sys.*, (Princeton, NJ), pp. 1–6, March 2010.

[75] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing estimation from quantized samples," in *Proc. 4th Workshop on Signal Process. with Adaptive Sparse Structured Representations (SPARS 2011)*, (Edinburgh, United Kingdom), p. 58, June 27-30, 2011.

[76] M. Unser and T. Blu, "Generalized smoothing splines and optimal discretization of the Wiener filter," *IEEE Trans. Signal Process.*, vol. 53, pp. 2146–2159, June 2005.

[77] T. Blu and M. Unser, "Self-similarity: Part II—Optimal estimation of fractal processes," *IEEE Trans. Signal Process.*, vol. 55, pp. 1364–1378, April 2007.

[78] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," *IEEE Trans. Inf. Theory*, vol. 60, pp. 2969–2985, May 2014.

[79] S. Billings and S. Fakhouri, "Identification of systems containing linear dynamic and static nonlinear elements," *Automatica*, vol. 18, no. 1, pp. 15–26, 1982.

[80] I. W. Hunter and M. J. Korenberg, "The identification of nonlinear biological systems: Wiener and Hammerstein cascade models," *Biol. Cybern.*, vol. 55, pp. 135–144, 1986.

[81] O. Schwartz, J. W. Pillow, N. C. Rust, and E. P. Simoncelli, "Spike-triggered neural characterization," *JVIS*, vol. 6, pp. 484–507, July 2006.

[82] D. Guo and C.-C. Wang, "Asymptotic mean-square optimality of belief propagation for sparse linear systems.," in *Proc. IEEE. Inform. Theory Workshop*, (Chengdu, China), pp. 194–198, October 2006.

[83] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *IEEE Inf. Theory Workshop*, (Dublin, Ireland), pp. 1–5, January 6-8, 2010.

[84] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: II. analysis and validation," in *IEEE Inf. Theory Workshop*, (Dublin, Ireland), pp. 1–5, January 6-8, 2010.

[85] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, pp. 764–785, February 2011.

[86] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Optimal quantization for compressive sensing under message passing reconstruction," in *Proc. IEEE Int. Symp. Information Theory*, (St. Petersburg, Russia), pp. 390–394, July 31-August 5, 2011.

[87] T. P. Minka, *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, January 2001.

[88] M. Seeger, "Bayesian inference and optimal design for the sparse linear model," *J. Machine Learning Research*, vol. 9, pp. 759–813, September 2008.

[89] J. Vila and P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing," in *Proc. Asilomar Conf. on Signals, Systems, and Computers*, (Pacific Grove, CA), November 6-9, 2011.

[90] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," in *Proc. Conf. on Inform. Sci. & Sys.*, (Princeton, NJ), March 2012.

[91] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," *Phys. Rev. X*, vol. 2, p. 021005, June 2012.

[92] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices.," *J. Stat. Mech.*, vol. 2012, p. P08009, August 2012.

[93] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–17, 1977.

[94] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, "Hybrid generalized approximation message passing with applications to structured sparsity," in *Proc. IEEE Int. Symp. Inform. Theory*, (Cambridge, MA), pp. 1241–1245, July 2012.

[95] D. L. Donoho and I. M. Johnstone, "Minimax risk over ℓ_p-balls for ℓ_q-error.," *Probab. Theory and Relat. Fields*, vol. 99, pp. 277–303, June 1994.

[96] R. Tibshirani, "Regression and selection via the lasso," *J. R. Stat. Soc. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[97] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Inf inference*, vol. 2, pp. 115–144, 2013.

[98] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, p. 12, December 2006.

[99] D. Donoho, I. Johnstone, A. Maleki, and A. Montanari, "Compressed sensing over ℓ^p-balls: Minimax mean square error," in *Proc. IEEE Int. Symp. Information Theory*, (St. Petersburg, Russia), July 31-August 5, 2011.

[100] A. K. Fletcher, S. Rangan, L. Varshney, and A. Bhargava, "Neural reconstruction with approximate message passing (NeuRAMP)," in *Proc. Neural Information Process. Syst.*, (Granada, Spain), pp. 2555–2563, December 12-14, 2011.

[101] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," 2013. arXiv:1301.6295 [cs.IT].

[102] V. A. Marčenko and L. A. Pastur, "Distribution of eigenvalues for some sets of random matrices," *Math. USSR–Sbornik*, vol. 1, no. 4, pp. 457–483, 1967.

[103] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.

[104] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, February 2006.

[105] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, April 2006.

[106] M. R. Teague, “Deterministic phase retrieval: a Green’s function solution,” *J. Opt. Soc. Am.*, vol. 73, no. 11, pp. 1434–1441, 1983.

[107] E. Bostan, E. Froustey, B. Rappaz, E. Shaffer, D. Sage, and M. Unser, “Phase retrieval by using transport-of-intensity equation and differential interference constrast microscopy,” in *Proc. IEEE Int. Conf. Image Process (ICIP’14)*, (Paris, France), October 27–30, 2014.

[108] S. Rangan, P. Schniter, and A. Fletcher, “On the convergence of approximate message passing with arbitrary matrices,” in *Proc. IEEE Int. Symp. Information Theory*, (Honolulu, HI, USA), pp. 236–240, June 29-July 4, 2014.

[109] A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *IEEE Trans. Image Process.*, vol. 21, pp. 1715–1728, April 2012.

[110] W. Choi, C. Fang-Yen, K. Badizadegan, S. Oh, N. Lue, R. R. Dasari, and M. S. Feld, “Tomographic phase microscopy,” *Nat. Methods*, vol. 4, pp. 717–719, September 2007.

[111] Y. Sung, W. Choi, C. Fang-Yen, K. Badizadegan, R. R. Dasari, and M. S. Feld, “Optical diffraction tomography for high resolution live cell imaging,” *Opt. Express*, vol. 17, pp. 266–277, December 2009.

[112] A. Goy, *Imaging and Microscopy in linear and nonlinear media using digital holography*. PhD thesis, École polytechnique fédérale de Lausanne, January 2013. Thesis number 5617.

[113] J. W. Goodman, *Introduction to Fourier Optics*. McGraw-Hill, 2 ed., 1996.

[114] R. Gilmore, “Baker-Campbell-Hausdorff formulas,” *J. Math. Phys.*, vol. 15, pp. 2090–2092, 1974.

[115] M. D. Feit and J. A. Fleck, “Bean nonparaxiality, filament formaform, and beam breakup in the self-focusing of optical beams,” *J. Opt. Soc. Am. B*, vol. 5, pp. 633–640, March 1988.

[116] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, 1995.

[117] L. Bottou, *Neural Networks: Tricks of the Trade*, ch. Stochastic Gradient Descent Tricks, pp. 421–437. Springer, 2 ed., September 2012.

[118] A. C. Kak and M. Slaney, *Principles of Computerized Tomography Imaging*. IEEE, 1988.

[119] A. Buades, B. Coll, and J. M. Morel, "Image denoising methods. A new nonlocal principle.," *SIAM Rev*, vol. 52, no. 1, pp. 113–147, 2010.

[120] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, pp. 2080–2095, August 2007.

[121] B. Tekin, U. S. Kamilov, E. Bostan, and M. Unser, "Benefits of consistency in image denoising with steerable wavelets," in *Proc. Int. Symp. Biomedical Imaging*, (Vancouver BC, Canada), pp. 1355–1358, May 26-31, 2013.

Curriculum Vitæ

Ulugbek Kamilov

EPFL STI IMT LIB
BM 4138, Station 17
CH-1015, Lausanne
Switzerland
ulugbek.kamilov@alumni.epfl.ch
http://www.ukamilov.com

Education

2011-2015	**École polytechnique fédérale de Lausanne (EPFL)**, Lausanne, Switzerland PhD in Electrical Engineering Project: "Sparsity-Driven Statistical Inference for Inverse Problems" Advisor: Prof. Michael Unser
2013	Stanford University, Stanford, CA, USA
2008-2011	**École polytechnique fédérale de Lausanne (EPFL)**, Lausanne, Switzerland MSc in Communication Systems Project: "Optimal Quantization for Compressive Sensing with Relaxed Belief Propagation" Advisors: Prof. Vivek Goyal (MIT) and Prof. Martin Vetterli (EPFL)
2010	MIT, Cambridge, MA, USA
2005-2008	**École polytechnique fédérale de Lausanne (EPFL)**, Lausanne, Switzerland BSc in Communication Systems
2007	Carnegie Mellon, Pittsburgh, PA, USA

Technical Skills

Software Development:	Java, C#, C/C++, Xcode/Objective-C, HTML/CSS
Scientific computing:	Matlab, ImageJ, Icy
Engineering coursework:	Signal and Image Processing, Software Engineering, Computer Networks, Machine Learning and Pattern Classification, Convex and Distributed Optimization

Languages

Russian, Uzbek	Native
English, French	Fluent
German	Competent

Professional Experience

2015-present	**MERL**, Cambridge, MA, USA (Member of research staff) Member of Multimedia group responsible for development of theory and algorithms for signal recovery from available physical measurements.
2009	**Microsoft**, Zurich, Switzerland (Software engineer intern) Developed an extension for the Response Group Service (RGS) of Microsoft Lync Server that enables administrators to create and configure small response groups for routing and queuing incoming instant messages to several agents. The extension I developed increases the response group usage by allowing agents to interact with several clients simultaneously.
2008	**Telecommunications Research Center**, Vienna, Austria (Intern) Developed density evolution for the analysis and design of LDPC codes. Also, created an optimized version of density evolution that reduced the computational time by several orders of magnitude.

Projects

2011-2015	**EPFL**, Lausanne, Switzerland (PhD Thesis in Biomedical Imaging Group) I am working on the development of fast algorithms for solving inverse problems in biomedical imaging applications such as MRI and optical microscopy. The algorithms I developed during my thesis, outperformed state-of-the-art methods on a wide range of applications in terms of both speed and quality. I published numerous peer-reviewed scientific publications (25 in total so far) available at: http://www.ukamilov.com.
2013	**Stanford University**, Stanford, USA (Visiting Student Researcher) Worked on the development of a low complexity algorithm based on belief propagation for finding rank one approximation of a large matrix. The applications of the method include finding large cliques in a random network, data clustering, and non-negative or sparse matrix factorization. This project was conducted in Information Systems Lab of Stanford.
2010-2011	**Massachusetts Institute of Technology**, Cambridge, USA (Master thesis) Developed a novel statistical algorithm for estimating signals from discretized measurements. The algorithm and the resulting publication were nominated for the best student paper award at the Fourth IEEE International Workshop in Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP). This project was conducted in the Research Laboratory of Electronics of MIT.
2010	**EPFL**, Lausanne (Semester Project in Biomedical Imaging Group) Developed a program for removing noise from biomicroscopy images. The program relied on the statistical characterization of the noise, as well as on a convex optimization algorithm for reducing it. The novel and creative aspect was that the program I developed extended the standard algorithm to the Poisson-type noise that is common in the low light regimes.
2010	**EPFL**, Lausanne (Pattern Classification and Machine Learning) The goal of this class project was to develop a program for learning and classifying handwritten digits from the MNIST database. In a group of two, we developed two classification algorithms based on back-projection in neural networks and on expectation-maximization algorithm. Our classifiers achieved classification error rates of 2% and 4%, respectively, during validation.
2009	**EPFL**, Lausanne (Semester Project in Audiovisual Communications Lab) Developed an adaptive beamformer, which is a program that can be used in conjunction with an array of microphones, for performing directional sound acquisition. While traditional techniques rely solely on the temporal processing of the signal, my extension did spatiotemporal processing of the acoustic field. This allowed the beamformer to also perform spatial localization of the sound sources.
2007-2008	**Carnegie Mellon University**, Pittsburgh, USA (Research Assistant) The goal of this project was to develop a computer vision software enabling high-speed eye tracking with a robust pupil center estimation. The software was part of a larger eye tracking system for diagnosis and early screening of diabetic retinopathy.

Publications

Journal

1. M. H. Shoreh, U. S. Kamilov, I. N. Papadopoulos, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, “A Learning Approach to Optical Tomography,” `arXiv:1502.01914 [physics.optics]`, February 2015.

2. S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, “Inference for Generalized Linear Models via Alternating Directions and Beth Free Energy Minimization,” `arXiv:1501.01797 [cs.IT]`, January 2015.

3. U. S. Kamilov, E. Bostan, and M. Unser, “Variational Justification of Cycle Spinning for Wavelet-Based Solutions of Inverse Problems,” *IEEE Signal Process. Letters*, vol. 21, no. 11, pp. 1326–1330, November 2014.

4. U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, “Approximate Message Passing with Consistent Parameter Estimation and Application to Sparse Learning,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2969–2985, May 2014.

5. E. Bostan, U. S. Kamilov, M. Nilchian, and M. Unser, “Sparse Stochastic Processes and Discretization of Linear Inverse Problems,” *IEEE Trans. Image Process.*, vol. 22, no. 7, July 2013.

Publications (cont.)

6. A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser, “Bayesian Denoising: From MAP to MMSE Using Consistent Cycle Spinning,” *IEEE Signal Process. Letters*, vol. 20, no. 3, pp. 249-252, March 2013.

7. A. Amini, U. S. Kamilov, E. Bostan, and M. Unser, “Bayesian Estimation for Continuous-Time Sparse Stochastic Processes,” *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 907–920, February 2013.

8. U. S. Kamilov, P. Pad, A. Amini, and M. Unser, “MMSE Estimation of Sparse Lévy Processes,” *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 137–147, January 2013.

9. U. S. Kamilov, V. K. Goyal, and S. Rangan, “Message-Passing De-Quantization with Applications to Compressed Sensing,” *IEEE Trans. Signal Process.*,vol. 60, no. 12, pp. 6270–6281, December 2012.

10. U. S. Kamilov, A. Bourquard, A. Amini, and M. Unser, “One-Bit Measurements with Adaptive Thresholds,” *IEEE Signal Process. Letters*, vol. 19, no. 10., pp. 607-610, October 2012.

11. U. S. Kamilov, E. Bostan, and M. Unser, “Wavelet Shrinkage with Consistent Cycle Spinning Generalizes Total Variation Denoising,” *IEEE Signal Process. Letters*, vol. 19, no. 4, pp. 187-190, April 2012.

Conference, symposium, and workshop

1. U. S. Kamilov, A. Bourquard, and M. Unser, "Sparse Image Deconvolution with Message Passing," Proc. 5th Workshop on Signal Process. with Adaptive Sparse Structured Representations (SPARS 2013) (Lausanne, Switzerland, July 8-11).

2. E. Bostan, U. S. Kamilov, M. Nilchian, and M. Unser, "Consistent Discretization of Linear Inverse Problems using Sparse Stochastic Processes," Proc. 5th Workshop on Signal Process. with Adaptive Sparse Structured Representations (SPARS 2013) (Lausanne, Switzerland, July 8-11).

3. E. Bostan, J. Fageot, U. S. Kamilov, and M. Unser, "MAP Estimators for Self-Similar Sparse Stochastic Models," Proc. 10th International Conference on Sampling Theory and Applications (SAMPTA 2013) (Bremen, Germany, July 1-5), pp. 197–199.

4. U. S. Kamilov, A. Bourquard, E. Bostan, and M. Unser, “Autocalibrated Signal Reconstruction from Linear Measurements using Adaptive GAMP,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. 2013 (Vancouver, Canada, May 26-31), pp. 5925–5928.

5. B. Tekin, U. S. Kamilov, E. Bostan, and M. Unser, “Benefits of Consistency in Image Denoising with Steerable Wavelets,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. 2013 (Vancouver, Canada, May 26-31), pp. 1355–1358.

6. U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, “Approximate Message Passing with Consistent Parameter Estimation and Applications to Sparse Learning,” Proc. 23rd Ann. Conf. Neural Information Processing Systems (NIPS 2012) (Lake Tahoe, Nevada, December 3-6), pp. 2447–2455.
   - ★ Acceptance rate of 370/1467 = 25%.

7. A. Amini, U. S. Kamilov, and M. Unser, “The Analog Formulation of Sparsity Implies Infinite Divisibility and Rules Out Bernoulli-Gaussian Priors,” Proc. IEEE Information Theory Workshop 2012 (Lausanne, Switzerland, September 3–7), pp. 687–691.

8. E. Bostan, U. S. Kamilov, and M. Unser, “Reconstruction of Biomedical Images and Sparse Stochastic Modelling,” Proc. Int. Symp. Biomedical Imaging 2012 (Barcelona, Spain, May 2-5), pp. 880–883.

9. U. S. Kamilov, A. Amini, and M. Unser, “MMSE Denoising of Sparse Lévy Processes via Message Passing,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. 2012 (Kyoto, Japan, March 25-30), pp. 3637–3640.

10. U. S. Kamilov, E. Bostan, and M. Unser, “Generalized Total Variation Denoising via Augmented Lagrangian Cycle Spinning with Haar Wavelets,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. 2012 (Kyoto, Japan, March 25-30), pp. 909–912.

11. A. Amini, U. S. Kamilov, and M. Unser, “Bayesian Denoising of Stochastic Processes with Finite Rate of Innovation,” Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process. 2012 (Kyoto, Japan, March 25-30), pp. 3629–3632.

Publications (cont.)

12. U. S. Kamilov, V. K. Goyal, and S. Rangan, “Generalized Approximate Message Passing Estimation from Quantized Samples," Proc. 4th Int. Workshop on Computational Advances in Multi-Sensor Adaptive Process. (CAMSAP 2011) (San Juan, Puerto Rico, December 13–16), pp. 401–404.
    ★ Student Paper Award finalist.

13. U. S. Kamilov, V. K. Goyal, and S. Rangan, “Message-Passing Estimation from Quantized Samples," Proc. 4th Workshop on Signal Process. with Adaptive Sparse Structured Representations (SPARS 2011) (Edinburgh, United Kingdom, June 27–June 30), p. 58.

14. U. S. Kamilov, V. K. Goyal, and S. Rangan, “Optimal Quantization for Compressive Sensing under Message Passing Reconstruction," Proc. 2011 IEEE Int. Symp. Inform. Theory (Saint-Petersburg, Russia, July 31–August 5), pp. 390–394.

Teaching

Teaching Assistant for the Courses:

- Signals and Systems I: Autumn 2011, Autumn 2012, Autumn 2013
  ★ Part of the team that won 2013 Education Award from EPFL’s Life Sciences Section.
- Signals and Systems II: Spring 2012, Spring 2013, Spring 2014
- Introduction to Object Oriented Programming: Autumn 2008
- Introduction to Communication Systems: Autumn 2007

Students Supervised

1. Bugra Tekin – masters student at EPFL, co-advised with Emrah Bostan and Prof. Michael Unser, 2012.
   Semester Project: “Solving Inverse Problems with Sparsifying Transforms.”
   Currently a PhD student at EPFL.
2. Ipek Baz – masters student at EPFL, co-advised with Aurélien Bourquard and Prof. Michael Unser, 2012.
   Semester Project: “Reconstruction of Signals from Sign Measurements”
3. Abbas Kazerouni – intern at EPFL from Sharif University, co-advised with Prof. Michael Unser, 2012.
   Topic: “MMSE Estimation of Sparse Stochastic Processes”
   Currently a PhD student at Stanford.
4. Julien Schwab – masters student at EPFL, co-advised with Ricard Delgado Gonzalo and Prof. Michael Unser, 2012.
   Semester Project: “Object Classification with Kinect”
5. Mamoun Benkirane – masters student at EPFL, co-advised with Prof. Michael Unser, 2014.
   Semester Project: “Interactive Demo for Signals & Systems Course”.
6. Sander Kromwijk – masters student at EPFL, co-advised with Prof. Michael Unser, 2014.
   Master Project: “Reconstruction of a 3D map of the refractive index from digital holography microscopy data.”

Professional Activity and Service

Student Member of IEEE (2011–present).

Full Member of Sigma Xi (2014–present).

Reviewer for
IEEE Int. Conf. Acoustics, Speech, & Signal Processing 2012,
IEEE Int. Symp. Biomedical Imaging 2012,
IEEE Trans. Image Processing,
IEEE Trans. Signal Process.,
IEEE Signal Process. Lett.,
IEEE Information Theory Workshop 2014,
NIPS 2014,
IEEE Trans. Med. Imaging.
IEEE J. Sel. Topics in Sign. Proces.,
IEEE Int. Symp. Inform. Theory 2015.

Short Biography

Ulugbek S. Kamilov received his PhD in Electrical Engineering from the École polytechnique fédérale de Lausanne (EPFL), Switzerland, in 2015.

In 2007-08, he was an exchange student in Electrical and Computer Engineering at Carnegie Mellon University (CMU). In 2008, he worked as a research intern at the Telecommunications Research Center in Vienna, Austria. In 2009, he worked as a software engineering intern at Microsoft. In 2010-11, he was a visiting student in the Research Laboratory of Electronics at Massachusetts Institute of Technology (MIT). In 2013, he was a visiting student researcher in the Information Systems Laboratory at Stanford University. In 2011-15, he was a research assistant in Biomedical Imaging Group at EPFL.

Since 2015, he is with the Multimedia Group at MERL. His research interest lies in the development of advanced algorithms for solving inverse problems.