ABSTRACT

ZHU, JUNAN. Statistical Physics and Information Theory Perspectives on Linear Inverse Problems. (Under the direction of Dror Baron.)

Many real-world problems in machine learning, signal processing, and communications assume that an unknown vector $\mathbf{x}$ is measured by a matrix $\mathbf{A}$, resulting in a vector $\mathbf{y}=\mathbf{A}\mathbf{x}+\mathbf{z}$, where $\mathbf{z}$ denotes the noise; we call this a single measurement vector (SMV) problem. Sometimes, multiple dependent vectors $\mathbf{x}^{(j)},\ j\in\{1,\cdots,J\}$, are measured at the same time, forming the so-called multi-measurement vector (MMV) problem. Both SMV and MMV are linear models (LM's), and the process of estimating the underlying vector(s) $\mathbf{x}$ from an LM given the matrices, noisy measurements, and knowledge of the noise statistics, is called a linear inverse problem. In some scenarios, the matrix $\mathbf{A}$ is stored in a single processor and this processor also records its measurements $\mathbf{y}$; this is called centralized LM. In other scenarios, multiple sites are measuring the same underlying unknown vector $\mathbf{x}$, where each site only possesses part of the matrix $\mathbf{A}$; we call this multi-processor LM. Recently, due to an ever-increasing amount of data and ever-growing dimensions in LM's, it has become more important to study large-scale linear inverse problems. In this dissertation, we take advantage of tools in statistical physics and information theory to advance the understanding of large-scale linear inverse problems. The intuition of the application of statistical physics to our problem is that statistical physics deals with large-scale problems, and we can make an analogy between an LM and a thermodynamic system [Tan02; GV05; Krz12a; Krz12b; MM09; BK15]. Therefore, we can apply statistical physics analysis tools as well as algorithmic tools into understanding large-scale LM's and their corresponding linear inverse problems. In terms of information theory [CT06], although it was originally developed to characterize the theoretic limits of digital communication systems, information theory was later found to be rather useful in analyzing and understanding other inference problems. We use some of the concepts and ideas of information theory to understand the theoretic performance limits in various aspects of linear inverse problems.

There exist numerous algorithms for solving linear inverse problems. However, only a partial understanding of the theoretic characterization of the minimum mean squared error (MMSE) when solving linear inverse problems appears in the literature [Ran12; Tan02; GV05]. Such a theoretic analysis helps practitioners appreciate the gap between their estimation quality and the theoretically optimal quality. Therefore, in this dissertation we use the replica analysis [Tan02; GV05; MT06; Krz12a; Krz12b; MM09; BK15; Les15] from statistical physics to study the MMSE in MMV problems. We obtain different performance regions in which the MMSE behaves differently. Besides the quality of the estimation, there are also other "costs" that practitioners might care about, especially in the big data era. Some prior art has focused on reducing certain costs such as the communication cost [Han14] and the computation cost [Ma14c], but there has been less progress relating different

costs and achieving optimal trade-offs among them. Despite the lack of such works, these trade-offs are important to system designers in order to produce efficient systems. To address these issues, in this dissertation we use a distributed algorithm as an example and study the behavior of the optimal communication scheme in the limit of low excess mean squared error beyond the MMSE for that distributed algorithm. Furthermore, we study the optimal trade-offs among the computation cost, the communication cost, and the quality of the estimate.

Finally, we discuss estimation algorithm design for an SMV setting. There are numerous estimation algorithms for SMV in the prior art, but they all require some statistical knowledge about the underlying vector $\mathbf{x}$; in a practical setting, such knowledge might be inaccurate or unavailable. Therefore, it is important to design a *universal* estimation algorithm that is more agnostic to the prior knowledge of the unknown vector $\mathbf{x}$. In this dissertation, we design an algorithmic framework based on Markov chain Monte Carlo (MCMC) borrowed from statistical physics, and in extensive numerical experiments the algorithm achieves a mean squared error that is close to the MMSE.

Statistical Physics and Information Theory Perspectives on Linear Inverse Problems

by
Junan Zhu

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2017

APPROVED BY:

Huaiyu Dai	Karen Daniels
Brian Hughes	David Ricketts

Dror Baron
Chair of Advisory Committee

DEDICATION

To people who care about me and people who I care about. To world peace.

BIOGRAPHY

Junan Zhu is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering (ECE) at North Carolina State University (NCSU), Raleigh, North Carolina, U.S. His research interests include compressed sensing, statistical signal processing, information theory, statistical physics, machine learning, optimization, distributed algorithms, and computational imaging. Before joining NCSU, Mr. Zhu received the B.E. degree in Electrical Engineering with a focus on optoelectronics from the University of Shanghai for Science and Technology (USST), Shanghai, China in 2011. His research in USST focused on Terahertz waveguides and black silicon.

Junan Zhu received the Graduate Student Fellowship at NCSU in 2011, which was awarded to the top 3 incoming ECE graduate students. He also received the National Scholarship in 2009 and the Baosteel Scholarship in 2010, both at USST.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Dr. Dror Baron. It is his patient guidance and advice in research that has enlightened me and made my research life easier. It is his helpful mentoring about life in the U.S. that has provided me with enough information to merge into this new society. It is his abundant financial support that has allowed me to focus on research. (In particular, I would like to thank the generous support of the National Science Foundation and Army Research Office.[1]) Dr. Baron is more than an academic advisor. He is a mentor and a friend. I am very grateful for Dr. Baron's help and advice, and I hope to work on research projects with him in the future as well.

Next, I would like to thank my committee, in alphabetical order: Dr. Huaiyu Dai, Dr. Karen Daniels, Dr. Brian Hughes, and Dr. David Ricketts, as well as former committee members Dr. W. Rhett Davis and Dr. Edgar Lobaton. Their helpful comments about my work and enlightening feedback greatly improved the quality of my work and dissertation. Besides my committee, I would like to thank Dr. Ahmad Beirami, Dr. Marco F. Duarte, Dr. Florent Krzakala, and Dr. Lenka Zdeborova for their advice and collaboration. I also want to thank the lecturers of all the courses I attended. It is their clear explanations that granted me a solid understanding of various subjects in my field.

I also want to thank my dear roommates, Dr. Shikai Luo and Shuiqing Wang, whom I started my endeavor in the U.S. with and whom I shared joy and sadness with. Also, I would like to thank them for their help on technical subjects. In addition, I would like to thank my colleagues and friends, in alphabetical order, Nicholas Casale, Miao Feng, Qian Ge, Dr. Fengyuan Gong, Dr. Xiaofan He, Yufan Huang, Richeng Jin, Nikhil Krishnan, Dr. Chengzhi Li, Wuyuan Li, Feier Lian, Dr. Juan Liu, Dr. Yuan Lu, Yanting Ma, Ryan Pilgrim, Macey Ruble, Rafael Silva, Dr. Jin Tan, Joseph Young, and Dr. Huazi Zhang. Without their help and friendship, I could not have lived a happy life while I am working toward my Ph.D.

Furthermore, I would like to thank my college buddies, Shijie Li and Jiaming Xu, who are now pursuing their Ph.D.s as well. Without their encouragement and help, I could not have even dreamed of coming to the U.S. to pursue my Ph.D. I hope their research progress goes well and that they graduate soon. I am also very grateful to Dr. Yiming Zhu, my advisor in China, who changed my life.

At last, I would like to thank my dear parents. Whenever I need them, they are ready to help. It is their unconditional love and support that enable the endeavor of my life. They give me the courage to conquer every difficulty in the pursuit of my dream and teach me to love this world so that I am not alone. My special thanks goes to my beloved Meizhu, who accompanied me when I felt lonely, encouraged me when I was lost, and shared happiness with me whenever there were good news; life is like a box of chocolate, and you are the sweetest one.

[1] More specifically, the author was supported in part by the National Science Foundation under the Grants CCF-1217749 and ECCS-1611112, and by the U.S. Army Research Office under the Grants W911NF-04-D-0003 and W911NF-14-1-0314.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1 INTRODUCTION

Many problems in science and engineering can be approximated as linear, where an unknown vector $\mathbf{x} \in \mathbb{R}^N$ is measured via a matrix multiplication, $\mathbf{w} = \mathbf{A}\mathbf{x}$, with $\mathbf{A}$ being an $M \times N$ matrix. The measurements $\mathbf{y}$ are collected after $\mathbf{w}$ is corrupted by measurement noise $\mathbf{z} \in \mathbb{R}^M$,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}. \tag{1.1}$$

In some machine learning problems, the training set consists of $\mathbf{A}$ and $\mathbf{y}$, where $\mathbf{A}$ contains the features and $\mathbf{y}$ contains the outcomes [Ric07; McM13]; $\mathbf{x}$ is usually called the coefficient vector that describes the relation between the features and the outcomes. In signal processing, $\mathbf{A}$ describes the signal acquisition system, $\mathbf{y}$ contains the measurements, and $\mathbf{x}$ is the underlying signal [Don06a]. For communication systems such as CDMA, the matrix $\mathbf{A}$ contains the spreading sequences that spread the input (channel) symbol from each user, and then the receiver mixes the spread symbols from different users and obtains $\mathbf{y}$ [GV05]. The input symbols from different users at a certain time interval form the vector $\mathbf{x}$. For ease of presentation, we call the underlying input vector $\mathbf{x}$ the *signal*, $\mathbf{A}$ the *measurement matrix*, and $\mathbf{y}$ the *measurements* vector. In the following, we introduce several variants of our setting (1.1) and then discuss the prior art in solving the linear models.

1.1 Linear Models and Linear Inverse Problems

1.1.1 Problem setting

There are some variants of linear models (LM's). Based on how the measurements $\mathbf{y}$ and the matrix $\mathbf{A}$ are stored, we form centralized LM's or multi-processor LM's. We can also define linear models based on the number of underlying unknown vectors $\mathbf{x}$: if there is only one unknown vector $\mathbf{x}$, then it is a single measurement vector (SVM) problem; if there are more than one unknown vector $\mathbf{x}$, then we form a multi-measurement vector (MMV) problem.

Centralized vs. multi-processor LM's: If the matrix $\mathbf{A}$ and the measurements $\mathbf{y}$ in (1.1) are stored in a single processor, then we call the LM a *centralized LM*. Recently, there is an increasing amount of data being generated in various applications. For example, the trend of relying on Internet services and social networks is more prevalent than ever before; users of web services are generating numerous log files daily. As another example, financial analysts need to predict the changes in prices based on historical price information. Given the amount of financial derivatives and the high frequency of changes in prices, financial institutions are also overwhelmed by a vast amount of data. Another example involves recent advances in wearable devices. Health care providers can provide patients with wearable sensors that record and report the health status of patients frequently, so that the health care providers can react quickly once there is an emergency. With these ever-growing amounts of data, it is no longer practical to fit these data into a single machine, and distributed and scalable file systems such as Hadoop Distributed File Systems (HDFS) [DG08] have been developed. For the case of LM, if the matrix $\mathbf{A}$ and the measurements $\mathbf{y}$ are so big that they have to be stored in a distributed file system such as HDFS, then we form a *multi-processor (MP) LM* [Mot12; Pat13; Pat14; Han14; Han15b; Rav15; Han15a; Han16]. Consider an MP-LM with P distributed *processor nodes* and a *fusion center*. Each distributed processor node stores $\frac{M}{P}$ rows of the matrix $\mathbf{A}$, and acquires the corresponding measurements of the underlying signal $\mathbf{x}$. Without loss of generality, the LM in distributed processor node $p \in \{1, \cdots, P\}$ can be written as

$$y_i = \mathbf{A}_i \mathbf{x} + z_i,\ i \in \left\{ \frac{M(p-1)}{P} + 1, \cdots, \frac{Mp}{P} \right\}, \tag{1.2}$$

where $\mathbf{A}_i$ is the i-th row of $\mathbf{A}$, and y_i and z_i are the i-th entries of $\mathbf{y}$ and $\mathbf{z}$, respectively.

Single measurement vector vs. multiple measurement vectors: Apart from the MP-LM, another type of distributed linear model involves multiple sensors. Using multiple sensors can accelerate the sensing speed by pointing different sensors at different regions of interest, which we call *distributed sensing* [Dua06; HN06; Bar06]. In distributed sensing, suppose that J sensors are measuring J signal vectors, $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(J)}$. Each signal vector $\mathbf{x}^{(j)}$ is measured by a matrix $\mathbf{A}^{(j)}$, which models the sensing mechanism of each sensor, and the measurements $\mathbf{y}^{(j)}$ are corrupted by independent and identically distributed (i.i.d.) noise $\mathbf{z}^{(j)}$,

$$\mathbf{y}^{(j)} = \mathbf{A}^{(j)}\mathbf{x}^{(j)} + \mathbf{z}^{(j)}, \quad j \in \{1, \cdots, J\}, \tag{1.3}$$

where the (j) in the super-script denotes the index of the corresponding sensor. Of particular interest in reducing the number of measurements while achieving similar signal estimation quality, distributed sensing leads to a proliferation of research on the MMV problem [CH06; Cot05; ME09; BF09], in which the J sparse signal vectors $\mathbf{x}^{(j)},\ j \in \{1, \cdots, J\}$, share common non-zero supports, as explained below. Let us construct a *super-symbol* $\mathbf{x}_l = \left[x_l^{(1)}, \cdots, x_l^{(J)}\right]^\top$, where $\{\cdot\}^\top$ denotes the transpose, and $x_l^{(j)}$ is the l-th entry of the signal vector $\mathbf{x}^{(j)}$. The super-symbols $\mathbf{x}_l,\ l \in \{1, \cdots, N\}$, follow an i.i.d. J-dimensional joint distribution,

$$f(\mathbf{x}_l) = \rho\phi(\mathbf{x}_l) + (1-\rho)\delta(\mathbf{x}_l), \tag{1.4}$$

where ρ is the *sparsity rate*, $\phi(\mathbf{x}_l)$ is a J-dimensional joint distribution, and $\delta(\mathbf{x}_l)$ is the Dirac delta function for J-dimensional vectors. When the number of signal vectors becomes 1, i.e., $J = 1$, this MMV problem (1.3) becomes an SMV problem. The MMV problem has many applications such as radar array signal processing, acoustic sensing with multiple speakers, magnetic resonance imaging with multiple coils [Jun07; Jun09], and diffuse optical tomography using multiple illumination patterns [Lee11].

Linear inverse problem: Usually, estimation algorithms need to be designed to estimate the signal $\mathbf{x}$ given the matrix $\mathbf{A}$, noisy measurements $\mathbf{y}$, and possible statistical knowledge about the noise $\mathbf{z}$. We call this a linear inverse problem.

In this work, we focus on the *large system limit* defined below.

Definition 1.1 (Large system limit [GW08])**.** *The signal length N scales to infinity, and the number of measurements $M = M(N)$ depends on N and also scales to infinity, where the ratio approaches a positive constant κ,*

$$\lim_{N\to\infty} \frac{M(N)}{N} = \kappa > 0.$$

We call κ the measurement rate.

1.1.2 Prior art and open questions

Linear models are widely studied and find extensive real-world applications. Over the years, people have developed various algorithms to solve the underlying signal vectors for linear models. Many estimation algorithms pose a sparsity prior on the signal $\mathbf{x}$ or the coefficient vector θ [Can06; Don06a; Fig07], where $\theta = \mathbf{W}^{-1}\mathbf{x}$, and $\mathbf{W}$ is called the *sparsifying transform* that renders a sparse coefficient vector θ. A second, separate class of Bayesian algorithms to solve the linear inverse problem poses a probabilistic prior for the coefficients of **x** in a known transform domain [Don10; Ran11; Ji08; SN08; Bar10]. Given a probabilistic model, some related message passing approaches learn the parameters of the signal model and achieve the minimum mean squared error (MMSE) in some settings; examples include EM-GM-AMP-MOS [VS13], turboGAMP [Zin12], and AMP-MixD [Ma14b]. As a third alternative, complexity-penalized least square methods [FN03; Don06b; HN06; HN12; RS12a] can use arbitrary prior information on the signal model and provide analytical guarantees, but are only computationally efficient for specific signal models, such as the independent-entry Laplacian model [HN06]. For example, Donoho et al. [Don06b] relies on Kolmogorov complexity, which cannot be computed [CT06; LV08]. As a fourth alternative, there exist algorithms that can formulate dictionaries that yield sparse representations for the signals of interest when a large amount of training data is available [RS12a; Aha06; Mai08; Zho12]. When the signal is non-i.i.d., existing algorithms require either prior knowledge of the probabilistic model [Zin12] or the use of training data [GO07]. In spite of the numerous algorithms to solve the linear inverse problem, there are many important gaps in the prior art, such as those listed below.

1. **What is the best we can do?** Along with existing algorithms for solving linear inverse problems, researchers often provide theoretic estimation accuracy guarantees for these algorithms. However, what is often missing is the optimal estimation quality associated with the linear inverse problem itself, instead of the optimal estimation quality for a specific algorithm. Such a theoretic analysis will help us evaluate the quality of each algorithm and identify the gap between a specific algorithm and the theoretically optimal estimation quality.

2. **What are the costs of running an algorithm?** Nowadays, due to the large amounts of data mentioned in Section 1.1.1, many systems are designed in a distributed fashion. Hence, estimation algorithms need to run in a distributed network and thus incur communication costs. There exists some work trying to save communication by designing cache systems so that each node in the network does not need to send every piece of data every time [Li15; Li16]. There are also some works using heuristics in reducing the precision of the floating-point numbers sent across the network [McM13; Tha13]. However, there is little prior art discussing the “optimal” communication scheme.

3. **Better algorithms?** At the beginning of this section, we briefly discussed some classes of algorithms. In certain cases, one might not be certain about the structure or statistics of the signal prior to estimation. Uncertainty about such structure may result in a sub-optimal choice of the sparsifying transform **W**, yielding a coefficient vector θ that requires more measurements to achieve reasonable estimation quality; uncertainty about the statistics of the signal will make it difficult to select a prior or model for Bayesian algorithms. Thus, we think that a “better” algorithm should be more agnostic to the particular statistics of the signal while still achieving reasonable estimation results.

1.1.3 Contributions

In the following, we briefly discuss our contributions corresponding to each of the unsolved problems raised in Section 1.1.2. Most of our contributions are made possible by taking advantage of statistical physics tools and information theory.

1. **Characterizing the optimal estimation quality:** In Chapter 3, we make an analogy between the MMV problem (1.3) and a thermodynamic system and use the replica analysis [Tan02; GV05; MT06; Krz12a; Krz12b; MM09; BK15; Les15] from statistical physics to analyze the information theoretic MMSE for MMV problems with i.i.d. Gaussian measurement matrices and i.i.d. Gaussian noise. Our analysis is readily extended to other i.i.d. measurement matrices and i.i.d. measurement noise. Note that the MMSE is associated with the MMV problem (1.3) itself and is not associated with any specific estimation algorithms. Realizing that mean squared error (MSE) might not be the only metric that is of interest, we propose a future direction to extend the work of Tan and coauthors [Tan14a; Tan14b] to analyze the average error based on arbitrary user-defined error metrics for MMV problems.

2. **Optimal trade-offs among different costs:** In Chapter 4, we apply rate-distortion theory [CT06; Ber71; GG93; WV12a] to optimize the communication cost in a specific distributed algorithm, and propose a method to find the optimal combined cost of computation and communication. In addition, we study the asymptotic behavior of the optimal communication scheme in the limit of low excess MSE beyond the MMSE. Also, recognizing that we cannot minimize the computation cost, communication cost, and the quality of the estimate simultaneously, we study the optimal trade-offs among these different costs.

3. **Designing better algorithms:** In Chapter 5, we propose a *universal* algorithm that is based on the mild assumption of the signal being “simple,” i.e., there is some structure in the signal that is simple. Our algorithm is based on “simulated annealing,” a mathematical analogy to a statistical physics concept, and achieves favorable estimation accuracy while using limited prior information about the signal models. In Chapter 5, we also briefly discuss another universal algorithm that is based on belief propagation [Don09; Bar10; BM11; Mon12; Krz12a; Krz12b; BK15], which originates from statistical physics and information theory. We refer interested readers to Ma et al. [Ma14a; Ma16].

The underlying intuition of why statistical physics and information theory can be useful in tackling our problems is that they both deal with large systems, and fortunately, the problems that we are targeting in this dissertation are indeed large systems. Moreover, the general formulations of our problems create analogies between our problems and thermodynamic systems and communication systems, so that we can take advantage of the existing analytical and algorithmic tools in the rich fields of statistical physics and information theory.

1.2 Organization, Notations, and Acronyms

1.2.1 Organization

This dissertation is organized as follows. Chapter 2 introduces some background on statistical physics and information theory. Chapter 3 studies the MMSE and its behavior for MMV problems; we also propose a future direction to study arbitrary user-defined error metrics for MMV problems. The limiting behavior of the optimal communication scheme and the optimal trade-offs among different costs in MP-LM's are discussed in Chapter 4. In Chapter 5, we propose a universal algorithmic framework that achieves favorable estimation quality. Chapter 6 concludes the dissertation and proposes some future directions. Details about some proofs appear in the appendices.

Note that Chapter 3 is based on our work with Baron [ZB13] and with Baron and Krzakala [Zhu16b]. Chapter 4 is based on our work with Han et al. [Han16] and with Baron and Beirami [Zhu16c; Zhu16a]. Chapter 5 is based on our work with Baron and Duarte [Zhu14; Zhu15].

1.2.2 Notations

In this dissertation, bold capital letters represent matrices, bold lower case letters represent vectors, and normal font letters represent scalars. The entry (scalar) in the i-th row, j-th column of a matrix $\mathbf{A}$ is denoted by $A_{i,j}$, where the comma is often omitted. The i-th entry (scalar) in a vector $\mathbf{z}$ is denoted by z_i. Following are some frequently used notations.

- $\mathbf{A}$: Measurement matrix
- $\mathbb{C}$: The set of complex numbers
- D: Distortion
- $\delta(\cdot)$: Dirac delta function
- $f(\cdot)$: Probability density function (continuous variable)
- $\mathbb{E}[\cdot]$: Expectation
- κ: Measurement rate
- M: Number of measurements
- N: Signal length
- $\mathbb{N}$: The set of natural numbers, i.e., $\{0, 1, \cdots\}$
- $\mathcal{N}(\mu, \sigma^2)$: Gaussian distribution with mean μ and variance σ^2

- R: Coding rate
- $\mathbb{R}$: The set of real numbers
- $\mathbb{P}$: Probability
- $\mathbb{P}(\cdot)$: Probability mass function (discrete variable)
- ρ: Sparsity rate (percentage of non-zeros in a vector)
- σ_Z^2: Variance of the noise $\mathbf{z}$
- t: Iteration index
- $\mathbf{A}^\top$: Transpose of matrix $\mathbf{A}$
- $\mathbf{x}$: Signal
- $||\mathbf{x}||_p$: ℓ_p norm of a vector $\mathbf{x}$; if p is not specified, then we refer to ℓ_2 norm
- $\mathbf{y}$: Measurements
- $\mathbf{z}$: Noise
- $[x_1, x_2, \cdots, x_N]$: The vector consists of $x_1, x_2, \cdots, x_N$
- $\{1, 2, \cdots, N\}$: The set consists of $1, 2, \cdots, N$

1.2.3 Acronyms

- AMP: Approximate message passing
- BP: Belief propagation
- CS: Compressed sensing
- i.i.d.: Independent and identically distributed
- LM: Linear model
- MMSE: Minimum mean squared error
- MMV: Multi-measurement vector
- MP: Multi-processor
- MSE: Mean squared error

- PMF: Probability mass function
- RD: Rate-distortion
- SDR: Signal-to-distortion ratio
- SMV: Single measurement vector
- SNR: Signal-to-noise ratio

CHAPTER

2

STATISTICAL PHYSICS AND INFORMATION THEORY BACKGROUND

In Chapter 1, we discussed the prior art and mentioned that our contributions are made possible by tools in statistical physics and information theory. Due to the interdisciplinary nature of this dissertation, this chapter briefly reviews some concepts and methodologies that are used in our work. We refer readers who are interested in delving into these subjects to the books by Mézard and Montanari [MM09] and by Cover and Thomas [CT06].

2.1 Relevant Statistical Physics Concepts

Statistical physics studies a disordered thermodynamic system containing a large number of particles that are interacting with each other by the internal force between (among) the particles as well as the external force applied to the entire disordered system.

2.1.1 Basics

In this section, we briefly introduce some concepts that are frequently used in statistical physics.

Entropy (thermodynamics): Entropy quantifies the amount of disorder of a thermodynamic system,

$$\mathscr{S}(\mathbf{x}) = -\sum_{\mathbf{x}} \mathbb{P}(\mathbf{x}) \log \mathbb{P}(\mathbf{x}), \tag{2.1}$$

where the vector **x** describes the *configuration* of a certain thermodynamic system and $\mathbb{P}(\mathbf{x})$ is the probability of a certain configuration existing in the disordered system. By summing over all possible configurations and accounting for their corresponding probability, we are able to obtain the level of disorder, or the *entropy* of this particular thermodynamic system.

Boltzmann distribution: In a thermodynamic system, the higher the temperature is, the more disordered the system is. The Boltzmann distribution is a probability distribution used to describe various possible configurations in a thermodynamic system,

$$\mathbb{P}(\mathbf{x}) = \frac{1}{Z} \exp\left(-\frac{H(\mathbf{x})}{T}\right), \tag{2.2}$$

where the vector **x** describes the configuration of a thermodynamic system, T is the temperature of this system, $H(\mathbf{x})$ is the energy for a certain configuration, and Z is a normalizer called the *partition function*. If the thermodynamic system is in a high temperature, i.e., T is large, then the probabilities for configurations with different energy are approximately the same and the system reaches the maximum entropy (2.1), which corresponds to the greatest amount of disorder.

Annealing and quench: The configuration associated with the lowest energy can be obtained through a process called annealing, where a disordered system gradually cools down. Intuitively, when the temperature T decreases, the configurations with lower energy becomes more and more likely in the disordered system, according to (2.2). Given enough time that allows a slow enough decrease in the temperature, we can guarantee to obtain the globally minimum energy configuration. A related concept is *quench*, in which the temperature is quickly decreased, so that the disordered system is likely to achieve a local minimum energy configuration. Since the temperature is quickly decreased, once a local minimum energy configuration appears, it will be difficult to generate other lower energy configurations according to (2.2).

2.1.2 Spin glass theory basics

A basic understanding of spin glass theory provides new perspectives when solving linear inverse problems. In the following, we introduce some basics of spin glass theory. The goal is to provide intuition, and we refer interested readers to Mézard and Montanari [MM09] for rigorous and detailed explanations.

Mean-field spin glasses: As discussed in Section 2.1.1, the thermodynamic system we are interested in contains many particles. A *simple model* in the mean-field spin glass theory models each of the particles as a spinning glass, where each glass has two spinning states. *In this simple*

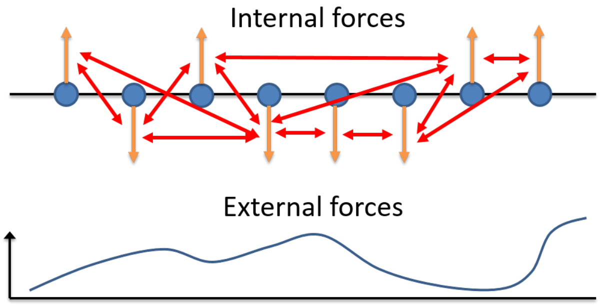


Figure 2.1 Illustration of spin glasses with internal and external forces. Each dot represents a spin glass. Vertical arrows denote the state of each glass. The remaining arrows illustrate the internal forces between pairs of spin glasses and the curve in the bottom panel illustrates the external force. Figure inspired by Ralf R. Müller.

model, there exist internal forces between *each pair* of the spinning glasses. Moreover, we assume that there is an external force that can affect the states of the glasses. Hence, the overall energy of a specific thermodynamic system for a specific *configuration* $\mathbf{x}$ is

$$H(\mathbf{x}) = -\sum_{i}\sum_{j<i} r_{ij} x_i x_j - \sum_{i} h_i x_i, \tag{2.3}$$

where x_i is the i-th element of the configuration (vector) $\mathbf{x}$ and it represents the state of the i-th glass, r_{ij} models the force between glass i and glass j, and h_i models the external force applied to glass i. This model is illustrated in Figure 2.1, where each dot represents a glass, and the vertical arrows denote the state of each glass. The remaining arrows illustrate the internal forces between pairs of spin glasses and the curve in the bottom panel illustrates the external force. The energy function (2.3) is often called the *Hamiltonian*. Note that the Hamiltonian (2.3) is *quenched*, because we assume that r_{ij} and h_i are constant.

One of the things that nature does is maximizing the entropy (2.1) of a thermodynamic system for a given energy (because energy is assumed to be conserved),

$$\mathscr{E} = \sum_{\mathbf{x}} \mathbb{P}(\mathbf{x}) H(\mathbf{x}). \tag{2.4}$$

It can be proved that the Boltzmann distribution (2.2) maximizes the entropy (2.1) for a given energy (2.4). Moreover, the energy $H(\mathbf{x})$ in the Boltzmann distribution (2.2) is the Hamiltonian for configuration $\mathbf{x}$ (2.3).

Free energy and self-averaging: Sometimes, instead of (mathematically) evaluating the maximum entropy (2.1), it is more convenient to evaluate the minimum *free energy* given by

$$\mathscr{F} = \mathscr{E} - T\mathscr{S}. \tag{2.5}$$

Using (2.1), (2.2), and (2.4) with normalization by the number of spin glasses N, we simplify (2.5) as

$$\mathscr{F} = -\frac{T}{N}\log Z, \tag{2.6}$$

where the partition function Z is the normalizer in (2.2). Note that because the Hamiltonian (2.3) is quenched, the free energy (2.6) is quenched.

The expression in (2.6) is undesirable, because we have to calculate the free energy for each of the quenched Hamiltonians. Physically, it means that we need to carry out this calculation for every specific piece of material. It turns out that when the size of the system is sufficiently large, the properties of the system do not depend on the specific settings of r_{ij} and h_i any more (2.3), which is the so-called *self-averaging* property of a thermodynamic system, given sufficiently many particles. Hence, we define the free energy as

$$\mathscr{F} = -\lim_{N\to\infty}\frac{T}{N}\mathbb{E}\left[\log Z\right]. \tag{2.7}$$

2.2 Information Theory and Coding Theory

This section discusses some important results from information theory and coding theory that are relevant to this dissertation. The author refers interested readers to the book by Cover and Thomas [CT06] for further details and more comprehensive explanations. Coding theory and information theory are quite related and are both widely used in digital communication systems, and we simply call them “information theory” for brevity. Seeing that information theory is widely used in digital communication systems, we start by introducing the components of a typical digital communication system. But before that, we must understand the most basic of concepts: the bit.

Bit: A bit is a unit that can represent two states. We could call these two states 0 and 1, or -1 and +1, and so on. Why are bits so important? Before entering the digital world, people used analog electronics. One of the key challenges was the noise in the signal. For example, in order to represent a number 1.2, a waveform of magnitude 1.2 needs to be formed and transmitted. However, due to various noise and distortions, what the receiver receives is not exactly 1.2, which is undesirable. In the digital world, devices use sequences of bits to represent a number such as 1.2. The advantage of digital electronics is that they use “bits” that only have two states: the circuit is either on or off. The recognition and identification of a bit are much easier than recognizing and identifying analog waveforms. Information theory provides theoretical bounds for various errors when using bits, and proves that by using bits a digital communication system can exploit the communication channel as well as an analog communication system does. Moreover, information theory develops many techniques to achieve these theoretical bounds.

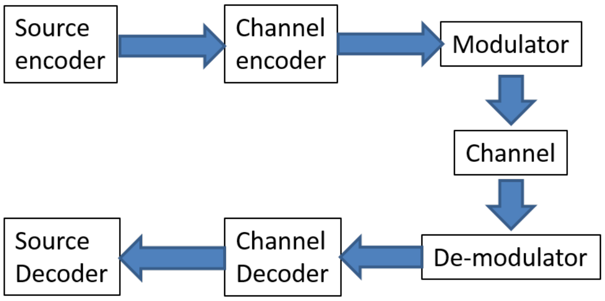


Figure 2.2 Illustration of a typical digital communication system. Figure inspired by Brian Hughes' slides.

Components of a digital communication system: As illustrated in Figure 2.2, there are 7 key components of a typical digital communication system. First, the signal is encoded (compressed), so that the communication system does not need to send as many bits as required by the original signal; this step is called *source encoding*. Then, the encoded (compressed) signal is passed through a channel encoder, in which redundancy is introduced to the bit sequence. This redundancy is crucial to better utilize the energy of the transmitter and the channel. Next, the redundant sequence is modulated to an analog waveform by one of the available modulation schemes. After modulation, the transmitter sends the modulated signals (analog) through a noisy channel and the receiver receives a noisy sequence that contains the information of the original signal. Then, the receiver demodulates the noisy analog waveform into a sequence of bits. After that, the receiver decodes (channel decoder) the sequence to remove redundancy.[1] Finally, with an error-free (hopefully) sequence of bits, the last step is to decompress the data.

Link between statistical physics and information theory:[2] In Section 2.1, we denote the configuration of a thermodynamic system by a vector $\mathbf{x}=[x_1,\cdots,x_N]$, where $x_i,\ i\in\{1,\cdots,N\}$, represents the state of the i-th spin glass. In information theory, we typically use $\mathbf{x}=[x_1,\cdots,x_N]$ to represent a length-N signal. This signal $\mathbf{x}$ is passed through a channel. The counterparts of the channel in digital communication systems for statistical physics are the internal and external forces that interact with the particles of the thermodynamic system. With this brief analogy, we start introducing some important concepts and results in information theory.

Entropy (information theory): We have introduced entropy (2.1) in statistical physics. In information theory, entropy quantifies the amount of information carried by a certain signal $\mathbf{x}$. If the entries of $\mathbf{x}$ take discrete values, then the expression for entropy in information theory is the same as (2.1), and the only difference is that $\mathbb{P}(\mathbf{x})$ represents the joint probability mass function of a signal

[1] There will be errors in the demodulated sequence. By introducing redundancy in the channel encoding step, the channel decoder can identify and correct errors due to the noisy channel.

[2] Interested readers may want to refer to Merhav [Mer10].

x. If the entries of **x** are continuous, then the entropy in information theory for a signal **x** is

$$\mathscr{S}(\mathbf{x}) = -\int_{\mathbf{x}} f(\mathbf{x}) \log[f(\mathbf{x})] d\mathbf{x}, \tag{2.8}$$

where $f(\mathbf{x})$ is the joint probability density function of **x**.

Coding rate (source encoder): Before transmitting the signal $\mathbf{x} \in \mathbb{R}^N$ to the receiver, a communication system typically first compresses the signal, so that it can save in communication load. The coding rate is defined as

$$R = \frac{\text{Number of bits after compression}}{N}. \tag{2.9}$$

Distortion: After receiving the encoded signal,[3] the receiver needs to decode it. There are two types of data compression that can be used in the source encoder. One is *lossless compression* and the other is *lossy compression*. In lossless compression, after the source decoder decodes the data sequence, it obtains a signal that is identical to the original signal. In lossy compression, the signal obtained after decoding is somewhat distorted from the original signal. The cause of this *distortion* is the *quantization* process when encoding the signal in a lossy way. A typical quantizer builds a "grid" in the space of value(s) to be quantized. Next, the quantizer rounds the value(s) to the nearest point on the grid. As an example, the scalar quantizer [GG93; CT06] rounds each (scalar) entry in the signal to the nearest grid point. The vector quantizer [Lin80; Gra84; GG93] rounds sequence of scalars to the nearest hyper-grid point.

Denote the distance between a certain entry in the original signal x_i and the corresponding entry in the decoded signal $\widehat{x}_i$ by $d(x_i, \widehat{x}_i)$, where we can use various distance functions [Kre89] for $d(\cdot,\cdot)$. The average distortion of the entire signal is given by

$$D = \frac{1}{N} \sum_{i=1}^{N} d(x_i, \widehat{x}_i). \tag{2.10}$$

Rate-distortion theory: There is a fundamental information theoretic relation between the rate (2.9) and distortion (2.10). With a certain quantization scheme and knowledge about the distribution of the signal, we can calculate the coding rate R (2.9) and the expected distortion D (2.10). Although this calculation is not always an easy task [Ari72; Bla72; Ros94], a pivotal message from rate-distortion theory is that we can save a lot in the coding rate R (2.9) by allowing a small distortion D (2.10).

Cavity method and belief propagation: We can regard the linear model in (1.1) as a communication channel, where **x** is the signal to be transmitted, **A** models the transmission scheme, **z** is the

[3] According to Figure 2.2, after data compression and before transmitting the sequence, there is typically a channel encoding step, which helps to exploit the channel to a greater extent. Here, we assume perfect channel decoding. Interested readers can refer to Cover and Thomas [CT06].

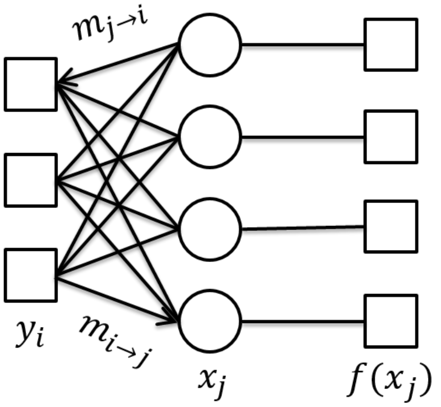


Figure 2.3 Illustration of belief propagation. The boxes are called the factor nodes and the circles are called the variable nodes.

noise in the receiver, and **y** is the received sequence. Belief propagation (BP) [Don09; Bar10; BM11; Mon12; Krz12a; Krz12b; BK15] is an algorithm that can be used to infer the underlying signal **x** in the channel (1.1). BP was invented independently by researchers in coding theory, statistical physics, and artificial intelligence. First of all, we represent the channel (1.1) as a Tanner graph in Figure 2.3, where we express each entry x_j of the signal **x** by a variable node (circles in Figure 2.3), driven by its distribution $f(x_j)$ from a factor node (boxes in Figure 2.3). Then, variable nodes are interacting with the factor nodes y_i's.

The messages $m_{i\to j}(x_j)$ and $m_{j\to i}(x_j)$ given by the canonical BP updating rules for the posterior distribution $f(\mathbf{x}|\mathbf{y})$ are as follows,

$$\begin{aligned} m_{i\to j}(x_j) &= \frac{1}{Z_{i\to j}} \int \left[\prod_{k\neq j} m_{k\to i}(x_k)\right] \mathrm{e}^{-\frac{1}{2\sigma_Z^2}\left(\sum_{k\neq j} A_{ik}x_k + A_{ik}x_k - y_i\right)^2} \left[\prod_{k\neq j} dx_k\right], \\ m_{j\to i}(x_j) &= \frac{1}{Z_{j\to i}} f(x_j) \prod_{q\neq j} m_{q\to j}(x_j). \end{aligned} \tag{2.11}$$

Note that in statistical physics, the factor nodes model the forces between (or among) spin glasses (variable nodes). When **A** is sparse or locally tree-like, BP yields an estimate that converges to the true posterior distribution $f(\mathbf{x}|\mathbf{y})$. With this posterior distribution, we obtain the estimate $\widehat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}]$ of the original signal that achieves the smallest mean squared error [Ran11].

CHAPTER

3

MINIMUM MEAN SQUARED ERROR FOR MULTI-MEASUREMENT VECTOR PROBLEM

The multi-measurement vector (MMV) problem (1.3) considers the estimation of a set of sparse signal vectors that share common supports, and has applications such as radar array signal processing, acoustic sensing with multiple speakers, magnetic resonance imaging with multiple coils [Jun07; Jun09], and diffuse optical tomography using multiple illumination patterns [Lee11]. In this chapter, which is based on our work with Baron [ZB13] and with Baron and Krzakala [Zhu16b], two related MMV settings are studied. In the first setting, each signal vector is measured by a different independent and identically distributed (i.i.d.) measurement matrix, while in the second setting, all signal vectors are measured by the same i.i.d. matrix. Although there are many algorithms [Dua13; Tro06b; CH06; Mal05; Tro06a; Cot05; ME09; Lee12; Ye15; ZS11] for solving the unknown vectors in the MMV problem (1.3), the performance limits of MMV signal estimation in the presence of measurement noise have not been studied. In this chapter, replica analysis [Tan02; GV05; MT06; Krz12a; Krz12b; MM09; BK15; Les15], borrowed from statistical physics, is performed for these two MMV settings, and the minimum mean squared error (MMSE), which turns out to be identical for both settings, is obtained as a function of the noise variance and number of measurements. To showcase the application of MMV models, the MMSE's of complex single measurement vector (SMV) problems with both real and complex measurement matrices are also analyzed. Multiple performance regions for MMV are identified where the MMSE behaves differently as a function of the noise variance and the number of measurements.

Belief propagation (BP) is a signal estimation framework for linear inverse problems that often achieves the MMSE asymptotically. A phase transition for BP is identified. This phase transition, verified by numerical results, separates the regions where BP achieves the MMSE and where it is sub-optimal. Numerical results also illustrate that more signal vectors in the jointly sparse signal ensemble lead to a better phase transition.

Realizing that the mean squared error might not be the only error metric that is of interest, we propose some future directions involving the study of optimal performance for arbitrary user-defined additive error metrics for MMV problems by extending the work of Tan and coauthors [Tan14a; Tan14b].

3.1 Related Work and Contributions

In multi-measurement vector (MMV) problems, thanks to the common support, the number of sparse coefficients that can be successfully estimated increases with the number of measurements. This property was evaluated rigorously for noiseless measurements using ℓ_0 minimization [Dua13]. To address measurement noise, estimation approaches for MMV problems have included greedy algorithms such as SOMP [Tro06b; CH06], ℓ_1 convex relaxation [Mal05; Tro06a], and M-FOCUSS [Cot05]. REduce MMV and BOost (ReMBo) has been shown to outperform conventional methods [ME09], and subspace methods have also been used to solve MMV problems [Lee12; Ye15]. Statistical approaches [ZS11] often achieve the oracle minimum mean squared error (MMSE). However, the performance limits of MMV signal estimation in the presence of measurement noise have not been studied.

Replica analysis is a statistical physics method that can be used to analyze the MMSE and phase transition for inverse problems [Tan02; GV05; MT06; Krz12a; Krz12b; MM09; BK15; Les15]. Barbier and Krzakala [BK15] studied the MMSE for estimating superposition codes using replica analysis. In this chapter, we extend the derivation in Barbier and Krzakala [BK15] to two related yet different MMV settings: (*i*) J jointly sparse signals are measured by J different dense matrices that are independent and identically distributed (i.i.d.), and (*ii*) J jointly sparse signals are measured by J identical i.i.d. matrices. We only consider dense i.i.d. Gaussian matrices in this work, while our analysis can be extended to other i.i.d. matrices easily.

We make several contributions in this chapter. First, we obtain the information theoretic MMSE for the two MMV settings above under the Bayesian setting. Second, we show that in the large system limit (defined in Definition 1.1) the MMSE's for these two settings are identical to the single measurement vector (SMV) problem with a dense measurement matrix and a block sparse signal with fixed length blocks. Third, we derive the MMSE for complex SMV problems by noticing that complex SMV is essentially an MMV problem. Fourth, we identify several performance regions for MMV, where the MMSE has different characteristics based on the channel noise variance and measurement rate. Finally, we find a phase transition for belief propagation algorithms (BP) [Don09; Bar10; BM11; Mon12; Krz12a; Krz12b; BK15] applied to MMV problems, which separates regions where BP achieves the MMSE asymptotically and where it is sub-optimal. BP simulation results confirm the phase transition results. Seeing that the mean squared error (MSE) might not be the only error metric that is of interest, we propose a future direction to extend the work of Tan and coauthors [Tan14a; Tan14b] to MMV settings, so that we can analyze the performance limits for arbitrary user-defined additive error metrics, as well as design an algorithmic framework that can achieve such performance limits.

The remainder of this chapter is organized as follows. We introduce our signal and measurement models in Section 3.2, followed by replica analysis for two MMV settings as well as two complex SMV problems in Section 3.3. Section 3.4 proves the results of Section 3.3. Numerical results are discussed in Section 3.5. Section 3.6 proposes some future directions to study the performance of arbitrary user-defined additive error metrics for MMV problems and we conclude in Section 3.7. Some detailed derivations appear in Appendix A.

3.2 Signal and Measurement Models

Signal model: We consider an ensemble of J signal vectors, $\underline{\mathbf{x}}^{(j)} \in \mathbb{R}^N,\ j \in \{1, \cdots, J\}$, where j is the index of the signal. As in Section 1.1.1, we consider a *super-symbol*$\mathbf{x}_l = \left[\underline{x}_l^{(1)}, \cdots, \underline{x}_l^{(J)}\right]^\top,\ l \in \{1, \cdots, N\}$, where $\{\cdot\}^\top$ denotes the transpose. The super-symbol $\mathbf{x}_l$ follows a J-dimensional Bernoulli-Gaussian distribution (defined in (1.4)),

$$f(\mathbf{x}_l) = \rho\,\phi(\mathbf{x}_l) + (1-\rho)\delta(\mathbf{x}_l), \tag{3.1}$$

where ρ is the sparsity rate, $\phi(\mathbf{x}_l)$ is a J-dimensional Gaussian distribution with zero mean and identity covariance matrix, and $\delta(\mathbf{x}_l)$ is the delta function for J-dimensional vectors.

Definition 3.1 (Jointly sparse)**.** *Ensembles of signals that obey* (3.1) *are called jointly sparse.*

Measurement models: Each signal $\underline{\mathbf{x}}^{(j)}$ is measured by an i.i.d. Gaussian measurement matrix $\underline{\mathbf{A}}^{(j)} \in \mathbb{R}^{M\times N}$, $\underline{A}_{\mu l}^{(j)} \sim \mathcal{N}(0, \frac{1}{N})$, where μ refers to the row index and l is the column index. The measurements $\underline{\mathbf{y}}^{(j)}$ are corrupted by i.i.d. Gaussian noise $\underline{\mathbf{z}}^{(j)}$ consisting of entries $\underline{z}_{\mu}^{(j)} \sim \mathcal{N}(0, \sigma_Z^2)$,

$$\underline{\mathbf{y}}^{(j)} = \underline{\mathbf{A}}^{(j)}\underline{\mathbf{x}}^{(j)} + \underline{\mathbf{z}}^{(j)}, \quad j \in \{1, \cdots, J\}. \tag{3.2}$$

When the number of signal vectors becomes $J = 1$, this MMV model (3.2) becomes an SMV problem. Note that SMV and MMV problems were motivated in (1.1) and (1.3), respectively. Our analysis in this chapter is readily extended to other i.i.d. matrices, jointly sparse signals (3.1), and other i.i.d. noise distributions.

Definition 3.2 (MMV-1)**.** *The setting MMV-1 refers to the measurement model in* (3.2) *with all matrices* $\underline{\mathbf{A}}^{(j)}$ *being different.*

Definition 3.3 (MMV-2)**.** *The setting MMV-2 refers to the measurement model in* (3.2) *with all matrices* $\underline{\mathbf{A}}^{(j)}$ *being equal.*

In the signal model (3.1) and measurement model (3.2), the sparsity rate ρ, channel noise variance σ_Z^2, and number of channels J are constant. We are interested in the large system limit, which has been defined in Definition 1.1 in Section 1.1.1. For readers' convenience, we restate the definition of the large system limit as follows.

Definition 3.4 (Large system limit [GW08])**.** *The signal length N scales to infinity, and the number of measurements $M = M(N)$ depends on N and also scales to infinity, where the ratio approaches a positive constant κ,*

$$\lim_{N\to\infty} \frac{M(N)}{N} = \kappa > 0. \tag{3.3}$$

We call κ the measurement rate.

3.3 Replica Analysis for MMV Settings

Section 3.2 discussed two MMV settings. Both settings have applications in real-world problems such as magnetic resonance imaging [Jun07; Jun09] and sensor networks [PK00]. Although numerous algorithms for MMV signal estimation have been proposed [Tro06b; CH06; Mal05; Tro06a; Cot05; ME09; ZS11], what is often missing is an information theoretic analysis of the best possible MSE performance. In this chapter, we only consider the MSE as our performance metric, except for Section 3.6.

3.3.1 Statistical physics background and replica method

In order to express (3.2) using a single channel, we transform it to an SMV form. One possible way to do so is illustrated in Figure 3.1. The equivalent SMV problem is

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \tag{3.4}$$

where $\mathbf{A} \in \mathbb{R}^{MJ\times NJ}$ is the matrix, $\mathbf{y} \in \mathbb{R}^{MJ}$ are the measurements, and the noise is $\mathbf{z} \in \mathbb{R}^{MJ}$. Entries of the signal vectors $\underline{\mathbf{x}}^{(j)}$, measurement vectors $\underline{\mathbf{y}}^{(j)}$, and noise vectors $\underline{\mathbf{z}}^{(j)}$ in (3.2) form the SMV signal

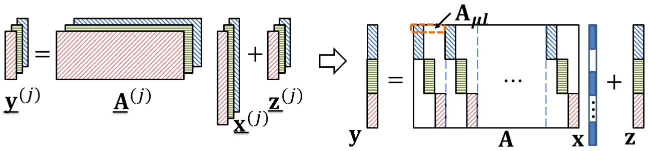

$\underline{\mathbf{y}}^{(j)}$
=
$\underline{\mathbf{A}}^{(j)}$
$\underline{\mathbf{x}}^{(j)}$
+
$\underline{\mathbf{z}}^{(j)}$
$\mathbf{A}_{\mu l}$
$\mathbf{y}$
=
$\cdots$
$\mathbf{A}$
$\mathbf{x}$
+
$\mathbf{z}$

Figure 3.1 Illustration of MMV channel (3.2) with $J = 3$ signal vectors (left), and one of its possible SMV forms (right). Different background patterns differentiate entries from different channels, and blank space denotes zeros.

x, measurements **y**, and noise **z** (3.4) with

$$x_{(l-1)J+j} = \underline{x}_l^{(j)},\ y_{(j-1)M+\mu} = \underline{y}_\mu^{(j)},\ \text{and}\ z_{(j-1)M+\mu} = \underline{z}_\mu^{(j)},$$

respectively. Entries of the matrix $\underline{\mathbf{A}}^{(j)}$ (3.2) form the SMV matrix **A** (3.4) with $A_{(j-1)M+\mu,(l-1)J+j} = \underline{A}_{\mu l}^{(j)}$; other entries of **A** are zeros. The posterior for the estimate $\widehat{\mathbf{x}} \in \mathbb{R}^{NJ}$, comprised of super-symbols $\widehat{\mathbf{x}}_l = \left[\widehat{x}_{(l-1)J+1}, \cdots, \widehat{x}_{lJ}\right]^\top$, $l \in \{1, \cdots, N\}$, is

$$f(\widehat{\mathbf{x}}|\mathbf{y}) = \frac{1}{Z} \prod_{l=1}^{N} f(\widehat{\mathbf{x}}_l) \prod_{\mu=1}^{MJ} \left[\frac{\mathrm{e}^{-\frac{1}{2\sigma_Z^2}\left(y_\mu - \sum_{l=1}^{N} \mathbf{A}_{\mu l}\widehat{\mathbf{x}}_l\right)^2}}{\sqrt{2\pi\sigma_Z^2}} \right], \tag{3.5}$$

where $\mathbf{A}_{\mu l} = [A_{\mu,(l-1)J+1}, \cdots, A_{\mu,lJ}]$ is a super-symbol highlighted by the dashed area in Figure 3.1, and the denominator Z is the partition function [Tan02; GV05; Krz12a; Krz12b; MM09; BK15],

$$Z = \int \prod_{l=1}^{N} f(\widehat{\mathbf{x}}_l) \prod_{\mu=1}^{MJ} \left[\frac{\mathrm{e}^{-\frac{1}{2\sigma_Z^2}\left(y_\mu - \sum_{l=1}^{N} \mathbf{A}_{\mu l}\widehat{\mathbf{x}}_l\right)^2}}{\sqrt{2\pi\sigma_Z^2}} \right] \prod_{l=1}^{N} d\widehat{\mathbf{x}}_l. \tag{3.6}$$

Note that multi-dimensional integrations such as (3.6) are denoted by a single $\int$ operator for brevity. Confining our attention to the Bayesian setting [Krz12a; Krz12b; BK15], $f(\widehat{\mathbf{x}}_l)$ follows the true distribution (3.1), $f(\widehat{\mathbf{x}}_l) = \rho\phi(\widehat{\mathbf{x}}_l) + (1-\rho)\delta(\widehat{\mathbf{x}}_l)$.

By creating an analogy between the channel (3.4) and a many-body thermodynamic system [Tan02; GV05; Krz12a; Krz12b; MM09; BK15], the posterior (3.5) can be interpreted as the Boltzmann measure on a disordered system with the following Hamiltonian,

$$H(\widehat{\mathbf{x}}) = \sum_{l=1}^{N} \log[f(\widehat{\mathbf{x}}_l)] + \sum_{\mu=1}^{MJ} \frac{1}{2\sigma_Z^2} \left(y_\mu - \sum_{l=1}^{N} \mathbf{A}_{\mu l}\widehat{\mathbf{x}}_l \right)^2. \tag{3.7}$$

The averaged free energy of the disordered system given by (3.7) characterizes the thermodynamic properties of the system. Evaluating the fixed points (local maxima) in the free energy expression provides the MMSE for the channel (3.4) [Tan02; GV05; Krz12a; Krz12b; MM09; BK15]. *Under the assumption of self-averaging* [Tan02; GV05; Krz12a; Krz12b; MM09; BK15], the free energy is defined as[1]

$$\mathscr{F} = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[\log(Z)], \tag{3.8}$$

which is difficult to evaluate. Note that $\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[\cdot]$ denotes expectation with respect to (w.r.t.) $\mathbf{A}, \mathbf{x}$, and $\mathbf{z}$. The replica method [Tan02; GV05; Krz12a; Krz12b; MM09; BK15] introduces n replicas of the estimate $\widehat{\mathbf{x}}$ as $\widehat{\mathbf{x}}^a,\ a \in \{1, \cdots, n\}$, and the free energy (3.8) can be approximated by the replica trick [Krz12a; Krz12b; MM09; BK15],

$$\mathscr{F} = \lim_{N\to\infty} \lim_{n\to 0} \frac{\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[Z^n] - 1}{Nn}. \tag{3.9}$$

Note that the self-averaging property that leads to (3.8) and the replica trick (3.9), as well as the replica symmetry assumptions that appear in latter parts of this chapter, are assumed to be valid in this work, and their rigorous justification is still an open problem in mathematical physics [Tan02; GV05; Krz12a; Krz12b; MM09; BK15].[2]

Evaluating the free energy: To evaluate the free energy (3.9), we calculate $\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[Z^n]$ as follows,

$$\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}\left[Z^n\right] = (2\pi\sigma_Z^2)^{-\frac{nMJ}{2}} \times \mathbb{E}_{\mathbf{x}}\left[\int \prod_{l=1}^{N}\prod_{a=1}^{n} f(\widehat{\mathbf{x}}_l^a) \prod_{\mu=1}^{M} \mathbb{X}_\mu \prod_{l=1}^{N}\prod_{a=1}^{n} d\widehat{\mathbf{x}}_l^a\right], \tag{3.10}$$

where Z is given in (3.6),

$$\mathbb{X}_\mu = \mathbb{E}_{\mathbf{A},\mathbf{z}}\left[\mathrm{e}^{-\frac{1}{2\sigma_Z^2}\sum_{j=1}^{J}\sum_{a=1}^{n}(v_{\mu j}^a)^2}\right], \tag{3.11}$$

a is the replica index, $\widehat{\mathbf{x}}_l^a$ is the l-th super-symbol of $\widehat{\mathbf{x}}^a$, and

$$v_{\mu j}^a = \sum_{l=1}^{N} \mathbf{A}_{\mu+M(j-1),l}(\mathbf{x}_l - \widehat{\mathbf{x}}_l^a) + z_{\mu+M(j-1)}. \tag{3.12}$$

Lemma 3.1. *In the large system limit, the quantity* $\mathbb{X}_\mu$ (3.11) *is the same for both MMV-1 and MMV-2.*

Lemma 3.1 is proved in Section 3.4. Because of Lemma 3.1, the free energy expressions for MMV-1 and MMV-2 should be identical in the large system limit. We state the result as a theorem and the detailed derivations appear in Appendix A.

[1] Part of the literature [Tan02; GV05], including (2.7) in this dissertation, defines the free energy as the negative of (3.8), so that fixed points of the free energy correspond to local minima.

[2] Recently, the replica Gibbs free energy has been proven rigorously for the SMV case by Barbier et al. [Bar16] and Reeves and Pfister [RP16]. We conjecture that by generalizing these two works [Bar16; RP16], our MMV analysis can be made rigorous; we leave it for future work.

Theorem 3.1 (Free energy for MMV)**.** *For settings MMV-1 and MMV-2, the free energy expressions as functions of E are identical in the large system limit and are given below,*

$$\begin{aligned}
\mathscr{F}(E) &= -\frac{J}{2}\kappa\left\{\log[2\pi(\sigma_Z^2+E)]+\frac{\rho+\sigma_Z^2}{E+\sigma_Z^2}\right\}+ \\
&\quad \int f(\mathbf{x}_1)\int \log\left[\int f(\widehat{\mathbf{x}}_1)\mathrm{e}^{-\frac{\widehat{Q}+\widehat{q}}{2}\widehat{\mathbf{x}}_1^\top\widehat{\mathbf{x}}_1+\widehat{m}\widehat{\mathbf{x}}_1^\top\mathbf{x}_1+\sqrt{\widehat{q}}\mathbf{h}^\top\widehat{\mathbf{x}}_1}\,d\widehat{\mathbf{x}}_1\right]\mathscr{D}\mathbf{h}\,d\mathbf{x}_1 \qquad (3.13)\\
&= -\frac{J}{2}\kappa\left\{\log[2\pi(\sigma_Z^2+E)]+\frac{\sigma_Z^2}{E+\sigma_Z^2}\right\}+\frac{JR(1-\rho)}{2(\kappa+E+\sigma_Z^2)}+ \\
&\quad \rho\int\log\left[\rho\left(\frac{E+\sigma_Z^2}{\kappa+E+\sigma_Z^2}\right)^{J/2}+(1-\rho)\mathrm{e}^{-\frac{\kappa}{2(E+\sigma_Z^2)}\mathbf{g}^\top\mathbf{g}}\right]\mathscr{D}\mathbf{g}+ \\
&\quad (1-\rho)\int\log\left[\rho\left(\frac{E+\sigma_Z^2}{\kappa+E+\sigma_Z^2}\right)^{J/2}+(1-\rho)\mathrm{e}^{-\frac{\kappa}{2(\kappa+E+\sigma_Z^2)}\mathbf{h}^\top\mathbf{h}}\right]\mathscr{D}\mathbf{h}, \qquad (3.14)
\end{aligned}$$

where $\mathbf{h},\mathbf{x}_1$*, and* $\mathbf{g}$ *are J-dimensional super-symbols, and the differential* $\mathscr{D}\mathbf{h}=\prod_{j=1}^{J}\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-h_j^2/2}\,dh_j$*; the same rule applies to* $\mathscr{D}\mathbf{g}$.[3]

MMSE: The E that maximizes the free energy (3.14) *corresponds to* the MMSE [Krz12a; Krz12b; BK15]. After finding the E_0 that maximizes the free energy (3.14), we obtain the MMSE, $D_0=E_0$, in the large system limit.

Corollary 3.2. *The MMSE for MMV-1 and MMV-2 is the same for the same measurement rate κ, noise variance σ_Z^2, and number of signal vectors J.*

Remark 3.1. *As the reader can see from the proof of Lemma 3.1 in Section 3.4, the key reason that both MMV-1 and MMV-2 have an identical MMSE is that the entries in the super-symbols* $\mathbf{x}_l$ *and* $\widehat{\mathbf{x}}_l^{\{\cdot\}}$ *are i.i.d. That said, we suspect that the MMSE for MMV-1 and MMV-2 could differ by some higher order terms. If the entries of these super-symbols are not i.i.d., which is true in some practical MMV applications [ZS13], then it becomes more difficult to analyze the covariance matrix* $\mathbf{G}_\mu$ *as in Section 3.4. Therefore, we do not have an analysis for non-i.i.d. entries within* $\mathbf{x}_l$ *and* $\widehat{\mathbf{x}}_l^{\{\cdot\}}$*. However, we speculate that MMV-1 might have lower MMSE than MMV-2 in that case.*

Link to SMV with block sparse signal: The signal $\mathbf{x}$ in (3.4) is a block sparse signal comprised of N blocks of length J. We study an SMV problem by replacing the measurement matrix $\mathbf{A}$ in (3.4) with an i.i.d. Gaussian matrix $\widehat{\mathbf{A}}\in\mathbb{R}^{MJ\times NJ}$, i.e., $\mathbf{y}=\widehat{\mathbf{A}}\mathbf{x}+\mathbf{z}$. The entries of $\widehat{\mathbf{A}}$ follow the distribution, $\widehat{A}_{\mu l}\sim\mathcal{N}(0,\frac{1}{NJ})$. This SMV is similar to the setting in Barbier and Krzakala [BK15], except for the different priors and different ℓ_2 norms in each row of $\widehat{\mathbf{A}}$. We consider these differences while following

[3] The J-dimensional integrals in (3.14) can be simplified to one-dimensional integrals using a change of coordinates to J-sphere coordinates. Note also that E approaches the MSE in the large system limit; details appear in Appendix A.

their derivation [BK15], and obtain the same free energy expression as (3.14). We have also shown that MMV-1 and MMV-2 have the same MMSE in the large system limit. Hence, the three settings have the same free energy expression and their MMSE's are the same under the same noise variance σ_Z^2 and measurement rate κ in the large system limit.

3.3.2 Extension to complex SMV

The MMV model with jointly sparse signals is a versatile model that can be adapted to other problems. As an example, we show how the MMV model can be used to analyze the MMSE of a complex SMV. Consider the complex SMV, $\mathbf{y}^{\mathscr{C}} = \mathbf{A}^{\mathscr{C}}\mathbf{x}^{\mathscr{C}} + \mathbf{z}^{\mathscr{C}}$, where $\mathbf{x}^{\mathscr{C}} = \mathbf{x}^{\mathscr{R}} + i\mathbf{x}^{\mathscr{I}} \in \mathbb{C}^N$, $\mathbf{A}^{\mathscr{C}} = \mathbf{A}^{\mathscr{R}} + i\mathbf{A}^{\mathscr{I}} \in \mathbb{C}^{M\times N}$, $\mathbf{z}^{\mathscr{C}} = \mathbf{z}^{\mathscr{R}} + i\mathbf{z}^{\mathscr{I}} \in \mathbb{C}^M$, $\mathbf{y}^{\mathscr{C}} = \mathbf{y}^{\mathscr{R}} + i\mathbf{y}^{\mathscr{I}} \in \mathbb{C}^M$, $i = \sqrt{-1}$, and $\mathscr{R}$ and $\mathscr{I}$ refer to the real and imaginary parts, respectively. The real and imaginary parts of the entries of $\mathbf{z}^{\mathscr{C}}$ both follow a Gaussian distribution, $z_l^{\mathscr{R}}, z_l^{\mathscr{I}} \sim \mathcal{N}(0, \sigma_Z^2)$, $l \in \{1, \cdots, M\}$. Assume that the complex signal $\mathbf{x}^{\mathscr{C}}$ is comprised of two jointly sparse signals, $\mathbf{x}^{\mathscr{R}}$ and $\mathbf{x}^{\mathscr{I}}$, that satisfy the $J = 2$ dimensional Bernoulli-Gaussian distribution (3.1). We can extend the analysis of Section 3.3.1 to two settings of complex SMV: (*i*) the measurement matrix $\mathbf{A}^{\mathscr{C}}$ is real and (*ii*) $\mathbf{A}^{\mathscr{C}}$ is complex.[4]

Real measurement matrix: Suppose that $\mathbf{A}^{\mathscr{C}}$ is real, $\mathbf{A}^{\mathscr{C}} = \mathbf{A}^{\mathscr{R}} \in \mathbb{R}^{M\times N}$, and the entries of $\mathbf{A}^{\mathscr{R}}$ follow a Gaussian distribution, $A_{\mu l}^{\mathscr{R}} \sim \mathcal{N}(0, \frac{1}{N})$. Complex SMV with a real measurement matrix can be written as real-valued MMV,

$$\mathbf{y}^{\mathscr{R}} = \mathbf{A}^{\mathscr{R}}\mathbf{x}^{\mathscr{R}} + \mathbf{z}^{\mathscr{R}} \text{ and } \mathbf{y}^{\mathscr{I}} = \mathbf{A}^{\mathscr{R}}\mathbf{x}^{\mathscr{I}} + \mathbf{z}^{\mathscr{I}}, \tag{3.15}$$

where $\mathbf{x}^{\mathscr{R}}$ and $\mathbf{x}^{\mathscr{I}}$ are jointly sparse and follow (3.1). This formulation (3.15) fits into MMV-2 for $J = 2$. Hence, we can obtain the MMSE according to (3.14).[5]

Complex measurement matrix: Consider a complex $\mathbf{A}^{\mathscr{C}} = \mathbf{A}^{\mathscr{R}} + i\mathbf{A}^{\mathscr{I}} \in \mathbb{C}^{M\times N}$ with entries $A_{\mu l}^{\mathscr{R}}, A_{\mu l}^{\mathscr{I}} \sim \mathcal{N}(0, \frac{1}{2N})$. Expanding out the complex channel, $\mathbf{y}^{\mathscr{C}} = \mathbf{A}^{\mathscr{C}}\mathbf{x}^{\mathscr{C}} + \mathbf{z}^{\mathscr{C}}$, we obtain the equivalent real-valued SMV channel,

$$\begin{bmatrix}\mathbf{y}^{\mathscr{R}}\\ \mathbf{y}^{\mathscr{I}}\end{bmatrix} = \begin{bmatrix}\mathbf{A}^{\mathscr{R}} & -\mathbf{A}^{\mathscr{I}}\\ \mathbf{A}^{\mathscr{I}} & \mathbf{A}^{\mathscr{R}}\end{bmatrix}\begin{bmatrix}\mathbf{x}^{\mathscr{R}}\\ \mathbf{x}^{\mathscr{I}}\end{bmatrix} + \begin{bmatrix}\mathbf{z}^{\mathscr{R}}\\ \mathbf{z}^{\mathscr{I}}\end{bmatrix}. \tag{3.16}$$

[4] A replica analysis for complex SMV with a real measurement matrix appears in Guo and Verdú [GV05]. Their derivation does not cover complex matrices.

[5] As a reminder, the free energy of MMV-2 is identical to that of MMV-1 in the large system limit.

We rearrange (3.16) as follows,

$$\underbrace{\begin{bmatrix}\mathbf{y}^{\mathscr{R}}\\ \mathbf{y}^{\mathscr{I}}\end{bmatrix}}_{\bar{\mathbf{y}}}=\underbrace{\begin{bmatrix}\mathbf{A}^{\mathscr{R}}_{:,1},-\mathbf{A}^{\mathscr{I}}_{:,1},\cdots,\mathbf{A}^{\mathscr{R}}_{:,N},-\mathbf{A}^{\mathscr{I}}_{:,N}\\ \mathbf{A}^{\mathscr{I}}_{:,1},\ \mathbf{A}^{\mathscr{R}}_{:,1},\cdots,\mathbf{A}^{\mathscr{I}}_{:,N},\ \mathbf{A}^{\mathscr{R}}_{:,N}\end{bmatrix}}_{\bar{\mathbf{A}}}\underbrace{\begin{bmatrix}x^{\mathscr{R}}_1\\ x^{\mathscr{I}}_1\\ \vdots\\ x^{\mathscr{R}}_N\\ x^{\mathscr{I}}_N\end{bmatrix}}_{\bar{\mathbf{x}}}+\underbrace{\begin{bmatrix}\mathbf{z}^{\mathscr{R}}\\ \mathbf{z}^{\mathscr{I}}\end{bmatrix}}_{\bar{\mathbf{z}}},\tag{3.17}$$

where {:} refers to all the rows. In the rearranged channel (3.17), the measurement matrix $\bar{\mathbf{A}}$ consists of super-symbols,

$$\bar{\mathbf{A}}_{\mu l}=\begin{cases}\quad [A^{\mathscr{R}}_{\mu l},-A^{\mathscr{I}}_{\mu l}],\ \mu\in\{1,\cdots,M\}\\ \quad [A^{\mathscr{I}}_{\mu l},A^{\mathscr{R}}_{\mu l}],\ \mu\in\{M+1,\cdots,2M\}\end{cases},\tag{3.18}$$

and the signal $\bar{\mathbf{x}}$ consists of $\bar{\mathbf{x}}_l=\begin{bmatrix}x^{\mathscr{R}}_l\\ x^{\mathscr{I}}_l\end{bmatrix}$, $l\in\{1,\cdots,N\}$. The measurements and noise are $\bar{\mathbf{y}}=\begin{bmatrix}\mathbf{y}^{\mathscr{R}}\\ \mathbf{y}^{\mathscr{I}}\end{bmatrix}$ and $\bar{\mathbf{z}}=\begin{bmatrix}\mathbf{z}^{\mathscr{R}}\\ \mathbf{z}^{\mathscr{I}}\end{bmatrix}$, respectively. Hence, $\bar{y}_\mu=\sum_{l=1}^{N}\bar{\mathbf{A}}_{\mu l}\bar{\mathbf{x}}_l+\bar{z}_\mu$, $\mu\in\{1,\cdots,2M\}$.

Section 3.4 shows that the free energy and MMSE for complex SMV with complex measurement matrices are the same as MMV-1 with $J=2$. Note that in the free energy expression (3.14), the MSE, $D=E$ (A.8), is the average MSE of the J entries of $\mathbf{x}_l$. Therefore, in this complex SMV setting, D is the average MSE of the real and imaginary parts of the signal entries.

3.4 Proof of Lemma 3.1

In this section, we show that the quantity $\mathbb{X}_\mu$ (3.11) is the same for MMV-1 and MMV-2. Moreover, we show that complex SMV with a complex measurement matrix also yields the same $\mathbb{X}_\mu$ with $J=2$.

First, we rewrite (3.11) in the vector form

$$\mathbb{X}_\mu=\mathbb{E}_{\mathbf{v}_\mu}\left[\mathrm{e}^{-\frac{1}{2\sigma_Z^2}\sum_{j=1}^{J}\sum_{a=1}^{n}(v^a_{\mu j})^2}\right]=\mathbb{E}_{\mathbf{v}_\mu}\left[\mathrm{e}^{-\frac{1}{2\sigma_Z^2}\mathbf{v}_\mu^\top\mathbf{v}_\mu}\right],\tag{3.19}$$

where $\mathbf{v}_\mu=[v^1_{\mu 1},\cdots,v^a_{\mu 1},\cdots,v^1_{\mu J},\cdots,v^n_{\mu J}]^\top$ and $v^a_{\mu j}$ is given in (3.12). In order to calculate the expectation w.r.t. $\mathbf{v}_\mu$ in (3.19), we calculate the distribution of $\mathbf{v}_\mu$, which is approximated by a Gaussian distribution, due to the central limit theorem. The mean is $\mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j}]=0$.

We now calculate the covariance matrix, $\mathbf{G}_\mu=\mathbb{E}[\mathbf{v}_\mu\mathbf{v}_\mu^\top]$. The matrix $\mathbf{G}_\mu$ is separated into $J\times J$ blocks of size $n\times n$, as shown in Figure 3.2. The main diagonal of $\mathbf{G}_\mu$ consists of entries $w_1=\mathbb{E}_{\mathbf{A},\mathbf{z}}[(v^a_{\mu j})^2]$. The entries in the blocks along the main diagonal (other than entries along the main diagonal itself) are $w_3=\mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j}v^b_{\mu j}]$. The main diagonals of other blocks have entries $w_2=\mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j}v^a_{\mu\eta}]$,

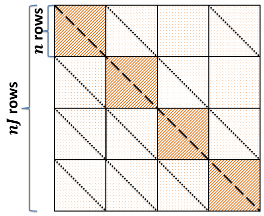


Figure 3.2 Covariance matrix $\mathbf{G}_\mu \in \mathbb{R}^{nJ\times nJ}$. Each block in $\mathbf{G}_\mu$ has a size of $n \times n$. The entries in the heavily marked blocks take the value w_3, except that entries along the dashed diagonal are w_1. The entries in the lightly marked blocks take the value w_4, except that entries along the dotted diagonals are w_2.

and other entries in these blocks are $w_4 = \mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j} v^b_{\mu\eta}]$. We now calculate each of these values as follows for MMV-1, MMV-2, and complex SMV with a complex measurement matrix.

MMV-1: We begin by calculating the diagonal entries of the covariance matrix $\mathbf{G}_\mu = \mathbb{E}[\mathbf{v}_\mu \mathbf{v}_\mu^\top]$,

$$w_1 = \mathbb{E}_{\mathbf{A},\mathbf{z}}\left[(v^a_{\mu j})^2\right] = \sum_{l,k=1}^{N,N}\left\{(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l)^\top \mathbb{E}_{\mathbf{A}}\left[\mathbf{A}^\top_{\mu+M(j-1),l}\mathbf{A}_{\mu+M(j-1),k}\right](\mathbf{x}_k - \widehat{\mathbf{x}}^a_k)\right\} + \sigma_Z^2. \tag{3.20}$$

In (3.20), $\mathbb{E}_{\mathbf{A}}\left[\mathbf{A}^\top_{\mu+M(j-1),l}\mathbf{A}_{\mu+M(j-1),k}\right] = \frac{\delta_{k,l}}{N}\widetilde{\mathbf{I}}_J$ (cf. Figure 3.1), where $\widetilde{\mathbf{I}}_J$ is a $J \times J$ matrix with only one 1 located at the j-th row, j-th column, and $\delta_{k,l} = 1$ when $k = l$, else zero. Hence, (3.20) becomes

$$\begin{aligned} w_1 &= \mathbb{E}_{\mathbf{A},\mathbf{z}}\left[(v^a_{\mu j})^2\right] = \frac{1}{N}\sum_{l=1}^{N}(x_{l,j} - \widehat{x}^a_{l,j})^2 + \sigma_Z^2 && (3.21)\\ &= \frac{1}{NJ}\sum_{l=1}^{N}(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l)^\top(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l) + \sigma_Z^2, && (3.22)\end{aligned}$$

where $x_{l,j}$ and $\widehat{x}^a_{l,j}$ (3.21) denote the j-th entries in super-symbols $\mathbf{x}_l$ and $\widehat{\mathbf{x}}^a_l$, respectively, and (3.22) holds because all J entries within the same super-symbol ($\mathbf{x}_l$ or $\widehat{\mathbf{x}}^a_l$) are i.i.d.

Similarly, we obtain

$$\begin{aligned} w_2 &= \mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j} v^a_{\mu\eta}] = \frac{1}{N}\sum_{l=1}^{N}(x_{l,j} - \widehat{x}^a_{l,j})(x_{l,\eta} - \widehat{x}^a_{l,\eta})\\ &= \frac{1}{NJ}\sum_{l=1}^{N}(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l)^\top(\mathbf{x}^a_l - \widehat{\mathbf{x}}^b_l),\end{aligned} \tag{3.23}$$

where entries of $\mathbf{x}^{\{\cdot\}}_l$ and $\widehat{\mathbf{x}}^{\{\cdot\}}_l$ follow the same distribution as entries of $\mathbf{x}_l$ given l, and (3.23) is due to (*i*) entries of $\mathbf{x}_l$ being i.i.d., (*ii*) entries of $\widehat{\mathbf{x}}^{\{\cdot\}}_l$ being i.i.d. for fixed l, and (*iii*) the replica symmetry assumption [Krz12a; Krz12b]. We also obtain

$$\begin{aligned} w_3 &= \mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j} v^b_{\mu j}] = \frac{1}{NJ}\sum_{l=1}^{N}(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l)^\top(\mathbf{x}_l - \widehat{\mathbf{x}}^b_l) + \sigma_Z^2, \\ w_4 &= \mathbb{E}_{\mathbf{A},\mathbf{z}}[v^a_{\mu j} v^b_{\mu \eta}] = \frac{1}{NJ}\sum_{l=1}^{N}(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l)^\top(\mathbf{x}^a_l - \widehat{\mathbf{x}}^b_l). \end{aligned} \tag{3.24}$$

We now define the following auxiliary parameters

$$m_a = \frac{\sum_{l=1}^{N}(\widehat{\mathbf{x}}^a_l)^\top \mathbf{x}_l}{NJ}, \quad Q_a = \frac{\sum_{l=1}^{N}(\widehat{\mathbf{x}}^a_l)^\top \widehat{\mathbf{x}}^a_l}{NJ}, \quad q_{ab} = \frac{\sum_{l=1}^{N}(\widehat{\mathbf{x}}^a_l)^\top \widehat{\mathbf{x}}^b_l}{NJ}, \quad q_0 = \frac{1}{NJ}\sum_{l=1}^{N}(\mathbf{x}^a_l)^\top \mathbf{x}_l, \tag{3.25}$$

which allow us to express (3.22)–(3.24) as

$$\begin{aligned} w_1 &= \rho - 2m_a + Q_a + \sigma_Z^2, \\ w_2 &= q_0 - (m_a + m_b) + q_{ab}, \qquad (3.26) \\ w_3 &= \rho - (m_a + m_b) + q_{ab} + \sigma_Z^2, \\ w_4 &= q_0 - (m_a + m_b) + q_{ab}. \qquad (3.27) \end{aligned}$$

Up to this point, we have obtained the entries of $\mathbf{G}_\mu$. Plugging the distribution of $\mathbf{v}_\mu$, approximated by $f(\mathbf{v}_\mu) = [(2\pi)^n \det(\mathbf{G}_\mu)]^{-\frac{1}{2}} \exp(-\frac{1}{2}\mathbf{v}_\mu^\top \mathbf{G}_\mu^{-1}\mathbf{v}_\mu)$, into (3.19), we obtain

$$\mathbb{X}_\mu = \left[\det\left(\mathbb{I}_n + \frac{1}{\sigma_Z^2}\mathbf{G}_\mu\right)\right]^{-1/2}, \tag{3.28}$$

where $\mathbb{I}_n$ denotes an identity matrix of size $n \times n$ and $\det(\cdot)$ is the determinant of a matrix.

MMV-2: For the matrix **A** (3.4) in MMV-2, rows $jM+1, \cdots, (j+1)M$, $2 \le j \le J$, will be the right-shift of rows $(j-1)M+1, \cdots, jM$. We express $v^a_{\mu j}$ (3.12) as

$$v^a_{\mu j} = \sum_{l=1}^{N}\mathbf{A}_{\mu l}\mathbf{T}_j(\mathbf{x}_l - \widehat{\mathbf{x}}^a_l) + z_{\mu + M(j-1)}, \; \mu \in \{1, \cdots, M\}, \tag{3.29}$$

where $\mathbf{T}_j$ is a $J \times J$ transform matrix with the j-th entry of the first row being one and all other entries in $\mathbf{T}_j$ being zeros. Using the same derivations as in MMV-1, it can be proved that the covariance matrix $\mathbf{G}_\mu = \mathbb{E}[\mathbf{v}_\mu \mathbf{v}_\mu^\top]$ in MMV-2 is identical to that of MMV-1. Therefore, $\mathbb{X}_\mu$ in MMV-1 and MMV-2 are identical in the large system limit.

Complex SMV with complex measurement matrix: The derivations are the same as in MMV-2 above, except that we need to change $\mathbf{A}_{\mu l}$ in (3.29) to $\overline{\mathbf{A}}_{\mu l}$ (3.18) and replace $\mathbf{T}_j$ by

$$\mathbf{T} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix},$$

because $\overline{\mathbf{A}}_{(\mu+M)l} = \overline{\mathbf{A}}_{\mu l}\mathbf{T}$, $\mu \in \{1, \cdots, M\}$. Using similar steps as above, we obtain that the covariance matrix $\mathbf{G}_\mu$ in this case is also the same as that of MMV-1 with $J = 2$.

Solving $\mathbb{X}_\mu$: For such a structured matrix $\mathbf{G}_\mu$ (Figure 3.2), elementary transforms show that the eigenvalues (EV's) are comprised of one EV equal to $\alpha_1 = [w_1+(J-1)w_2]+(n-1)[w_3+(J-1)w_4]$, $(J-1)$ EV's equal to $\alpha_2 = (w_1-w_2)+(n-1)(w_3-w_4)$, $(n-1)$ EV's equal to $\alpha_3 = [w_1+(J-1)w_2]-[w_3+(J-1)w_4]$, and $(J-1)(n-1)$ EV's equal to $\alpha_4 = (w_1-w_2)-(w_3-w_4)$.

Owing to replica symmetry [Krz12a; Krz12b], we have $m_a = m_b = m$, $Q_a = Q$, and $q_{ab} = q$, cf. (3.25). Also, in the Bayesian setting, we have $m = q_0 = q$ and $Q = \rho$. Thus, $w_2 = w_4 = 0$ ((3.26) and (3.27)), and

$$\begin{aligned} \det\left(\mathbb{I}_{nJ} + \frac{1}{\sigma_Z^2}\mathbf{G}_\mu\right) &= \left(1+\frac{\alpha_1}{\sigma_Z^2}\right)\left(1+\frac{\alpha_2}{\sigma_Z^2}\right)^{J-1}\left(1+\frac{\alpha_1}{\sigma_Z^2}\right)^{n-1}\left(1+\frac{\alpha_1}{\sigma_Z^2}\right)^{(n-1)(J-1)} \\ &= \left(1+n\frac{w_3}{\sigma_Z^2+\alpha_4}\right)^{J}\left(1+\frac{1}{\sigma_Z^2}\alpha_4\right)^{Jn}. \end{aligned} \tag{3.30}$$

Considering (3.30), we simplify (3.28),

$$\lim_{n\to 0}\mathbb{X}_\mu = \mathrm{e}^{-\frac{nJ}{2}\left[\frac{\rho-2m+\sigma_Z^2+q}{Q-q+\sigma_Z^2}+\log(Q-q+\sigma_Z^2)-\log(\sigma_Z^2)\right]}, \tag{3.31}$$

where we rely on the following Taylor series,

$$\mathrm{e}^{nk} \approx 1+nk \Rightarrow \mathrm{e}^{-\frac{n}{2}k} \approx (1+nk)^{-1/2},\ n \to 0.$$

3.5 Numerical Results

Given a free energy expression for an MMV problem, the MMSE can be obtained by evaluating the largest free energy [Tan02; GV05; Krz12a; Krz12b; MM09; BK15]. Having derived the free energy for the two MMV settings in Section 3.3, this section calculates the MMSE under various cases. Different performance regions of MMV are identified, where the MMSE behaves differently as a function of the noise variance σ_Z^2 and measurement rate κ. We identify a phase transition of belief propagation (BP) that separates regions where BP is optimal asymptotically or not. Simulation results match the performance predicted for BP.

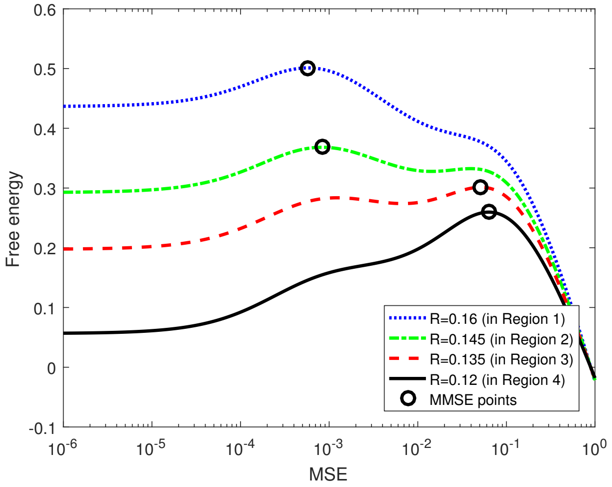


Figure 3.3 Free energy as a function of the MSE for different measurement rates κ (number of jointly sparse signal vectors $J = 3$ and noise variance $\sigma_Z^2 = -35$ dB). The black circles mark the largest free energy, and so they correspond to the MMSE.

3.5.1 Performance regions: Definitions and numerical results

When calculating the MMSE (A.8) for different settings from the free energy expression (3.14), four different *performance regions* will appear, as discussed below; the free energy as a function of the MSE is shown in Figure 3.3 for different performance regions.

Regions 1 and 4: The free energy (3.14) has one local maximum point w.r.t. the MSE D (A.8). This D leads to the globally maximum free energy and is the MMSE.

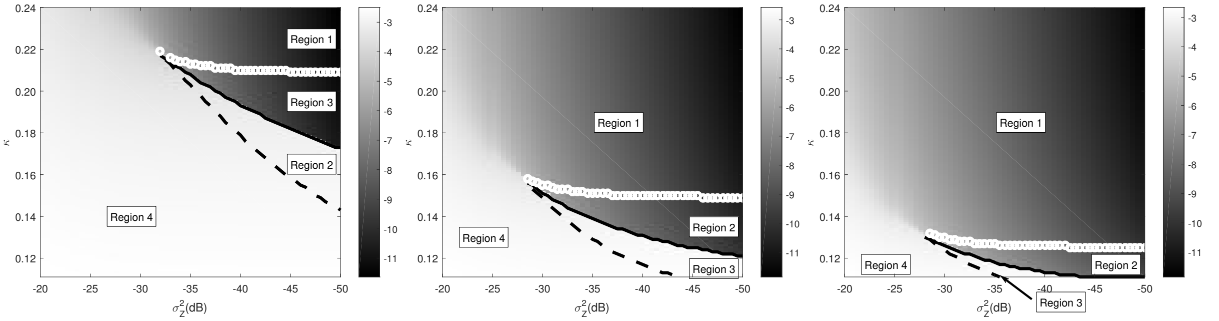
Regions 2 and 3: There are 2 local maxima in the free energy, D_1 and D_2, where $D_1 < D_2$. In Region 2, the smaller MSE, D_1, leads to the larger local maximum free energy (3.14) (hence, $\mathscr{F}(D_1)$ is the global maximum), and is the MMSE. In Region 3, the larger MSE, D_2, is the MMSE.

Boundaries between regions: We denote the boundary separating regions 1 and 2 by the *BP threshold* $\kappa_{BP}(\sigma_Z^2)$, the boundary separating regions 2 and 3 by the *low noise threshold* $\kappa_l(\sigma_Z^2)$, and the boundary separating regions 3 and 4 by the *critical threshold* $\kappa_c(\sigma_Z^2)$.

Numerical results: Consider J-dimensional Bernoulli-Gaussian signals (3.1) with sparsity rate $\rho = 0.1$. Evaluating the free energy (3.14) with the noise variance σ_Z^2 from -20 dB to -50 dB and measurement rate κ from 0.11 to 0.24, we obtain the MMSE as a function of σ_Z^2 and κ for $J = 1, 3$, and 5, as shown in Figure 3.4.[6] The darkness of the shades represents the natural logarithm of the MMSE, ln(MMSE). In all panels, the critical threshold $\kappa_c(\sigma_Z^2)$, low noise threshold $\kappa_l(\sigma_Z^2)$, and BP threshold $\kappa_{BP}(\sigma_Z^2)$, as well as Regions 1-4, are marked.

In Regions 3 and 4, the best-possible algorithm yields a large MMSE for all noise variances. In

[6] The MMV with $J = 1$ becomes an SMV. The MMSE results in Figure 3.4a match with the SMV MMSE in Krzakala et. al. [Krz12a; Krz12b] and Zhu and Baron [ZB13].



(a) $J = 1$ **(b)** $J = 3$ **(c)** $J = 5$

Figure 3.4 Performance regions for MMV with different J. The darkness of the shades corresponds to ln(MMSE) for a certain noise variance σ_Z^2 and measurement rate κ. There are 4 regions, Regions 1 to 4, where the MMSE as a function of the noise variance σ_Z^2 and measurement rate κ behaves differently. Regions 1 to 4 are separated by 3 thresholds, $\kappa_c(\sigma_Z^2)$ (the dashed curves), $\kappa_l(\sigma_Z^2)$ (the solid curves), and $\kappa_{BP}(\sigma_Z^2)$ (the curves comprised of little white circles); note that Section 3.5.1 discusses how to obtain these thresholds. (a) MMV with $J = 1$, (b) MMV with $J = 3$, and (c) MMV with $J = 5$.

contrast, in Regions 1 and 2, the optimal algorithm yields an MMSE that decreases with the noise variance σ_Z^2. To summarize, the optimal algorithm yields poor estimation performance below the low noise threshold $\kappa_l(\sigma_Z^2)$, and good performance above $\kappa_l(\sigma_Z^2)$.

We further examine the MMSE as a function of the number of jointly sparse signal vectors J and the measurement rate κ. We plot the MMSE in dB scale in Figure 3.5. The noise variance is -35 dB. We can see that the MMSE decreases with more signal vectors J and greater measurement rate κ. However, the MMSE depends less on J as J is increased. Note that the discontinuity in the MMSE surface in Figure 3.5 is a result of the different performance regions that the various settings (different J and κ) lie in.

3.5.2 BP phase transition

Belief propagation (BP) [Don09; Bar10; Mon12; BM11; Krz12a; Krz12b; BK15] is an algorithmic framework invented independently by researchers in coding theory, statistical physics, and artificial intelligence, which can often achieve the optimal estimation performance (MMSE) for linear inverse problems. The canonical BP updating rules appeared in (2.11). When there are multiple local maxima $D_1 < D_2$ in the free energy (3.14), BP converges to the local maximum with the larger MSE, D_2 [Don09; Mon12; BM11; Krz12a; Krz12b]. Hence, D_2 characterizes the MSE *predicted* for BP. Moving from Region 1 to Region 2 by decreasing the measurement rate κ with fixed noise variance σ_Z^2, the number of local maxima increases from 1 to 2. Therefore, BP estimation performance experiences a sudden deterioration (increase in MSE) when the measurement rate κ drops such that the combination of the noise variance σ_Z^2 and measurement rate κ moves from Region 1 to Region 2. The BP threshold,

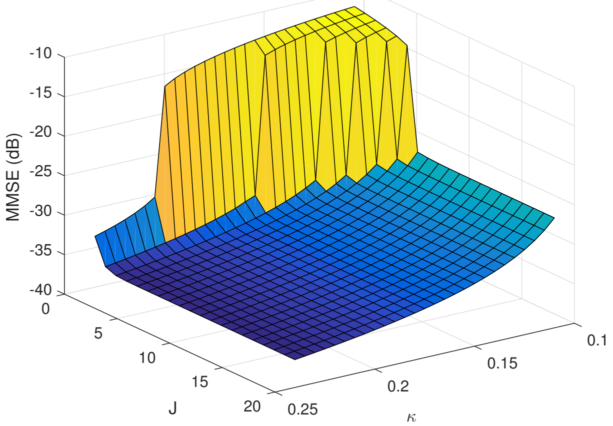


Figure 3.5 MMSE in dB as a function of the number of jointly sparse signal vectors J and the measurement rate κ (noise variance $\sigma_Z^2 = -35$ dB).

$\kappa_{BP}(\sigma_Z^2)$, is the boundary between Regions 1 and 2, and is where the BP phase transition happens. That is, BP achieves poor estimation performance below $\kappa_{BP}(\sigma_Z^2)$, and good performance above $\kappa_{BP}(\sigma_Z^2)$.

Remark 3.2. *In Figure 3.4, we see that increasing J reduces the BP threshold $\kappa_{BP}(\sigma_Z^2)$. Since BP achieves the MMSE when $\kappa > \kappa_{BP}(\sigma_Z^2)$, increasing J is beneficial to applications that use BP as the estimation algorithm.*

Remark 3.3. *We further numerically analyzed the low noise ($\sigma_Z^2 \to 0$) and zero noise ($\sigma_Z^2 = 0$) cases. The low noise threshold $\kappa_l(\sigma_Z^2)$ converges to ρ as the noise variance σ_Z^2 is decreased for $J = 1, 3$, and 5. We believe that this numerical result holds for every J. Moreover, this result matches the theoretical robust threshold of Wu and Verdú [WV12b] for $J = 1$ in the low noise limit. Our numerical results also show that the BP threshold $\kappa_{BP}(\sigma_Z^2)$ converges to some value for different J as $\sigma_Z^2 \to 0$. Analyzing these observations rigorously is left for future work.*

3.5.3 BP simulation

After obtaining the theoretic MMSE for MMV, as well as the MSE predicted for BP, we run some simulations to estimate the $\underline{\mathbf{x}}^{(j)}$ of channel (3.2) in a Bayesian setting. The algorithm we use is approximate message passing (AMP) [Don09; Mon12; BM11; Krz12a; Krz12b; BK15], which is an approximation to the BP algorithm; related algorithms have been proposed by Ziniel and Schniter [ZS13] and Kim et al. [Kim11]. In the SMV case, when the measurement matrix and the signal have i.i.d. entries, AMP has the state evolution (SE) formalism [Don11; BM11; JM12; Don13; Bay15] that tracks the evolution of the MSE at each iteration. Recently, Javanmard and Montanari proved that SE tracks AMP rigorously

Algorithm 3.1 AMP for MMV

```
**Inputs:** Maximum number of iterations $T$, threshold $\epsilon$, sparsity rate $\rho$, noise variance $\sigma_Z^2$, measurements $\mathbf{y}^{(j)}$, and measurement matrices $\mathbf{A}^{(j)}, \forall j$
**Initialize:** $t = 1, \delta = \infty, \mathbf{w}^{(j)} = \mathbf{y}^{(j)}, \Theta_j = 0, v_l^{(j)} = \rho\sigma_Z^2, a_l^{(j)} = 0, \forall l, j$
**while** $t < T$ and $\delta > \epsilon$ **do**
    **for** $j \leftarrow 1$ to $J$ **do**
        $\mathbf{q}^{(j)} = \frac{\mathbf{y}^{(j)} - \mathbf{w}^{(j)}}{\sigma_Z^2 + \Theta_j}$
        $\Theta_j = \frac{1}{N}\sum_{l=1}^{N} v_l^{(j)}$
        $\mathbf{w}^j = \mathbf{A}^{(j)}\mathbf{a}^{(j)} - \Theta_j \mathbf{q}^{(j)}$
        $\Sigma_j = \frac{N(\sigma_Z^2 + \Theta_j)}{M}$    ▷ Scalar channel noise variance
        $\mathbf{R}^{(j)} = \mathbf{a}^{(j)} + \Sigma_j \left(\mathbf{A}^{(j)}\right)^\top \frac{\mathbf{y}^{(j)} - \mathbf{w}^{(j)}}{\sigma_Z^2 + \Theta_j}$    ▷ Pseudodata
        $\widehat{\mathbf{a}}^{(j)} = \mathbf{a}^{(j)}$    ▷ Save current estimate
    **end for**
    **for** $l \leftarrow 1$ to $N$ **do**
        $\left\{v_l^{(j)}\right\}_{j=1}^{J} = f_{v_l}\left(\{\Sigma_j\}_{j=1}^{J}, \left\{R_l^{(j)}\right\}_{j=1}^{J}\right)$    ▷ Variance
        $\left\{a_l^{(j)}\right\}_{j=1}^{J} = f_{a_l}\left(\{\Sigma_j\}_{j=1}^{J}, \left\{R_l^{(j)}\right\}_{j=1}^{J}\right)$    ▷ Estimate
    **end for**
    $t = t + 1$    ▷ Increment iteration index.
    $\delta = \frac{1}{NJ}\sum_{l=1}^{N}\sum_{j=1}^{J}\left(\widehat{a}_l^{(j)} - a_l^{(j)}\right)^2$    ▷ Change in estimate
**end while**
**Outputs:** Estimate $\mathbf{a}^{(j)}, \forall j$
```

in an SMV setting with a spatially coupled measurement matrix [JM12]. According to our transform in Figure 3.1, we can see that the proof [JM12] could be extended to the MMV setting. Note that SE allows to compute the highest equilibrium of Gibbs free energy [Don11; BM11; JM12; Don13; Bay15], which corresponds to the local optimum D_2 in Section 3.5.2. Hence, AMP often achieves the same MSE as BP and we use AMP simulation results to demonstrate that the MMSE can often be achieved.[7] Considering (3.2), we simplify the AMP algorithm in Barbier and Krzakala [BK15] to obtain Algorithm 3.1,[8] where $\{\Sigma_j\}_{j=1}^J$, $\left\{R_l^{(j)}\right\}_{j=1}^J$, $\left\{a_l^{(j)}\right\}_{j=1}^J$ and $\left\{v_l^{(j)}\right\}_{j=1}^J$ refer to sets of all intermediate variables Σ_j, pseudodata $R_l^{(j)}$, estimates $a_l^{(j)}$, and variances $v_l^{(j)},\ j \in \{1, \cdots, J\},\ l \in \{1, \cdots, N\}$, respectively. The current iteration t, change in the estimate δ, and intermediate variables $\Theta_j,\ j \in \{1, \cdots, J\}$, are scalars. The intermediate variables $\mathbf{q}^{(j)}$ and $\mathbf{w}^{(j)}$ are vectors of length M. The functions $f_{a_l}\left(\{\Sigma_j\}_{j=1}^J, \left\{R_l^{(j)}\right\}_{j=1}^J\right)$

[7] When the assumptions about the measurement matrix and signal [Don09; Mon12; BM11; Krz12a; Krz12b; BK15] are violated, AMP might suffer from divergence issues.

[8] Note that Algorithm 3.1 is a straightforward simplification of the AMP algorithm by Barbier and Krzakala [BK15].

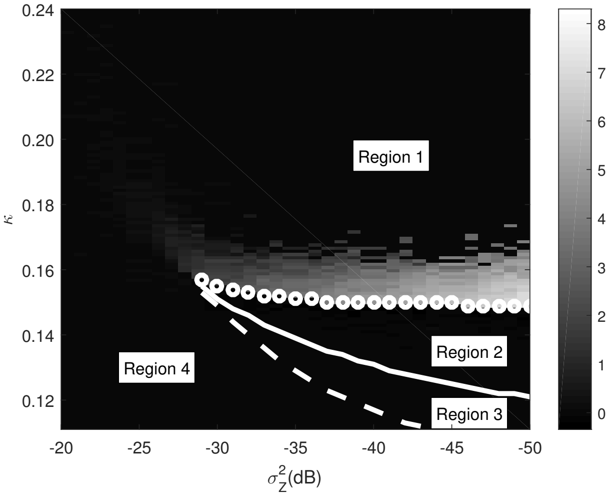


Figure 3.6 AMP simulation results (MSE$_{\text{AMP}}$) compared to the MSE predicted for BP (MSE$_{\text{BP}}$) with $J = 3$ jointly sparse signal vectors. The dashed curve, solid curve, and the curve comprised of little circles correspond to thresholds $\kappa_c(\sigma_Z^2)$, $\kappa_l(\sigma_Z^2)$, and $\kappa_{BP}(\sigma_Z^2)$, respectively. Regions 1-4 are also marked. The darkness of the shades denotes $\ln\left(\frac{\text{MSE}_{\text{AMP}}}{\text{MSE}_{\text{BP}}}\right)$, which we expect to be zero (completely dark shades) in the entire κ versus σ_Z^2 plane. The narrow bright band above the BP threshold indicates the mismatch between the MSE from the simulation and the MSE predicted for BP.

and $f_{v_l}\left(\{\Sigma_j\}_{j=1}^J, \left\{R_l^{(j)}\right\}_{j=1}^J\right)$ are given by

$$f_{a_l}\left(\{\Sigma_j\}_{j=1}^J, \left\{R_l^{(j)}\right\}_{j=1}^J\right) = \frac{\rho\frac{1}{\Sigma_j+1}\left\{R_l^{(j)}\right\}_{j=1}^J}{\rho+(1-\rho)\prod_{j=1}^J\left\{\sqrt{1+\frac{1}{\Sigma_j}}\exp\left[-\frac{\left(R_l^{(j)}\right)^2}{2\Sigma_j(\Sigma_j+1)}\right]\right\}},$$

$$f_{v_l}\left(\{\Sigma_j\}_{j=1}^J, \left\{R_l^{(j)}\right\}_{j=1}^J\right) = -\left[f_{a_l}\left(\{\Sigma_j\}_{j=1}^J, \left\{R_l^{(j)}\right\}_{j=1}^J\right)\right]^2 + \frac{\rho\frac{1}{\Sigma_j+1}\left[\left(\left\{R_l^{(j)}\right\}_{j=1}^J\right)^2\frac{1}{\Sigma_j+1}+\Sigma_j\right]}{\rho+(1-\rho)\prod_{j=1}^J\left\{\sqrt{1+\frac{1}{\Sigma_j}}\exp\left[-\frac{\left(R_l^{(j)}\right)^2}{2\Sigma_j(\Sigma_j+1)}\right]\right\}},$$

for J-dimensional Bernoulli-Gaussian signals (3.1).

We simulated the signals in (3.1) with $J = 3$ signal vectors and sparsity rate $\rho = 0.1$ measured by a channel (3.2) with measurement rate $\kappa \in [0.11, 0.24]$ and noise variance $\sigma_Z^2 \in [-20, -50]$ dB. For each setting, we generated 50 signals of length $N = 5000$, and the resulting MSE compared to the MSE predicted for BP is shown in Figure 3.6.[9] The labels of the thresholds are omitted for brevity. We can see that AMP simulation results match with the MSE predicted for BP and BP phase transition from the replica analysis of Section 3.5.2. Note that there is a narrow band of light shades above the BP

[9] We simulated both J different measurement matrices $\underline{\mathbf{A}}^{(j)}$ and J identical $\underline{\mathbf{A}}^{(j)}$. Both results match the MSE predicted for BP, which support our conclusion that the MMSE's of both settings are the same. Figure 3.6 is with J different $\underline{\mathbf{A}}^{(j)}$.

threshold, $\kappa_{BP}(\sigma_Z^2)$ (the top threshold), meaning that the MSE from the simulation is greater than the MSE predicted for BP; this is due to randomness in our generated signals and channels. Note that we also compared the AMP simulation results to that of the M-SBL algorithm [Ye15], a widely used algorithm to solve the MMV problem. The M-SBL results were not as good. Indeed, because AMP is often an approach that achieves the MMSE, other algorithms are expected to provide greater MSE.

3.6 Extension to Arbitrary Error Metrics

In this chapter, we have obtained the MMSE for MMV problems. As mentioned in Section 3.1, there are many estimation approaches for MMV problems [Tro06b; CH06; Mal05; Tro06a; Cot05; ME09; Lee12; Ye15; ZS11]. However, when running estimation algorithms for MMV problems, people might be interested in obtaining an estimate whose "user-defined" error is as small as possible. For example, if estimating the underlying signal is important, people may use the MSE metric; when there might be outliers in the estimate, using the mean absolute error metric might be more appropriate. For applications such as compressive diffuse optical tomography [Lee11], estimating the support set of the jointly sparse underlying signals is of more interest. Seeing that there are different algorithms minimizing different error metrics, but there is no prior work discussing the optimal performance with user-defined (arbitrary) error metrics in MMV, it is of interest to study the optimal performance with user-defined error metrics in MMV problems and also design algorithms to achieve such optimal performance.

Tan and coauthors [Tan14a; Tan14b] studied the optimal performance for arbitrary additive error metrics for an SMV problem (1.1) by taking advantage of the properties of BP [Don09; Bar10; BM11; Mon12; Krz12a; Krz12b; BK15]: BP yields an equivalent scalar channel

$$\widetilde{\mathbf{y}} = \mathbf{x} + \widetilde{\mathbf{z}}, \tag{3.32}$$

whose posterior $f(\mathbf{x}|\widetilde{\mathbf{y}})$ approaches the true posterior distribution $f(\mathbf{x}|\mathbf{y})$ under certain conditions [Ran11]. Using $f(\mathbf{x}|\widetilde{\mathbf{y}})$, Tan and coauthors designed the denoiser that minimizes the (additive) user-defined error metrics for (3.32).

According to Section 3.2 and Figure 3.1, we can transform the MMV problem (3.2) into an SMV problem (3.4). Hence, we can extend the work of Tan and coauthors [Tan14a; Tan14b] to study the optimal performance for arbitrary additive error metrics, as well as to build algorithms that achieve the optimal performance for MMV (3.2). The details are left for future work.

3.7 Conclusion

We analyzed the minimum mean squared error (MMSE) for two settings of multi-measurement vector (MMV) problems, where the entries in the signal vectors are independent and identically distributed (i.i.d.), and share the same support. One MMV setting has i.i.d. Gaussian measurement matrices, while the other MMV setting has identical i.i.d. Gaussian measurement matrices. Replica analysis yields identical free energy expressions for these two settings in the large system limit when the signal length goes to infinity and the number of measurements scales with the signal length. Because of the identical free energy expressions, the MMSE's for both MMV settings are identical. By numerically evaluating the free energy expression, we identified different performance regions for MMV where the MMSE as a function of the channel noise variance and the measurement rate behaves differently. We also identified a phase transition for belief propagation algorithms (BP) that separates regions where BP achieves the MMSE asymptotically and where it is sub-optimal. Simulation results of an approximated version of BP matched with the mean squared error (MSE) predicted by replica analysis. As a special case of MMV, we extended our replica analysis to complex single measurement vector (SMV) problems, so that we can calculate the MMSE for complex SMV with real or complex measurement matrices. Seeing that the MSE might not be the only error metric that is of interest, we proposed to extend the work of Tan and coauthors [Tan14a; Tan14b] to MMV problems, so that we can optimize over different user-defined additive error metrics in MMV applications.

CHAPTER

4

PERFORMANCE TRADE-OFFS IN MULTI-PROCESSOR APPROXIMATE MESSAGE PASSING

In Chapter 3, we focused on analyzing the information theoretic performance limits for multi-measurement vector problems (1.3). Our analysis is readily extended to single measurement vector problems (1.1). In practice, many algorithms run in distributed networks, especially as we are entering the "big data" era. Running estimation algorithms across distributed networks can incur different costs besides the quality of the estimation. Some prior art has focused on reducing certain costs such as the communication cost [Han14] and the computation cost [Ma14c], but there has been less progress relating different costs and achieving optimal trade-offs among them. Despite the lack of such works, these trade-offs are important to system designers in order to produce efficient systems. Studying the relation between different costs is a broad problem with a rich design space. Therefore, in this chapter, we focus our discussion on one specific distributed algorithm as an example: the "multi-processor approximate message passing" algorithm (MP-AMP) [Han14; Han16], and study the optimal trade-offs among different costs. In each MP-AMP iteration, nodes of the multi-processor system and its fusion center exchange lossily compressed messages pertaining to their estimates of the input. In this setup, we derive the optimal per-iteration coding rates using dynamic programming. We analyze the excess mean squared error (EMSE) beyond the minimum mean squared error, and prove that, in the limit of low EMSE, the optimal coding rates increase approximately linearly per iteration. Additionally, we obtain that the combined cost of computation and communication scales with the desired estimation quality according to $O(\log^2(1/\text{EMSE}))$. Finally, we study trade-offs between the physical costs of the estimation process including computation time, communication loads, and the estimation quality as a multi-objective optimization problem, and characterize the properties of the Pareto optimal surfaces. This chapter is based on our work with Han et al. [Han16] and with Baron and Beirami [Zhu16c; Zhu16a].

4.1 Related Work and Contributions

4.1.1 Related work

Many scientific and engineering problems [Don06a; Can06] can be approximated using a linear model,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \tag{4.1}$$

where $\mathbf{x} \in \mathbb{R}^N$ is the unknown input signal, $\mathbf{A} \in \mathbb{R}^{M\times N}$ is the matrix that characterizes the linear model, and $\mathbf{z} \in \mathbb{R}^M$ is measurement noise. The goal is to estimate $\mathbf{x}$ from the noisy measurements $\mathbf{y}$ given $\mathbf{A}$ and statistical information about $\mathbf{z}$; this is a *linear inverse problem*. Alternately, one could view the estimation of $\mathbf{x}$ as fitting or learning a linear model for the data comprised of $\mathbf{y}$ and $\mathbf{A}$.

When $M \ll N$, the setup (4.1) is known as compressed sensing (CS) [Don06a; Can06]; by posing a sparsity or compressibility requirement on the signal, it is indeed possible to accurately recover $\mathbf{x}$ from the ill-posed linear model [Don06a; Can06] when the number of measurements M is large enough, and the noise level is modest. However, we might need $M > N$ when the signal is dense or the noise is substantial. Hence, we do not constrain ourselves to the case of $M \ll N$.

Approximate message passing (AMP) [Don09; Mon12; BM11; RV16] is an iterative framework that solves linear inverse problems by successively decoupling [Tan02; GV05; GW08] the problem in (4.1) into scalar denoising problems with additive white Gaussian noise (AWGN). AMP has received considerable attention, because of its fast convergence and the state evolution (SE) formalism [Don09; BM11; RV16], which offers a precise characterization of the AWGN denoising problem in each iteration. In the Bayesian setting, AMP often achieves the minimum mean squared error (MMSE) [Guo09; Ran12; ZB13; Krz12a] in the limit of large linear systems ($N \to \infty$, $\frac{M}{N} \to \kappa$, cf. Definition 1.1).

In real-world applications, a multi-processor (MP) version of the linear model could be of interest, due to either storage limitations in each individual processor node, or the need for fast computation. This chapter considers multi-processor linear model (MP-LM) [Mot12; Pat14; Han14; Rav15; Han15a; Han16], in which there are P *processor nodes* and a *fusion center*. Recall from (1.2) that in an MP-LM, each processor node stores $\frac{M}{P}$ rows of the matrix $\mathbf{A}$, and acquires the corresponding linear measurements of the underlying signal $\mathbf{x}$. Without loss of generality, we model the measurement system in processor node $p \in \{1, \cdots, P\}$ as

$$y_i = \mathbf{A}_i \mathbf{x} + z_i, \ i \in \left\{ \frac{M(p-1)}{P} + 1, \cdots, \frac{Mp}{P} \right\}, \tag{4.2}$$

where $\mathbf{A}_i$ is the i-th row of $\mathbf{A}$, and y_i and z_i are the i-th entries of $\mathbf{y}$ and $\mathbf{z}$, respectively. Once every y_i is collected, we run distributed algorithms among the fusion center and P processor nodes to estimate the signal $\mathbf{x}$. MP versions of AMP (MP-AMP) for MP-LM have been studied in the literature [Han14; Han16]. Usually, MP platforms are designed for distributed settings such as sensor networks [PK00; Est02] or large-scale “big data" computing systems [Ec2], where the computational and communication burdens can differ among different settings. We reduce the communication costs of MP platforms by applying lossy compression [Ber71; CT06; GG93] to the communication portion of MP-AMP. Our key idea in this work is to minimize the total communication and computation costs by varying the lossy compression schemes in different iterations of MP-AMP.

4.1.2 Contributions

Rate-distortion (RD) theory suggests that we can transmit data with greatly reduced coding rates, if we allow some distortion at the output. However, the MP-AMP problem does not directly fall into the RD framework, because the quantization error in the current iteration feeds into estimation errors in future iterations. We quantify the interaction between these two forms of error by studying the excess mean squared error (EMSE) of MP-AMP above the MMSE (EMSE=MSE-MMSE, where MSE denotes the mean squared error). Our first contribution (Section 4.3) is to use dynamic programming (DP, cf. Bertsekas [Ber95]) to find a sequence of coding rates that yields a desired EMSE while achieving the smallest combined cost of communication and computation; our DP-based scheme is proved to yield optimal coding rates.

Our second contribution (Section 4.4) is to pose the task of finding the optimal coding rate at each iteration in the low EMSE limit as a convex optimization problem. We prove that the optimal coding rate grows approximately linearly in the low EMSE limit. At the same time, we also provide the theoretic asymptotic growth rate of the optimal coding rates in the limit of low EMSE. This provides practitioners with a heuristic to find a near-optimal coding rate sequence without solving the optimization problem. The linearity of the optimal coding rate sequence (defined in Section 4.3) is also illustrated numerically. With the rate being approximately linear, we obtain that the combined cost of computation and communication scales as $O(\log^2(1/\text{EMSE}))$.

In Section 4.5, we further consider a rich design space that includes various costs, such as the number of iterations T, aggregate coding rate R_{agg}, which is the sum of the coding rates in all iterations and is formally defined in (4.14), and the MSE achieved by the estimation algorithm. In such a rich design space, reducing any cost is likely to incur an increase in other costs, and it is impossible to simultaneously minimize all the costs. Han et al. [Han14] reduce the communication costs, and Ma et al. [Ma14c] develop an algorithm with reduced computation; both works [Han14; Ma14c] achieve a reasonable MSE. However, the optimal trade-offs in this rich design space have not been studied. Our third contribution is to pose the problem of finding the best trade-offs among the individual costs T, R_{agg}, and MSE as a multi-objective optimization problem (MOP), and study the properties of Pareto optimal tuples [DD98] of this MOP. These properties are verified numerically using the DP-based scheme developed in this chapter.

Finally, we emphasize that although this chapter is presented for the specific framework of MP-AMP, similar methods could be applied to other iterative distributed algorithms, such as consensus averaging [Fra08; Tha13], to obtain the optimal coding rate as well as optimal trade-offs between communication and computation costs.

Organization: The rest of the chapter is organized as follows. Section 4.2 provides background content. Section 4.3 formulates a DP scheme that finds an optimal coding rate. Section 4.4 proves that any optimal coding rate in the low EMSE limit grows approximately linearly as iterations proceed. Section 4.5 studies the optimal trade-offs among the computation cost, communication cost, and the MSE of the estimate. Section 4.6 uses some real-world examples to showcase the different trade-offs between communication and computation costs, and Section 4.7 concludes the chapter.

4.2 Background

4.2.1 Centralized linear model using AMP

In our linear model (4.1), we consider an independent and identically distributed (i.i.d.) Gaussian measurement matrix $\mathbf{A}$, i.e., $A_{i,j} \sim \mathcal{N}(0, \frac{1}{M})$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2. The signal entries follow an i.i.d. distribution, $f_X(x)$. The noise entries obey $z_i \sim \mathcal{N}(0, \sigma_Z^2)$, where σ_Z^2 is the noise variance.

Starting from $\mathbf{x}_0 = \mathbf{0}$, the AMP framework [Don09] proceeds iteratively according to[1]

$$\mathbf{x}_{t+1} = \eta_t(\mathbf{A}^\top \mathbf{r}_t + \mathbf{x}_t), \tag{4.3}$$

$$\mathbf{r}_t = \mathbf{y} - \mathbf{A}\mathbf{x}_t + \frac{1}{\kappa}\mathbf{r}_{t-1}\langle \eta'_{t-1}(\mathbf{A}^\top \mathbf{r}_{t-1} + \mathbf{x}_{t-1})\rangle, \tag{4.4}$$

where $\eta_t(\cdot)$ is a denoising function, $\eta'_t(\cdot) = \frac{d\eta_t(\cdot)}{d\{\cdot\}}$ is the derivative of $\eta_t(\cdot)$, and $\langle \mathbf{u} \rangle = \frac{1}{N}\sum_{i=1}^N u_i$ for any vector $\mathbf{u} \in \mathbb{R}^N$. The subscript t represents the iteration index, $\{\cdot\}^\top$ denotes the matrix transpose operation, and $\kappa = \frac{M}{N}$ is the measurement rate. Owing to the decoupling effect [Tan02; GV05; GW08], in each AMP iteration [BM11; Mon12; RV16], the vector $\mathbf{f}_t = \mathbf{A}^\top \mathbf{r}_t + \mathbf{x}_t$ in (4.3) is statistically equivalent

[1] AMP is an approximation to the belief propagation algorithm (2.11).

to the input signal $\mathbf{x}$ corrupted by AWGN $\mathbf{w}_t$ generated by a source $W \sim \mathcal{N}(0,\sigma_t^2)$,

$$\mathbf{f}_t = \mathbf{x} + \mathbf{w}_t. \tag{4.5}$$

We call (4.5) the *equivalent scalar channel*. In large systems ($N \to \infty$, $\frac{M}{N} \to \kappa$),[2] a useful property of AMP [BM11; Mon12; RV16] is that the noise variance σ_t^2 evolves following state evolution (SE):

$$\sigma_{t+1}^2 = \sigma_Z^2 + \frac{1}{\kappa}\mathrm{MSE}(\eta_t,\sigma_t^2), \tag{4.6}$$

where $\mathrm{MSE}(\eta_t,\sigma_t^2) = \mathbb{E}_{X,W}\left[\left(\eta_t\left(X+W\right)-X\right)^2\right]$, $\mathbb{E}_{X,W}(\cdot)$ is expectation with respect to (w.r.t.) X and W, and X is the source that generates $\mathbf{x}$. Note that $\sigma_1^2 = \sigma_Z^2 + \frac{\mathbb{E}[X^2]}{\kappa}$, because of the all-zero initial estimate for $\mathbf{x}$. Formal statements for SE appear in prior work [BM11; Mon12; RV16].

In this chapter, we confine ourselves to the Bayesian setting, in which we assume knowledge of the true prior, $f_X(x)$, for the signal $\mathbf{x}$. Therefore, throughout this chapter we use conditional expectation, $\eta_t(\cdot) = \mathbb{E}[\mathbf{x}|\mathbf{f}_t]$, as the MMSE-achieving denoiser.[3] The derivative of $\eta_t(\cdot)$, which is continuous, can be easily obtained, and is omitted for brevity. Other denoisers such as soft thresholding [Don09; Mon12; BM11] yield MSE's that are larger than that of the MMSE denoiser, $\eta_t(\cdot) = \mathbb{E}[\mathbf{x}|\mathbf{f}_t]$. When the true prior for $\mathbf{x}$ is unavailable, parameter estimation techniques can be used [Ma16]; Ma et al. [Ma15] study the behavior of AMP when the denoiser uses a mismatched prior.

4.2.2 MP-LM using lossy MP-AMP

In the sensing problem formulated in (4.2), the measurement matrix is stored in a distributed manner in each processor node. Lossy MP-AMP [Han16] iteratively solves MP-LM using lossily compressed messages:

$$\text{Processor nodes: } \mathbf{r}_t^p = \mathbf{y}^p - \mathbf{A}^p\mathbf{x}_t + \frac{1}{\kappa}\mathbf{r}_{t-1}^p\omega_{t-1}, \tag{4.7}$$

$$\mathbf{f}_t^p = \frac{1}{P}\mathbf{x}_t + (\mathbf{A}^p)^\top \mathbf{r}_t^p, \tag{4.8}$$

$$\text{Fusion center: } \mathbf{f}_{Q,t} = \sum_{p=1}^{P} Q(\mathbf{f}_t^p),\ \omega_t = \langle d\eta_t(\mathbf{f}_{Q,t})\rangle, \tag{4.9}$$

$$\mathbf{x}_{t+1} = \eta_t(\mathbf{f}_{Q,t}), \tag{4.10}$$

[2] Note that the results of this chapter only hold for large systems.

[3] Tan et al. [Tan14a] showed that AMP with MMSE-achieving denoisers can be used as a building block for algorithms that minimize arbitrary user-defined error metrics.

where $Q(\cdot)$ denotes quantization, and an MP-AMP iteration refers to the process from (4.7) to (4.10). The processor nodes send quantized (lossily compressed) messages, $Q(\mathbf{f}_t^p)$, to the fusion center. The reader might notice that the fusion center also needs to transmit the denoised signal vector $\mathbf{x}_t$ and a scalar ω_{t-1} to the processor nodes. The transmission of ω_{t-1} is negligible, and the fusion center may broadcast $\mathbf{x}_t$ so that naive compression of $\mathbf{x}_t$, such as compression with a fixed quantizer, is sufficient. Hence, we will not discuss possible compression of messages transmitted by the fusion center.

Assume that we quantize $\mathbf{f}_t^p, \forall p$, and use C bits to encode the quantized vector $Q(\mathbf{f}_t^p) \in \mathbb{R}^N$. According to (2.9), the *coding rate* is $R = \frac{C}{N}$. We incur an *expected distortion*

$$D_t^p = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(Q(f_{t,i}^p) - f_{t,i}^p)^2\right]$$

at iteration t in each processor node,[4] where $Q(f_{t,i}^p)$ and $f_{t,i}^p$ are the i-th entries of the vectors $Q(\mathbf{f}_t^p)$ and $\mathbf{f}_t^p$, respectively, and the expectation is over $\mathbf{f}_t^p$. When the size of the problem grows, i.e., $N \to \infty$, the rate-distortion (RD) function, denoted by $R(D)$, offers the fundamental information theoretic limit on the coding rate R for communicating a long sequence up to distortion D [CT06; Ber71; GG93; WV12a]. A pivotal conclusion from RD theory is that coding rates can be greatly reduced even if D is small. The function $R(D)$ can be computed in various ways [Ari72; Bla72; Ros94], and can be achieved by an RD-optimal quantization scheme in the limit of large N. Other quantization schemes may require larger coding rates to achieve the same expected distortion D.

The goal of this chapter is to understand the fundamental trade-offs for MP-LM using MP-AMP. Hence, unless otherwise stated, we assume that appropriate vector quantization (VQ) schemes [Lin80; Gra84; GG93], which achieve $R(D)$, are applied within each MP-AMP iteration, although our analysis is readily extended to practical quantizers such as entropy coded scalar quantization (ECSQ) [GG93; CT06]. (Note that the cost of running quantizers in each processor node is not considered, because the cost of processing a bit is usually much smaller than the cost of transmitting it.) Therefore, the signal *at the fusion center* before denoising can be modeled as

$$\mathbf{f}_{Q,t} = \sum_{p=1}^{P} Q(\mathbf{f}_t^p) = \mathbf{x} + \mathbf{w}_t + \mathbf{n}_t, \tag{4.11}$$

where $\mathbf{w}_t$ is the equivalent scalar channel noise (4.5) and $\mathbf{n}_t$ is the overall quantization error whose entries follow $\mathcal{N}(0, PD_t)$. Because the quantization error, $\mathbf{n}_t$, is a sum of quantization errors in the P processor nodes, $\mathbf{n}_t$ resembles Gaussian noise due to the central limit theorem. Han et al. suggest

[4] Because we assume that $\mathbf{A}$ and $\mathbf{z}$ are both i.i.d., the expected distortions are the same over all P nodes, and can be denoted by D_t for simplicity. Note also that $D_t = \mathbb{E}[(Q(f_{t,i}^p) - f_{t,i}^p)^2]$ due to $\mathbf{x}$ being i.i.d.

that SE for lossy MP-AMP [Han16] (called lossy SE) follows

$$\sigma_{t+1}^2 = \sigma_Z^2 + \frac{1}{\kappa}\text{MSE}(\eta_t, \sigma_t^2 + PD_t), \tag{4.12}$$

where σ_t^2 can be estimated by $\widehat{\sigma}_t^2 = \frac{1}{M}\|\mathbf{r}_t\|_2^2$ with $\|\cdot\|_p$ denoting the ℓ_p norm [BM11; Mon12], and σ_{t+1}^2 is the variance of $\mathbf{w}_{t+1}$.

The rigorous justification of (4.12) by extending the framework put forth by Bayati and Montanari [BM11] and Rush and Venkataramanan [RV16] is left for future work. Instead, we argue that lossy SE (4.12) asymptotically tracks the evolution of σ_t^2 in lossy MP-AMP in the limit of $\frac{PD_t}{\sigma_t^2} \to 0$. Our argument is comprised of three parts: (*i*) $\mathbf{w}_t$ and $\mathbf{n}_t$ (4.11) are approximately independent in the limit of $\frac{PD_t}{\sigma_t^2} \to 0$, (*ii*) $\mathbf{w}_t + \mathbf{n}_t$ is approximately independent of $\mathbf{x}$ in the limit of $\frac{PD_t}{\sigma_t^2} \to 0$, and (*iii*) lossy SE (4.12) holds if (*i*) and (*ii*) hold. The first part ($\mathbf{w}_t$ and $\mathbf{n}_t$ are independent) ensures that we can track the variance of $\mathbf{w}_t + \mathbf{n}_t$ with $\sigma_t^2 + PD_t$. The second part ($\mathbf{w}_t + \mathbf{n}_t$ is independent of $\mathbf{x}$) ensures that lossy MP-AMP follows lossy SE (4.12) as it falls under the general framework discussed in Bayati and Montanari [BM11] and Rush and Venkataramanan [RV16]. Hence, the third part of our argument holds. The first two parts are backed up by extensive numerical evidence in Appendix B.1, where ECSQ [GG93; CT06] is used; ECSQ approaches $R(D)$ within 0.255 bits in the high rate limit (corresponds to small distortion) [GG93]. Furthermore, Appendix B.2 provides extensive numerical evidence to show that lossy SE (4.12) indeed tracks the evolution of the MSE when $\mathbf{w}_t$ and $\mathbf{n}_t$ are independent and $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ are independent.

Although lossy SE (4.12) requires $\frac{PD_t}{\sigma_t^2} \to 0$, if scalar quantization is used in a practical implementation, then lossy SE approximately holds when $\gamma < \frac{2\sigma_t}{\sqrt{P}}$, where γ is the quantization bin size of the scalar quantizer (details in Appendices B.1 and B.2). Note that the condition $\gamma < \frac{2\sigma_t}{\sqrt{P}}$ is motivated by Widrow and Kollár [WK08]. If appropriate VQ schemes [Lin80; Gra84; GG93] are used, then we might need milder requirements than $\frac{PD_t}{\sigma_t^2} \to 0$ in the scalar quantizer case, in order for $\mathbf{w}_t$ and $\mathbf{n}_t$ to be independent and for $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ to be independent.

Denote the coding rate used to transmit $Q(\mathbf{f}_t^p)$ at iteration t by R_t. The sequence $\mathbf{R} = (R_1, \cdots, R_T)$ is called the *coding rate sequence*, where T is the total number of MP-AMP iterations. Given $\mathbf{R}$, the distortion D_t can be evaluated with $R(D)$, and the scalar channel noise variance σ_t^2 can be evaluated with (4.12). Hence, the MSE for $\mathbf{R}$ can be predicted. The MSE at the last iteration is called the *final MSE*.

4.3 Optimal Rates Using Dynamic Programming

In this section, we first define the cost of running MP-AMP. We then use DP to find an optimal coding rate sequence with minimum cost, while achieving a desired EMSE.

Definition 4.1 (Combined cost)**.** *Define the cost of estimating a signal in an MP system as*

$$C^b(\mathbf{R}) = b\|\mathbf{R}\|_0 + \|\mathbf{R}\|_1, \tag{4.13}$$

where $\|\mathbf{R}\|_0 = T$ *is the number of iterations to run, and* $\|\mathbf{R}\|_1$ *is the aggregate coding rate, denoted also by* R_{agg}*,*

$$R_{agg} = \|\mathbf{R}\|_1 = \sum_{t=1}^{T} R_t. \tag{4.14}$$

The parameter b *is the cost of computation in one MP-AMP iteration normalized by the cost of transmitting* $Q(\mathbf{f}_t^p)$ (4.9) *at a coding rate of 1 bit/entry. Also, the cost at iteration* t *is*

$$C_t^b(R_t) = b \times \mathbb{1}_{R_t \neq 0} + R_t, \tag{4.15}$$

where the indicator function $\mathbb{1}_{\mathscr{A}}$ *is 1 if the condition* $\mathscr{A}$ *is met, else 0. Hence,* $C^b(\mathbf{R}) = \sum_{t=1}^{T} C_t^b(R_t)$*.*

In some applications, we may want to obtain a sufficiently small EMSE at minimum cost (4.13), where the physical meaning of the cost varies in different problems (cf. Section 4.6). Denote the EMSE at iteration t by ϵ_t. Hence, the *final EMSE* at the output of MP-AMP is ϵ_T.

Let us formally state the problem. Our goal is to obtain a coding rate sequence $\mathbf{R}$ for MP-AMP iterations, which is the solution of the following optimization problem:

$$\text{minimize } C^b(\mathbf{R}) \qquad \text{subject to } \epsilon_T \leq \Delta. \tag{4.16}$$

We now have a definition for the optimal coding rate sequence.

Definition 4.2 (Optimal coding rate sequence)**.** *An optimal coding rate sequence* $\mathbf{R}^*$ *is a solution of* (4.16).

To compute $\mathbf{R}^*$, we derive a dynamic programming (DP) [Ber95] scheme, and then prove that it is optimal.

Dynamic programming scheme: Suppose that MP-AMP is at iteration t. Define the smallest cost for the $(T-t)$ remaining iterations to achieve the EMSE constraint, $\epsilon_T \leq \Delta$, as $\Phi_{T-t}(\sigma_t^2)$, which is a function of the scalar channel noise variance at iteration t, σ_t^2 (4.11). Hence, $\Phi_{T-1}(\sigma_1^2)$ is the cost for solving (4.16), where $\sigma_1^2 = \sigma_Z^2 + \frac{1}{\kappa}\mathbb{E}[X^2]$ is due to the all-zero initialization of the signal estimate.

DP uses a base case and recursion steps to find $\Phi_{T-1}(\sigma_1^2)$. In the base case of DP, $T-t=0$, the cost of running MP-AMP is $C_T^b(R_T) = b \times \mathbb{1}_{R_T \neq 0} + R_T$ (4.15). If σ_T^2 is not too large, then there exist some values for R_T that satisfy $\epsilon_T \leq \Delta$; for these σ_T^2 and R_T, we have $\Phi_0(\sigma_T^2) = \min_{R_T} C_T^b(R_T)$. If σ_T^2 is too large, even lossless transmission of $\mathbf{f}_T^p$ during the single remaining MP-AMP iteration (4.12) does not yield an EMSE that satisfies the constraint, $\epsilon_T \leq \Delta$, and we assign $\Phi_0(\sigma_T^2) = \infty$ for such σ_T^2.

Next, in the recursion steps of DP, we iterate back in time by decreasing t (equivalently, increasing $T-t$),

$$\Phi_{T-t}(\sigma_t^2) = \min_{\widehat{R}} \left\{ C_t^b(\widehat{R}) + \Phi_{T-(t+1)}(\sigma_{t+1}^2(\widehat{R})) \right\}, \tag{4.17}$$

where $\widehat{R}$ is the coding rate used in the current MP-AMP iteration t, the equivalent scalar channel noise variance at the fusion center is σ_t^2 (4.11), and $\sigma_{t+1}^2(\widehat{R})$, which is obtained from (4.12), is the variance of the scalar channel noise (4.11) in the next iteration after transmitting $\mathbf{f}_t^p$ at rate $\widehat{R}$. The terms on the right hand side are the current cost of MP-AMP (4.15) (including computational and communication costs) and the minimum combined cost in all later iterations, $t+1, \cdots, T$.

The coding rates $\widehat{R}$ that yield the smallest cost $\Phi_{T-t}(\sigma_t^2)$ for different t and σ_t^2 are stored in a table $\mathscr{R}(t, \sigma_t^2)$. After DP finishes, we obtain the coding rate for the first MP-AMP iteration as $R_1 = \mathscr{R}(1, \sigma_Z^2 + \frac{1}{\kappa}\mathbb{E}[X^2])$. Using R_1, we calculate σ_t^2 from (4.12) for $t=2$ and find $R_2 = \mathscr{R}(2, \sigma_2^2)$. Iterating from $t=1$ to T, we obtain $\mathbf{R} = (R_1, \cdots, R_T)$.

To be computationally tractable, the proposed DP scheme should operate in discretized search spaces for $\sigma_{\{\cdot\}}^2$ and $R_{\{\cdot\}}$. Details about the resolutions of $\sigma_{\{\cdot\}}^2$ and $R_{\{\cdot\}}$ appear in Appendix B.3.

In the following, we state that our DP scheme yields the optimal solution. The proof appears in Appendix B.4.

Lemma 4.1. *The dynamic programming formulation in* (4.17) *yields an optimal coding rate sequence* $\mathbf{R}^*$*, which is a solution of* (4.16) *for the discretized search spaces of* R_t *and* σ_t^2, $\forall t$.

Lemma 4.1 focuses on the optimality of our DP scheme in discretized search spaces for R_t and σ_t^2. It can be shown that we can achieve a desired accuracy level in $\mathbf{R}^*$ by adjusting the resolutions of the discretized search spaces for R_t and σ_t^2. Suppose that the discretized search spaces for $\sigma_{\{\cdot\}}^2$ and $R_{\{\cdot\}}$ have K_1 and K_2 different values, respectively. Then, the computational complexity of our DP scheme is $O(TK_1K_2)$.

Optimal coding rate sequence given by DP: Consider estimating a *Bernoulli-Gaussian* signal,

$$X = X_B X_G, \tag{4.18}$$

where $X_B \sim \text{Ber}(\rho)$ is a Bernoulli random variable, ρ is called the *sparsity rate* of the signal, and $X_G \sim \mathcal{N}(0,1)$; here we use $\rho = 0.1$. Note that the results in this chapter apply to priors, $f_X(x)$, other than (4.18).

We run our DP scheme on a problem with relatively small desired EMSE, $\Delta = 5 \times 10^{-5}$, in the last iteration T. The signal is measured in an MP platform with $P = 100$ processor nodes according to (4.2). The measurement rate is $\kappa = \frac{M}{N} = 0.4$, and the noise variance is $\sigma_Z^2 = \frac{1}{400}$. The parameter $b = 2$ (4.13). We use ECSQ [GG93; CT06] as the quantizer in each processor node, and use the corresponding relation between the rate R_t and distortion D_t of ECSQ in our DP scheme. Note that we require the quantization bin size to be smaller than $\frac{2\sigma_t}{\sqrt{P}}$, according to Section 4.2.2. Figure 4.1

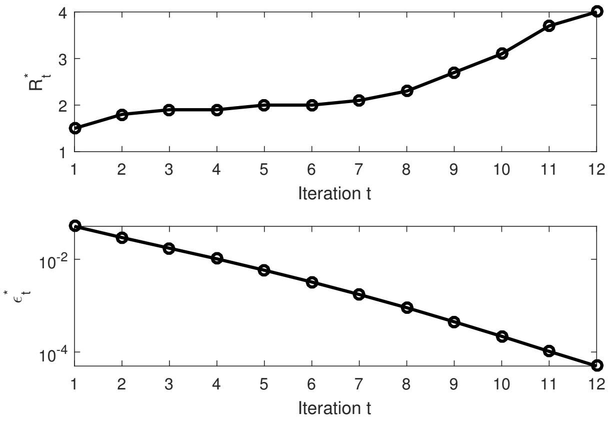


Figure 4.1 The optimal coding rate sequence $\mathbf{R}^*$ (top panel) and optimal EMSE ϵ_t^* (bottom) given by DP are shown as functions of t. (Bernoulli-Gaussian signal (4.18) with $\rho = 0.1$, $\kappa = 0.4$, $P = 100$, $\sigma_Z^2 = \frac{1}{400}$, and $b = 2$.)

illustrates the optimal coding rate sequence $\mathbf{R}^*$ and optimal EMSE ϵ_t^* given by DP as functions of the iteration number t.

It is readily seen that after the first 5–6 iterations the coding rate seems near-linear. The next section proves that any optimal coding rate sequence $\mathbf{R}^*$ is approximately linear in the limit of EMSE→ 0. However, our proof involves the large t limit, and does not provide insights for small t. We ran DP for various configurations. Examining all $\mathbf{R}^*$ from our DP results, we notice that the coding rate is monotone non-decreasing, i.e., $R_1^* \leq R_2^* \leq \cdots \leq R_T^*$. This seems intuitive, because in early iterations of (MP-)AMP, the scalar channel noise $\mathbf{w}_t$ is large, which does not require transmitting $\mathbf{f}_t^p$ (cf. (4.8)) at high fidelity. Hence, a low rate R_t^* suffices. As the iterations proceed, the scalar channel noise $\mathbf{w}_t$ in (4.11) decreases, and the large quantization error $\mathbf{n}_t$ would be unfavorable for the final MSE. Hence, higher rates are needed in later iterations.

4.4 Properties of Optimal Coding Rate Sequences

4.4.1 Intuition

We start this section by providing some brief intuitions about why optimal coding rate sequences are approximately linear when the EMSE is small.

Consider a case where we aim to reach a low EMSE. Montanari [Mon12] provided a geometric interpretation of the relation between the MSE performance of AMP at iteration t and the denoiser $\eta_t(\cdot)$ being used.[5] In the limit of small EMSE, the EMSE decreases by a nearly-constant multiplicative factor per AMP iteration, yielding a geometric decay of the EMSE. In MP-AMP, in addition to the

[5] We will also provide such an interpretation in Section 4.4.2.

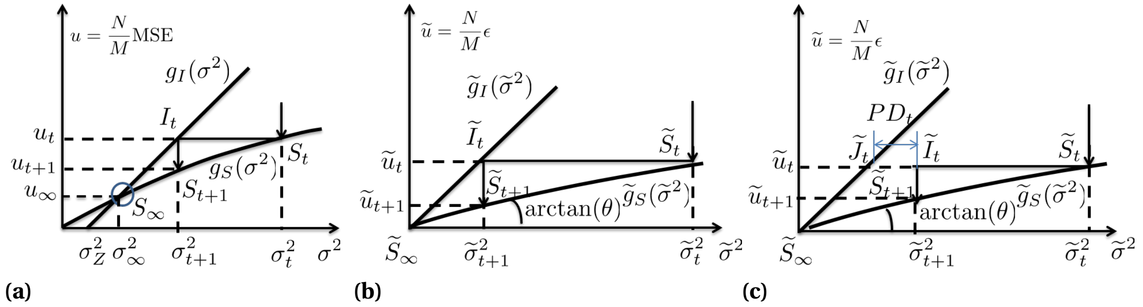

$u = \frac{N}{M}\mathrm{MSE}$
$g_I(\sigma^2)$
I_t
u_t
u_{t+1}
u_∞
S_t
$g_S(\sigma^2)$
S_{t+1}
S_∞
σ_Z^2 σ_∞^2 σ_{t+1}^2 σ_t^2 σ^2
(a)
$\widetilde{u} = \frac{N}{M}\epsilon$
$\widetilde{g}_I(\widetilde{\sigma}^2)$
$\widetilde{I}_t$
$\widetilde{S}_t$
$\widetilde{u}_t$
$\widetilde{u}_{t+1}$
$\widetilde{S}_{t+1}$
$\arctan(\theta)$
$\widetilde{g}_S(\widetilde{\sigma}^2)$
$\widetilde{S}_\infty$ $\widetilde{\sigma}_{t+1}^2$ $\widetilde{\sigma}_t^2$ $\widetilde{\sigma}^2$
(b)
$\widetilde{u} = \frac{N}{M}\epsilon$
$\widetilde{g}_I(\widetilde{\sigma}^2)$
PD_t
$\widetilde{J}_t$
$\widetilde{I}_t$
$\widetilde{S}_t$
$\widetilde{u}_t$
$\widetilde{u}_{t+1}$
$\widetilde{S}_{t+1}$
$\arctan(\theta)$
$\widetilde{g}_S(\widetilde{\sigma}^2)$
$\widetilde{S}_\infty$ $\widetilde{\sigma}_{t+1}^2$ $\widetilde{\sigma}_t^2$ $\widetilde{\sigma}^2$
(c)

Figure 4.2 Geometric interpretation of SE. In all panels, the thick solid curves correspond to $g_I(\cdot)$ and $g_S(\cdot)$, and their offset versions $\widetilde{g}_I(\cdot)$ and $\widetilde{g}_S(\cdot)$. The solid lines with arrows correspond to the SE of AMP. Dashed lines without arrows are auxiliary lines. Panel (a): Illustration of centralized SE. Panel (b): Zooming in to the small region just above point S_∞. Panel (c): Illustration of lossy SE.

equivalent scalar channel noise $\mathbf{w}_t$, we have additive quantization error $\mathbf{n}_t$ (4.11). In order for the EMSE in an MP-AMP system to decay geometrically, the distortion D_t must decay at least as quickly. To obtain this geometric decay in D_t, recall that in the high rate limit, the distortion-rate function typically takes the form $D(R) \approx C_1 2^{-2R}$ [GN98] for some positive constant C_1. We propose for R_t to have the form, $R_t \approx C_2 + C_3 t$, where C_2 and C_3 are constants. In the remainder of this section, we first discuss the geometric interpretation of AMP state evolution, followed by our results about the linearity of optimal coding rate sequences. The detailed proofs appear in the appendices.

4.4.2 Geometric interpretation of AMP state evolution

Centralized SE: The equivalent scalar channel of AMP is given by (4.5). We rewrite the centralized AMP SE (4.6) as follows [Don09; BM11; RV16],

$$\underbrace{\sigma_{t+1}^2 - \sigma_Z^2}_{g_I(\sigma_{t+1}^2)} = \underbrace{\frac{N}{M}\mathrm{MSE}_{\eta_t}(\sigma_t^2)}_{g_S(\sigma_t^2)}, \tag{4.19}$$

where $\mathrm{MSE}_{\eta_t}(\sigma_t^2)$ denotes the MSE after denoising $\mathbf{f}_t$ (4.5) using $\eta_t(\cdot)$. The functions $g_I(\cdot)$ and $g_S(\cdot)$ are illustrated in Figure 4.2a with solid curves; the meanings of I and S will become clear below. We see that $g_I(\sigma_t^2)$ is an affine function with unit slope, whereas $g_S(\sigma_t^2)$ is generally a non-linear function of σ_t^2 (see Figure 4.2a). The lines with arrows illustrate the state evolution (SE). Details appear below.

In Figure 4.2a, we present a geometric interpretation of SE. The horizontal axis is the scalar channel noise variance σ^2 and the vertical axis represents the scaled MSE, $u = \frac{N}{M}\mathrm{MSE}$. Let $S_t = (\sigma_t^2, u_t)$ be the *state* point that is reached by SE in iteration t. We follow the SE trajectory $S_t \to I_t \to$

$S_{t+1} \to \cdots$ in Figure 4.2a, where $I_t = (\sigma^2_{t+1}, u_t)$ represents the *intermediate* point in the transition between states S_t and S_{t+1} corresponding to iterations t and $t+1$, respectively. Observe that the points S_t and I_t have the same ordinate (u_t), while S_{t+1} and I_t have the same abscissa (σ^2_{t+1}), which are related as $\sigma^2_{t+1} = g_I^{-1}(u_t)$ and $u_{t+1} = g_S(\sigma^2_{t+1})$. As t grows, σ^2_t converges to σ^2_∞, which is the abscissa of the point S_∞. The ordinate of point S_∞ is $u_\infty = \frac{N}{M}\mathrm{MSE}_\infty$, where $\mathrm{MSE}_\infty = \mathrm{MMSE}$. If we stop the algorithm at iteration T, or equivalently at point $S_T = (\sigma^2_T, u_T)$, the corresponding MSE, MSE_T, has an EMSE of $\epsilon_T = \mathrm{MSE}_T - \mathrm{MMSE}$.

In Figure 4.2b, we zoom into the neighborhood of point S_∞. To make the presentation more concise, we vertically offset $g_I(\cdot)$ and $g_S(\cdot)$ by $\frac{N}{M}\mathrm{MMSE}$ and horizontally offset them by σ^2_∞; we call the resulting functions $\widetilde{g}_I(\cdot)$ and $\widetilde{g}_S(\cdot)$, respectively. Hence, the vertical axis in Figure 4.2b represents the scaled EMSE, $\widetilde{u} = \frac{N}{M}\mathrm{EMSE} = \frac{N}{M}\epsilon$, and we have $\widetilde{g}_I(\widetilde{\sigma}^2_t) = g_I(\widetilde{\sigma}^2_t + \sigma^2_\infty) - \frac{N}{M}\mathrm{MMSE}$ and $\widetilde{g}_S(\widetilde{\sigma}^2_t) = g_S(\widetilde{\sigma}^2_t + \sigma^2_\infty) - \frac{N}{M}\mathrm{MMSE}$. Observe that $\widetilde{g}_I(0) = \widetilde{g}_S(0) = 0$. Additionally, the slope of $\widetilde{g}_I(\widetilde{\sigma}^2_t)$ is $\widetilde{g}_I'(\widetilde{\sigma}^2_t) = 1$, where $\widetilde{g}_I'(\cdot)$ is the first-order derivative of $\widetilde{g}_I(\cdot)$ w.r.t. $\widetilde{\sigma}^2_t$ (Figure 4.2b). Because the MSE function for the MMSE-achieving denoiser is continuous and differentiable twice [WV11], we can invoke Taylor's theorem to express

$$\widetilde{g}_S(\widetilde{\sigma}^2_t) = \widetilde{g}_S'(0)\widetilde{\sigma}^2_t + \frac{1}{2}\widetilde{g}_S''(\zeta_t)\widetilde{\sigma}^4_t, \tag{4.20}$$

where $\zeta_t \in (0, \widetilde{\sigma}^2_t)$, and $\widetilde{g}_S'(\widetilde{\sigma}^2_t)$ and $\widetilde{g}_S''(\widetilde{\sigma}^2_t)$ are the first- and second-order derivatives of $\widetilde{g}_S(\cdot)$ w.r.t. $\widetilde{\sigma}^2_t$, respectively. Due to continuity and differentiability of the denoising function, $\widetilde{g}_S(\cdot)$ is invertible in a neighborhood around 0, and its inverse is denoted by $\widetilde{g}_S^{-1}(\cdot)$. Invoking Taylor's theorem,

$$\widetilde{g}_S^{-1}(\widetilde{u}_t) = (\widetilde{g}_S^{-1})'(0)\widetilde{u}_t + \frac{1}{2}(\widetilde{g}_S^{-1})''(\zeta_t)\widetilde{u}_t^2, \tag{4.21}$$

where $\zeta_t \in (0, \widetilde{u}_t)$, and $(\widetilde{g}_S^{-1})'(\widetilde{u}_t)$ and $(\widetilde{g}_S^{-1})''(\widetilde{u}_t)$ are the first- and second-order derivatives of $\widetilde{g}_S^{-1}(\cdot)$ w.r.t. $\widetilde{u}_t$, respectively. When $t \to \infty$, $\widetilde{\sigma}^2_t \to 0$ and $\widetilde{u}_t \to 0$, and the higher-order terms become $\frac{1}{2}\widetilde{g}_S''(\xi_t)\widetilde{\sigma}^4_t = O(\widetilde{\sigma}^4_t)$ and $\frac{1}{2}(\widetilde{g}_S^{-1})''(\zeta_t)\widetilde{u}_t^2 = O(\widetilde{u}_t^2)$. In other words, both $\widetilde{g}_S(\widetilde{\sigma}^2_t)$ and $\widetilde{g}_S^{-1}(\widetilde{u}_t)$ become approximately linear functions, as shown in Figure 4.2b. We further denote the slope of $\widetilde{g}_S(0)$ by θ, i.e.,

$$\theta = \widetilde{g}_S'(0) = \frac{1}{(\widetilde{g}_S^{-1})'(0)}. \tag{4.22}$$

To calculate the slope θ, we first calculate the scalar channel noise variance for point S_∞, σ^2_∞, by using replica analysis [ZB13; Krz12a],[6] and obtain $\theta = g_S'(\sigma^2_\infty) = \widetilde{g}_S'(0)$. Moreover, the slope of $\widetilde{g}_S(0)$ satisfies $\theta = \widetilde{g}_S'(0) \in (0,1)$; otherwise, the curves $\widetilde{g}_I(\cdot)$ and $\widetilde{g}_S(\cdot)$ would not intersect at point S_∞.

[6] The outcome of replica analysis [ZB13; Krz12a] is close to simulating SE (4.19) with a large number of iterations.

Lossy SE: Considering lossy SE (4.12), we have

$$\underbrace{\sigma_{t+1}^2 - \sigma_Z^2}_{g_I(\sigma_{t+1}^2)} = \underbrace{\frac{N}{M}\text{MSE}_{\eta_t}(\sigma_t^2 + PD_t)}_{g_S(\sigma_t^2 + PD_t)}, \tag{4.23}$$

where P is the number of processor nodes in an MP network, and D_t is the expected distortion incurred by each node at iteration t. Note that lossy SE has not been rigorously proved in the literature, although we argued in Section 4.2.2 that it tracks the evolution of the equivalent scalar channel noise variance σ_t^2 when $D_t \ll \frac{1}{P}\sigma_t^2$.

We notice the additional term PD_t, which corresponds to the distortion *at the fusion center*. Because the P nodes transmit their signals $\mathbf{f}_t^p$ with distortion D_t, and their messages are independent, the fusion center's signal has distortion PD_t. The lines with arrows in Figure 4.2c illustrate the lossy SE after vertically offsetting $g_I(\cdot)$ and $g_S(\cdot)$ by $\frac{N}{M}$MMSE and horizontally offsetting $g_I(\cdot)$ and $g_S(\cdot)$ by σ_∞^2. After arriving at point $\widetilde{S}_t$, we move horizontally to $\widetilde{J}_t$, and obtain the ordinate of $\widetilde{I}_t$, $\widetilde{u}_t$, from $\widetilde{g}_S(\widetilde{\sigma}_t^2 + PD_t) = \widetilde{u}_t$. Geometrically, SE is dragged to the right by distance PD_t from point $\widetilde{J}_t$ to $\widetilde{I}_t$, and then SE descends from $\widetilde{I}_t$ to $\widetilde{S}_{t+1}$.

4.4.3 Asymptotic linearity of the optimal coding rate sequence

Recall from (4.20) that $\lim_{t\to\infty} \widetilde{\sigma}_t^2 = 0$. Hence, as t grows, $f_{t,i}$ (4.5) converges in distribution to $x_i + \mathcal{N}(0, \sigma_\infty^2)$. Therefore, the RD function converges to some fixed function as t grows. For large coding rate R, this function has the form

$$R_t = \frac{1}{2}\log_2\left(\frac{C_1}{D_t}\right)(1 + o_t(1)), \tag{4.24}$$

for some constant C_1 that does not depend on t [GN98]. Note that the assumption of $\widetilde{\sigma}_t^2$ being small implicitly requires the coding rate used in the corresponding iteration to be large.

For an optimal coding rate sequence $\mathbf{R}^*$, we call the distortion D_t^*, derived from (4.24), incurred by the optimal coding rate R_t^* at a certain iteration t the *optimal distortion*. Correspondingly, we call the EMSE achieved by MP-AMP with $\mathbf{R}^*$, denoted by ϵ_t^*, the *optimal EMSE* at iteration t. In the following, we state our main results on the optimal coding rate, the optimal distortion, and the optimal EMSE.

Theorem 4.1 (Linearity of the optimal coding rate sequence)**.** *Supposing that lossy SE* (4.23) *holds, we have*

$$\lim_{t\to\infty} \frac{D_{t+1}^*}{D_t^*} = \theta, \tag{4.25}$$

where θ is defined in (4.22). *Furthermore,*

$$\lim_{t\to\infty}\left(R^*_{t+1}-R^*_t\right)=\frac{1}{2}\log_2\left(\frac{1}{\theta}\right). \tag{4.26}$$

Theorem 4.1 is proved in Appendix B.5.

Remark 4.1. *Define the additive growth rate of an optimal coding rate sequence* $\mathbf{R}^*$ *at iteration* t *as* $R^*_{t+1}-R^*_t$. *Theorem 4.1 not only shows that any optimal coding rate sequence grows approximately linearly in the low EMSE limit, but also provides a way to calculate its additive growth rate in the low EMSE limit. Hence, if the goal is to achieve a low EMSE, practitioners could simply use a coding rate sequence that has a fixed coding rate in the first few iterations and then increases linearly with additive growth rate* $\frac{1}{2}\log_2\left(\frac{1}{\theta}\right)$.

The following theorem provides (*i*) the relation between the optimal distortion D^*_{t+1} and the optimal EMSE ϵ^*_t in the large t limit, and (*ii*) the convergence rate of the optimal EMSE ϵ^*_t.

Theorem 4.2. *Assuming that lossy SE* (4.23) *holds, we have*

$$\lim_{t\to\infty}\frac{D^*_t}{\epsilon^*_t}=0. \tag{4.27}$$

Furthermore, the convergence rate of the optimal EMSE is

$$\lim_{t\to\infty}\frac{\epsilon^*_{t+1}}{\epsilon^*_t}=\theta. \tag{4.28}$$

Theorem 4.2 is proved in Appendix B.6. Note that $\lim_{t\to\infty}\frac{D^*_t}{\epsilon^*_t}=0$ meets the requirement $\frac{PD_t}{\sigma^2_t}\to 0$ discussed in Section 4.2.2. Extending Theorems 4.1 and 4.2, we have the following result.

Corollary 4.3. *Assuming that lossy SE* (4.12) *holds, the combined computation and communication cost* (4.13) *scales as* $O(\log^2(1/\Delta))$, $\forall b>0$, *where* Δ *is the desired EMSE.*

Proof. Given Theorem 4.2, we obtain that the optimal EMSE, ϵ^*_t, indeed decreases geometrically in the large t limit (as a reminder, we provided such intuition in Section 4.4.1). Considering (4.14) and Theorem 4.1, the total computation and communication cost (4.13) for running T iterations is $C^b(\mathbf{R}^*)=O(T^2)=O(\log^2(1/\epsilon^*_T))=O(\log^2(1/\Delta))$. □

Remark 4.2. *The key to the proofs of Theorems 4.1 and 4.2 is lossy SE* (4.23). *We expect that the linearity of the optimal coding rate sequence could be extended to other iterative distributed algorithms provided that (i) they have formulations similar to lossy SE* (4.23) *that track their estimation errors and (ii) their estimation errors converge geometrically. Moreover, formulations that track the estimation error in such algorithms might require less restrictive constraints than AMP. For example, consensus*

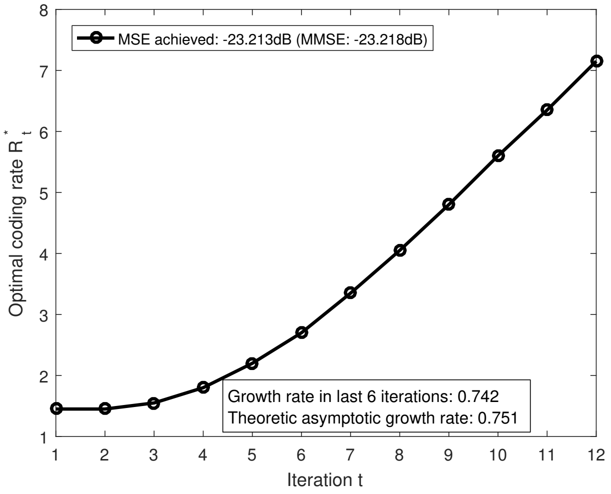


Figure 4.3 Comparison of the additive growth rate of the optimal coding rate sequence given by DP at low EMSE and the asymptotic additive growth rate $\frac{1}{2}\log_2\left(\frac{1}{\theta}\right)$. (Bernoulli-Gaussian signal (4.18) with $\rho = 0.2$, $\kappa = 1$, $P = 100, \sigma_Z^2 = 0.01$, $b = 0.782$.)

averaging [Fra08; Tha13] only requires i.i.d. entries in the vector that each node in the network averages.

4.4.4 Comparison of DP results to Theorem 4.1

We run DP (cf. Section 4.3) to find an optimal coding rate sequence $\mathbf{R}^*$ for the setting of $P = 100$ nodes, a Bernoulli-Gaussian signal (4.18) with sparsity rate $\rho = 0.2$, measurement rate $\kappa = 1$, noise variance $\sigma_Z^2 = 0.01$, and parameter $b = 0.782$. The goal is to achieve a desired EMSE of 0.005 dB, i.e., $10\log_{10}\left(1+\frac{\Delta}{\text{MMSE}}\right) = 0.005$. We use ECSQ [GG93; CT06] as the quantizer in each processor node and use the corresponding relation between the rate R_t and distortion D_t of ECSQ in the DP scheme. Note that we require the quantization bin size γ to be smaller than $\frac{2\sigma_t}{\sqrt{P}}$, according to Section 4.2.2. We know that ECSQ achieves a coding rate within an additive constant of the RD function $R(D)$ [GG93]. Therefore, the additive growth rate of the optimal coding rate sequence obtained for ECSQ will be the same as the additive growth rate if the RD relation is modeled by $R(D)$ [CT06; Ber71; GG93; WV12a].

The resulting optimal coding rate sequence is plotted in Figure 4.3. The additive growth rate of the last six iterations is $\frac{1}{6}(R_{12}^* - R_6^*) = 0.742$, and the asymptotic additive growth rate according to Theorem 4.1 is $\frac{1}{2}\log_2\left(\frac{1}{\theta}\right) \approx 0.751$. Note that we use $\Delta R_t = 0.05$ in the discretized search space for R_t. Hence, the discrepancy of 0.009 between the additive growth rate from the simulation and the asymptotic additive growth rate is within our numerical precision. In conclusion, our numerical result matches the theoretical prediction of Theorem 4.1.

4.5 Achievable Performance Region

Following the discussion of Section 4.2, we can see that the lossy compression of $\mathbf{f}_t^p, \forall p \in \{1, \cdots, P\}$, can reduce communication costs. On the other hand, the greater the savings in the coding rate sequence **R**, the worse the final MSE is expected to be. If a certain level of final MSE is desired despite a small coding rate budget, then more iterations T will be needed. As mentioned above, there is a trade-off between T, R_{agg}, and the final MSE, i.e., MMSE $+ \Delta$, and there is no solution that minimizes them simultaneously. To deal with such trade-offs, which implicitly correspond to sweeping b in (4.13) in a multi-objective optimization (MOP) problem, it is customary to think about *Pareto optimality* [DD98].

4.5.1 Properties of achievable region

For notational convenience, denote the set of all MSE values achieved by the pair (T, R_{agg}) for some parameter b (4.13) by $\mathscr{E}(T, R_{agg})$. Within (T, R_{agg}), let the smallest MSE be MSE$^*(T, R_{agg})$. We now define the achievable set $\mathscr{C}$,

$$\mathscr{C} := \{(T, R_{agg}, \text{MSE}) \in \mathbb{R}^3_{\geq 0} : \text{MSE} \in \mathscr{E}(T, R_{agg})\},$$

where $\mathbb{R}_{\geq 0}$ is the set of non-negative real numbers. That is, $\mathscr{C}$ contains all tuples (T, R_{agg}, MSE) for which some instantiation of MP-AMP estimates the signal at the desired MSE level using T iterations and aggregate coding rate R_{agg}.

Definition 4.3. *The point $\mathscr{X}_1 \in \mathscr{C}$ is said to dominate another point $\mathscr{X}_2 \in \mathscr{C}$, denoted by $\mathscr{X}_1 \prec \mathscr{X}_2$, if $T_1 \leq T_2$, $R_{agg_1} \leq R_{agg_2}$, and $MSE_1 \leq MSE_2$. A point $\mathscr{X}^* \in \mathscr{C}$ is Pareto optimal if there does not exist $\mathscr{X} \in \mathscr{C}$ satisfying $\mathscr{X} \prec \mathscr{X}^*$. Furthermore, let $\mathscr{P}$ denote the set of all Pareto optimal points,*

$$\mathscr{P} := \{\mathscr{X} \in \mathscr{C} : \mathscr{X} \textit{ is Pareto optimal}\}. \tag{4.29}$$

In words, the tuple (T, R_{agg}, MSE) is Pareto optimal if no other tuple $(\widehat{T}, \widehat{R}_{agg}, \widehat{\text{MSE}})$ exists such that $\widehat{T} \leq T$, $\widehat{R}_{agg} \leq R_{agg}$, and $\widehat{\text{MSE}} \leq \text{MSE}$. Thus, the Pareto optimal tuples belong to the boundary of $\mathscr{C}$.

We extend the definition of the number of iterations T to a probabilistic one. To do so, suppose that the number of iterations is drawn from a probability distribution π over $\mathbb{N}$, such that $\sum_{i=1}^{\infty} \pi_i = 1$. Of course, this definition contains a deterministic $T = j$ as a special case with $\pi_j = 1$ and $\pi_i = 0$ for all $i \neq j$. Armed with this definition of Pareto optimality and the probabilistic definition of the number of iterations, we have the following lemma.

Lemma 4.2. *For a fixed noise variance σ_Z^2, measurement rate κ, and P processor nodes in MP-AMP, the achievable set $\mathscr{C}$ is a convex set.*

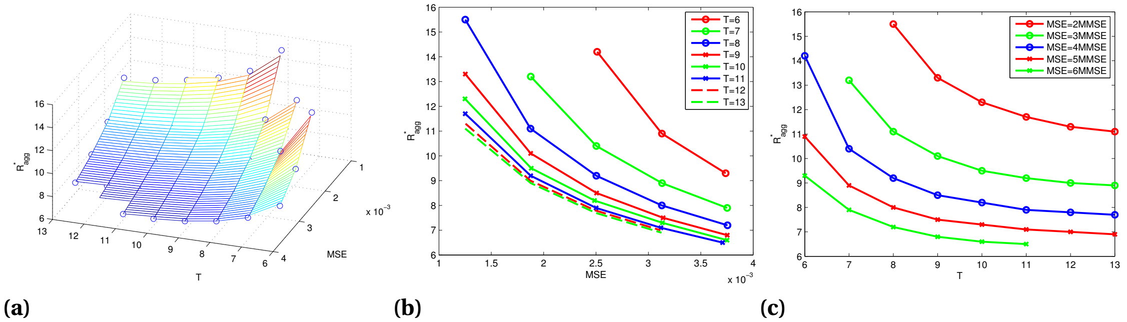


Figure 4.4 Pareto optimal results provided by DP under a variety of parameters b (4.13): (a) Pareto optimal surface, (b) Pareto optimal aggregate coding rate R^*_{agg} (4.14) versus the achieved MSE for different optimal MP-AMP iterations T, and (c) Pareto optimal R^*_{agg} (4.14) versus the number of iterations T for different optimal MSE's. The signal is Bernoulli-Gaussian (4.18) with $\rho = 0.1$. ($\kappa = 0.4$, $P = 100$, and $\sigma_Z^2 = \frac{1}{400}$.)

Proof. We need to show that for any $(T^{(1)}, R^{(1)}_{agg}, \text{MSE}^{(1)})$, $(T^{(2)}, R^{(2)}_{agg}, \text{MSE}^{(2)}) \in \mathscr{C}$ and any $0 < \lambda < 1$,

$$(\lambda T^{(1)} + (1-\lambda)T^{(2)}, \lambda R^{(1)}_{agg} + (1-\lambda)R^{(2)}_{agg}, \lambda \text{MSE}^{(1)} + (1-\lambda)\text{MSE}^{(2)}) \in \mathscr{C}. \tag{4.30}$$

This result is shown using time-sharing arguments (see Cover and Thomas [CT06]). Assume that $(T^{(1)}, R^{(1)}_{agg}, \text{MSE}^{(1)})$, $(T^{(2)}, R^{(2)}_{agg}, \text{MSE}^{(2)}) \in \mathscr{C}$ are achieved by probability distributions $\pi^{(1)}$ and $\pi^{(2)}$, respectively. Let us select all parameters of the first tuple with probability λ and those of the second with probability $(1-\lambda)$. Hence, we have $\pi = \lambda\pi^{(1)} + (1-\lambda)\pi^{(2)}$. Due to the linearity of expectation, $T = \lambda T^{(1)} + (1-\lambda)T^{(2)}$ and $\text{MSE} = \lambda\text{MSE}^{(1)} + (1-\lambda)\text{MSE}^{(2)}$. Again, due to the linearity of expectation, $R_{agg} = \lambda R^{(1)}_{agg} + (1-\lambda)R^{(2)}_{agg}$, implying that (4.30) is satisfied, and the proof is complete. □

Definition 4.4. *Let the function $R^*(T, MSE) : \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the Pareto optimal rate function, which is implicitly described as $R^*(T, MSE) = R^*_{agg} \Leftrightarrow (T, R^*_{agg}, MSE) \in \mathscr{P}$. We further define implicit functions $T^*(R_{agg}, MSE)$ and $MSE^*(T, R_{agg})$ in a similar way.*

Corollary 4.4. *The functions $R^*(T, MSE)$, $T^*(R_{agg}, MSE)$, and $MSE^*(T, R_{agg})$ are convex in their arguments.*

Note that our proof for the convexity of the set $\mathscr{C}$ might be extended to other iterative distributed learning algorithms that transmit lossily compressed messages.

4.5.2 Pareto optimal points via DP

After proving that the achievable set $\mathscr{C}$ is convex, we apply DP in Section 4.3 to find the Pareto optimal points, and validate the convexity of the achievable set.

According to Definition 4.3, the resulting tuple (T, R_{agg}, MSE) computed using DP (Section 4.3) is Pareto optimal on the discretized search spaces. Hence, in this subsection, we run DP to obtain the Pareto optimal points for a certain distributed linear model by sweeping the parameter b (4.13).

Consider the same setting as in Figure 4.1, except that we analyze MP platforms [PK00; Est02; Ec2] for different b (4.13). Running the DP scheme of Section 4.3, we obtain the optimal coding rate sequence $\mathbf{R}^*$ that yields the lowest combined cost while providing a desired EMSE that is at most $\Delta \in \{1, 2, \cdots, 5\} \times \text{MMSE}$ or equivalently $\text{MSE} \in \{2, 3, \cdots, 6\} \times \text{MMSE}$. In Figure 4.4a, we draw the Pareto optimal surface obtained by our DP scheme, where the circles are Pareto optimal points. Figure 4.4b plots the aggregate coding rate R_{agg} as a function of MSE for different optimal numbers of MP-AMP iterations T. Finally, Figure 4.4c plots the aggregate coding rate R_{agg} as a function of T for different optimal MSE's. We can see that the surface comprised of the Pareto optimal points is indeed convex. Note that when running DP to generate Figure 4.4, we used the RD function [CT06; Ber71; GG93; WV12a] to model the relation between the rate R_t and distortion D_t at each iteration, which could be approached by VQ at sufficiently high rates. We also ignored the constraint on the quantization bin size (Section 4.2.2). Therefore, we only present Figure 4.4 for illustration purposes.

When a smaller MSE (or equivalently smaller EMSE) is desired, more iterations T and greater aggregate coding rates R_{agg} (4.14) are needed. Optimal coding rate sequences increase R_{agg} to reduce T when communication costs are low (examples are commercial cloud computing systems [Ec2], multi-processor CPUs, and graphic processing units), whereas more iterations allow to reduce the coding rate when communication is costly (for example, in sensor networks [PK00; Est02]). These applications are discussed in Section 4.6.

Discussion of corner points: We further discuss the corners of the Pareto optimal surface (Figure 4.4) below.

1. First, consider the corner points along the MSE coordinate.

   - If $\text{MSE}^* \to \text{MMSE}$ (or equivalently $\Delta \to 0$), then MP-AMP needs to run infinite iterations with infinite coding rates. Hence, $R^*_{agg} \to \infty$ and $T^* \to \infty$. The rate of growth of R^*_{agg} can be deduced from Theorem 4.1.
   - If $\text{MSE}^* \to \rho$ (the variance of the signal (4.18)), then MP-AMP does not need to run any iterations at all. Instead, MP-AMP outputs an all-zero estimate. Therefore, $\lim_{\text{MSE}^* \to \rho} R^*_{agg} = 0$ and $\lim_{\text{MSE}^* \to \rho} T^* = 0$.

2. Next, we discuss the corner points along the T coordinate.

   - If $T^* \to 0$, then the best MP-AMP can do is to output an all-zero estimate. Hence, $\lim_{T^* \to 0} \text{MSE}^* = \rho$ and $\lim_{T^* \to 0} R^*_{agg} = 0$.
   - The other extreme, $T^* \to \infty$, occurs only when we want to achieve an $\text{MSE}^* \to \text{MMSE}$. Hence, $R^*_{agg} \to \infty$.

3. We conclude with corner points along the R_{agg} coordinate.

   - If $R^*_{agg} \to 0$, then the best MP-AMP can do is to output an all-zero estimate without running any iterations at all. Hence, $\lim_{R^*_{agg}\to 0} \text{MSE}^* = \rho$ and $\lim_{R^*_{agg}\to 0} T^* = 0$.
   - If $R^*_{agg} \to \infty$, then the optimal scheme will use high rates in all iterations, and MP-AMP resembles centralized AMP. Therefore, the MSE* as a function of T^* converges to that of centralized AMP SE (4.6).

4.6 Real-world Case Study

To showcase the difference between optimal coding rate sequences in different platforms, this section discusses several MP platforms including sensor networks [PK00; Est02] and large-scale cloud servers [Ec2]. The costs in these platforms are quite different due to the different constraints in these platforms, and we will see how they affect the optimal coding rate sequence $\mathbf{R}^*$. The changes in the optimal $\mathbf{R}^*$ highlight the importance of optimizing for the correct costs.

4.6.1 Sensor networks

In sensor networks [PK00; Est02], distributed sensors are typically dispatched to remote locations where they collect data and communicate with the fusion center. However, distributed sensors may have severe power consumption constraints. Therefore, low power chips such as the CC253X from Texas Instruments [Cc2] are commonly used in distributed sensors. Some typical parameters for such low power chips are: central processing unit (CPU) clock frequency 32MHz, data transmission rate 250Kbps, voltage between 2V-3.6V, and transceiver current 25mA [Cc2], where the CPU current resembles the transceiver current. Because these chips are generally designed to be low power, when transmitting and receiving data, the CPU helps the transceiver and cannot carry out computing tasks. Therefore, the power consumption can be viewed as constant. Hence, in order to minimize the power consumption, we minimize the total runtime when estimating a signal from MP-LM measurements (4.2) collected by the distributed sensors.

The runtime in each MP-AMP iteration (4.7)-(4.10) consists of (*i*) time for computing (4.7) and (4.8), (*ii*) time for encoding $\mathbf{f}_t^p$ (4.8), and (*iii*) data transmission time for $Q(\mathbf{f}_t^p)$ (4.9). As discussed in Section 4.2.2, the fusion center may broadcast $\mathbf{x}_t$ (4.10), and simple compression schemes can reduce the coding rate. Therefore, we consider the data reception time in the P processor nodes to be constant. The overall computational complexity for (4.7) and (4.8) is $O(\frac{MN}{P})$. Suppose further that (*i*) each processor node needs to carry out two matrix-vector products in each iteration, (*ii*) the overhead of moving data in memory is assumed to be 10 times greater than the actual computation, and (*iii*) the clock frequency is 32MHz. Hence, we assume that the actual time needed for computing (4.7) and (4.8) is $C_4 = \frac{20MN}{32\times 10^6 P}$ sec. Transmitting $Q(\mathbf{f}_t^p)$ of length N at coding rate R requires $\frac{RN}{250\times 10^3}$ sec, where the denominator is the data transmission rate of the transceiver. Assuming that the overhead in communication is approximately the same as the communication load caused by the actual messages, we obtain that the time requested for transmitting $Q(\mathbf{f}_t^p)$ at coding rate R is $C_5 R$ sec, where $C_5 = \frac{2N}{250\times10^3}$. Therefore, the total cost can be calculated from (4.13) with $b = \frac{C_4}{C_5}$ (4.13).

Because low power chips equipped in distributed sensors have limited memory (around 10KB, although sometimes external flash is allowed) [Cc2], the signal length N and number of measurements M cannot be too large. We consider $N = 1000$ and $M = 400$ spread over $P = 100$ sensors, sparsity rate $\rho = 0.1$, and $\sigma_Z^2 = \frac{1}{400}$. We set the desired MSE to be 0.5 dB above the MMSE, i.e., $10\log_{10}\left(1+\frac{\Delta}{\text{MMSE}}\right) = 0.5$, and run DP as in Section 4.3.[7] The coding rate sequence provided by DP is $\mathbf{R}^* = (0.1, 0.1, 0.6, 0.8, 1.0, 1.0, 1.1, 1.1, 1.2, 1.4, 1.6, 1.9, 2.3, 2.7, 3.1)$. In total we have $T = 15$ MP-AMP iterations with $R_{agg} = 20.0$ bits aggregate coding rate (4.14). The final MSE (MMSE $+\,\Delta$) is 7.047×10^{-4}, which is 0.5 dB from the MMSE (6.281×10^{-4}) [ZB13; Krz12a; Guo09; Ran12].

4.6.2 Large-scale cloud server

Having discussed sensor networks [PK00; Est02], we now discuss an application of DP (cf. Section 4.3) to large-scale cloud servers. Consider the dollar cost for users of Amazon EC2 [Ec2], a commercial cloud computing service. A typical cost for CPU time is \$0.03/hour, and the data transmission cost is \$0.03/GB. Assuming that the CPU clock frequency is 2.0GHz and considering various overheads, we need a runtime of $\frac{20MN}{2\times10^9 P}$ sec and the computation cost is $C_4 = \$\frac{20MN}{2\times10^9 P}\times\frac{0.03}{3600}$ per MP-AMP iteration. Similar to Section 4.6.1, the communication cost for coding rate R is $C_5 R = \$2RN\frac{0.03}{8\times10^9}$. Note that the multiplicative factors of 20 in C_4 and 2 in C_5 are due to the same considerations as in Section 4.6.1, and the 8×10^9 in C_5 is the number of bits per GB. Therefore, the total cost with T MP-AMP iterations can still be modeled as in (4.13), where $b = \frac{C_4}{C_5}$.

We consider a problem with the same signal and channel model as the setting of Section 4.6.1, while the size of the problem grows to $N = 50000$ and $M = 20000$ spread over $P = 100$ computing nodes. Running DP, we obtain the coding rate sequence $\mathbf{R}^* = (1.3, 1.6, 1.8, 1.8, 1.8, 1.9, 2.1, 2.3, 2.6, 3.1, 3.7)$ for a total of $T = 11$ MP-AMP iterations with $R_{agg} = 24.0$ bits aggregate coding rate. The final MSE is 7.031×10^{-4}, which is 0.49 dB above the MMSE. Note that this final MSE is 0.01 dB better than our goal of 0.5 dB above the MMSE due to the discretized search spaces used in DP.

Settings with even cheaper communication costs: Compared to large-scale cloud servers, the relative cost of communication is even cheaper in multi-processor CPU and graphics processing unit (GPU) systems. We reduce b by a factor of 100 compared to the large-scale cloud server case above. We rerun DP, and obtain the coding rate sequence $\mathbf{R}^* = (2.3, 2.5, 2.6, 2.7, 2.7, 2.8, 3.0, 3.4, 3.7, 4.5)$ for $T = 10$ and $R_{agg} = 30.2$ bits. Note that 10 iterations are needed for centralized AMP to

[7] Throughout Section 4.6, we use the RD function [CT06; Ber71; GG93; WV12a] to model the relation between rate R_t and distortion D_t at each iteration. We also ignore the constraint on the quantizer (Section 4.2.2). Therefore, the optimal coding rate sequences in Section 4.6 are only for illustration purposes.

converge in this setting. With the low-cost communication of this setting, DP yields a coding rate sequence $\mathbf{R}^*$ within 0.5 dB of the MMSE with the same number of iterations as centralized AMP, while using an average coding rate of only 3.02 bits per iteration.

Remark 4.3. *Let us review the cost tuples* (T, R_{agg}, MSE) *for our three cases. For sensor networks,* $(T, R_{agg}, MSE)_{sensornet} = (15, 20, 7.047 \times 10^{-4})$*; for cloud servers,* $(T, R_{agg}, MSE)_{cloud} = (11, 24, 7.031 \times 10^{-4})$*; and for GPUs,* $(T, R_{agg}, MSE)_{GPU} = (10, 30.2, 7.047 \times 10^{-4})$*. These cost tuples are different points in the Pareto optimal set* $\mathscr{P}$ (4.29)*. We can see for sensor networks that the optimal coding rate sequence reduces* R_{agg} *while adding iterations, because sensor networks have relatively expensive communications. The optimal coding rate sequences use higher rates in cloud servers and GPUs, because their communication costs are relatively lower. Indeed, different trade-offs between computation and communication lead to different aggregate coding rates* R_{agg} *and numbers of MP-AMP iterations* T*. Moreover, the optimal coding rate sequences for sensor networks, cloud servers, and GPUs use average coding rates of* 1.33*,* 2.18*, and* 3.02 *bits/entry/iteration, respectively. Compared to* 32 *bits/entry/iteration single-precision floating point communication schemes, optimal coding rate sequences reduce the communication costs significantly.*

4.7 Conclusion

This chapter used lossy compression in multi-processor (MP) approximate message passing (AMP) for solving MP linear inverse problems. Dynamic programming (DP) was used to obtain the optimal coding rate sequence for MP-AMP that incurs the lowest combined cost of communication and computation while achieving a desired mean squared error (MSE). We posed the problem of finding the optimal coding rate sequence in the low excess MSE (EMSE=MSE-MMSE, where MMSE refers to the minimum MSE) limit as a convex optimization problem and proved that optimal coding rate sequences are approximately linear when the EMSE is small. Additionally, we obtained that the combined cost of computation and communication scales with $O(\log^2(1/\text{EMSE}))$. Furthermore, realizing that there is a trade-off among the communication cost, computation cost, and MSE, we formulated a multi-objective optimization problem (MOP) for these costs and studied the Pareto optimal points that exploit this trade-off. We proved that the achievable region of the MOP is convex.

We further emphasize that there is little work in the prior art discussing the optimization of communication schemes in iterative distributed algorithms. Although we focused on the MP-AMP algorithm, our conclusions such as the linearity of the optimal coding rate sequence and the convexity of the achievable set of communication/computation trade-offs could be extended to other iterative distributed algorithms including consensus averaging [Fra08; Tha13].

CHAPTER

5 UNIVERSAL ALGORITHM

Previous chapters discussed the information theoretic performance limits for multi-measurement vector problems (1.3) and also studied the optimal trade-offs among different costs in multi-processor linear models (1.2). When the number of rows M is smaller than the number of columns N in the measurement matrix $\mathbf{A}$, we call the corresponding linear model a compressed sensing (CS) problem. In this chapter, we study the CS signal estimation problem. While CS usually assumes sparsity or compressibility in the input signal during estimation, the signal structure that can be leveraged is often not known a priori. In this chapter, we consider universal CS signal estimation, where the statistics of a stationary ergodic signal source are estimated simultaneously with the signal itself. Inspired by Kolmogorov complexity and minimum description length, we focus on a maximum a posteriori (MAP) estimation framework that leverages universal priors to match the complexity of the source. Our framework can also be applied to general linear inverse problems where more measurements than the signal length might be needed. We provide theoretical results that support the algorithmic feasibility of universal MAP estimation using a Markov chain Monte Carlo implementation (an algorithmic framework mimicking the annealing process in statistical physics, cf. Section 2.1), which is computationally challenging. We incorporate some techniques to accelerate the algorithm while providing comparable and in many cases better estimation quality than existing algorithms. Experimental results show the promise of universality in CS, particularly for low-complexity sources that do not exhibit standard sparsity or compressibility. This chapter is based on our work with Baron and Duarte [Zhu14; Zhu15].

5.1 Motivation and Contributions

Since many systems in science and engineering are approximately linear (1.1), linear inverse problems have attracted great attention in the signal processing community. Recall from (1.1) that an input signal $\mathbf{x} \in \mathbb{R}^N$ is recorded via a linear operator under additive noise:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \tag{5.1}$$

where $\mathbf{A}$ is an $M \times N$ matrix and $\mathbf{z} \in \mathbb{R}^M$ denotes the noise. The goal is to estimate $\mathbf{x}$ from the measurements $\mathbf{y}$ given knowledge of $\mathbf{A}$ and a model for the noise $\mathbf{z}$. When $M \ll N$, the setup is known as compressed sensing (CS) and the estimation problem is commonly referred to as recovery or reconstruction; by posing a sparsity or compressibility[1] requirement on the signal and using this requirement as a prior during estimation, it is indeed possible to accurately estimate $\mathbf{x}$ from $\mathbf{y}$ [Can06; Don06a]. On the other hand, we might need more measurements than the signal length when the signal is dense or the noise is substantial.

Wu and Verdú [WV12b] have shown that independent and identically distributed (i.i.d.) Gaussian sensing matrices achieve the same phase transition threshold as the optimal (potentially non-linear) measurement operator, for any i.i.d. signals following the discrete/continuous mixture distribution $f_X(x) = \rho \cdot f_c(x) + (1-\rho) \cdot \mathbb{P}_d(x)$, where ρ is the probability for a scalar x to take a continuous distribution $f_c(x)$ and $\mathbb{P}_d(x)$ is an arbitrary discrete distribution. For non-i.i.d. signals, Gaussian matrices also work well [Don13; Tan15; Ma14a]. Hence, in CS the acquisition can be designed independently of the particular signal prior through the use of randomized Gaussian matrices $\mathbf{A}$. Nevertheless, the majority of (if not all) existing estimation algorithms require knowledge of the sparsity structure of $\mathbf{x}$, i.e., the choice of a *sparsifying transform* $\mathbf{W}$ that renders a sparse coefficient vector $\theta = \mathbf{W}^{-1}\mathbf{x}$ for the signal.

The large majority of CS signal estimation algorithms pose a sparsity prior on the signal $\mathbf{x}$ or the coefficient vector θ, e.g., [Can06; Don06a; Fig07]. A second, separate class of Bayesian CS signal estimation algorithms poses a probabilistic prior for the coefficients of $\mathbf{x}$ in a known transform domain [Don10; Ran11; Ji08; SN08; Bar10]. Given a probabilistic model, some related message passing approaches learn the parameters of the signal model and achieve the minimum mean squared error (MMSE) in some settings; examples include EM-GM-AMP-MOS [VS13], turboGAMP [Zin12], and AMP-MixD [Ma14b]. As a third alternative, complexity-penalized least square methods [FN03; Don06b; HN06; HN12; RS12a] can use arbitrary prior information on the signal model and provide analytical guarantees, but are only computationally efficient for specific signal models, such as the independent-entry Laplacian model [HN06]. For example, Donoho et al. [Don06b] relies on

[1] We use the term compressibility in this chapter as defined by Candès et al. [Can06] to refer to signals whose sparse approximation error decays sufficiently quickly.

Kolmogorov complexity, which cannot be computed [CT06; LV08]. As a fourth alternative, there exist algorithms that can formulate dictionaries that yield sparse representations for the signals of interest when a large amount of training data is available [RS12a; Aha06; Mai08; Zho12]. When the signal is non-i.i.d., existing algorithms require either prior knowledge of the probabilistic model [Zin12] or the use of training data [GO07].

In certain cases, one might not be certain about the structure or statistics of the source prior to estimation. Uncertainty about such structure may result in a sub-optimal choice of the sparsifying transform $\mathbf{W}$, yielding a coefficient vector θ that requires more measurements to achieve reasonable estimation quality; uncertainty about the statistics of the source will make it difficult to select a prior or model for Bayesian algorithms. Thus, it would be desirable to formulate algorithms to estimate $\mathbf{x}$ that are more agnostic to the particular statistics of the signal. Therefore, we shift our focus from the standard sparsity or compressibility priors to *universal* priors [ZL77; Ris83; RS12b]. Such concepts have been previously leveraged in the Kolmogorov sampler universal denoising algorithm [Don02], which minimizes Kolmogorov complexity [Cha66; Sol64; Kol65; LV08; JM11; Jal14; Bar11; BD11]. Related approaches based on minimum description length (MDL) [Ris78; Sch78; WB68; Bar98] minimize the complexity of the estimated signal with respect to (w.r.t.) some class of sources.

Approaches for non-parametric sources based on Kolmogorov complexity are not computable in practice [CT06; LV08]. To address this computational problem, we confine our attention to the class of stationary ergodic sources and develop an algorithmic framework for *universal* signal estimation in CS systems that will approach the MMSE as closely as possible for the class of stationary ergodic sources. Our framework can be applied to general linear inverse problems where more measurements might be needed. Our framework leverages the fact that for stationary ergodic sources, both the per-symbol empirical entropy and Kolmogorov complexity converge asymptotically almost surely to the entropy rate of the source [CT06]. We aim to minimize the empirical entropy; our minimization is regularized by introducing a log likelihood for the noise model, which is equivalent to the standard least squares under additive white Gaussian noise. Other noise distributions are readily supported.

We make the following contributions toward our universal CS framework.

- We apply a specific quantization grid to a maximum *a posteriori* (MAP) estimator driven by a universal prior, providing a finite-computation universal estimation scheme; our scheme can also be applied to general linear inverse problems where more measurements might be needed.

- We propose an estimation algorithm based on Markov chain Monte Carlo (MCMC) [GG84] to approximate this estimation procedure.

- We prove that for a sufficiently large number of iterations the output of our MCMC estimation algorithm converges to the correct MAP estimate.

- We identify computational bottlenecks in the implementation of our MCMC estimator and show approaches to reduce their complexity.
- We develop an adaptive quantization scheme that tailors a set of reproduction levels to minimize the quantization error within the MCMC iterations and that provides an accelerated implementation.
- We propose a framework that adaptively adjusts the cardinality (size) of the adaptive quantizer to match the complexity of the input signal, in order to further reduce the quantization error and computation.
- We note in passing that averaging over the outputs of different runs of the same signal with the same measurements will yield lower mean squared error (MSE) for our proposed algorithm.

This chapter is organized as follows. Section 5.2 provides background content. Section 5.3 overviews MAP estimation, quantization, and introduces universal MAP estimation. Section 5.4 formulates an initial MCMC algorithm for universal MAP estimation, Section 5.5 describes several improvements to this initial algorithm, and Section 5.6 presents experimental results. We conclude in Section 5.8. The proof of our main theoretical result appears in Appendix C.

5.2 Background and Related Work

5.2.1 Compressed sensing

Consider the noisy measurement setup via a linear operator (5.1). The input signal $\mathbf{x} \in \mathbb{R}^N$ is generated by a stationary ergodic source X, and must be estimated from $\mathbf{y}$ and $\mathbf{A}$. Note that the stationary ergodicity assumption enables us to model the potential memory in the source. *The distribution $f_X(\cdot)$ that generates $\mathbf{x}$ is unknown.* The matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ has i.i.d. Gaussian entries, $A_{m,n} \sim \mathcal{N}(0, \frac{1}{M})$.[2] These moments ensure that the columns of the matrix have unit norm on average. For concrete analysis, we assume that the noise $\mathbf{z} \in \mathbb{R}^M$ is i.i.d. Gaussian, with mean zero and known[3] variance σ_Z^2 for simplicity.

We focus on the large system limit (cf. Definition 1.1 in Chapter 1). Similar settings have been discussed in the literature [Ran10; GW08]. When $M \ll N$, this setup is known as CS; otherwise, it is a general linear inverse problem setting. Since $\mathbf{x}$ is generated by an unknown source, we must search for an estimation mechanism that is agnostic to the specific distribution $f_X(\cdot)$.

[2] In contrast to our analytical and numerical results, the algorithm presented in Section 5.4 is not dependent on a particular choice for the matrix $\mathbf{A}$.

[3] We assume that the noise variance is known or can be estimated [Don10; Ma14b].

5.2.2 Related work

For a scalar channel with a discrete-valued signal $\mathbf{x}$, e.g., $\mathbf{A}$ is an identity matrix and $\mathbf{y}=\mathbf{x}+\mathbf{z}$, Donoho proposed the Kolmogorov sampler for denoising [Don02],

$$\mathbf{x}_{KS} \triangleq \arg\min_{\mathbf{w}} K(\mathbf{w}), \text{ subject to } \|\mathbf{w}-\mathbf{y}\|^2 < \tau, \tag{5.2}$$

where $K(\mathbf{x})$ denotes the Kolmogorov complexity of $\mathbf{x}$, defined as the length of the shortest input to a Turing machine [Tur50] that generates the output $\mathbf{x}$ and then halts,[4] and $\tau = N\sigma_Z^2$ controls for the presence of noise. It can be shown that $K(\mathbf{x})$ asymptotically captures the statistics of the stationary ergodic source X, and the per-symbol complexity achieves the entropy rate $H \triangleq H(X)$, i.e., $\lim_{N\to\infty} \frac{1}{N} K(\mathbf{x}) = H$ almost surely [[CT06], p. 154, Theorem 7.3.1]. Noting that universal lossless compression algorithms [ZL77; Ris83] achieve the entropy rate for any discrete-valued finite state machine source X, we see that these algorithms achieve the per-symbol Kolmogorov complexity almost surely.

Donoho et al. expanded Kolmogorov sampler to the linear CS measurement setting $\mathbf{y} = \mathbf{Ax}$ but did not consider measurement noise [Don06b]. Recent papers by Jalali and coauthors [JM11; Jal14], which appeared simultaneously with Baron [Bar11] and Baron and Duarte [BD11], provide an analysis of a modified Kolmogorov sampler suitable for measurements corrupted by noise of bounded magnitude. Inspired by Donoho et al. [Don06b], we estimate $\mathbf{x}$ from noisy measurements $\mathbf{y}$ using the empirical entropy as a proxy for the Kolmogorov complexity (cf. Section 5.4.1).

Separate notions of complexity-penalized least squares have also been shown to be well suited for denoising and CS signal estimation [FN03; Don06b; Ris78; Sch78; WB68; HN06; HN12; RS12a]. For example, minimum description length (MDL) [Ris78; Sch78; WB68; RS12a] provides a framework composed of classes of models for which the signal complexity can be defined sharply. In general, complexity-penalized least square approaches can yield MDL-flavored CS signal estimation algorithms that are adaptive to parametric classes of sources [Don06b; FN03; HN06; HN12]. An alternative universal denoising approach computes the universal conditional expectation of the signal [Bar11; Ma14b].

5.3 Universal MAP Estimation and Discretization

This section briefly reviews MAP estimation and then applies it over a quantization grid, where a universal prior is used for the signal. Additionally, we provide a conjecture for the MSE achieved by our universal MAP scheme.

[4] For real-valued $\mathbf{x}$, Kolmogorov complexity can be approximated using a fine quantizer. Note that the algorithm developed in this chapter uses a coarse quantizer and does not rely on Kolmogorov complexity due to the absence of a feasible method for its computation [CT06; LV08] (cf. Section 5.5).

5.3.1 Discrete MAP estimation

In this subsection, we assume for exposition purposes that we know the signal distribution $f_X(\cdot)$. Given the measurements $\mathbf{y}$, the MAP estimator for $\mathbf{x}$ has the form

$$\mathbf{x}_{MAP} \triangleq \arg\max_{\mathbf{w}} f_X(\mathbf{w}) f_{Y|X}(\mathbf{y}|\mathbf{w}). \tag{5.3}$$

Because $\mathbf{z}$ is i.i.d. Gaussian with mean zero and known variance σ_Z^2,

$$f_{Y|X}(\mathbf{y}|\mathbf{w}) = c_1\, \mathrm{e}^{-c_2\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2},$$

where $c_1 = (2\pi\sigma_Z^2)^{-M/2}$ and $c_2 = \frac{1}{2\sigma_Z^2}$ are constants, and $\|\cdot\|$ denotes the Euclidean norm.[5] Plugging into (5.3) and taking log likelihoods, we obtain $\mathbf{x}_{MAP} = \arg\min_{\mathbf{w}} \Psi^X(\mathbf{w})$, where $\Psi^X(\cdot)$ denotes the objective function (risk)

$$\Psi^X(\mathbf{w}) \triangleq -\ln(f_X(\mathbf{w})) + c_2\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2;$$

our ideal risk would be $\Psi^X(\mathbf{x}_{MAP})$.

Instead of performing continuous-valued MAP estimation, we optimize for the MAP in the discretized domain $\mathscr{R}^N$, with $\mathscr{R}$ being defined as follows. Adapting the approach of Baron and Weissman [BW12], we define the set of data-independent reproduction levels for quantizing $\mathbf{x}$ as

$$\mathscr{R} \triangleq \left\{\cdots, -\frac{1}{\gamma}, 0, \frac{1}{\gamma}, \cdots\right\}, \tag{5.4}$$

where $\gamma = \lceil \ln(N) \rceil$. As N increases, $\mathscr{R}$ will quantize $\mathbf{x}$ to a greater resolution. These reproduction levels simplify the estimation problem from continuous to discrete.

Having discussed our reproduction levels in the set $\mathscr{R}$, we provide a technical condition on boundedness of the signal.

Condition 5.1. *We require that the probability density $f_X(\cdot)$ has bounded support, i.e., there exists $\Lambda = [x_{min}, x_{max}]$ such that (s.t.) $f_X(\mathbf{x}) = 0$ for $\mathbf{x} \notin \Lambda^N$.*

A limitation of the data-independent reproduction level set (5.4) is that $\mathscr{R}$ has infinite cardinality (or size for short). Thanks to Condition 5.1, for each value of γ there exists a constant $c_3 > 0$ s.t. a finite set of reproduction levels

$$\mathscr{R}_F \triangleq \left\{-\frac{c_3\gamma^2}{\gamma}, -\frac{c_3\gamma^2 - 1}{\gamma}, \cdots, \frac{c_3\gamma^2}{\gamma}\right\} \tag{5.5}$$

will quantize the range of values Λ to the same accuracy as that of (5.4). We call $\mathscr{R}_F$ the *repro-*

[5] Other noise distributions are readily supported, e.g., for i.i.d. Laplacian noise, we need to change the ℓ_2 norm to an ℓ_1 norm and adjust c_1 and c_2 accordingly.

duction alphabet, and each element in it a (*reproduction*) *level*. This finite quantizer reduces the complexity of the estimation problem from infinite to combinatorial. In fact, $x_i \in [x_{\min}, x_{\max}]$ under Condition 5.1. Therefore, for all $c_3 > 0$ and sufficiently large N, this set of levels will cover the range $[x_{\min}, x_{\max}]$. The resulting reduction in complexity is due to the structure in $\mathscr{R}_F$ and independent of the particular statistics of the source X.

Now that we have set up a quantization grid $(\mathscr{R}_F)^N$ for $\mathbf{x}$, we convert the distribution $f_X(\cdot)$ to a probability mass function (PMF) $\mathbb{P}_X(\cdot)$ over $(\mathscr{R}_F)^N$. Let $f_{\mathscr{R}_F} \triangleq \sum_{\mathbf{w}\in(\mathscr{R}_F)^N} f_X(\mathbf{w})$, and define a PMF $\mathbb{P}_X(\cdot)$ as $\mathbb{P}_X(\mathbf{w}) \triangleq \frac{f_X(\mathbf{w})}{f_{\mathscr{R}_F}}$. Then,

$$\mathbf{x}_{MAP}(\mathscr{R}_F) \triangleq \arg\min_{\mathbf{w}\in(\mathscr{R}_F)^N} \left[-\ln(\mathbb{P}_X(\mathbf{w})) + c_2\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2\right]$$

gives the MAP estimate of $\mathbf{x}$ over $(\mathscr{R}_F)^N$. Note that we use the PMF formulation above, instead of the more common bin integration formulation, in order to simplify our presentation and analysis. Luckily, as N increases, $\mathbb{P}_X(\cdot)$ will approximate $f_X(\cdot)$ more closely under (5.5).

5.3.2 Universal MAP estimation

We now describe a universal estimator for CS over a quantized grid. Consider a prior $\mathbb{P}_U(\cdot)$ that might involve Kolmogorov complexity [Cha66; Sol64; Kol65], e.g., $\mathbb{P}_U(\mathbf{w}) = 2^{-K(\mathbf{w})}$, or MDL complexity w.r.t. some class of parametric sources [Ris78; Sch78; WB68]. We call $\mathbb{P}_U(\cdot)$ a *universal prior* if it has the fortuitous property that for every stationary ergodic source X and fixed $\epsilon > 0$, there exists some minimum $N_0(X,\epsilon)$ s.t.

$$-\frac{\ln(\mathbb{P}_U(\mathbf{w}))}{N} < -\frac{\ln(\mathbb{P}_X(\mathbf{w}))}{N} + \epsilon$$

for all $\mathbf{w} \in (\mathscr{R}_F)^N$ and $N > N_0(X,\epsilon)$ [ZL77; Ris83]. We optimize over an objective function that incorporates $\mathbb{P}_U(\cdot)$ and the presence of additive white Gaussian noise in the measurements:

$$\Psi^U(\mathbf{w}) \triangleq -\ln(\mathbb{P}_U(\mathbf{w})) + c_2\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2, \tag{5.6}$$

resulting in[6] $\mathbf{x}^U_{MAP} \triangleq \arg\min_{\mathbf{w}\in(\mathscr{R}_F)^N} \Psi^U(\mathbf{w})$. Our universal MAP estimator does not require $M \ll N$, and $\mathbf{x}^U_{MAP}$ can be used in general linear inverse problems.

5.3.3 Conjectured MSE performance

Donoho [Don02] showed for the scalar channel $\mathbf{y} = \mathbf{x} + \mathbf{z}$ that: (i) the Kolmogorov sampler $\mathbf{x}_{KS}$ (5.2) is drawn from the posterior distribution $\mathbb{P}_{X|Y}(\mathbf{x}|\mathbf{y})$; and (ii) the MSE of this estimate $\mathbb{E}_{X,Z,\mathbf{A}}[\|\mathbf{y}-$

[6] This formulation of $\mathbf{x}^U_{MAP}$ corresponds to a Lagrangian relaxation of the approach studied in [JM11; Jal14].

$\mathbf{x}_{KS}||^2]$ is no greater than twice the MMSE. Based on this result, which requires a large reproduction alphabet, we now present a conjecture on the quality of the estimate $\mathbf{x}^U_{MAP}$. Our conjecture is based on observing that (*i*) in the setting (5.1), Kolmogorov sampling achieves optimal rate-distortion performance; (*ii*) the Bayesian posterior distribution is the solution to the rate-distortion problem; and (*iii*) sampling from the Bayesian posterior yields a squared error that is no greater than twice the MMSE. Hence, $\mathbf{x}^U_{MAP}$ behaves as if we sample from the Bayesian posterior distribution and yields no greater than twice the MMSE; some experimental evidence to assess this conjecture is presented in Figures 5.2 and 5.4.

Conjecture 5.1. *Assume that* $\mathbf{A} \in \mathbb{R}^{M\times N}$ *is an i.i.d. Gaussian measurement matrix where each entry has mean zero and variance* $\frac{1}{M}$. *Suppose that Condition 5.1 holds, the aspect ratio* $\kappa = \frac{M}{N}$, *and the noise* $\mathbf{z} \in \mathbb{R}^M$ *is i.i.d. zero-mean Gaussian with finite variance. Then for all* $\epsilon > 0$, *the mean squared error of the universal MAP estimator* $\mathbf{x}^U_{MAP}$ *satisfies*

$$\frac{\mathbb{E}_{X,Z,\mathbf{A}}\left[||\mathbf{x}-\mathbf{x}^U_{MAP}||^2\right]}{N} < \frac{2\mathbb{E}_{X,Z,\mathbf{A}}\left[||\mathbf{x}-\mathbb{E}_X[\mathbf{x}|\mathbf{y},\mathbf{A}]||^2\right]}{N} + \epsilon$$

for sufficiently large N.

5.4 Fixed Reproduction Alphabet Algorithm

Although the results of the previous section are theoretically appealing, a brute force optimization of $\mathbf{x}^U_{MAP}$ is computationally intractable. Instead, we propose an algorithmic approach based on MCMC methods [GG84]. Our approach is reminiscent of the framework for lossy data compression [JW08; JW12; BW12; Yan97].

5.4.1 Universal compressor

We propose a universal lossless compression formulation following the conventions of Weissman and coauthors [JW08; JW12; BW12]. We refer to the estimate as $\mathbf{w}$ in our algorithm. Our goal is to characterize $-\ln(\mathbb{P}_U(\mathbf{w}))$, cf. (5.6). Although we are inspired by the Kolmogorov sampler approach [Don02], Kolmogorov complexity cannot be computed [CT06; LV08], and we instead use empirical entropy. For stationary ergodic sources, the empirical entropy converges to the per-symbol entropy rate almost surely [CT06].

To define the empirical entropy, we first define the empirical symbol counts:

$$n_q(\mathbf{w},\alpha)[\beta] \triangleq \left|\{i \in [q+1,N] : \mathbf{w}^{i-1}_{i-q} = \alpha, w_i = \beta\}\right|, \tag{5.7}$$

where q is the context depth [Ris83; Wil95], $\beta \in \mathscr{R}_F$, $\alpha \in (\mathscr{R}_F)^q$, w_i is the i-th symbol of $\mathbf{w}$, and $\mathbf{w}^j_i$ is the string comprising symbols i through j within $\mathbf{w}$. We now define the order q conditional empirical probability for the context α as

$$\mathbb{P}_q(\mathbf{w},\alpha)[\beta] \triangleq \frac{n_q(\mathbf{w},\alpha)[\beta]}{\sum_{\beta'\in\mathscr{R}_F} n_q(\mathbf{w},\alpha)[\beta']}, \tag{5.8}$$

and the order q conditional empirical entropy,[7]

$$H_q(\mathbf{w}) \triangleq -\frac{1}{N} \sum_{\alpha\in(\mathscr{R}_F)^q,\beta\in\mathscr{R}_F} n_q(\mathbf{w},\alpha)[\beta] \log_2\left(\mathbb{P}_q(\mathbf{w},\alpha)[\beta]\right), \tag{5.9}$$

where the sum is only over non-zero counts and probabilities.

Allowing the context depth $q \triangleq q_N = o(\log(N))$ to grow slowly with N, various universal compression algorithms can achieve the empirical entropy $H_q(\cdot)$ asymptotically [Ris83; Wil95; ZL77]. On the other hand, no compressor can outperform the entropy rate. Additionally, for large N, the empirical symbol counts with context depth q provide a sufficiently precise characterization of the source statistics. Therefore, H_q provides a concise approximation to the per-symbol coding length of a universal compressor.

5.4.2 Markov chain Monte Carlo

Having approximated the coding length, we now describe how to optimize our objective function. We define the energy $\Psi^{H_q}(\mathbf{w})$ in an analogous manner to $\Psi^U(\mathbf{w})$ (5.6), using $H_q(\mathbf{w})$ as our universal coding length:

$$\Psi^{H_q}(\mathbf{w}) \triangleq N H_q(\mathbf{w}) + c_4\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2, \tag{5.10}$$

where $c_4 = c_2\log_2(\mathrm{e})$. The minimization of this energy is analogous to minimizing $\Psi^U(\mathbf{w})$.

Ideally, our goal is to compute the globally minimum energy solution $\mathbf{x}_{MAP}^{H_q} \triangleq \arg\min_{\mathbf{w}\in(\mathscr{R}_F)^N} \Psi^{H_q}(\mathbf{w})$. We use a stochastic MCMC relaxation [GG84] to achieve the globally minimum solution in the limit of infinite computation. To assist the reader in appreciating how MCMC is used to compute $\mathbf{x}_{MAP}^{H_q}$, we include pseudocode for our approach in Algorithm 5.1. The algorithm, called basic MCMC (B-MCMC), will be used as a building block for our latter Algorithms 5.2 and 4 in Section 5.5. The initial estimate $\mathbf{w}$ is obtained by quantizing the *initial point* $\mathbf{x}^* \in \mathbb{R}^N$ to $(\mathscr{R}_F)^N$. The initial point $\mathbf{x}^*$ could be the output of any CS signal estimation algorithm, and because $\mathbf{x}^*$ is a preliminary estimate of the signal that does not require high fidelity, we let $\mathbf{x}^* = \mathbf{A}^\top\mathbf{y}$ for simplicity, where $\{\cdot\}^\top$ denotes transpose. We refer to the processing of a single entry of $\mathbf{w}$ as an iteration and group the processing of all entries of $\mathbf{w}$, randomly permuted, into super-iterations.

The Boltzmann PMF for a thermodynamic system was defined in (2.2). Similarly, we define the

[7] Interested readers can refer to the definitions of entropy for thermodynamics and information theory in (2.1) and (2.8), respectively.

Boltzmann PMF for the energy $\Psi^{H_q}(\mathbf{w})$ (5.10) as

$$\mathbb{P}_s(\mathbf{w}) \triangleq \frac{1}{\zeta_s} \exp\left(-s\Psi^{H_q}(\mathbf{w})\right), \tag{5.11}$$

where $s > 0$ is inversely related to the temperature in simulated annealing and ζ_s is a normalization constant. MCMC samples from the Boltzmann PMF (5.11) using a *Gibbs sampler*: in each iteration, a single element w_n is generated while the rest of $\mathbf{w}$, $\mathbf{w}^{\backslash n} \triangleq \{w_i : n \neq i\}$, remains unchanged. We denote by $\mathbf{w}_1^{n-1}\beta\mathbf{w}_{n+1}^N$ the concatenation of the initial portion of the output vector $\mathbf{w}_1^{n-1}$, the symbol $\beta \in \mathscr{R}_F$, and the latter portion of the output $\mathbf{w}_{n+1}^N$. The Gibbs sampler updates w_n by resampling from the PMF:

$$\begin{aligned} \mathbb{P}_s(w_n = a|\mathbf{w}^{\backslash n}) &= \frac{\exp\left(-s\Psi^{H_q}(\mathbf{w}_1^{n-1}a\mathbf{w}_{n+1}^N)\right)}{\sum_{b\in\mathscr{R}_F}\exp\left(-s\Psi^{H_q}(\mathbf{w}_1^{n-1}b\mathbf{w}_{n+1}^N)\right)} \\ &= \frac{1}{\sum_{b\in\mathscr{R}_F}\exp\left[-s\left(N\Delta H_q(\mathbf{w},n,b,a)+c_4\Delta d(\mathbf{w},n,b,a)\right)\right]}, \end{aligned}$$

where

$$\Delta H_q(\mathbf{w},n,b,a) \triangleq H_q(\mathbf{w}_1^{n-1}b\mathbf{w}_{n+1}^N) - H_q(\mathbf{w}_1^{n-1}a\mathbf{w}_{n+1}^N)$$

is the change in empirical entropy $H_q(\mathbf{w})$ (5.9) when $w_n = a$ is replaced by b, and

$$\Delta d(\mathbf{w},n,b,a) \triangleq \|\mathbf{y}-\mathbf{A}(\mathbf{w}_1^{n-1}b\mathbf{w}_{n+1}^N)\|^2 - \|\mathbf{y}-\mathbf{A}(\mathbf{w}_1^{n-1}a\mathbf{w}_{n+1}^N)\|^2 \tag{5.12}$$

is the change in $\|\mathbf{y}-\mathbf{A}\mathbf{w}\|^2$ when $w_n = a$ is replaced by b. The maximum change in the energy within an iteration of Algorithm 5.1 is then bounded by

$$\Delta_q = \max_{1\le n\le N}\max_{\mathbf{w}\in(\mathscr{R}_F)^N}\max_{a,b\in\mathscr{R}_F}\left|N\Delta H_q(\mathbf{w},n,b,a)+c_4\Delta d(\mathbf{w},n,b,a)\right|. \tag{5.13}$$

Note that $\mathbf{x}$ is assumed bounded (cf. Condition 5.1) so that (5.12–5.13) are bounded as well.

In MCMC, the space $\mathbf{w} \in (\mathscr{R}_F)^N$ is analogous to a thermodynamic system, and at low temperatures the system tends toward low energies. Therefore, during the execution of the algorithm, we set a sequence of decreasing temperatures that takes into account the maximum change given in (5.13):

$$s_t \triangleq \ln(t + r_0)/(cN\Delta_q) \text{ for some } c > 1, \tag{5.14}$$

where r_0 is a temperature offset. At low temperatures, i.e., large s_t, a small difference in energy $\Psi^{H_q}(\mathbf{w})$ drives a big difference in probability, cf. (5.11). Therefore, we begin at a high temperature where the Gibbs sampler can freely move around $(\mathscr{R}_F)^N$. As the temperature is reduced, the PMF

Algorithm 5.1 Basic MCMC for universal CS – Fixed alphabet

```
**Inputs**: Initial estimate $\mathbf{w}$, reproduction alphabet $\mathscr{R}_F$, noise variance $\sigma_Z^2$, number of super-iterations $r$, temperature constant $c > 1$, and context depth $q$
Compute $n_q(\mathbf{w}, \alpha)[\beta]$, $\forall\ \alpha \in (\mathscr{R}_F)^q$, $\beta \in \mathscr{R}_F$
**for** $t = 1$ to $r$ **do**                                          ▷ super-iteration
    $s \leftarrow \ln(t)/(cN\Delta_q)$                                 ▷ $s = s_t$, cf. (5.14)
    Draw permutation $\{1, \cdots, N\}$ at random
    **for** $t' = 1$ to $N$ **do**                                      ▷ iteration
        Let $n$ be component $t'$ in permutation
        **for** all $\beta$ in $\mathscr{R}_F$ **do**                   ▷ possible new $w_n$
            Compute $\Delta H_q(\mathbf{w}, n, \beta, w_n)$
            Compute $\Delta d(\mathbf{w}, n, \beta, w_n)$
            Compute $\mathbb{P}_s(w_n = \beta | \mathbf{w}^{\backslash n})$
        **end for**
        Generate $w_n$ using $\mathbb{P}_s(\cdot | \mathbf{w}^{\backslash n})$          ▷ Gibbs
        Update $n_q(\mathbf{w}, \alpha)[\beta]$, $\forall\ \alpha \in (\mathscr{R}_F)^q$, $\beta \in \mathscr{R}_F$
    **end for**
**end for**
**Output:** Return approximation $\mathbf{w}$ of $\mathbf{x}_{MAP}^U$
```

becomes more sensitive to changes in energy (5.11), and the trend toward **w** with lower energy grows stronger. In each iteration, the Gibbs sampler modifies w_n in a random manner that resembles heat bath concepts in thermodynamics. Although MCMC could sink into a local minimum, Geman and Geman [GG84] proved that if we decrease the temperature according to (5.14), then the randomness of Gibbs sampling will eventually drive MCMC out of the locally minimum energy and it will converge to the globally optimal energy w.r.t. $\mathbf{x}_{MAP}^U$. Note that Geman and Geman proved that MCMC will converge, although the proof states that it will take infinitely long to do so. In order to help B-MCMC approach the global minimum with reasonable runtime, we will refine B-MCMC in Section 5.5.

The following theorem is proven in Appendix C.1, following the framework established by Jalali and Weissman [JW08; JW12].

Theorem 5.1. *Let X be a stationary ergodic source that obeys Condition 5.1. Then the outcome $\mathbf{w}^r$ of Algorithm 5.1 in the limit of an infinite number of super-iterations r obeys*

$$\lim_{r\to\infty} \Psi^{H_q}(\mathbf{w}^r) = \min_{\widetilde{\mathbf{w}}\in(\mathscr{R}_F)^N} \Psi^{H_q}(\widetilde{\mathbf{w}}) = \Psi^{H_q}\left(\mathbf{x}_{MAP}^{H_q}\right).$$

Theorem 5.1 shows that Algorithm 5.1 matches the best-possible performance of the universal MAP estimator as measured by the objective function Ψ^{H_q}, which should yield an MSE that is twice the MMSE (cf. Conjecture 5.1). We want to remind the reader that Theorem 5.1 is based on the stationarity and ergodicity of the source, which could have memory. To gain some insight about the convergence process of MCMC, we focus on a fixed arbitrary sub-optimal sequence $\mathbf{w} \in (\mathscr{R}_F)^N$. Suppose that at super-iteration t the energy for the algorithm's output $\Psi^{H_q}(\mathbf{w})$ has converged to the steady state (see Appendix C.1 for details on convergence). We can then focus on the probability ratio $\rho_t = \mathbb{P}_{s_t}(\mathbf{w})/\mathbb{P}_{s_t}\left(\mathbf{x}_{MAP}^{H_q}\right)$; $\rho_t < 1$ because $\mathbf{x}_{MAP}^{H_q}$ is the global minimum and has the largest Boltzmann probability over all $\mathbf{w} \in (\mathscr{R}_F)^N$, whereas $\mathbf{w}$ is sub-optimal. We then consider the same sequence $\mathbf{w}$ at super-iteration t^2; the inverse temperature is $2s_t$ and the corresponding ratio at super-iteration t^2 is (cf. (5.11))

$$\frac{\mathbb{P}_{2s_t}(\mathbf{w})}{\mathbb{P}_{2s_t}\left(\mathbf{x}_{MAP}^{H_q}\right)} = \frac{\exp\left(-2s_t\Psi^{H_q}(\mathbf{w})\right)}{\exp\left(-2s_t\Psi^{H_q}\left(\mathbf{x}_{MAP}^{H_q}\right)\right)} = \left(\frac{\mathbb{P}_{s_t}(\mathbf{w})}{\mathbb{P}_{s_t}\left(\mathbf{x}_{MAP}^{H_q}\right)}\right)^2.$$

That is, between super-iterations t and t^2 the probability ratio ρ_t is also squared, and the Gibbs sampler is less likely to generate samples whose energy differs significantly from the minimum energy w.r.t. $\mathbf{x}_{MAP}^{H_q}$. We infer from this argument that the probability concentration of our algorithm around the globally optimal energy w.r.t. $\mathbf{x}_{MAP}^{H_q}$ is linear in the number of super-iterations.

5.4.3 Computational challenges

Studying the pseudocode of Algorithm 5.1, we recognize that Lines 9–11 must be implemented efficiently, as they run $rN|\mathscr{R}_F|$ times. Lines 9 and 10 are especially challenging.

For Line 9, a naive update of $H_q(\mathbf{w})$ has complexity $O(|\mathscr{R}_F|^{q+1})$, cf. (5.9). To address this problem, Jalali and Weissman [JW08; JW12] recompute the empirical conditional entropy in $O(q|\mathscr{R}_F|)$ time only for the $O(q)$ contexts whose corresponding counts are modified [JW08; JW12]. The same approach can be used in Line 14, again reducing computation from $O(|\mathscr{R}_F|^{q+1})$ to $O(q|\mathscr{R}_F|)$. Some straightforward algebra allows us to convert Line 10 to a form that requires aggregate runtime of $O(Nr(M+|\mathscr{R}_F|))$. Combined with the computation for Line 9, and since $M \gg q|\mathscr{R}_F|^2$ (because $|\mathscr{R}_F| = \gamma^2, \gamma = \lceil \ln(N) \rceil, q = o(\log(N))$, and $M = O(N)$) in practice, the entire runtime of our algorithm is $O(rMN)$.

The practical value of Algorithm 5.1 may be reduced due to its high computational cost, dictated by the number of super-iterations r required for convergence to $\mathbf{x}_{MAP}^{H_q}$ and the large size of the reproduction alphabet. Nonetheless, Algorithm 5.1 provides a starting point toward further performance gains of more practical algorithms for computing $\mathbf{x}_{MAP}^{H_q}$, which are presented in Section 5.5. Furthermore, our experiments in Section 5.6 will show that the performance of the algorithm of Section 5.5 is comparable to and in many cases better than existing algorithms.

5.5 Adaptive Reproduction Alphabet

While Algorithm 5.1 is a first step toward universal signal estimation in CS, N must be large enough to ensure that $\mathscr{R}_F$ quantizes a broad enough range of values of $\mathbb{R}$ finely enough to represent the estimate $\mathbf{x}_{MAP}^{H_q}$ well. For large N, the estimation performance using the reproduction alphabet (5.5) could suffer from high computational complexity. On the other hand, for small N the number of reproduction levels employed is insufficient to obtain acceptable performance. Nevertheless, using an excessive number of levels will slow down the convergence. Therefore, in this section, we explore techniques that tailor the reproduction alphabet adaptively to the signal being observed.

5.5.1 Adaptivity in reproduction levels

To estimate better with finite N, we utilize reproduction levels that are *adaptive* instead of the fixed levels in $\mathscr{R}_F$. To do so, instead of $\mathbf{w} \in (\mathscr{R}_F)^N$, we optimize over a sequence $\mathbf{u} \in \mathscr{Z}^N$, where $|\mathscr{Z}| < |\mathscr{R}_F|$ and $|\cdot|$ denotes the size. The new reproduction alphabet $\mathscr{Z}$ does not directly correspond to real numbers. Instead, there is an adaptive mapping $\mathscr{A} : \mathscr{Z} \to \mathbb{R}$, and the reproduction levels are $\mathscr{A}(\mathscr{Z})$. Therefore, we call $\mathscr{Z}$ the *adaptive* reproduction alphabet. Since the mapping $\mathscr{A}$ is one-to-one, we also refer to $\mathscr{Z}$ as reproduction levels. Considering the energy function (5.10), we now compute the empirical symbol counts $n_q(\mathbf{u}, \alpha)[\beta]$, order q conditional empirical probabilities $\mathbb{P}_q(\mathbf{u}, \alpha)[\beta]$, and order q conditional empirical entropy $H_q(\mathbf{u})$ using $\mathbf{u} \in \mathscr{Z}^N$, $\alpha \in \mathscr{Z}^q$, and $\beta \in \mathscr{Z}$, cf. (5.7), (5.8), and (5.9). Similarly, we use $\|\mathbf{y} - \mathbf{A}\mathscr{A}(\mathbf{u})\|^2$ instead of $\|\mathbf{y} - \mathbf{A}\mathbf{w}\|^2$, where $\mathscr{A}(\mathbf{u})$ is the straightforward vector extension of $\mathscr{A}$. These modifications yield an adaptive energy function $\Psi_a^{H_q}(\mathbf{u}) \triangleq N H_q(\mathbf{u}) + c_4 \|\mathbf{y} - \mathbf{A}\mathscr{A}(\mathbf{u})\|^2$.

We choose $\mathscr{A}_{opt}$ to optimize for minimum squared error,

$$\mathscr{A}_{opt} \triangleq \arg\min_{\mathscr{A}} \|\mathbf{y} - \mathbf{A}\mathscr{A}(\mathbf{u})\|^2 = \arg\min_{\mathscr{A}} \left[\sum_{m=1}^{M} (y_m - [\mathbf{A}\mathscr{A}(\mathbf{u})]_m)^2 \right],$$

where $[\mathbf{A}\mathscr{A}(\mathbf{u})]_m$ denotes the m-th entry of the vector $\mathbf{A}\mathscr{A}(\mathbf{u})$. The optimal mapping depends entirely on $\mathbf{y}$, $\mathbf{A}$, and $\mathbf{u}$. From a coding perspective, describing $\mathscr{A}_{opt}(\mathbf{u})$ requires $H_q(\mathbf{u})$ bits for $\mathbf{u}$ and $|\mathscr{Z}| b \log\log(N)$ bits for $\mathscr{A}_{opt}$ to match the resolution of the non-adaptive $\mathscr{R}_F$, with $b > 1$ an arbitrary constant [BW12]. The resulting coding length defines our universal prior.

Optimization of reproduction levels: We now describe the optimization procedure for $\mathscr{A}_{opt}$, which must be computationally efficient. Write

$$\Upsilon(\mathscr{A}) \triangleq \|\mathbf{y} - \mathbf{A}\mathscr{A}(\mathbf{u})\|^2 = \sum_{m=1}^{M} \left(y_m - \sum_{n=1}^{N} A_{mn} \mathscr{A}(u_n) \right)^2,$$

where A_{mn} is the entry of $\mathbf{A}$ at the m-th row, n-th column. For $\Upsilon(\mathscr{A})$ to be minimum, we need zero-valued derivatives as follows,

$$\frac{d\Upsilon(\mathscr{A})}{d\mathscr{A}(\beta)} = -2\sum_{m=1}^{M}\left(y_m - \sum_{n=1}^{N} A_{mn}\mathscr{A}(u_n)\right)\left(\sum_{n=1}^{N} A_{mn}\mathbb{1}_{u_n=\beta}\right) = 0,\ \forall\ \beta \in \mathscr{X},$$

where the indicator function $\mathbb{1}_A$ is 1 if the condition A is met, else 0. Define the location sets $\mathscr{L}_\beta \triangleq \{n : 1 \le n \le N, u_n = \beta\}$ for each $\beta \in \mathscr{X}$, and rewrite the derivatives of $\Upsilon(\mathscr{A})$,

$$\frac{d\Upsilon(\mathscr{A})}{d\mathscr{A}(\beta)} = -2\sum_{m=1}^{M}\left(y_m - \sum_{\lambda\in\mathscr{X}}\sum_{n\in\mathscr{L}_\lambda} A_{mn}\mathscr{A}(\lambda)\right)\left(\sum_{n\in\mathscr{L}_\beta} A_{mn}\right). \tag{5.15}$$

Let the per-character sum column values be

$$\mu_{m\beta} \triangleq \sum_{n\in\mathscr{L}_\beta} A_{mn}, \tag{5.16}$$

for each $m \in \{1, \cdots, M\}$ and $\beta \in \mathscr{X}$. We desire the derivatives to be zero, cf. (5.15):

$$0 = \sum_{m=1}^{M}\left(y_m - \sum_{\lambda\in\mathscr{X}} \mathscr{A}(\lambda)\mu_{m\lambda}\right)\mu_{m\beta}.$$

Thus, the system of equations must be satisfied,

$$\sum_{m=1}^{M} y_m\mu_{m\beta} = \sum_{m=1}^{M}\left(\sum_{\lambda\in\mathscr{X}} \mathscr{A}(\lambda)\mu_{m\lambda}\right)\mu_{m\beta} \tag{5.17}$$

for each $\beta \in \mathscr{X}$. Consider now the right hand side,

$$\sum_{m=1}^{M}\left(\sum_{\lambda\in\mathscr{X}} \mathscr{A}(\lambda)\mu_{m\lambda}\right)\mu_{m\beta} = \sum_{\lambda\in\mathscr{X}} \mathscr{A}(\lambda)\sum_{m=1}^{M} \mu_{m\lambda}\mu_{m\beta},$$

for each $\beta \in \mathscr{X}$. The system of equations can be described in matrix form as follows,

$$\overbrace{\begin{bmatrix} \sum_{m=1}^{M}\mu_{m\beta_1}\mu_{m\beta_1} & \cdots & \sum_{m=1}^{M}\mu_{m\beta_{|\mathscr{X}|}}\mu_{m\beta_1} \\ \vdots & \ddots & \vdots \\ \sum_{m=1}^{M}\mu_{m\beta_1}\mu_{m\beta_{|\mathscr{X}|}} & \cdots & \sum_{m=1}^{M}\mu_{m\beta_{|\mathscr{X}|}}\mu_{m\beta_{|\mathscr{X}|}} \end{bmatrix}}^{\Omega} \overbrace{\begin{bmatrix} \mathscr{A}(\beta_1) \\ \vdots \\ \mathscr{A}(\beta_{|\mathscr{X}|}) \end{bmatrix}}^{\mathscr{A}(\mathscr{X})} = \overbrace{\begin{bmatrix} \sum_{m=1}^{M} y_m\mu_{m\beta_1} \\ \vdots \\ \sum_{m=1}^{M} y_m\mu_{m\beta_{|\mathscr{X}|}} \end{bmatrix}}^{\Theta}.$$

Note that by writing μ as a matrix with entries indexed by row m and column β given by (5.16), we can write Ω as a Gram matrix, $\Omega = \mu^\top\mu$, and we also have $\Theta = \mu^\top\mathbf{y}$, cf. (5.17). The optimal $\mathscr{A}$ can be computed as a $|\mathscr{X}| \times 1$ vector $\mathscr{A}_{opt} = \Omega^{-1}\Theta = (\mu^\top\mu)^{-1}\mu^\top\mathbf{y}$ if $\Omega \in \mathbb{R}^{|\mathscr{X}|\times|\mathscr{X}|}$ is invertible. We note in

Algorithm 5.2 Level-adaptive MCMC

```
1: *Inputs: Initial mapping $\mathscr{A}$, sequence $\mathbf{u}$, adaptive alphabet $\mathscr{X}$, noise variance $\sigma_Z^2$, number of
   super-iterations $r$, temperature constant $c > 1$, context depth $q$, and temperature offset $r_0$
2: Compute $n_q(\mathbf{u}, \alpha)[\beta]$, $\forall\ \alpha \in \mathscr{X}^q$, $\beta \in \mathscr{X}$
3: *Initialize $\Omega$
4: for $t = 1$ to $r$ do                                                    ▷ super-iteration
5:     $s \leftarrow \ln(t + r_0)/(cN\Delta_q)$                                  ▷ $s = s_t$, cf. (5.14)
6:     Draw permutation $\{1, \cdots, N\}$ at random
7:     for $t' = 1$ to $N$ do                                               ▷ iteration
8:         Let $n$ be component $t'$ in permutation
9:         for all $\beta$ in $\mathscr{X}$ do                                  ▷ possible new $u_n$
10:            Compute $\Delta H_q(\mathbf{u}, n, \beta, u_n)$
11:            *Compute $\mu_{m\beta}, \forall\ m \in \{1, \cdots, M\}$
12:            *Update $\Omega$                                         ▷ $O(1)$ rows and columns
13:            *Compute $\mathscr{A}_{opt}$                                        ▷ invert $\Omega$
14:            Compute $||\mathbf{y} - \mathbf{A}\mathscr{A}(\mathbf{u}_1^{n-1}\beta\mathbf{u}_{n+1}^N)||^2$
15:            Compute $\mathbb{P}_s(u_n = \beta|\mathbf{u}^{\backslash n})$
16:        end for
17:        *$\widetilde{u}_n \leftarrow u_n$                                      ▷ save previous value
18:        Generate $u_n$ using $\mathbb{P}_s(\cdot|\mathbf{u}^{\backslash n})$                        ▷ Gibbs
19:        Update $n_q(\cdot)[\cdot]$ at $O(q)$ relevant locations
20:        *Update $\mu_{m\beta}$, $\forall\ m, \beta \in \{u_n, \widetilde{u}_n\}$
21:        *Update $\Omega$                                             ▷ $O(1)$ rows and columns
22:    end for
23: end for
24: *Outputs: Return approximation $\mathscr{A}(\mathbf{u})$ of $\mathbf{x}_{MAP}^U$, $\mathscr{X}$, and temperature offset $r_0 + r$
```

passing that numerical stability can be improved by regularizing Ω. Note also that

$$||\mathbf{y} - \mathbf{A}\mathscr{A}(u)||^2 = \sum_{m=1}^{M} \left(y_m - \sum_{\beta \in \mathscr{X}} \mu_{m\beta} \mathscr{A}_{opt}(\beta) \right)^2, \tag{5.18}$$

which can be computed in $O(M|\mathscr{X}|)$ time instead of $O(MN)$.

Computational complexity: Pseudocode for level-adaptive MCMC (L-MCMC) appears in Algorithm 5.2, which resembles Algorithm 5.1. The initial mapping $\mathscr{A}$ is inherited from a quantization of the initial point $\mathbf{x}^*$, $r_0 = 0$ (r_0 takes different values in Section 5.5.2), and other minor differences between B-MCMC and L-MCMC appear in lines marked by asterisks.

We discuss computational requirements for each line of the pseudocode that is run within the inner loop.

- Line 10 can be computed in $O(q|\mathscr{X}|)$ time (see discussion of Line 9 of B-MCMC in Section 5.4.3).

- Line 11 updates $\mu_{m\beta}$ for $m \in \{1, \cdots, M\}$ in $O(M)$ time.
- Line 12 updates Ω. Because we only need to update $O(1)$ columns and $O(1)$ rows, each such column and row contains $O(|\mathscr{Z}|)$ entries, and each entry is a sum over $O(M)$ terms, we need $O(M|\mathscr{Z}|)$ time.
- Line 13 requires inverting Ω in $O(|\mathscr{Z}|^3)$ time.
- Line 14 requires $O(M|\mathscr{Z}|)$ time, cf. (5.18).
- Line 15 requires $O(|\mathscr{Z}|)$ time.

In practice we typically have $M \gg |\mathscr{Z}|^2$, and so the aggregate complexity is $O(rMN|\mathscr{Z}|)$, which is greater than the computational complexity of Algorithm 5.1 by a factor of $O(|\mathscr{Z}|)$.

5.5.2 Adaptivity in reproduction alphabet size

While Algorithm 5.2 adaptively maps $\mathbf{u}$ to $\mathbb{R}^N$, the signal estimation quality heavily depends on $|\mathscr{Z}|$. Denote the true alphabet of the signal by $\mathscr{X}$, $\mathbf{x} \in \mathscr{X}^N$; if the signal is continuous-valued, then $|\mathscr{X}|$ is infinite. Ideally we want to employ as many levels as the runtime allows for continuous-valued signals, whereas for discrete-valued signals we want $|\mathscr{Z}| = |\mathscr{X}|$. Inspired by this observation, we propose to begin with some initial $|\mathscr{Z}|$, and then adaptively adjust $|\mathscr{Z}|$ hoping to match $|\mathscr{X}|$. Hence, we propose the size- and level-adaptive MCMC algorithm (Algorithm 5.3), which invokes L-MCMC (Algorithm 5.2) several times.

Three basic procedures: In order to describe the size- and level-adaptive MCMC (SLA-MCMC) algorithm in detail, we introduce three alphabet adaptation procedures as follows.

- *MERGE*: First, find the closest adjacent levels $\beta_1, \beta_2 \in \mathscr{Z}$. Create a new level β_3 and add it to $\mathscr{Z}$. Let $\mathscr{A}(\beta_3) = (\mathscr{A}(\beta_1) + \mathscr{A}(\beta_2))/2$. Replace u_i by β_3 whenever $u_i \in \{\beta_1, \beta_2\}$. Next, remove β_1 and β_2 from $\mathscr{Z}$.
- *ADD-out*: Define the range $R_{\mathscr{A}} = [\min \mathscr{A}(\mathscr{Z}), \max \mathscr{A}(\mathscr{Z})]$, and $\mathscr{I}_{R_{\mathscr{A}}} = \max \mathscr{A}(\mathscr{Z}) - \min \mathscr{A}(\mathscr{Z})$. Add a *lower* level β_3 and/or *upper level* β_4 to $\mathscr{Z}$ with

  $$\mathscr{A}(\beta_3) = \min \mathscr{A}(\mathscr{Z}) - \frac{\mathscr{I}_{R_{\mathscr{A}}}}{|\mathscr{Z}| - 1},$$
  $$\mathscr{A}(\beta_4) = \max \mathscr{A}(\mathscr{Z}) + \frac{\mathscr{I}_{R_{\mathscr{A}}}}{|\mathscr{Z}| - 1}.$$

  Note that $\left|\{u_i : u_i = \beta_3 \text{ or } \beta_4, i = 1, \cdots, N\}\right| = 0$, i.e., the new levels are empty.

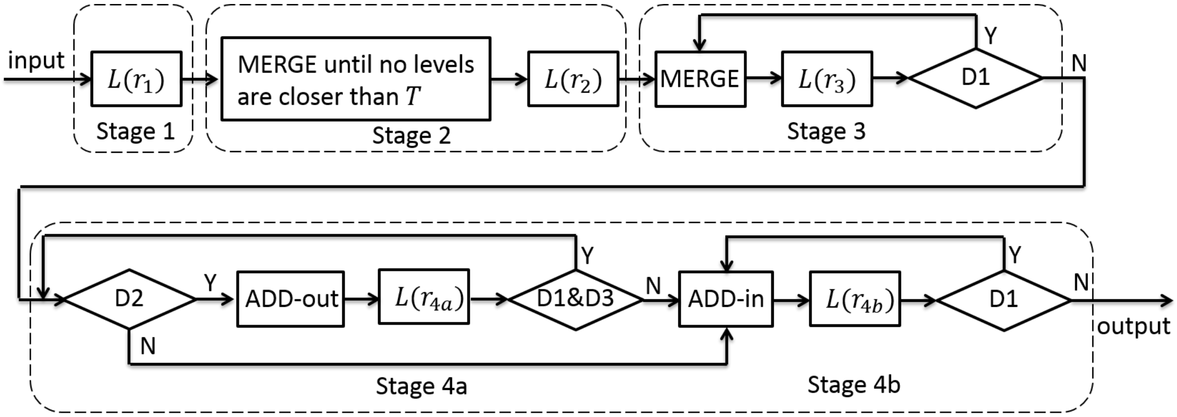


Figure 5.1 Flowchart of Algorithm 5.3 (size- and level-adaptive MCMC). L(r) denotes running L-MCMC for r super-iterations. The parameters r_1, r_2, r_3, r_{4a}, and r_{4b} are the number of super-iterations used in Stages 1 through 4, respectively. Criteria $D1-D3$ are described in the text.

- *ADD-in*: First, find the most distant adjacent levels, β_1 and β_2. Then, add a level β_3 to $\mathscr{Z}$ with $\mathscr{A}(\beta_3) = (\mathscr{A}(\beta_1)+\mathscr{A}(\beta_2))/2$. For $i \in \{1,\cdots,|\mathscr{Z}|\}$ s.t. $u_i = \beta_1$, replace u_i by β_3 with probability

$$\frac{\mathbb{P}_s(u_i=\beta_2)}{\mathbb{P}_s(u_i=\beta_1)+\mathbb{P}_s(u_i=\beta_2)},$$

where $\mathbb{P}_s(\cdot)$ is given in (5.11); for $i \in \{1,\cdots,|\mathscr{Z}|\}$ s.t. $u_i = \beta_2$, replace u_i by β_3 with probability

$$\frac{\mathbb{P}_s(u_i=\beta_1)}{\mathbb{P}_s(u_i=\beta_1)+\mathbb{P}_s(u_i=\beta_2)}.$$

Note that $\left|\{u_i : u_i = \beta_3, i = 1,\cdots,N\}\right|$ is typically non-zero, i.e., β_3 tends not to be empty.

We call the process of running one of these procedures followed by running L-MCMC a *round*.

Size- and level-adaptive MCMC: SLA-MCMC is conceptually illustrated in the flowchart in Figure 5.1. It has four stages, and in each stage we will run L-MCMC for several super-iterations; we denote the execution of L-MCMC for r super-iterations by L(r). The parameters r_1, r_2, r_3, r_{4a}, and r_{4b} are the number of super-iterations used in Stages 1 through 4, respectively. The choice of these parameters reflects a trade-off between runtime and estimation quality.

In Stage 1, SLA-MCMC uses a fixed-size adaptive reproduction alphabet $\mathscr{Z}$ to tentatively estimate the signal. The initial point of Stage 1 is obtained in the same way as L-MCMC. After Stage 1, the initial point and temperature offset for each instance of L-MCMC correspond to the respective outputs of the previous instance of L-MCMC. If the source is discrete-valued and $|\mathscr{Z}| > |\mathscr{X}|$ in Stage 1, then multiple levels in the output $\mathscr{Z}$ of Stage 1 may correspond to a single level in $\mathscr{X}$. To alleviate this problem, in Stage 2 we merge levels closer than $T = \mathscr{I}_{R_{\mathscr{A}}}/(K_1 \times (|\mathscr{Z}|-1))$, where K_1 is a parameter.

However, $|\mathscr{Z}|$ might still be larger than needed; hence in Stage 3 we tentatively merge the closest adjacent levels. The criterion $D1$ evaluates whether the current objective function is lower (better) than in the previous round; we do not leave Stage 3 until $D1$ is violated. Note that if $|\mathscr{X}| > |\mathscr{Z}|$ (this always holds for continuous-valued signals), then ideally SLA-MCMC should not merge any levels in Stage 3, because the objective function would increase if we merge any levels.

Define the outlier set $S = \{x_i : x_i \notin R_{\mathscr{A}}, i = 1, \cdots, N\}$. Under Condition 5.1, S might be small or even empty. When S is small, L-MCMC might not assign levels to represent the entries of S. To make SLA-MCMC more robust to outliers, in Stage 4a we add empty levels outside the range $R_{\mathscr{A}}$ and then allow L-MCMC to change entries of **u** to the new levels during Gibbs sampling; we call this *populating* the new levels. If a newly added outside level is not populated, then we remove it from $\mathscr{Z}$. Seeing that the optimal mapping $\mathscr{A}_{opt}$ in L-MCMC tends not to map symbols to levels with low population, we consider a criterion $D2$ where we will add an outside upper (lower) level if the population of the current upper (lower) level is smaller than $N/(K_2|\mathscr{Z}|)$, where K_2 is a parameter. That is, the criterion $D2$ is violated if both populations of the current upper and lower levels are sufficient (at least $N/(K_2|\mathscr{Z}|)$); in this case we do not need to add outside levels because $\mathscr{A}_{opt}$ will map some of the current levels to represent the entries in S. The criterion $D3$ is violated if all levels added outside are not populated by the end of the round. SLA-MCMC keeps adding levels outside $R_{\mathscr{A}}$ until it is wide enough to cover most of the entries of **x**.

Next, SLA-MCMC considers adding levels inside $R_{\mathscr{A}}$ (Stage 4b). If the signal is discrete-valued, this stage should stop when $|\mathscr{Z}| = |\mathscr{X}|$. Else, for continuous-valued signals SLA-MCMC can add levels until the runtime expires.

In practice, SLA-MCMC runs L-MCMC at most a constant number of times, and the computational complexity is in the same order of L-MCMC, i.e., $O(rMN|\mathscr{Z}|)$. On the other hand, SLA-MCMC allows varying $|\mathscr{Z}|$, which often improves the estimation quality.

5.5.3 Mixing

Donoho proved for the scalar channel setting that $\mathbf{x}_{KS}$ is sampled from the posterior $\mathbb{P}_{X|Y}(\mathbf{x}|\mathbf{y})$ [Don02]. Seeing that the Gibbs sampler used by MCMC (cf. Section 5.4.2) generates random samples, and the outputs of our algorithm will be different if its random number generator is initialized with different *random seeds*, we speculate that running SLA-MCMC several times will also yield independent samples from the posterior, where we note that the runtime grows linearly in the number of times that we run SLA-MCMC. By mixing (averaging over) several outputs of SLA-MCMC, we obtain $\widehat{\mathbf{x}}_{\text{avg}}$, which may have lower squared error w.r.t. the true **x** than the average squared error obtained by a single SLA-MCMC output. Numerical results suggest that mixing indeed reduces the MSE (cf. Figure 5.8); this observation suggests that mixing the outputs of multiple algorithms, including running a random signal estimation algorithm several times, may reduce the squared error.

5.6 Numerical Results

In this section, we demonstrate that SLA-MCMC is comparable and in many cases better than existing algorithms in estimation quality, and that SLA-MCMC is applicable when $M > N$. Additionally, some numerical evidence is provided to justify Conjecture 5.1 in Section 5.3.3. Then, the advantage of SLA-MCMC in estimating low-complexity signals is demonstrated. Finally, we compare B-MCMC, L-MCMC, and SLA-MCMC performance.

We implemented SLA-MCMC in Matlab[8] and tested it using several stationary ergodic sources. Except when noted, for each source, signals $\mathbf{x}$ of length $N = 10000$ were generated. Each such $\mathbf{x}$ was multiplied by a Gaussian random matrix $\mathbf{A}$ with normalized columns and corrupted by i.i.d. Gaussian measurement noise $\mathbf{z}$. Except when noted, the number of measurements M varied between 2000 and 7000. The noise variance σ_Z^2 was selected to ensure that the signal-to-noise ratio (SNR) was 5 or 10 dB; SNR was defined as $\text{SNR} = 10\log_{10}\left[(N\mathbb{E}[x^2])/(M\sigma_Z^2)\right]$. According to Section 5.4.1, the context depth $q = o(\log(N))$, where the base of the logarithm is the alphabet size; using typical values such as $N = 10000$ and $|\mathcal{Z}| = 10$, we have $\log(N) = 4$ and set $q = 2$. While larger q will slow down the algorithm, it might be necessary to increase q when N is larger. The numbers of super-iterations in different stages of SLA-MCMC are $r_1 = 50$ and $r_2 = r_3 = r_{4a} = r_{4b} = 10$, the maximum total number of super-iterations is set to 240, the initial number of levels is $|\mathcal{Z}| = 7$, and the tuning parameters from Section 5.5.2 are $K_1, K_2 = 10$; these parameters seem to work well on an extensive set of numerical experiments. SLA-MCMC was not given the true alphabet $\mathcal{X}$ for any of the sources presented in this chapter; our expectation is that it should adaptively adjust $|\mathcal{Z}|$ to match $|\mathcal{X}|$. The final estimate $\widehat{\mathbf{x}}_{\text{avg}}$ of each signal was obtained by averaging over the outputs $\widehat{\mathbf{x}}$ of 5 runs of SLA-MCMC, where in each run we initialized the random number generator with another random seed, cf. Section 5.5.3. These choices of parameters seemed to provide a reasonable compromise between runtime and estimation quality.

We chose our performance metric as the mean signal-to-distortion ratio (MSDR) defined as $\text{MSDR} = 10\log_{10}\left(\mathbb{E}[x^2]/\text{MSE}\right)$. For each M and SNR, the MSE was obtained after averaging over the squared errors of $\widehat{\mathbf{x}}_{\text{avg}}$ for 50 draws of $\mathbf{x}$, $\mathbf{A}$, and $\mathbf{z}$. We compared the performance of SLA-MCMC to that of (*i*) compressive sensing matching pursuit (CoSaMP) [NT09], a greedy method; (*ii*) gradient projection for sparse reconstruction (GPSR) [Fig07], an optimization-based method; (*iii*) message passing approaches (for each source, we chose best-matched algorithms between EM-GM-AMP-MOS (EGAM for short) [VS13] and turboGAMP (tG for short) [Zin12]); and (*iv*) Bayesian compressive sensing [Ji08] (BCS). Note that EGAM [VS13] places a Gaussian mixture (GM) prior on the signal, and tG [Zin12] builds a prior set including the priors for the signal, the support set of the signal, the channel, and the amplitude structure. Both algorithms learn the parameters of their assumed priors

[8] A toolbox that runs the simulations in this chapter is available at http://people.engr.ncsu.edu/dzbaron/software/UCS_BaronDuarte/

Table 5.1 Computational complexity

Algorithms	Complexity
SLA-MCMC	$O(rMN\|\mathscr{X}\|)$
CoSaMP	$O(L\log\frac{\|\mathbf{x}\|}{\epsilon})$
GPSR	$O(r_P MN)$
EGAM	$O(r_M r_E T_1 + r_M r_E r_G MN)$
tG	$O(r_E T_2 + r_E r_G MN)$

online from the measurements. We compare the computational complexities of the algorithms above in Table 5.1, where L bounds the cost of a matrix-vector multiply with **A** or the Hermitian transpose of **A**, and ϵ is a given precision parameter [NT09]; r_P, r_E, r_G, r_M are the number of GPSR [Fig07], expectation maximization (EM), GAMP [Ran11], and model selection [VS13] iterations, respectively; T_1 and T_2 are the average complexities for the EM algorithm and the turbo updating scheme [Zin12]. Because all these algorithms are iterative algorithms and require different number of iterations to converge or reach a satisfactory estimation quality, we also report their typical runtimes here. Typical runtimes are 1 hour (for continuous-valued signals) and 15 minutes (discrete-valued) per random seed for SLA-MCMC, 30 minutes for EGAM [VS13] and tG [Zin12], and 10 minutes for CoSaMP [NT09] and GPSR [Fig07] on an Intel(R) Core(TM) i7 CPU 860 @ 2.8GHz with 16.0GB RAM running 64 bit Windows 7. The performance of BCS was roughly 5 dB below SLA-MCMC results. Hence, BCS results are not shown in the sequel. We emphasize that algorithms that use training data (such as dictionary learning) [RS12a; Aha06; Mai08; Zho12] will find our problem size $N = 10000$ too large, because they need a training set that has more than N signals. On the other hand, SLA-MCMC does not need to train itself on any training set, and hence is advantageous.

Among these baseline algorithms designed for i.i.d. signals, GPSR [Fig07] and EGAM [VS13] only need **y** and **A**, and CoSaMP [NT09] also needs the number of non-zeros in **x**. Only tG [Zin12] is designed for non-i.i.d. signals; however, it must be aware of the probabilistic model of the source. Finally, GPSR [Fig07] performance was similar to that of CoSaMP [NT09] for all sources considered in this section, and thus is not plotted.

5.6.1 Performance on discrete-valued sources

Bernoulli source: We first present results for an i.i.d. Bernoulli source. The Bernoulli source followed the distribution $f_X(x) = 0.03\delta(x-1) + 0.97\delta(x)$, where $\delta(\cdot)$ is the Dirac delta function. Note that SLA-MCMC did not know $\mathscr{X} = \{0,1\}$ and had to estimate it on the fly. We chose EGAM [VS13] for message passing algorithms because it fits the signal with GM's, which can accurately characterize signals from an i.i.d. Bernoulli source. The resulting MSDR's for SLA-MCMC, EGAM [VS13], and CoSaMP [NT09] are plotted in Figure 5.2. We can see that when SNR = 5 dB, EGAM [VS13] approaches

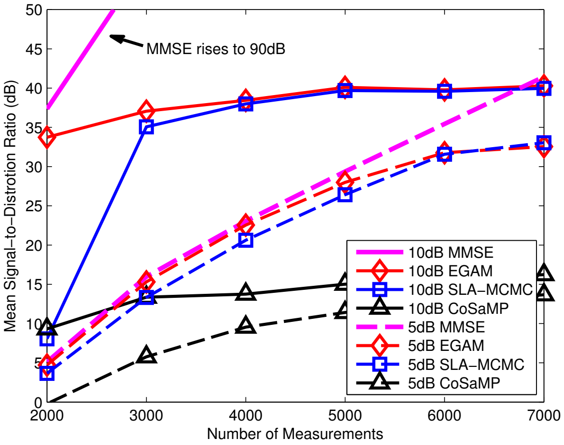


Figure 5.2 SLA-MCMC, EGAM, and CoSaMP estimation results for a source with i.i.d. Bernoulli entries with non-zero probability of 3% as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

the MMSE [ZB13] performance for low to medium M; although SLA-MCMC is often worse than EGAM [VS13], it is within 3 dB of the MMSE performance. This observation that SLA-MCMC approaches the MMSE for SNR = 5 dB partially substantiates Conjecture 5.1 in Section 5.3.3. When SNR = 10 dB, SLA-MCMC is comparable to EGAM [VS13] when $M \geq 3000$. CoSaMP [NT09] has worse MSDR.

Dense Markov-Rademacher source: Considering that most algorithms are designed for i.i.d. sources, we now illustrate the performance of SLA-MCMC on non-i.i.d. sources by simulating a dense Markov-Rademacher (MRad for short) source. The non-zero entries of the dense MRad signal were generated by a two-state Markov state machine (non-zero and zero states). The transition from zero to non-zero state for adjacent entries had probability $\mathbb{P}_{01} = \frac{3}{70}$, while the transition from non-zero to zero state for adjacent entries had probability $\mathbb{P}_{10} = 0.10$; these parameters yielded 30% non-zero entries on average. The non-zeros were drawn from a Rademacher distribution, which took values ±1 with equal probability. With such denser signals, we may need to take more measurements and/or require higher SNR's to achieve similar performance to previous examples. The number of measurements varied from 6000 to 16000, with SNR = 10 and 15 dB. Although tG [Zin12] does not provide an option that accurately characterize the MRad source, we still chose to compare against its performance because it is applicable to non-i.i.d. signals. The MSDR's for SLA-MCMC and tG [Zin12] are plotted in Figure 5.3. CoSaMP [NT09] performs poorly as it is designed for sparse signal estimation, and its results are not shown. Although tG [Zin12] is designed for non-i.i.d. sources, it is nonetheless outperformed by SLA-MCMC. This example shows that SLA-MCMC estimates non-i.i.d. signals well and is applicable to general linear inverse problems. However, recall that the

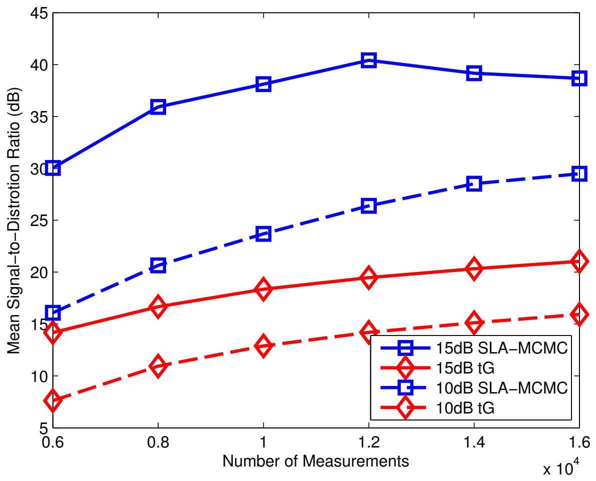


Figure 5.3 SLA-MCMC and tG estimation results for a dense two-state Markov source with non-zero entries drawn from a Rademacher (±1) distribution as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

computational complexity of SLA-MCMC is $O(rMN|\mathscr{X}|)$. Hence, despite the appealing performance of SLA-MCMC shown in this example, we will suffer from high computational time when we have to apply SLA-MCMC in the case when $M > N$.

5.6.2 Performance on continuous sources

We now discuss the performance of SLA-MCMC in estimating continuous sources.

Sparse Laplace (i.i.d.) source: For unbounded continuous-valued signals, which do not adhere to Condition 5.1, we simulated an i.i.d. sparse Laplace source following the random variable $X = X_B X_L$, where $X_B \sim \text{Ber}(0.03)$ is a Bernoulli random variable and X_L follows a Laplace distribution with mean zero and variance one. We chose EGAM [VS13] for message passing algorithms because it fits the signal with GM, which can accurately characterize signals from an i.i.d. sparse Laplace source. The MSDR's for SLA-MCMC, EGAM [VS13], and CoSaMP [NT09] are plotted in Figure 5.4. We can see that EGAM [VS13] approaches the MMSE [ZB13] performance in all settings; SLA-MCMC outperforms CoSaMP [NT09], while it is approximately 2 dB worse than the MMSE. Recall from Conjecture 5.1 that we expect to achieve twice the MMSE, which is approximately 3 dB below the signal-to-distortion ratio of MMSE, and thus SLA-MCMC performance is reasonable. This example of SLA-MCMC performance approaching the MMSE further substantiates Conjecture 5.1.

Markov-Uniform source: For bounded continuous-valued signals, which adhere to Condition 5.1, we simulated a Markov-Uniform (MUnif for short) source, whose non-zero entries were generated by a two-state Markov state machine (non-zero and zero states) with $\mathbb{P}_{01} = \frac{3}{970}$ and

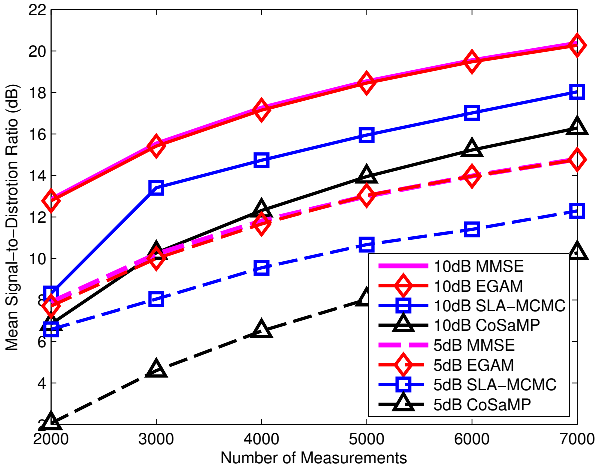


Figure 5.4 SLA-MCMC, EGAM, and CoSaMP estimation results for an i.i.d. sparse Laplace source as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

$\mathbb{P}_{10} = 0.10$; these parameters yielded 3% non-zero entries on average. The non-zero entries were drawn from a uniform distribution between 0 and 1. We chose tG with Markov support and GM model options [Zin12] for message passing algorithms. We plot the resulting MSDR's for SLA-MCMC, tG [Zin12], and CoSaMP [NT09] in Figure 5.5. We can see that the CoSaMP [NT09] lags behind in MSDR. The SLA-MCMC curve is close to that of tG [Zin12] when SNR = 10 dB, and it is slightly better than tG [Zin12] when SNR = 5 dB.

When the signal model is known, the message passing approaches EGAM [VS13] and tG [Zin12] achieve quite low MSE's, because they can get close to the Bayesian MMSE. Sometimes the model is only known imprecisely, and SLA-MCMC can improve over message passing; for example, it is better than tG [Zin12] in estimating MUnif signals (Figure 5.5), because tG [Zin12] approximates the uniformly distributed non-zeros by GM.

5.6.3 Comparison between discrete and continuous sources

When the source is continuous (Figures 5.4 and 5.5), SLA-MCMC might be worse than the existing message passing approaches (EGAM [VS13] and tG [Zin12]). One reason for the under-performance of SLA-MCMC is the 3 dB gap of Conjecture 5.1. The second reason is that SLA-MCMC can only assign finitely many levels to approximate continuous-valued signals, leading to under-representation of the signal. However, when it comes to discrete-valued signals that have finite size alphabets (Figures 5.2 and 5.3), SLA-MCMC is comparable to and in many cases better than existing algorithms. Nonetheless, we observe in the figures that SLA-MCMC is far from the state-of-the-art when the SNR is high and measurement rate is low. Additionally, the dense MRad source in Figure 5.3 has

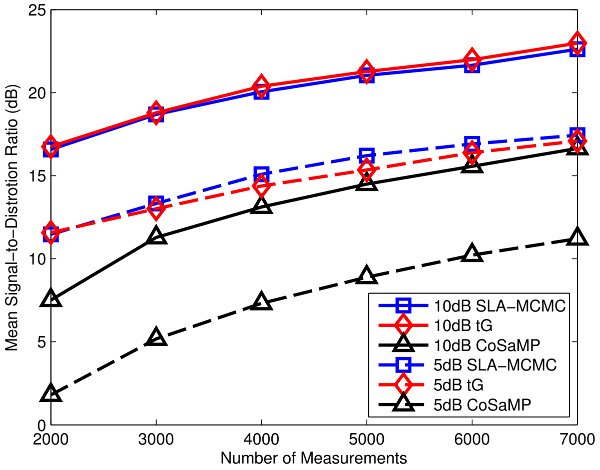


Figure 5.5 SLA-MCMC, tG, and CoSaMP estimation results for a two-state Markov source with non-zero entries drawn from a uniform distribution $U[0,1]$ as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

only a limited number of discrete levels and may not provide a general enough example.

5.6.4 Performance on low-complexity signals

SLA-MCMC promotes low complexity due to the complexity-penalized term in the objective function (5.10). Hence, it tends to perform well for signals with low complexity such as the signals in Figures 5.2 and 5.3 (note that the Bernoulli signal is sparse while the MRad signal is denser). In this subsection, we simulated a non-sparse low-complexity signal. We show that complexity-penalized approaches such as SLA-MCMC might estimate low-complexity signals well.

Four-state Markov source: To evaluate the performance of SLA-MCMC for discrete-valued non-i.i.d. and non-sparse signals, we examined a four-state Markov source (Markov4 for short) that generated the pattern $+1,+1,-1,-1,+1,+1,-1,-1\cdots$ with 3% errors in state transitions, resulting in the signal switching from -1 to $+1$ or vice versa either too early or too late. Note that the estimation algorithm did not know that this source is a binary source. While it is well known that sparsity-promoting CS signal estimation algorithms [Zin12; NT09; Fig07] can estimate sparse sources from linear measurements, the aforementioned switching source is not sparse in conventional sparsifying bases (e.g., Fourier, wavelet, and discrete cosine transforms), rendering such sparsifying transforms not applicable. Signals generated by this Markov source can be sparsified using an averaging analysis matrix [Can11] whose diagonal and first three lower sub-diagonals are filled with $+1$, and all other entries are 0; this transform yields 6% non-zeros in the sparse coefficient vector. However, even if this matrix had been known *a priori*, existing algorithms based on analysis sparsity [Can11] did

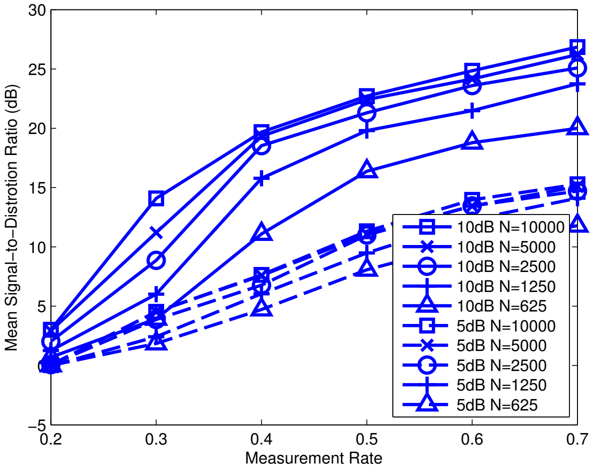


Figure 5.6 SLA-MCMC estimation results for a four-state Markov switching source as a function of the measurement rate κ for different SNR values and signal lengths. Existing CS algorithms fail at estimating this signal, because this source is not sparse.

not perform satisfactorily, yielding MSDR's below 5 dB. Thus, we did not include the results for these baseline algorithms in Figure 5.6. On the other hand, Markov4 signals have low complexity in the time domain, and hence, SLA-MCMC successfully estimated Markov4 signals with reasonable quality even when M was relatively small. This Markov4 source highlights the special advantage of our approach in estimating low-complexity signals.

The MSDR's for shorter Markov4 signals are also plotted in Figure 5.6. We can see that SLA-MCMC performs better when the signal to be estimated is longer. Indeed, SLA-MCMC needs a signal that is long enough to learn the statistics of the signal.

5.6.5 Performance on real world signals

Our experiments up to this point use synthetic signals, where SLA-MCMC has shown comparable and in many cases better results than existing algorithms. This subsection evaluates how well SLA-MCMC estimates a real world signal. We use the "Chirp" sound clip from Matlab: we cut a consecutive part with length 9600 out of the "Chirp" (denoted by $\mathbf{x}$) and performed a short-time discrete cosine transform (DCT) with window size, number of DCT points, and hop size all being 32. Then we vectorized the resulting short-time DCT coefficients matrix to form a coefficient vector θ of length 9600. By denoting the short-time DCT matrix by $\mathbf{W}^{-1}$, we have $\theta = \mathbf{W}^{-1}\mathbf{x}$. Therefore, we can rewrite (5.1) as $\mathbf{y} = \widetilde{\mathbf{A}}\theta + \mathbf{z}$, where $\widetilde{\mathbf{A}} = \mathbf{A}\mathbf{W}$. We want to estimate θ from the measurements $\mathbf{y}$ and the matrix $\widetilde{\mathbf{A}}$. After we obtain the estimate $\widehat{\theta}$, we obtain the estimated signal by $\widehat{\mathbf{x}} = \mathbf{W}\widehat{\theta}$. Although the coefficient vector θ may exhibit some type of memory, it is not readily modeled in closed form, and

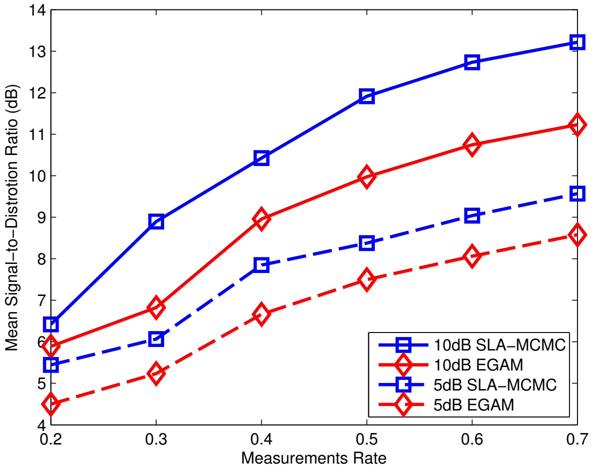


Figure 5.7 SLA-MCMC and EGAM estimation results for a Chirp signal as a function of the measurement rate κ for different SNR values ($N = 9600$).

so we cannot provide a valid model for tG [Zin12]. Instead, we use EGAM [VS13] as our benchmark algorithm. We do not compare to CoSaMP [NT09] because it falls behind in performance as we have seen from other examples. The MSDR's for SLA-MCMC and EGAM [VS13] are plotted in Figure 5.7, where SLA-MCMC outperforms EGAM by 1–2 dB.

5.6.6 Comparison of B-MCMC, L-MCMC, and SLA-MCMC

We compare the performance of B-MCMC, L-MCMC, and SLA-MCMC with different numbers of seeds (cf. Section 5.5.3) by examining the MUnif source (cf. Section 5.6.2). We ran B-MCMC with the fixed uniform alphabet $\mathscr{R}_F$ in (5.5) with $|\mathscr{R}_F| = 10$ levels. L-MCMC was initialized in the same way as Stage 1 of SLA-MCMC. B-MCMC and L-MCMC ran for 100 super-iterations before outputting the estimates; this number of super-iterations was sufficient because it was greater than $r_1 = 50$ in Stage 1 of SLA-MCMC. The results are plotted in Figure 5.8. B-MCMC did not perform well given the $\mathscr{R}_F$ in (5.5) and is not plotted. We can see that SLA-MCMC outperforms L-MCMC. Averaging over more seeds provides an increase of 1 dB in MSDR.[9] It is likely that averaging over more seeds with each seed running fewer super-iterations will decrease the squared error. We leave the optimization of the number of seeds and the number of super-iterations in each seed for future work. Finally, we tried a "good" reproduction alphabet in B-MCMC, $\widetilde{\mathscr{R}}_F = \dfrac{1}{|\mathscr{R}_F|-1/2}\{0, \cdots, |\mathscr{R}_F|-1\}$, and the results were close to those of SLA-MCMC. Indeed, B-MCMC is quite sensitive to the reproduction alphabet, and Stages 2–4 of SLA-MCMC find a good set of levels. Example output levels $\mathscr{A}(\mathscr{X})$ of SLA-MCMC were: $\{-0.001, 0.993\}$ for Bernoulli signals, $\{-0.998, 0.004, 1.004\}$ for dense MRad signals,

[9] For other sources, we observed an increase in MSDR of up to 2 dB.

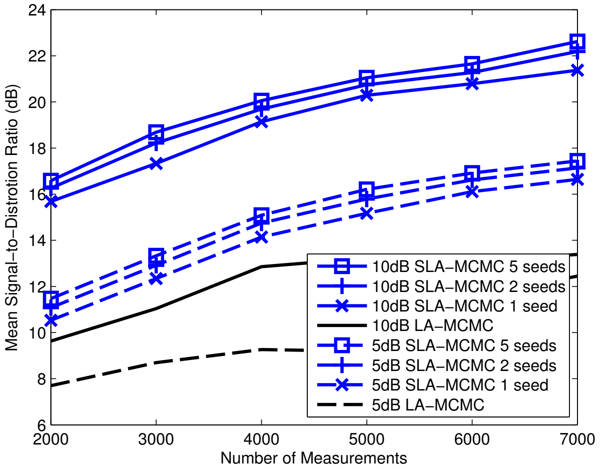


Figure 5.8 SLA-MCMC with different number of random seeds and L-MCMC estimation results for the Markov-Uniform source described in Figure 5.5 as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

21 levels spread in the range $[-3.283, 4.733]$ for i.i.d. sparse Laplace signals, 22 levels spread in the range $[-0.000, 0.955]$ for MUnif signals, and $\{-1.010, 0.996\}$ for Markov4 signals; we can see that SLA-MCMC adaptively adjusted $|\mathscr{Z}|$ to match $|\mathscr{X}|$ so that these levels represented each signal well. Also, we can see from Figures 5.2–5.4 that SLA-MCMC did not perform well in the low measurements and high SNR setting, which was due to mismatch between $|\mathscr{Z}|$ and $|\mathscr{X}|$.

5.7 Approximate Message Passing with Universal Denoising

We note in passing another universal algorithm, approximate message passing with universal denoising (AMP-UD) [Ma14a; Ma16], for CS signal estimation, of which the author of this dissertation is a coauthor. The signal $\mathbf{x}$ is assumed to be stationary and ergodic, but the input statistics are unknown. AMP-UD is a novel algorithmic framework that combines: (*i*) the approximate message passing CS signal estimation framework [Don09; Mon12; BM11; Krz12a; Krz12b; BK15], which solves the CS signal estimation problem by iterative scalar channel denoising; (*ii*) a universal denoising scheme based on context quantization [SW08; SW09], which partitions the stationary ergodic signal denoising into i.i.d. sub-sequence denoising; and (*iii*) a density estimation approach that approximates the probability distribution of an i.i.d. sequence by fitting a GM model [FJ02]. In addition to the algorithmic framework, Ma et al. [Ma14a; Ma16] provide three contributions: (*i*) numerical results showing that state evolution [Don11; BM11; JM12; Don13; Bay15] holds for non-separable Bayesian sliding-window denoisers; (*ii*) an i.i.d. denoiser based on a modified GM learning algorithm; and (*iii*) a universal denoiser that does not need information about the range where

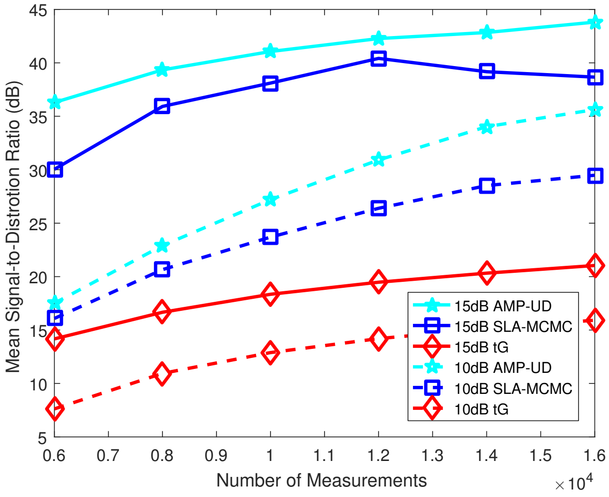


Figure 5.9 AMP-UD [Ma14a; Ma16], SLA-MCMC, and tG estimation results for a dense two-state Markov source with non-zero entries drawn from a Rademacher (±1) distribution as a function of the number of Gaussian random measurements M for different SNR values ($N = 10000$).

the input takes values from or require the input signal to be bounded. Ma et al. [Ma14a; Ma16] provide two implementations of AMP-UD with one being faster and the other being more accurate. The two implementations compare favorably with existing universal signal estimation algorithms (including the SLA-MCMC algorithm discussed in this chapter) in terms of both estimation quality and runtime.

To highlight the advantages of AMP-UD relative to SLA-MCMC, Figure 5.9 compares the AMP-UD simulation results to the SLA-MCMC and tG [Zin12] results for the setting in Figure 5.3. We see that AMP-UD outperforms both algorithms. Moreover, the runtime of AMP-UD is around 5 minutes to estimate this MRad signal of length 10000, while it usually takes SLA-MCMC an hour and tG [Zin12] 30 minutes to estimate this signal. Therefore, we see that AMP-UD is indeed promising.

5.8 Conclusion

This chapter provided universal algorithms for signal estimation from linear measurements. Here, universality denotes the property that the algorithm need not be informed of the probability distribution for the recorded signal prior to acquisition; rather, the algorithm simultaneously builds estimates both of the observed signal and its distribution. Inspired by the Kolmogorov sampler [Don02] and motivated by the need for a computationally tractable framework, our contribution focused on stationary ergodic signal sources and relied on a maximum a posteriori estimation algorithm. The algorithm was then implemented via a Markov chain Monte Carlo formulation that is proven to be convergent in the limit of infinite computation. We reduced the computational complexity and improve the estimation quality of the proposed algorithm by adapting the reproduction alphabet to match the complexity of the input signal. Our experiments have shown that the performance of the proposed algorithm is comparable to and in many cases better than existing algorithms, particularly for low-complexity sources that do not exhibit standard sparsity or compressibility.

As we were finishing this work, Jalali and Poor [JP14] have independently shown that our formulation (5.10) also provides an implementable version of Rényi entropy minimization. Their theoretical findings further motivated our proposed universal MCMC formulation. We noted in passing another universal algorithm that often achieves better estimation quality than the SLA-MCMC algorithm discussed in this chapter.

CHAPTER

6

DISCUSSION

This chapter concludes the dissertation. We begin by summarizing the previous chapters, and then we list our contributions. Finally, we propose some possible future directions.

6.1 Summary and Contributions

Linear models find wide applications in the real world, and the problem of estimating the underlying signal(s) from a linear model is called a linear inverse problem. Depending on the number of underlying signals, we have the single measurement vector problem (SMV) and the multi-measurement vector problem (MMV); depending on how the measurement matrix and the measurements are stored, we have the centralized linear model and the multi-processor linear model. Prior art includes algorithms for linear inverse problems and their corresponding performance characterizations. There are many remaining issues in the prior art. First, there is little work discussing the performance characterization for the linear inverse problems themselves. Second, when dealing with the distributed setting, there is little work studying the relations of different costs. At last, the existing algorithms for linear inverse problems require the prior knowledge of the unknown signal to some extent. These issues are important to practitioners. In this dissertation, we took advantage of the tools in statistical physics and information theory to address these issues in the large system limit, i.e., the length of the signal and the number of measurements go to infinity while the measurement rate (ratio between the number of measurements and the length of the signal) stays constant.

We started with providing background materials on statistical physics and information theory in Chapter 2, and we also discussed the link between statistical physics and information theory. Then, we studied the minimum mean squared error (MMSE) for MMV problem in Chapter 3 by using the replica analysis from statistical physics. We analyzed the MMSE for two settings of MMV problems, where the entries in the signal vectors are independent and identically distributed (i.i.d.), and share the same support. One MMV setting has i.i.d. Gaussian measurement matrices, while the other MMV setting has identical i.i.d. Gaussian measurement matrices. Replica analysis yields identical free energy expressions for these two settings in the large system limit. Because of the identical free energy expressions, the MMSE's for both MMV settings are identical. By numerically evaluating the free energy expression, we identified different performance regions for MMV where the MMSE as a function of the channel noise variance and the measurement rate behaves differently. We also identified a phase transition for belief propagation algorithms (BP) that separates regions where BP achieves the MMSE asymptotically and where it is sub-optimal. Simulation results of an approximated version of BP matched with the mean squared error (MSE) predicted by replica analysis. As a special case of MMV, we extended our replica analysis to complex SMV, so that we can calculate the MMSE for complex SMV with real or complex measurement matrices. Chapter 3 is based on our work with Baron [ZB13] and with Baron and Krzakala [Zhu16b].

In Chapter 4, we studied the optimization of different costs in running a distributed algorithm; these costs include (but are not limited to) the computation cost, the communication cost, and the quality of the estimate. We focused our discussion on a certain distributed algorithm, multiprocessor approximate message passing (MP-AMP). Our results might be extended to some other distributed and iterative algorithms. We proposed to use lossy compression (from information theory) on the messages being transmitted across the network, and we allowed the coding rate to vary from iteration to iteration for MP-AMP. Also, we proposed an algorithmic method to find the optimal coding rate for the messages being transmitted in the network for MP-AMP, so that we can achieve the smallest combined cost of computation and communication. In addition, we theoretically analyzed the optimal coding rate sequence in the limit of low excess mean squared error (EMSE=MSE-MMSE) and it turns out that the optimal coding rate sequence is approximately linear when the EMSE is low. At last, we proved the existence of trade-offs among these different costs for MP-AMP. Chapter 4 is based on our work with Han et al. [Han16] and with Baron and Beirami [Zhu16c; Zhu16a].

In Chapter 5, we proposed a universal algorithm, size- and level-adaptive Markov chain Monte Carlo (SLA-MCMC), to solve the linear inverse problem. Inspired by the Kolmogorov sampler [Don02] and motivated by the need for a computationally tractable framework, our contribution focused on stationary ergodic signal sources and relied on a maximum a posteriori estimation algorithm. The algorithm was then implemented via a Markov chain Monte Carlo formulation (motivated from thermodynamics) that is proven to be convergent in the limit of infinite computation. We reduced the computational complexity and improved the estimation quality of the proposed algorithm by adapting the reproduction alphabet to match the complexity of the input signal. Our experiments have shown that the performance of the proposed algorithm is comparable to and in many cases better than existing algorithms, particularly for low-complexity signals that do not exhibit standard sparsity or compressibility. Chapter 5 is based on our work with Baron and Duarte [Zhu14; Zhu15].

6.2 Future Directions

Along the line of this dissertation, we list some possible future directions.

1. Our replica analysis in Chapter 3 assumes that the non-zero entries of the jointly sparse signals are i.i.d. However, in real-world application, sometimes the non-zero entries that share the same support are dependent. Our derivation could possibly be generalized to such settings. When the non-zero entries of the signals are dependent, we suspect that the MMV setting with different matrices will yield lower MMSE than the MMV setting with identical matrices.

2. As is discussed in Chapter 3, studying other error metrics than the MSE could also be of interest. We could extend the work of Tan and coauthors [Tan14a; Tan14b], so that we can both study the theoretic optimal performance for user-defined additive error metric and design algorithms that can achieve the theoretic optimal performance.

3. In Chapter 4, our study of different costs is within the MP-AMP algorithm. One possible future direction could be to find a generic class of algorithms to which our analyses can apply. Another possible direction is to incorporate such ideas in a real-world software package design, which could be of great interest to industry.

4. Although both SLA-MCMC and AMP-UD from Chapter 5 seem promising, they are not so resilient to measurement matrices that are far from i.i.d. In order to make a larger impact, we need to design universal algorithms that are more resilient to non-i.i.d. matrices.

[Cc2] *A true system-on-chip solution for 2.4-GHz IEEE 802.15.4 and ZigBee applications*. SWRS081B. Rev. B. Texas Instruments. Apr. 2009.

[Aha06] Aharon, M. et al. “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation”. *IEEE Trans. Signal Process.* **54**.11 (Nov. 2006), pp. 4311–4322.

[Ec2] “Amazon EC2”. https://aws.amazon.com/ec2/.

[Ari72] Arimoto, S. “An algorithm for computing the capacity of an arbitrary discrete memoryless channel”. *IEEE Trans. Inf. Theory* **18**.1 (Jan. 1972), pp. 14–20.

[BK15] Barbier, J. & Krzakala, F. “Approximate message-passing decoder and capacity-achieving sparse superposition codes”. *Arxiv preprint arXiv:1503.08040* (Mar. 2015).

[Bar16] Barbier, J. et al. “The mutual information in random linear estimation”. *Arxiv preprint arXiv:1607.02335* (July 2016).

[Bar11] Baron, D. “Information complexity and estimation”. *Workshop Inf. Theoretic Methods Sci. Eng. (WITMSE)*. Helsinki, Finland, Aug. 2011.

[BD11] Baron, D. & Duarte, M. F. “Universal MAP estimation in compressed sensing”. *Proc. Allerton Conf. Commun., Control, and Comput.* Sept. 2011, pp. 768–775.

[BW12] Baron, D. & Weissman, T. “An MCMC approach to universal lossy compression of analog sources”. *IEEE Trans. Signal Process.* **60**.10 (Oct. 2012), pp. 5230–5240.

[Bar06] Baron, D. et al. *Distributed compressed sensing*. Tech. rep. ECE-0612. Rice University, Dec. 2006.

[Bar10] Baron, D. et al. “Bayesian compressive sensing via belief propagation”. *IEEE Trans. Signal Process.* **58**.1 (Jan. 2010), pp. 269–280.

[Bar98] Barron, A. et al. “The minimum description length principle in coding and modeling”. *IEEE Trans. Inf. Theory* **44**.6 (Oct. 1998), pp. 2743–2760.

[BM11] Bayati, M. & Montanari, A. “The dynamics of message passing on dense graphs, with applications to compressed sensing”. *IEEE Trans. Inf. Theory* **57**.2 (Feb. 2011), pp. 764–785.

[Bay15] Bayati, M. et al. “Universality in polytope phase transitions and message passing algorithms”. *Ann. Appl. Probability* **25**.2 (Feb. 2015), pp. 753–822.

[BF09] Berg, E. & Friedlander, M. P. “Joint-sparse recovery from multiple measurements”. *Arxiv preprint arXiv:0904.2051* (Apr. 2009).

[Ber71] Berger, T. *Rate Distortion Theory: Mathematical Basis for Data Compression*. Prentice-Hall Englewood Cliffs, NJ, 1971, xiii, 311 p.

[Ber95] Bertsekas, D. P. *Dynamic Programming and Optimal Control*. Vol. 1. Athena Scientific Belmont, MA, 1995.

[Bla72] Blahut, R. E. “Computation of channel capacity and rate-distortion functions”. *IEEE Trans. Inf. Theory* **18**.4 (July 1972), pp. 460–473.

[Bré99] Brémaud, P. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Vol. 31. Springer Verlag, 1999.

[Can06] Candès, E. et al. “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. *IEEE Trans. Inf. Theory* **52**.2 (Feb. 2006), pp. 489–509.

[Can11] Candès, E. J. et al. “Compressed sensing with coherent and redundant dictionaries”. *Appl. Computational Harmonic Anal.* **31**.1 (July 2011), pp. 59–73.

[Cha66] Chaitin, G. J. “On the length of programs for computing finite binary sequences”. *J. ACM* **13**.4 (1966), pp. 547–569.

[CH06] Chen, J. & Huo, X. “Theoretical results on sparse representations of multiple measurement vectors”. *IEEE Trans. Signal Process.* **54**.12 (Dec. 2006), pp. 4634–4643.

[Cot05] Cotter, S. F. et al. “Sparse solutions to linear inverse problems with multiple measurement vectors”. *IEEE Trans. Signal Process.* **53**.7 (July 2005), pp. 2477–2488.

[CT06] Cover, T. M. & Thomas, J. A. *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 2006.

[DD98] Das, I. & Dennis, J. E. “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems”. *SIAM J. Optimization* **8**.3 (Aug. 1998), pp. 631–657.

[DG08] Dean, J. & Ghemawat, S. “MapReduce: Simplified data processing on large clusters”. *Commun. ACM* **51**.1 (Jan. 2008), pp. 107–113.

[Don06a] Donoho, D. “Compressed sensing”. *IEEE Trans. Inf. Theory* **52**.4 (Apr. 2006), pp. 1289–1306.

[Don06b] Donoho, D. et al. “The simplest solution to an underdetermined system of linear equations”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Seattle, WA, July 2006, pp. 1924–1928.

[Don13] Donoho, D. et al. “Accurate prediction of phase transitions in compressed sensing via a connection to minimax denoising”. *IEEE Trans. Inf. Theory* **59**.6 (June 2013), pp. 3396–3433.

[Don02] Donoho, D. L. *The Kolmogorov sampler*. Department of Statistics Technical Report 2002-4. Stanford, CA: Stanford University, Jan. 2002.

[Don09] Donoho, D. L. et al. “Message passing algorithms for compressed sensing”. *Proc. Nat. Academy Sci.* **106**.45 (Nov. 2009), pp. 18914–18919.

[Don10] Donoho, D. L. et al. “Message passing algorithms for compressed sensing: I. Motivation and construction”. *IEEE Inf. Theory Workshop*. Jan. 2010.

[Don11] Donoho, D. L. et al. “The noise-sensitivity phase transition in compressed sensing”. *IEEE Trans. Inf. Theory* **57**.10 (Oct. 2011), pp. 6920–6941.

[Dua06] Duarte, M. F. et al. “Universal distributed sensing via random projections”. *Proc. IEEE Int. Conf. Inf. Process. Sensor Networks (IPSN)*. Nashville, TN, Apr. 2006, pp. 177–185.

[Dua13] Duarte, M. F. et al. “Measurement bounds for sparse signal ensembles via graphical models”. *IEEE Trans. Inf. Theory* **59**.7 (July 2013), pp. 4280–4289.

[Est02] Estrin, D. et al. “Connecting the physical world with pervasive networks”. *IEEE Pervasive Comput.* **1**.1 (Jan. 2002), pp. 59–69.

[FJ02] Figueiredo, M. & Jain, A. “Unsupervised learning of finite mixture models”. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**.3 (Mar. 2002), pp. 381–396.

[Fig07] Figueiredo, M. et al. “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems”. *IEEE J. Sel. Topics Signal Proces.* **1**.4 (Dec. 2007), pp. 586–597.

[FN03] Figueiredo, M. A. T. & Nowak, R. D. “An EM algorithm for wavelet-based image restoration”. *IEEE Trans. Image Process.* **12**.8 (Aug. 2003), pp. 906–916.

[Fra08] Frasca, P. et al. “Average consensus on networks with quantized communication”. *Int. J. Robust Nonlinear Control* **19**.16 (Nov. 2008), pp. 1787–1816.

[GO07] Garrigues, P. J. & Olshausen, B. A. “Learning horizontal connections in a sparse coding model of natural images”. *Workshop Neural Inf. Process. Syst. (NIPS)*. Vancouver, B.C., Canada, Dec. 2007, pp. 1–8.

[GG84] Geman, S. & Geman, D. “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”. *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (Nov. 1984), pp. 721–741.

[GG93] Gersho, A. & Gray, R. M. *Vector Quantization and Signal Compression*. Kluwer, 1993.

[Gra84] Gray, R. M. “Vector quantization”. *IEEE ASSP Mag.* **1**.2 (Apr. 1984), pp. 4–29.

[GN98] Gray, R. M. & Neuhoff, D. L. “Quantization”. *IEEE Trans. Inf. Theory* **IT-44** (Oct. 1998), pp. 2325–2383.

[GV05] Guo, D. & Verdú, S. “Randomly spread CDMA: Asymptotics via statistical physics”. *IEEE Trans. Inf. Theory* **51**.6 (June 2005), pp. 1983–2010.

[GW08] Guo, D. & Wang, C. C. “Multiuser detection of sparsely spread CDMA”. *IEEE J. Sel. Areas Commun.* **26**.3 (Apr. 2008), pp. 421–431.

[Guo09] Guo, D. et al. “A Single-letter Characterization of Optimal Noisy Compressed Sensing”. *Proc. Allerton Conf. Commun., Control, and Comput.* Sept. 2009, pp. 52–59.

[Han14] Han, P. et al. “Distributed approximate message passing for sparse signal recovery”. *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*. Atlanta, GA, Dec. 2014, pp. 497–501.

[Han15a] Han, P. et al. “Communication-efficient distributed IHT”. *Proc. Signal Process. with Adaptive Sparse Structured Representations Workshop (SPARS)*. Cambridge, United Kingdom, July 2015.

[Han15b] Han, P. et al. “Modified distributed iterative hard thresholding”. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*. Brisbane, Australia, Apr. 2015, pp. 3766–3770.

[Han16] Han, P. et al. “Multi-processor approximate message passing using lossy compression”. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*. Shanghai, China, Mar. 2016, pp. 6240–6244.

[HN06] Haupt, J. & Nowak, R. “Signal reconstruction from noisy random projections”. *IEEE Trans. Inf. Theory* **52**.9 (Sept. 2006), pp. 4036–4048.

[HN12] Haupt, J. D. & Nowak, R. “Adaptive sensing for sparse recovery”. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[Str] “Hubbard–Stratonovich transformation”. https://en.wikipedia.org/wiki/Hubbard-Stratonovich_transformation.

[JM11] Jalali, S. & Maleki, A. “Minimum complexity pursuit”. *Proc. Allerton Conf. Commun., Control, Comput.* Sept. 2011, pp. 1764–1770.

[JP14] Jalali, S. & Poor, H. V. “Universal compressed sensing of Markov sources”. *Arxiv preprint arXiv:1406.7807* (June 2014).

[JW08] Jalali, S. & Weissman, T. “Rate-distortion via Markov chain Monte Carlo”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Toronto, Ontario, Canada, July 2008, pp. 852–856.

[JW12] Jalali, S. & Weissman, T. “Block and sliding-block lossy compression via MCMC”. *IEEE Trans. Commun.* **60**.8 (Aug. 2012), pp. 2187–2198.

[Jal14] Jalali, S. et al. “Minimum complexity pursuit for universal compressed sensing”. *IEEE Trans. Inf. Theory* **60**.4 (Apr. 2014), pp. 2253–2268.

[JM12] Javanmard, A. & Montanari, A. “State evolution for general approximate message passing algorithms, with applications to spatial coupling”. *Arxiv preprint arXiv:1211.5164* (Dec. 2012).

[Ji08] Ji, S. et al. “Bayesian compressive sensing”. *IEEE Trans. Signal Process.* **56**.6 (June 2008), pp. 2346–2356.

[Jun07] Jung, H. et al. “Improved k-t BLAST and k-t SENSE using FOCUSS”. *Physics in Medicine and Biology* **52**.11 (May 2007), pp. 3201–3226.

[Jun09] Jung, H. et al. “k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI”. *J. Magnetic Resonance in Medicine* **61**.1 (Jan. 2009), pp. 103–116.

[Kim11] Kim, J. et al. “Belief propagation for jointly sparse recovery”. *Arxiv preprint arXiv:1102.3289* (Feb. 2011).

[Kol65] Kolmogorov, A. N. “Three approaches to the quantitative definition of information”. *Problems Inf. Transmission* **1**.1 (1965), pp. 1–7.

[Kre89] Kreyszig, E. *Introductory Functional Analysis with Applications*. Wiley, 1989.

[Krz12a] Krzakala, F. et al. “Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices”. *J. Stat. Mech. – Theory E.* **2012**.08 (Aug. 2012), P08009.

[Krz12b] Krzakala, F. et al. “Statistical-physics-based reconstruction in compressed sensing”. *Phys. Rev. X* **2**.2 (May 2012), p. 021005.

[Lee12] Lee, K. et al. “Subspace methods for joint sparse recovery”. *IEEE Trans. Inf. Theory* **58**.6 (June 2012), pp. 3613–3641.

[Lee11] Lee, O. et al. “Compressive diffuse optical tomography: Noniterative exact reconstruction using joint sparsity”. *IEEE Trans. Medical Imaging* **30**.5 (May 2011), pp. 1129–1142.

[Les15] Lesieur, T. et al. “Phase transitions in sparse PCA”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Hong Kong, China, July 2015, pp. 1635–1639.

[LV08] Li, M. & Vitanyi, P. M. B. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, 2008.

[Li15] Li, S. et al. “Coded MapReduce”. *Proc. Allerton Conf. Commun., Control, and Comput.* Sept. 2015, pp. 964–971.

[Li16] Li, S. et al. “Fundamental tradeoff between computation and communication in distributed computing”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Barcelona, Spain, July 2016, pp. 1814–1818.

[Lin80] Linde, Y. et al. “An algorithm for vector quantizer design”. *IEEE Trans. Commun.* **28**.1 (Jan. 1980), pp. 84–95.

[Ma14a] Ma, Y. et al. “Compressed sensing via universal denoising and approximate message passing”. *Proc. Allerton Conf. Commun., Control, and Comput.* Oct. 2014.

[Ma14b] Ma, Y. et al. “Empirical Bayes and full Bayes for signal estimation”. *Arxiv preprint arxiv:1405.2113v1* (May 2014).

[Ma14c] Ma, Y. et al. “Two-part reconstruction with noisy-sudocodes”. *IEEE Trans. Signal Process.* **62**.23 (Dec. 2014), pp. 6323–6334.

[Ma15] Ma, Y. et al. “Mismatched Estimation in Large Linear Systems”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Hong Kong, China, July 2015, pp. 760–764.

[Ma16] Ma, Y. et al. “Approximate message passing algorithm with universal denoising and Gaussian mixture learning”. *IEEE Trans. Signal Process.* **65**.21 (Nov. 2016), pp. 5611–5622.

[Mai08] Mairal, J. et al. “Supervised dictionary learning”. *Workshop Neural Inf. Process. Syst. (NIPS)*. Vancouver, B.C., Canada, Dec. 2008.

[Mal05] Malioutov, D. et al. “A sparse signal reconstruction perspective for source localization with sensor arrays”. *IEEE Trans. Signal Process.* **53**.8 (Aug. 2005), pp. 3010–3022.

[McM13] McMahan, H. B. et al. “Ad click prediction: A view from the trenches”. *Proc. ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*. Chicago, IL, Aug. 2013, pp. 1222–1230.

[Mer10] Merhav, N. “Statistical physics and information theory”. *Foundations and Trends in Communications and Information Theory* **6**.1–2 (2010), pp. 1–212.

[MM09] Mézard, M. & Montanari, A. *Information, Physics, and Computation*. Oxford University press, 2009.

[ME09] Mishali, M. & Eldar, Y. C. “Reduce and boost: Recovering arbitrary sets of jointly sparse vectors”. *IEEE Trans. Signal Process.* **56**.10 (Oct. 2009), pp. 4692–4702.

[Mon12] Montanari, A. “Graphical models concepts in compressed sensing”. *Compressed Sensing: Theory and Applications* (2012), pp. 394–438.

[MT06] Montanari, A. & Tse, D. “Analysis of belief propagation for non-linear problems: The example of CDMA (or: How to prove Tanaka’s formula)”. *IEEE Inf. Theory Workshop*. Mar. 2006, pp. 160–164.

[Mot12] Mota, J. et al. “Distributed basis pursuit”. *IEEE Trans. Signal Process.* **60**.4 (Apr. 2012), pp. 1942–1956.

[NT09] Needell, D. & Tropp, J. A. “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. *Appl. Computational Harmonic Anal.* **26**.3 (May 2009), pp. 301–321.

[Pat13] Patterson, S. et al. “Distributed sparse signal recovery for sensor networks”. *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*. Vancouver, B.C., Canada, May 2013, pp. 4494–4498.

[Pat14] Patterson, S. et al. “Distributed compressed sensing for static and time-varying networks”. *IEEE Trans. Signal Process.* **62**.19 (Oct. 2014), pp. 4931–4946.

[Pcc] “Pearson product-moment correlation coefficient”. https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient.

[PK00] Pottie, G. J. & Kaiser, W. J. “Wireless integrated network sensors”. *Commun. ACM* **43**.5 (May 2000), pp. 51–58.

[RS12a] Ramírez, I. & Sapiro, G. “An MDL framework for sparse coding and dictionary learning”. *IEEE Trans. Signal Process.* **60**.6 (June 2012), pp. 2913–2927.

[RS12b] Ramirez, I. & Sapiro, G. “Universal regularizers for robust sparse coding and modeling”. *IEEE Trans. Image Process.* **21**.9 (Sept. 2012), pp. 3850–3864.

[Ran10] Rangan, S. “Estimation with random linear mixing, belief propagation and compressed sensing”. *Proc. IEEE Conf. Inf. Sci. Syst. (CISS)*. Princeton, NJ, Mar. 2010.

[Ran11] Rangan, S. “Generalized approximate message passing for estimation with random linear mixing”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. St. Petersburg, Russia, July 2011, pp. 2168–2172.

[Ran12] Rangan, S. et al. “Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing”. *IEEE Trans. Inf. Theory* **58**.3 (Mar. 2012), pp. 1902–1923.

[Rav15] Ravazzi, C. et al. “Distributed iterative thresholding for ℓ_0/ℓ_1 -regularized linear inverse problems”. *IEEE Trans. Inf. Theory* **61**.4 (Apr. 2015), pp. 2081–2100.

[RP16] Reeves, G. & Pfister, H. D. “The replica-symmetric prediction for compressed sensing with Gaussian matrices is exact”. *Arxiv preprint arXiv:1607.02524* (July 2016).

[Ric07] Richardson, M. et al. “Predicting clicks: Estimating the click-through rate for new ads”. *Proc. Int. World Wide Web Conf. (WWW)*. Banff, Alberta, Canada, May 2007, pp. 521–530.

[Ris78] Rissanen, J. “Modeling by shortest data description”. *Automatica* **14**.5 (Sept. 1978), pp. 465–471.

[Ris83] Rissanen, J. “A universal data compression system”. *IEEE Trans. Inf. Theory* **29**.5 (Sept. 1983), pp. 656–664.

[Ros94] Rose, K. “A mapping approach to rate-distortion computation and analysis”. *IEEE Trans. Inf. Theory* **40**.6 (Nov. 1994), pp. 1939–1952.

[RV16] Rush, C. & Venkataramanan, R. “Finite-Sample Analysis of Approximate Message Passing”. *Arxiv preprint arXiv:1606.01800* (June 2016).

[Sch78] Schwarz, G. “Estimating the dimension of a model”. *Ann. Stat.* **6**.2 (Mar. 1978), pp. 461–464.

[SN08] Seeger, M. W. & Nickisch, H. “Compressed sensing and Bayesian experimental design”. *Proc. Int. Conf. Mach. Learning*. Helsinki, Finland, Aug. 2008, pp. 912–919.

[SW08] Sivaramakrishnan, K. & Weissman, T. “Universal denoising of discrete-time continuous-amplitude signals”. *IEEE Trans. Inf. Theory* **54**.12 (Dec. 2008), pp. 5632–5660.

[SW09] Sivaramakrishnan, K. & Weissman, T. “A context quantization approach to universal denoising”. *IEEE Trans. Signal Process.* **57**.6 (June 2009), pp. 2110–2129.

[Sol64] Solomonoff, R. J. “A formal theory of inductive inference. Part I”. *Inf. and Control* **7**.1 (Mar. 1964), pp. 1–22.

[Tan14a] Tan, J. et al. “Signal estimation with additive error metrics in compressed sensing”. *IEEE Trans. Inf. Theory* **60**.1 (Jan. 2014), pp. 150–158.

[Tan14b] Tan, J. et al. “Wiener filters in Gaussian mixture signal estimation with ℓ_∞-norm error”. *IEEE Trans. Inf. Theory* **60**.10 (Oct. 2014), pp. 6626–6635.

[Tan15] Tan, J. et al. “Compressive imaging via approximate message passing with image denoising”. *IEEE Trans. Signal Process.* **63**.8 (Apr. 2015), pp. 2085–2092.

[Tan02] Tanaka, T. “A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors”. *IEEE Trans. Inf. Theory* **48**.11 (Nov. 2002), pp. 2888–2910.

[Tha13] Thanou, D. et al. “Distributed average consensus with quantization refinement”. *IEEE Trans. Signal Process.* **61**.1 (Jan. 2013), pp. 194–205.

[Tro06a] Tropp, J. A. “Algorithms for simultaneous sparse approximation. Part II: Convex relaxation”. *Signal Process.* **86**.3 (Mar. 2006), pp. 589–602.

[Tro06b] Tropp, J. A. et al. “Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit.” *Signal Process.* **86**.3 (Mar. 2006), pp. 572–588.

[Tur50] Turing, A. M. “Computing machinery and intelligence”. *Mind* **59**.236 (Oct. 1950), pp. 433–460.

[VS13] Vila, J. & Schniter, P. “Expectation-maximization Gaussian-mixture approximate message passing”. *IEEE Trans. Signal Process.* **61**.19 (Oct. 2013), pp. 4658–4672.

[WB68] Wallace, C. S. & Boulton, D. M. “An information measure for classification”. *Comput. J.* **11**.2 (1968), pp. 185–194.

[WV12a] Weidmann, C. & Vetterli, M. “Rate distortion behavior of sparse sources”. *IEEE Trans. Inf. Theory* **58**.8 (Aug. 2012), pp. 4969–4992.

[WK08] Widrow, B. & Kollár, I. *Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications*. Cambridge University press, 2008.

[Wil95] Willems, F. M. J. et al. “The context tree weighting method: Basic properties”. *IEEE Trans. Inf. Theory* **41**.3 (May 1995), pp. 653–664.

[WV11] Wu, Y. & Verdú, S. “MMSE Dimension”. *IEEE Trans. Inf. Theory* **57**.8 (Aug. 2011), pp. 4857–4879.

[WV12b] Wu, Y. & Verdú, S. “Optimal phase transitions in compressed sensing”. *IEEE Trans. Inf. Theory* **58**.10 (Oct. 2012), pp. 6241–6263.

[Yan97] Yang, E. et al. “Fixed-slope universal lossy data compression”. *IEEE Trans. Inf. Theory* **43**.5 (Sept. 1997), pp. 1465–1476.

[Ye15] Ye, J. C. et al. “Improving M-SBL for joint sparse recovery using a subspace penalty”. *IEEE Trans. Signal Process.* **63**.24 (Dec. 2015), pp. 6595–6605.

[Zho12] Zhou, M. et al. “Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images”. *IEEE Trans. Image Process.* **21**.1 (Jan. 2012), pp. 130–144.

[ZB13] Zhu, J. & Baron, D. “Performance regions in compressed sensing from noisy measurements”. *Proc. IEEE Conf. Inf. Sci. Syst. (CISS)*. Baltimore, MD, Mar. 2013.

[Zhu14] Zhu, J. et al. “Complexity–adaptive universal signal estimation for compressed sensing”. *Proc. IEEE Stat. Signal Process. Workshop (SSP)*. Gold Coast, Australia, June 2014, pp. 416–419.

[Zhu15] Zhu, J. et al. “Recovery from linear measurements with complexity–matching universal signal estimation”. *IEEE Trans. Signal Process.* **63**.6 (Mar. 2015), pp. 1512–1527.

[Zhu16a] Zhu, J. et al. “Optimal trade-offs in multi-processor approximate message passing”. *Arxiv preprint arXiv:1601.03790* (Nov. 2016).

[Zhu16b] Zhu, J. et al. “Performance limits for noisy multi-measurement vector problems”. *Arxiv preprint arXiv:1604.02475v2* (Aug. 2016).

[Zhu16c] Zhu, J. et al. “Performance trade-offs in multi-processor approximate message passing”. *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. Barcelona, Spain, July 2016, pp. 680–684.

[ZS11] Ziniel, J. & Schniter, P. “Efficient message passing-based inference in the multiple measurement vector problem”. *Proc. IEEE Asilomar Conf. Signals, Syst., and Comput.* Nov. 2011, pp. 1447–1451.

[ZS13] Ziniel, J. & Schniter, P. “Efficient high-dimensional inference in the multiple measurement vector problem”. *IEEE Trans. Signal Process.* **61**.2 (Jan. 2013), pp. 340–354.

[Zin12] Ziniel, J. et al. “A generalized framework for learning and recovery of structured sparse signals”. *Proc. IEEE Stat. Signal Process. Workshop (SSP)*. Ann Arbor, MI, Aug. 2012, pp. 325–328.

[ZL77] Ziv, J. & Lempel, A. “A universal algorithm for sequential data compression”. *IEEE Trans. Inf. Theory* **23**.3 (May 1977), pp. 337–343.

APPENDICES

APPENDIX

A

APPENDIX FOR CHAPTER 3

This appendix follows the derivation of Barbier and Krzakala [BK15], except for some nuances. Our compressed derivation makes the presentation self-contained.

Plugging (3.31) and the following identity [BK15; Krz12a],

$$1 = \int \exp\Bigg\{ -\sum_{a=1}^{n}\Bigg[\widehat{m}_a\Bigg(m_a N J - \sum_{l=1}^{N}(\widehat{\mathbf{x}}_l^a)^\top \mathbf{x}_l\Bigg)\Bigg] + \sum_{a=1}^{n}\Bigg[\widehat{Q}_a\Bigg(Q_a\frac{NJ}{2} - \frac{1}{2}\sum_{l=1}^{N}(\widehat{\mathbf{x}}_l^a)^\top \widehat{\mathbf{x}}_l^a\Bigg)\Bigg] - \sum_{1\le a<b\le n}\Bigg[\widehat{q}_{ab}\Bigg(q_{ab}NJ - \sum_{l=1}^{N}(\widehat{\mathbf{x}}_l^a)^\top \widehat{\mathbf{x}}_l^b\Bigg)\Bigg]\Bigg\} \prod_{a=1}^{n} dQ_a\, d\widehat{Q}_a\, dm_a\, d\widehat{m}_a \prod_{1\le a<b\le n} dq_{ab}\, d\widehat{q}_{ab},$$

into (3.10), we obtain

$$\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[Z^n] = (2\pi\sigma_Z^2)^{-\frac{nMJ}{2}} \int \exp\Bigg[NJ\Bigg(\frac{1}{2}\sum_{a=1}^{n}\widehat{Q}_a Q_a - \frac{1}{2}\sum_{\substack{1\le a,b\le n\\ a\neq b}}\widehat{q}_{ab}q_{ab} - \sum_{a=1}^{n}\widehat{m}_a m_a\Bigg)\Bigg]\Bigg[\prod_{\mu=1}^{M}\mathbb{X}_\mu\Bigg]\times \Gamma^N \prod_{a=1}^{n} dQ_a\, d\widehat{Q}_a\, dm_a\, d\widehat{m}_a \prod_{\substack{1\le a,b\le n\\ a\neq b}} dq_{ab}\, d\widehat{q}_{ab}, \tag{A.1}$$

where

$$\Gamma = \int f(\mathbf{x}_1)\left[\prod_{a=1}^{n} f(\widehat{\mathbf{x}}_1^a)\right]\exp\left[-\frac{1}{2}\sum_{a=1}^{n}\widehat{Q}_a(\widehat{\mathbf{x}}_1^a)^\top\widehat{\mathbf{x}}_1^a + \frac{1}{2}\sum_{\substack{1\le a,b\le n\\ a\neq b}}\widehat{q}_{ab}(\widehat{\mathbf{x}}_1^a)^\top\widehat{\mathbf{x}}_1^b + \sum_{a=1}^{n}\widehat{m}_a(\widehat{\mathbf{x}}_1^a)^\top\mathbf{x}_1\right]d\mathbf{x}_1\prod_{a=1}^{n}d\widehat{\mathbf{x}}_1^a. \tag{A.2}$$

Further simplification of (3.10): The Stratanovitch transform [Str] in J dimensions is given by

$$\begin{aligned}\exp\left[\frac{\widehat{q}}{2}\sum_{\substack{1\le a,b\le n\\ a\neq b}}(\widehat{\mathbf{x}}_1^a)^\top\widehat{\mathbf{x}}_1^b\right] &= \prod_{j=1}^{J}\exp\left[\frac{\widehat{q}}{2}\sum_{\substack{1\le a,b\le n\\ a\neq b}}\widehat{x}_{1,j}^a\widehat{x}_{1,j}^b\right]\\ &= \prod_{j=1}^{J}\int\exp\left[\sqrt{\widehat{q}}h_j\sum_{a=1}^{n}\widehat{x}_{1,j}^a - \frac{\widehat{q}}{2}\sum_{a=1}^{n}\left(\widehat{x}_{1,j}^a\right)^2\right]\mathscr{D}h_j \\ &= \int\exp\left[\sqrt{\widehat{q}}\mathbf{h}^\top\sum_{a=1}^{n}\widehat{\mathbf{x}}_1^a - \frac{\widehat{q}}{2}\sum_{a=1}^{n}\left(\widehat{\mathbf{x}}_1^a\right)^\top\widehat{\mathbf{x}}_1^a\right]\mathscr{D}\mathbf{h},\end{aligned} \tag{A.3}$$

where $\mathbf{h} = [h_1, ..., h_J]^\top$, and the differential $\mathscr{D}h_j = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-h_j^2/2}\,dh_j$. With the Stratanovitch transform (A.3), we simplify Γ (A.2) as follows,

$$\Gamma = \int f(\mathbf{x}_1)\int\left[f(\mathbf{h})\right]^n\mathscr{D}\mathbf{h}\;d\mathbf{x}_1, \tag{A.4}$$

where $f(\mathbf{h}) = \int f(\mathbf{x}_1)\mathrm{e}^{-\frac{\widehat{Q}+\widehat{q}}{2}\widehat{\mathbf{x}}_1^\top\widehat{\mathbf{x}}_1+\widehat{m}\widehat{\mathbf{x}}_1^\top\mathbf{x}_1+\sqrt{\widehat{q}}\mathbf{h}^\top\widehat{\mathbf{x}}_1}\,d\widehat{\mathbf{x}}_1$, and we drop the super-script a of $\widehat{\mathbf{x}}_1^a$ owing to the replica symmetry assumption [Krz12a; Krz12b]. In the limit of $n \to 0$, using another Taylor series $[f(\mathbf{h})]^n \approx 1 + n\log[f(\mathbf{h})]$, we have $\int[f(\mathbf{h})]^n\mathscr{D}\mathbf{h} \approx 1 + n\int\log[f(\mathbf{h})]\mathscr{D}\mathbf{h} \approx \mathrm{e}^{n\int\log[f(\mathbf{h})]\mathscr{D}\mathbf{h}}$, so that $\mathbb{E}\left\{\int[f(\mathbf{h})]^n\mathscr{D}\mathbf{h}\right\} \approx \mathbb{E}\left\{1 + n\int\log[f(\mathbf{h})]\mathscr{D}\mathbf{h}\right\} \approx \mathrm{e}^{\mathbb{E}\{n\int\log[f(\mathbf{h})]\mathscr{D}\mathbf{h}\}}$. Hence, we can approximate (A.4) as

$$\Gamma = \exp\left\{n\int f(\mathbf{x}_1)\int\log[f(\mathbf{h})]\mathscr{D}\mathbf{h}\;d\mathbf{x}_1\right\}. \tag{A.5}$$

Considering (A.5), we rewrite (A.1) as

$$\mathbb{E}_{\mathbf{A},\mathbf{x},\mathbf{z}}[Z^n] = \int\mathrm{e}^{nN\widetilde{\Phi}_J(m,\widehat{m},q,\widehat{q},Q,\widehat{Q})}\,dm\;d\widehat{m}\;dq\;d\widehat{q}\;dQ\;d\widehat{Q}, \tag{A.6}$$

where $\widetilde{\Phi}_J(m,\widehat{m},q,\widehat{q},Q,\widehat{Q})$ is given below,

$$\widetilde{\Phi}_J(m,\widehat{m},q,\widehat{q},Q,\widehat{Q}) = \frac{J}{2}(Q\widehat{Q}+q\widehat{q}-2m\widehat{m}) - \frac{MJ}{2N}\left[\frac{\rho-2m+\sigma_Z^2+q}{Q-q+\sigma_Z^2}+\log(Q-q+\sigma_Z^2)-\log(\sigma_Z^2)\right]+$$
$$\int f(\mathbf{x}_1)\left\{\int \log\left\{\int f(\widehat{\mathbf{x}}_1)\exp\left[-\frac{1}{2}(\widehat{Q}+\widehat{q})\widehat{\mathbf{x}}_1^\top\widehat{\mathbf{x}}_1+\widehat{m}\widehat{\mathbf{x}}_1^\top\mathbf{x}_1+\sqrt{\widehat{q}}\,\mathbf{h}^\top\widehat{\mathbf{x}}_1\right]d\widehat{\mathbf{x}}_1\right\}\mathscr{D}\mathbf{h}\right\}d\mathbf{x}_1 - \frac{MJ}{2N}\log(2\pi\sigma_Z^2). \tag{A.7}$$

Free energy expression: We now substitute (A.6) into (3.9). Assuming that the limits in (3.9) commute and that we only evaluate (3.9) at optimum points of $\widetilde{\Phi}_J$ (A.7) [BK15; Krz12a; Krz12b], we have $\mathscr{F} = \widetilde{\Phi}_J(m^*,\widehat{m}^*,q^*,\widehat{q}^*,Q^*,\widehat{Q}^*)$, where the asterisks denote stationary points. Next, we calculate the stationary points:

$$\frac{\partial\widetilde{\Phi}_J}{\partial m} = 0 \Rightarrow \widehat{m}^* = \frac{\kappa}{Q^*-q^*+\sigma_Z^2},$$
$$\frac{\partial\widetilde{\Phi}_J}{\partial q} = 0 \Rightarrow \widehat{q}^* = \kappa\frac{\sigma_Z^2+\rho-2m^*+q^*}{(Q^*-q^*+\sigma_Z^2)^2},$$
$$\frac{\partial\widetilde{\Phi}_J}{\partial Q} = 0 \Rightarrow \widehat{Q}^* = \kappa\frac{2m^*-\rho-2q^*+Q^*}{(Q^*-q^*+\sigma_Z^2)^2},$$

where κ (3.3) is the measurement rate. Because we are analyzing the MMSE, we must assume that the estimated prior matches the true underlying prior, which is a Bayesian setting. Thus, $q^* = m^*$ and $Q^* = \rho$ (3.25). Let $E = q^*-2m^*+Q^* = Q^*-q^*$, then we obtain $\widehat{q}^* = \widehat{m}^* = \frac{\kappa}{E+\sigma_Z^2}$ and $\widehat{Q}^* = 0$. Therefore, we solve for the free energy as a function of E in (3.13). Using a change of variables, we obtain (3.14), which is a function of E. Using (3.25), the MSE is

$$D = E+Q-q = E+\frac{\rho}{N} \overset{N\to\infty}{\longrightarrow} E. \tag{A.8}$$

Hence, in the large system limit, we can regard the free energy (3.14) as a function of the MSE, D.

APPENDIX

B

APPENDICES FOR CHAPTER 4

B.1 Impact of the Quantization Error

This appendix provides numerical evidence that (*i*) the quantization error $\mathbf{n}_t$ is independent of the scalar channel noise $\mathbf{w}_t$ (4.11) in the fusion center and (*ii*) $\mathbf{w}_t + \mathbf{n}_t$ is independent of the signal $\mathbf{x}$. In the following, we simulate the AMP equivalent scalar channel in each processor node and in the fusion center. In the interest of simple implementation, we use scalar quantization (SQ) to quantize $\mathbf{f}_t^p$ (4.8) (in each processor node) and hypothesis testing to evaluate (*i*) whether $\mathbf{w}_t$ and $\mathbf{n}_t$ (in the fusion center) are independent and (*ii*) whether $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ are independent. Both parts are necessary for lossy SE (4.12) to hold: part (*i*) ensures that we can predict the variance of $\mathbf{w}_t + \mathbf{n}_t$ by $\sigma_t^2 + PD_t$ and part (*ii*) ensures that lossy MP-AMP falls within the general framework of Bayati and Montanari [BM11] and Rush and Venkataramanan [RV16], so that lossy SE (4.12) holds. Details about our simulation appear below.

Considering (4.5) and (4.8), we obtain that the AMP equivalent scalar channel in each processor node can be expressed as

$$\mathbf{f}_t^p = \frac{1}{P}\mathbf{x} + \mathbf{w}_t^p, \tag{B.1}$$

where $\sum_{p=1}^{P} \mathbf{w}_t^p = \mathbf{w}_t$ (4.5), and the variances of $\mathbf{w}_t^p$ and $\mathbf{w}_t$ can be expressed as $(\sigma_t^p)^2$ and σ_t^2,

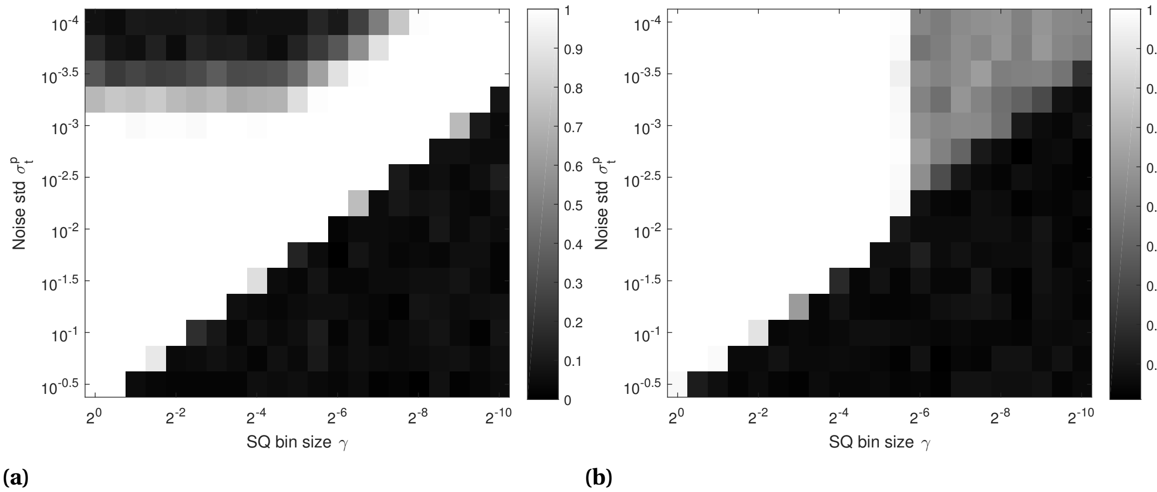


Figure B.1 PCC test results. The darkness of the shades shows the fraction of 100 tests where we reject the null hypothesis (random variables being tested are uncorrelated) with 5% confidence. The horizontal and vertical axes represent the quantization bin size γ of the SQ and the scalar channel noise standard deviation (std) σ_t^p in each processor node, respectively. Panel (a): Test the correlation between $\mathbf{w}_t$ and $\mathbf{n}_t$. Panel (b): Test the correlation between $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$.

respectively (4.12). Hence, we obtain $\sigma_t^2 = \sum_{p=1}^P (\sigma_t^p)^2$. The signal $\mathbf{x}$ follows (4.18) with $\rho = 0.1$. The entries of $\mathbf{w}_t^p$ are i.i.d. and follow $\mathcal{N}(0, (\sigma_t^p)^2)$. Next, we apply an SQ to $\mathbf{f}_t^p$ (B.1),

$$Q(\mathbf{f}_t^p) = \frac{1}{P}\mathbf{x} + \mathbf{w}_t^p + \mathbf{n}_t^p, \tag{B.2}$$

where $Q(\cdot)$ denotes the quantization process, $\mathbf{n}_t^p$ is the quantization error in processor node p, and recall that the variance of $\mathbf{n}_t^p$ is D_t. We simulate the fusion center by calculating

$$\mathbf{f}_t = \sum_{p=1}^{P} Q(\mathbf{f}_t^p) = \mathbf{x} + \mathbf{w}_t + \mathbf{n}_t, \tag{B.3}$$

where $\mathbf{n}_t = \sum_{p=1}^P \mathbf{n}_t^p$. Note that $\mathbf{w}_t$ is Gaussian due to properties of AMP [Don09; Mon12; BM11]. The total quantization error at the fusion center, $\mathbf{n}_t$, is also Gaussian, due to the central limit theorem. Hence, in order to test the independence of $\mathbf{w}_t$ and $\mathbf{n}_t$ (B.3), we need only test whether $\mathbf{w}_t$ and $\mathbf{n}_t$ are uncorrelated. We also test whether $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ are uncorrelated.

We study the settings $\sigma_t^p \in \{10^{-0.5}, \cdots, 10^{-4}\}$ and $\gamma \in \{2^0, \cdots, 2^{-10}\}$, where γ denotes the SQ bin size. In each setting, we simulate (B.1)–(B.3) 100 times and perform 100 Pearson correlation coefficient (PCC) tests [Pcc] for $\mathbf{w}_t$ and $\mathbf{n}_t$, respectively. The null hypothesis of the PCC tests [Pcc] is that $\mathbf{w}_t$ and $\mathbf{n}_t$ are uncorrelated. The null hypothesis is rejected if the resulting p-value is smaller than 0.05.

For each setting, we record the fraction of 100 tests where the null hypothesis is rejected, which is shown by the darkness of the shades in Figure B.1a. The horizontal and vertical axes represent the quantization bin size γ and the standard deviation (std) σ_t^p, respectively. Similarly, we test $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$; results appear in Figure B.1b. We can see that when $\gamma \ll \sigma_t^p$ (bottom right corner), (*i*) $\mathbf{w}_t$ and $\mathbf{n}_t$ tend to be independent and (*ii*) $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ tend to be independent.

Now consider Figure B.1b, which provides PCC test results evaluating possible correlations between $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$. There appears to be a phase transition that separates regions where $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ seem independent or dependent. We speculate that this phase transition is related to the pdf of $\frac{1}{P}\mathbf{x} + \mathbf{w}_t^p$. To explain our hypothesis, note that when the noise $\mathbf{w}_t^p$ is low (top part of Figure B.1b), the phase transition is less affected by noise, and the role of γ is smaller. By contrast, large noise (bottom) sharpens the phase transition.

In summary, it appears that when $\gamma < 2\sigma_t^p = \frac{2\sigma_t}{\sqrt{P}}$, we can regard (*i*) $\mathbf{w}_t$ and $\mathbf{n}_t$ to be independent and (*ii*) $\mathbf{w}_t + \mathbf{n}_t$ and $\mathbf{x}$ to be independent. The requirement $\gamma < 2\sigma_t^p = \frac{2\sigma_t}{\sqrt{P}}$ is motivated by Widrow and Kollár [WK08]; we leave the study of this phase transition for future work.

B.2 Numerical Evidence for Lossy SE

This appendix provides numerical evidence for lossy SE (4.23). We simulate two signal types, one is the Bernoulli-Gaussian signal (4.18) and the other is a mixture Gaussian.

Bernoulli-Gaussian signals: We generate 50 signals of length 10000 according to (4.18). These signals are measured by $M = 5000$ measurements spread over $P = 100$ distributed nodes. We estimate each of these signals by running $T = 10$ MP-AMP iterations. ECSQ is used to quantize $\mathbf{f}_t^p$ (B.1), and $Q(\mathbf{f}_t^p)$ (B.2) is encoded at coding rate R_t. We simulate settings with sparsity rate $\rho \in \{0.1, 0.2\}$ and noise variance $\sigma_Z^2 \in \{0.01, 0.001\}$. In each setting, we randomly generate the coding rate sequence $\mathbf{R}$, s.t. the quantization bin size at each iteration satisfies $\gamma < \frac{2\sigma_t}{\sqrt{P}}$ (details in Appendix B.1).[1] A Bayesian denoiser, $\eta_t(\cdot) = \mathbb{E}[\mathbf{x}|\mathbf{f}_t]$, is used in (4.10). The resulting MSE's from the MP-AMP simulation averaged over the 50 signals, along with MSE's predicted by lossy SE (4.23), are plotted in Figure B.2a. We can see that the simulated MSE's are close to the MSE's predicted by lossy SE.

Mixture Gaussian signals: We independently generate 50 signals of length 10000 according to $X = \sum_{i \in \{0,1,2\}} \mathbb{1}_{X_B = i} X_{G,i}$ where $X_B \sim \text{cat}(0.5, 0.3, 0.2)$ follows a categorical distribution on alphabet $\{0, 1, 2\}$, $X_{G,0} \sim \mathcal{N}(0, 0.1)$, $X_{G,1} \sim \mathcal{N}(-1.5, 0.8)$, and $X_{G,2} \sim \mathcal{N}(2, 1)$. We simulate settings with $T = 10$,

[1] Note that the constraint on γ implies that $\mathbf{R}$ is likely monotone non-decreasing.

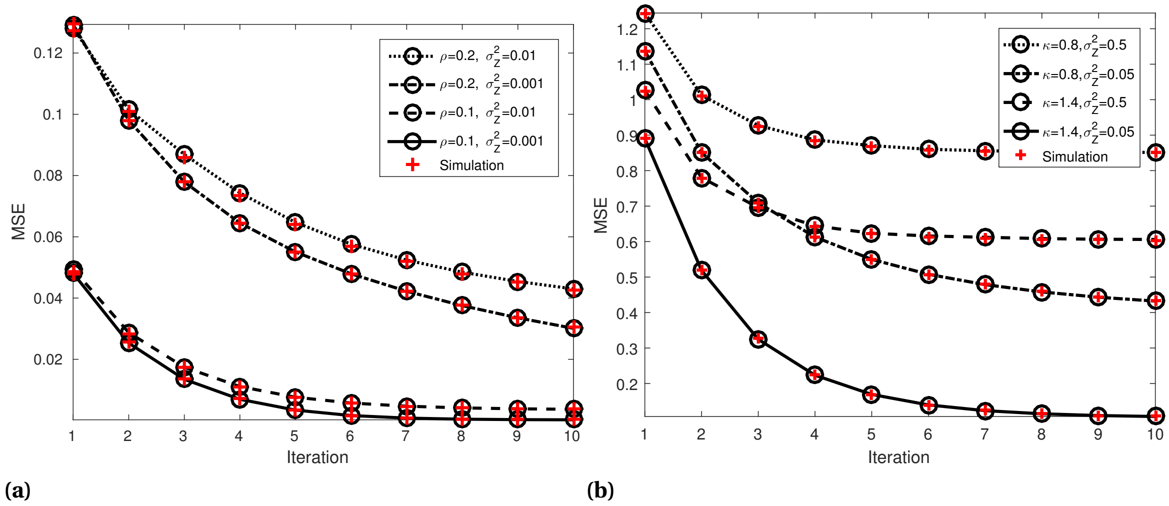


Figure B.2 Comparison of the MSE predicted by lossy SE (4.12) and the MSE of MP-AMP simulations for various settings. The round markers represent MSE's predicted by lossy SE, and the (red) crosses represent simulated MSE's. Panel (a): Bernoulli-Gaussian signal. Panel (b): Mixture Gaussian signal.

$P = 100$, $\kappa = \frac{M}{N} \in \{0.8, 1.6\}$, and $\sigma_Z^2 \in \{0.5, 0.05\}$. In each setting, we randomly generate the coding rate sequence $\mathbf{R}$, s.t. the quantization bin size at each iteration satisfies $\gamma < \frac{2\sigma_t}{\sqrt{P}}$. The results are plotted in Figure B.2b. The simulation results match well with the lossy SE predictions.

B.3 Integrity of Discretized Search Space

When a coding rate $\widehat{R}$ is selected in MP-AMP iteration t, DP calculates the equivalent scalar channel noise variance σ_{t+1}^2 (4.11) for the next MP-AMP iteration according to (4.12). The variance σ_{t+1}^2 is unlikely to lie on the discretized search space for σ_t^2, denoted by the *grid* $\mathscr{G}(\sigma^2)$. Therefore, $\Phi_{T-(t+1)}(\sigma_{t+1}^2(\widehat{R}))$ in (4.17) does not reside in memory. Instead of brute-force calculation of $\Phi_{\{\cdot\}}(\cdot)$, we estimate it by fitting a function to the closest neighbors of σ_{t+1}^2 that lie on the grid $\mathscr{G}(\sigma^2)$ and finding $\Phi_{\{\cdot\}}(\cdot)$ according to the fit function. We evaluate a linear interpolation scheme.

Interpolation in $\mathscr{G}(\sigma^2)$: We run DP over the original coarse grid $\mathscr{G}^c(\sigma^2)$ with resolution $\Delta\sigma^2 = 0.01$ dB, and a 4× finer grid $\mathscr{G}^f(\sigma^2)$ with $\Delta\sigma^2 = 0.0025$ dB. We obtain the cost function with the coarse grid $\Phi_{T-t}^c((\sigma_t^2)_c)$ and the cost function with the fine grid $\Phi_{T-t}^f((\sigma_t^2)_f)$, $\forall t \in \{1, ..., T\}, (\sigma_t^2)_c \in \mathscr{G}^c(\sigma^2), (\sigma_t^2)_f \in \mathscr{G}^f(\sigma^2)$. Next, we interpolate $\Phi_{T-t}^c((\sigma_t^2)_c)$ over the fine grid $\mathscr{G}^f(\sigma^2)$ and obtain the interpolated $\Phi_{T-t}^i((\sigma_t^2)_c)$. In order to compare $\Phi_{T-t}^i((\sigma_t^2)_c)$ with $\Phi_{T-t}^f((\sigma_t^2)_c)$ in a comprehensive way, we consider the settings given by the Cartesian product of the following variables: (*i*) the

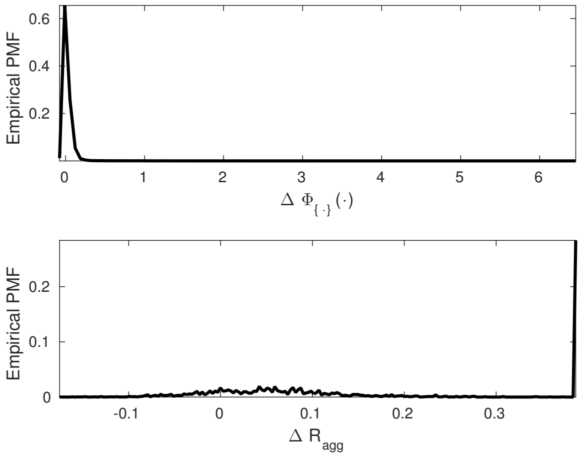


Figure B.3 Justification of the discretized search space used in DP. Top panel: Empirical PMF of the error in the cost function $\Delta\Phi_{\{\cdot\}}(\cdot)$ used to verify the integrity of the linear interpolation in the discretized search space of σ^2. Bottom panel: Empirical PMF of ΔR_{agg}; used to verify the integrity of the choice of $\Delta R = 0.1$.

number of distributed nodes $P \in \{50, 100\}$, (*ii*) sparsity rate $\rho \in \{0.1, 0.2\}$, (*iii*) measurement rate $\kappa = \frac{M}{N} \in \{3\rho, 5\rho\}$, (*iv*) EMSE $\epsilon_T \in \{1, 0.5\}$dB, (*v*) parameter $b \in \{0.5, 2\}$, and (*vi*) noise variance $\sigma_Z^2 \in \{0.01, 0.001\}$. In total, there are 64 different settings. We calculate the error $\Delta\Phi_{T-t}\left((\sigma_t^2)_c\right) = \Phi_{T-t}^i\left((\sigma_t^2)_c\right) - \Phi_{T-t}^f\left((\sigma_t^2)_c\right)$ and plot the empirical probability mass function (PMF) of $\Delta\Phi_{T-t}\left((\sigma_t^2)_c\right)$ over all t, $(\sigma_t^2)_c$, and all 64 settings. The resulting empirical PMF of $\Delta\Phi_{\{\cdot\}}(\cdot)$ is plotted in the top panel of Figure B.3. We see that with 99% probability, the error satisfies $\Delta\Phi_{\{\cdot\}}(\cdot) \leq 0.2$, which corresponds to an inaccuracy of approximately 0.2 in the aggregate coding rate R_{agg}.[2] In the simulation, we used a resolution of $\Delta R = 0.1$. Hence, the inaccuracy of 0.2 in R_{agg} (over roughly 10 iterations) is negligible. Therefore, we use linear interpolation with a coarse grid $\mathscr{G}^c(\sigma^2)$ with $\Delta\sigma^2 = 0.01$ dB.

Integrity of choice of ΔR: We tentatively select resolution $\Delta R = 0.1$, and investigate the integrity of this ΔR over the 64 different settings above. After the coding rate sequence $\mathbf{R}^* = (R_1^*, \cdots, R_T^*)$ is obtained by DP for each setting, we randomly perturb R_t^* by $R_p(t) = R_t^* + \beta_t,\ t = 1, ..., T$, where $R_p(t)$ is the *perturbed coding rate*, the bias is $\beta_t \in \left[-\frac{\Delta R}{2}, +\frac{\Delta R}{2}\right]$, and $\mathbf{R}_p = (R_p(1), \cdots, R_p(T))$ is called the *perturbed coding rate sequence*. After randomly generating 100 different perturbed coding rate sequences $\mathbf{R}_p$, we calculate the aggregate coding rate (4.14), R_{agg}^p, of each $\mathbf{R}_p$; we only consider the perturbed coding rate sequences that achieve EMSE no greater than the optimal coding rate sequence $\mathbf{R}^*$ given by DP. The bottom panel of Figure B.3 plots the empirical PMF of ΔR_{agg}, where $\Delta R_{agg} = R_{agg}^p - R_{agg}^*$ and $R_{agg}^* = ||\mathbf{R}^*||_1$. Roughly 15% of cases in our simulation yield $\Delta R_{agg} < 0$

[2] Note that when calculating Φ^f, we are still using the corresponding interpolation scheme. Although this comparison is not ideal, we believe it still provides the reader with enough insight.

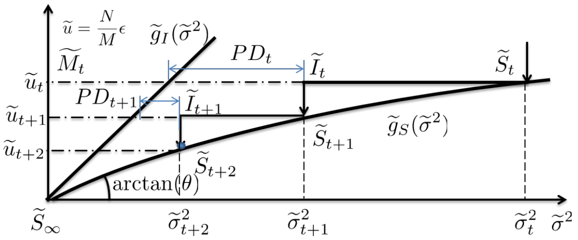


Figure B.4 Illustration of the evolution of $\widetilde{u}_t$. The vertical axis shows $\widetilde{u}_t = \frac{N}{M}\text{EMSE} = \frac{N}{M}\epsilon_t$. The solid lines with arrows denote the lossy SE associated with a coding rate sequence and dashed-dotted lines are auxiliary lines.

(meaning that the perturbed coding rate sequence has lower R_{agg}), while for the other 85% cases, $\mathbf{R}^*$ has lower R_{agg}. Considering the resolution $\Delta R = 0.1$, we can see that the perturbed sequences are only marginally better than $\mathbf{R}^*$. Hence, we verified the integrity of $\Delta R = 0.1$.

B.4 Proof of Lemma 4.1

Proof. We show that our DP scheme (4.17) fits into Bertsekas' formulation [Ber95], which has been proved to be optimal. Under Bertsekas' formulation, our decision variable is the coding rate R_t and our state is the scalar channel noise variance σ_t^2. Our next-state function is the lossy SE (4.12) with the distortion D_t being calculated from the RD function given the decision variable R_t. Our additive cost associated with the dynamic system is $b \times \mathbb{1}_{R_t \neq 0} + R_t$. Our control law maps the state σ_t^2 to a decision (the coding rate R_t). Therefore, our DP formulation (4.17) fits into the optimal DP formulation of Bertsekas [Ber95]. Hence, our DP formulation (4.17) is also optimal *for the discretized search spaces of R_t and σ_t^2.* □

B.5 Proof of Theorem 4.1

Proof. Our proof is based on the assumption that lossy SE (4.12) holds. Consider the geometry of the SE incurred by $\mathbf{R}^*$ for arbitrary iterations t and $t+1$, as shown in Figure B.4. Let $\widetilde{S}_t = (\widetilde{\sigma}_t^2, \widetilde{u}_t)$ and R_t^* be the state and the optimal coding rate at iteration t, respectively. We know that the slope of $\widetilde{g}_I(\cdot)$ is $\widetilde{g}_I'(\cdot) = 1$. Hence, the length of line segment $\widetilde{M}_t\widetilde{I}_t$ is $\widetilde{\sigma}_{t+1}^2 = \widetilde{u}_t + PD_t$. That is

$$PD_t = \widetilde{\sigma}_{t+1}^2 - \widetilde{u}_t. \tag{B.4}$$

Similarly, we obtain

$$PD_{t+1} = \widetilde{\sigma}^2_{t+2} - \widetilde{u}_{t+1}, \tag{B.5}$$

where $\widetilde{u}_{t+1}$ and $\widetilde{\sigma}^2_{t+1}$ obey

$$\widetilde{\sigma}^2_{t+1} = \widetilde{g}_S^{-1}(\widetilde{u}_{t+1}). \tag{B.6}$$

Recall that, according to Taylor's theorem (4.21), we obtain that

$$\widetilde{g}_S^{-1}(\widetilde{u}_{t+1}) = \frac{1}{\theta}\widetilde{u}_{t+1} + C\widetilde{u}^2_{t+1}, \tag{B.7}$$

with θ defined in (4.22). Although C depends on $\widetilde{u}_{t+1}$, it is uniformly bounded, i.e., $C \in [-B, B]$ for some $0 \le B < \infty$.

Fixing $\widetilde{u}_t = \frac{N}{M}\epsilon^*_t$ and $\widetilde{u}_{t+2} = \frac{N}{M}\epsilon^*_{t+2}$, we explore different distortions D_t and D_{t+1} that obey (B.4)–(B.6). According to Definition 4.2, among distortions that obey (B.4)–(B.6), the optimal D^*_t and D^*_{t+1} correspond to the smallest aggregate rate at iterations t and $t+1$, $R_t + R_{t+1}$. Considering (4.24), we have

$$R_t + R_{t+1} = \left[\frac{1}{2}\log_2\left(\frac{C_1}{D_t}\right) + \frac{1}{2}\log_2\left(\frac{C_1}{D_{t+1}}\right)\right](1 + o_t(1)).$$

Therefore, in the large t limit, minimizing $R_t + R_{t+1}$ is identical to maximizing the product $D_t D_{t+1}$. Considering (B.4)–(B.6), our optimization problem becomes maximization over $F(\widetilde{u}_{t+1})$, where

$$F(\widetilde{u}_{t+1}) = (\widetilde{\sigma}^2_{t+2} - \widetilde{u}_{t+1})(\widetilde{g}_S^{-1}(\widetilde{u}_{t+1}) - \widetilde{u}_t). \tag{B.8}$$

Invoking Taylor's theorem (B.7) and considering that $C \in [-B, B]$, we solve the optimization problem (B.8) in two extremes: one with $C = B$ and the other with $C = -B$.

In the case of $C = B$, we obtain

$$F(\widetilde{u}_{t+1}) = -\frac{1}{\theta}\widetilde{u}^2_{t+1} + \frac{1}{\theta}\widetilde{u}_{t+1}\widetilde{\sigma}^2_{t+2} + B\widetilde{\sigma}^2_{t+2}\widetilde{u}^2_{t+1} - B\widetilde{u}^3_{t+1} - \widetilde{u}_t\widetilde{\sigma}^2_{t+2} + \widetilde{u}_t\widetilde{u}_{t+1}.$$

The maximum of $F(\widetilde{u}_{t+1})$ is achieved when $F'(\widetilde{u}_{t+1}) = 0$. That is,

$$F'(\widetilde{u}_{t+1}) = -3B\widetilde{u}^2_{t+1} + \left(2B\widetilde{\sigma}^2_{t+2} - \frac{2}{\theta}\right)\widetilde{u}_{t+1} + \frac{\widetilde{\sigma}^2_{t+2}}{\theta} + \widetilde{u}_t = 0. \tag{B.9}$$

Considering that $0 < \widetilde{u}_{t+1} < \widetilde{u}_t$, the root of the quadratic equation (B.9) is

$$\widetilde{u}^*_{t+1} = \frac{1}{3B}\left[\left(B\widetilde{\sigma}^2_{t+2} - \frac{1}{\theta}\right) + A\right], \tag{B.10}$$

where

$$A = \sqrt{\left(B\widetilde{\sigma}_{t+2}^2 - \frac{1}{\theta}\right)^2 + 3B\left(\frac{\widetilde{\sigma}_{t+2}^2}{\theta} + \widetilde{u}_t\right)}. \tag{B.11}$$

We can further simplify (B.11) as

$$\begin{aligned} A =& \frac{1}{\theta}\sqrt{1 + B(\theta\widetilde{\sigma}_{t+2}^2 + B\theta^2\widetilde{\sigma}_{t+2}^4 + 3\theta^2\widetilde{u}_t)} \\ =& \frac{1}{\theta}\left[1 + \frac{B}{2}(\theta\widetilde{\sigma}_{t+2}^2 + B\theta^2\widetilde{\sigma}_{t+2}^4 + 3\theta^2\widetilde{u}_t)\right] + O(\widetilde{u}_t^2), \end{aligned} \tag{B.12}$$

Plugging (B.12) into (B.10),

$$\widetilde{u}_{t+1}^* = \frac{1}{2}(\widetilde{\sigma}_{t+2}^2 + \theta\,\widetilde{u}_t) + O(\widetilde{u}_t^2). \tag{B.13}$$

Plugging (B.13) into (B.4) and (B.5),

$$\begin{aligned} PD_t^* =& \frac{1}{2\theta}(\widetilde{\sigma}_{t+2}^2 - \widetilde{u}_t\theta) + O(\widetilde{u}_t^2), \\ PD_{t+1}^* =& \frac{1}{2}(\widetilde{\sigma}_{t+2}^2 - \widetilde{u}_t^2\theta) + O(\widetilde{u}_t^2), \end{aligned}$$

which leads to

$$\frac{D_{t+1}^*}{D_t^*} = \theta(1 + O(\widetilde{u}_t)). \tag{B.14}$$

These steps provided the optimal relation between D_t^* and D_{t+1}^* when $C = B$. For the other extreme case, $C = -B$, similar steps will lead to (B.14), where the differences between the results are higher order terms. Note that for any $C \in [-B, B]$ the higher order term is bounded between the two extremes. Hence, the optimal D_t^* and D_{t+1}^* follow (B.14) leading to the first part of the claim (4.25). Considering (4.24) and (B.14),

$$R_{t+1}^* - R_t^* = \frac{1}{2}\log_2\left(\frac{1}{\theta}\right)(1 + o_t(1)).$$

Therefore, we obtain the second part of the claim (4.26). □

B.6 Proof of Theorem 4.2

Proof. Our proof is based on the assumption that lossy SE (4.12) holds. Let us focus on an optimal coding rate sequence $\mathbf{R}^* = (R_1^*, \cdots, R_T^*)$. Applying Taylor's theorem to calculate the ordinate of point

$\widetilde{S}_{t+1}$ using its abscissa (Figure B.4), we obtain

$$\widetilde{u}^*_{t+1} = \theta(\widetilde{u}^*_t + PD^*_t) + O((\widetilde{u}^*_t)^2). \tag{B.15}$$

Therefore,

$$\frac{\widetilde{u}^*_{t+1}}{\widetilde{u}^*_t} = \theta + \frac{\theta PD^*_t}{\widetilde{u}^*_t} + O(\widetilde{u}^*_t). \tag{B.16}$$

Similarly, we obtain

$$\frac{\widetilde{u}^*_{t+2}}{\widetilde{u}^*_{t+1}} = \theta + \frac{\theta PD^*_{t+1}}{\widetilde{u}^*_{t+1}} + O(\widetilde{u}^*_t). \tag{B.17}$$

Plugging (B.14) and (B.15) into (B.17), we obtain

$$\begin{aligned}\frac{\widetilde{u}^*_{t+2}}{\widetilde{u}^*_{t+1}} &= \theta + \frac{\theta PD^*_t(1+O(u^*_t))}{\widetilde{u}^*_t + PD^*_t + O((\widetilde{u}^*_t)^2)} + O(\widetilde{u}^*_t)\\ &= \theta + \frac{\theta PD^*_t}{\widetilde{u}^*_t + PD^*_t} + O(\widetilde{u}^*_t).\end{aligned} \tag{B.18}$$

On the other hand, $\lim_{t\to\infty} \frac{\widetilde{u}^*_{t+1}}{\widetilde{u}^*_t} = \lim_{t\to\infty} \frac{\widetilde{u}^*_{t+2}}{\widetilde{u}^*_{t+1}}$. Therefore, considering (B.16) and (B.18), we obtain

$$\lim_{t\to\infty} \frac{\theta PD^*_t}{\widetilde{u}^*_t} = \lim_{t\to\infty} \frac{\theta PD^*_t}{\widetilde{u}^*_t + PD^*_t},$$

which leads to $\lim_{t\to} \frac{D^*_t}{\widetilde{u}^*_t} = 0$. We obtain (4.27) by noting that the optimal EMSE at iteration t is $\epsilon^*_t = \frac{M}{N}\widetilde{u}^*_t$. Plugging (4.27) into (B.16), we obtain (4.28). □

APPENDIX

C

APPENDICES FOR CHAPTER 5

C.1 Proof of Theorem 5.1

Our proof mimics a very similar proof presented in [JW08; JW12] for lossy source coding; we include all details for completeness. The proof technique relies on mathematical properties of non-homogeneous (e.g., time-varying) Markov chains (MC's) [Bré99]. Through the proof, $\mathscr{S} \triangleq (\mathscr{R}_F)^N$ denotes the state space of the MC of codewords generated by Algorithm 5.1, with size $|\mathscr{S}| = |\mathscr{R}_F|^N$. We define a stochastic transition matrix $\mathbf{P}_{(t)}$ from $\mathscr{S}$ to itself given by the Boltzmann distribution for super-iteration t in Algorithm 5.1. Similarly, $\pi_{(t)}$ defines the stable-state distribution on $\mathscr{S}$ for $\mathbf{P}_{(t)}$, satisfying $\pi_{(t)}\mathbf{P}_{(t)} = \pi_{(t)}$.

Definition C.1. *[Bré99] Dobrushin's ergodic coefficient of an MC transition matrix* $\mathbf{P}$ *is denoted by* $\xi(\mathbf{P})$ *and defined as* $\xi(\mathbf{P}) \triangleq \max_{1\le i,j\le N} \frac{1}{2}\|\mathbf{P}_i - \mathbf{P}_j\|_1$, *where* $\mathbf{P}_i$ *denotes the* i*-th row of* $\mathbf{P}$.

From the definition, $0 \le \xi(\mathbf{P}) \le 1$. Moreover, the ergodic coefficient can be rewritten as

$$\xi(\mathbf{P}) = 1 - \min_{1\le i,j\le N} \sum_{k=1}^{N} \min(P_{ik}, P_{jk}), \tag{C.1}$$

where P_{ij} denotes the entry of $\mathbf{P}$ at the i-th row, j-th column.

We group the product of transition matrices across super-iterations as $\mathbf{P}_{(t_1\to t_2)} = \prod_{t=t_1}^{t_2} \mathbf{P}_{(t)}$. There are two common characterizations for the stable-state behavior of a non-homogeneous MC.

Definition C.2. *[Bré99] A non-homogeneous MC is called weakly ergodic if for any distributions η and ν over the state space $\mathscr{S}$, and any $t_1 \in \mathbb{N}$, $\limsup_{t_2\to\infty} \|\eta\mathbf{P}_{(t_1\to t_2)} - \nu\mathbf{P}_{(t_1\to t_2)}\|_1 = 0$, where $\|\cdot\|_1$ denotes the ℓ_1 norm. Similarly, a non-homogeneous MC is called strongly ergodic if there exists a distribution π over the state space $\mathscr{S}$ such that for any distribution η over $\mathscr{S}$, and any $t_1 \in \mathbb{N}$, $\limsup_{t_2\to\infty} \|\eta\mathbf{P}_{(t_1\to t_2)} - \pi\|_1 = 0$. We will use the following two theorems from [Bré99] in our proof.*

Theorem C.1. *[Bré99] An MC is weakly ergodic if and only if there exists a sequence of integers $0 \le t_1 \le t_2 \le \cdots$ such that $\sum_{i=1}^{\infty}\left(1 - \xi\left(\mathbf{P}_{(t_i\to t_{i+1})}\right)\right) = \infty$.*

Theorem C.2. *[Bré99] Let an MC be weakly ergodic. Assume that there exists a sequence of probability distributions $\{\pi_{(t)}\}_{i=1}^{\infty}$ on the state space $\mathscr{S}$ such that $\pi_{(t)}\mathbf{P}_{(t)} = \pi_{(t)}$. Then the MC is strongly ergodic if $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$.*

The rest of proof is structured as follows. First, we show that the sequence of stable-state distributions for the MC used by Algorithm 5.1 converges to a uniform distribution over the set of sequences that minimize the energy function as the iteration count t increases. Then, we show using Theorems C.1 and C.2 that the non-homogeneous MC used in Algorithm 5.1 is strongly ergodic, which by the definition of strong ergodicity implies that Algorithm 5.1 always converges to the stable distribution found above. This implies that the outcome of Algorithm 5.1 converges to a minimum-energy solution as $t \to \infty$, completing the proof of Theorem 5.1.

We therefore begin by finding the stable-state distribution for the non-homogeneous MC used by Algorithm 5.1. At each super-iteration t, the distribution defined as

$$\pi_{(t)}(\mathbf{w}) \triangleq \frac{\exp\left(-s_t \Psi^{H_q}(\mathbf{w})\right)}{\sum_{\mathbf{z}\in\mathscr{S}} \exp\left(-s_t \Psi^{H_q}(\mathbf{z})\right)} = \frac{1}{\sum_{\mathbf{z}\in\mathscr{S}} \exp\left(-s_t \left(\Psi^{H_q}(\mathbf{z}) - \Psi^{H_q}(\mathbf{w})\right)\right)} \tag{C.2}$$

satisfies $\pi_{(t)}\mathbf{P}_{(t)} = \pi_{(t)}$, cf. (5.12). We can show that the distribution $\pi_{(t)}$ converges to a uniform distribution over the set of sequences that minimize the energy function, i.e.,

$$\lim_{t\to\infty} \pi_{(t)}(\mathbf{w}) = \begin{cases} 0 & \mathbf{w} \notin \mathscr{H}, \\ \frac{1}{|\mathscr{H}|} & \mathbf{w} \in \mathscr{H}, \end{cases} \tag{C.3}$$

where $\mathscr{H} = \{\mathbf{w} \in \mathscr{S}$ subject to $\Psi^{H_q}(\mathbf{w}) = \min_{\mathbf{z}\in\mathscr{S}} \Psi^{H_q}(\mathbf{z})\}$. To show (C.3), we will show that $\pi_{(t)}(\mathbf{w})$ is increasing for $\mathbf{w} \in \mathscr{H}$ and eventually decreasing for $\mathbf{w} \in \mathscr{H}^C$. Since for $\mathbf{w} \in \mathscr{H}$ and $\widetilde{\mathbf{w}} \in \mathscr{S}$ we have

$\Psi^{H_q}(\widetilde{\mathbf{w}}) - \Psi^{H_q}(\mathbf{w}) \geq 0$, for $t_1 < t_2$ we have

$$\sum_{\widetilde{\mathbf{w}} \in \mathscr{S}} \exp\left[-s_{t_1}\left(\Psi^{H_q}(\widetilde{\mathbf{w}}) - \Psi^{H_q}(\mathbf{w})\right)\right] \geq \sum_{\widetilde{\mathbf{w}} \in \mathscr{S}} \exp\left[-s_{t_2}\left(\Psi^{H_q}(\widetilde{\mathbf{w}}) - \Psi^{H_q}(\mathbf{w})\right)\right],$$

which together with (C.2) implies $\pi_{(t_1)}(\mathbf{w}) \leq \pi_{(t_2)}(\mathbf{w})$. On the other hand, if $\mathbf{w} \in \mathscr{H}^C$, then we obtain

$$\pi_{(t)}(\mathbf{w}) = \left\{ \sum_{\widetilde{\mathbf{w}}:\Psi^{H_q}(\widetilde{\mathbf{w}}) \geq \Psi^{H_q}(\mathbf{w})} \exp\left[-s_t\left(\Psi^{H_q}(\widetilde{\mathbf{w}}) - \Psi^{H_q}(\mathbf{w})\right)\right] + \sum_{\widetilde{\mathbf{w}}:\Psi^{H_q}(\widetilde{\mathbf{w}}) < \Psi^{H_q}(\mathbf{w})} \exp\left[-s_t\left(\Psi^{H_q}(\widetilde{\mathbf{w}}) - \Psi^{H_q}(\mathbf{w})\right)\right] \right\}^{-1}. \tag{C.4}$$

For sufficiently large s_t, the denominator of (C.4) is dominated by the second term, which increases when s_t increases, and therefore $\pi_{(t)}(\mathbf{w})$ decreases for $\mathbf{w} \in \mathscr{H}^C$ as t increases. Finally, since all sequences $\mathbf{w} \in \mathscr{H}$ have the same energy $\Psi^{H_q}(\mathbf{w})$, it follows that the distribution is uniform over the symbols in $\mathscr{H}$.

Having shown convergence of the non-homogenous MC's stable-state distributions, we now show that the non-homogeneous MC is strongly ergodic. The transition matrix $\mathbf{P}_{(t)}$ of the MC at iteration t depends on the temperature s_t in (5.14) used within Algorithm 5.1. We first show that the MC used in Algorithm 5.1 is weakly ergodic via Theorem C.1; the proof of the following Lemma is given in C.2.

Lemma C.1. *The ergodic coefficient of*$\mathbf{P}_{(t)}$ *for any* $t \geq 0$ *is upper bounded by* $\xi\left(\mathbf{P}_{(t)}\right) \leq 1 - exp(-s_t N \Delta_q)$*, where* Δ_q *is defined in (5.13).*

We note in passing that Condition 5.1 ensures that Δ_q is finite. Using Lemma C.1 and (5.14), we can evaluate the sum given in Theorem C.1 as

$$\sum_{j=1}^{\infty}\left[1 - \xi\left(\mathbf{P}_{(j)}\right)\right] \geq \sum_{j=1}^{\infty} \exp(-s_j N \Delta_q) = \sum_{j=1}^{\infty} \frac{1}{j^{1/c}} = \infty,$$

and so the non-homogeneous MC defined by $\{\mathbf{P}_{(t)}\}_{t=1}^{\infty}$ is weakly ergodic. Now we use Theorem C.2 to show that the MC is strongly ergodic by proving that $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$. Since we know from earlier in the proof that $\pi_{(t)}(\mathbf{w})$ is increasing for $\mathbf{w} \in \mathscr{H}$ and eventually decreasing for $\mathbf{w} \in \mathscr{H}^C$, there exists a $t_0 \in \mathbb{N}$ such that for any $t_1 > t_0$, we have

$$\begin{aligned}\sum_{t=t_0}^{t_1} \|\pi_{(t)} - \pi_{(t+1)}\|_1 &= \sum_{\mathbf{w}\in\mathscr{H}} \sum_{t=t_0}^{t_1} \left(\pi_{(t+1)}(\mathbf{w}) - \pi_{(t)}(\mathbf{w})\right) + \sum_{\mathbf{w}\notin\mathscr{H}} \sum_{t=t_0}^{t_1} \left(\pi_{(t)}(\mathbf{w}) - \pi_{(t+1)}(\mathbf{w})\right) \\ &= \sum_{\mathbf{w}\in\mathscr{H}} \left(\pi_{(t_1+1)}(\mathbf{w}) - \pi_{(t_0)}(\mathbf{w})\right) + \sum_{\mathbf{w}\notin\mathscr{H}} \left(\pi_{(t_0)}(\mathbf{w}) - \pi_{(t_1+1)}(\mathbf{w})\right) \\ &= \|\pi_{(t_1+1)} - \pi_{(t_0)}\|_1 \le \|\pi_{(t_1+1)}\|_1 + \|\pi_{(t_0)}\|_1 = 2.\end{aligned}$$

Since the right hand side does not depend on t_1, we have that $\sum_{t=1}^{\infty} \|\pi_{(t)} - \pi_{(t+1)}\|_1 < \infty$. This implies that the non-homogeneous MC used by Algorithm 5.1 is strongly ergodic, and thus completes the proof of Theorem 5.1.

C.2 Proof of Lemma C.1

Let $\mathbf{w}', \mathbf{w}''$ be two arbitrary sequences in $\mathscr{S}$. The probability of transitioning from a given state to a neighboring state within iteration t' of super-iteration t of Algorithm 5.1 is given by (5.12), and can be rewritten as

$$\begin{aligned}\mathbb{P}_{(t,t')}(\mathbf{w}_1{}^{t'-1} a \mathbf{w}_{t'+1}^{N} | \mathbf{w}_1^{t'-1} b \mathbf{w}_{t'+1}^{N}) = \mathbb{P}_{s_t}(w_{t'} = a | \mathbf{w}^{\setminus t'}) &= \frac{\exp\left(-s_t \Psi^{H_q}\left(\mathbf{w}_1^{t'-1} a \mathbf{w}_{t'+1}^{N}\right)\right)}{\sum_{b\in\mathscr{R}_F} \exp\left(-s_t \Psi^{H_q}\left(\mathbf{w}_1^{t'-1} b \mathbf{w}_{t'+1}^{N}\right)\right)} \\ = \frac{\exp\left[-s_t\left(\Psi^{H_q}\left(\mathbf{w}_1^{t'-1} a \mathbf{w}_{t'+1}^{N}\right) - \Psi^{H_q}_{\min,t'}\left(\mathbf{w}_1^{t'-1}, \mathbf{w}_{t'+1}^{N}\right)\right)\right]}{\sum_{b\in\mathscr{R}_F} \exp\left[-s_t\left(\Psi^{H_q}\left(\mathbf{w}_1^{t'-1} b \mathbf{w}_{t'+1}^{N}\right) - \Psi^{H_q}_{\min,t'}\left(\mathbf{w}_1^{t'-1}, \mathbf{w}_{t'+1}^{N}\right)\right)\right]} &\ge \frac{\exp(-s_t \Delta_q)}{|\mathscr{R}_F|},\end{aligned}$$

where $\Psi^{H_q}_{\min,t'}(\mathbf{w}_1^{t'-1}, \mathbf{w}_{t'+1}^{N}) = \min_{\beta\in\mathscr{R}_F} \Psi^{H_q}(\mathbf{w}_1^{t'-1} \beta \mathbf{w}_{t'+1}^{N})$. Therefore, the smallest probability of transition from $\mathbf{w}'$ to $\mathbf{w}''$ within super-iteration t of Algorithm 5.1 is bounded by

$$\min_{\mathbf{w}',\mathbf{w}''\in\mathscr{R}_F} \mathbb{P}_{(t)}(\mathbf{w}''|\mathbf{w}') \ge \prod_{t'=1}^{N} \frac{\exp(-s_t \Delta_q)}{|\mathscr{R}_F|} = \frac{\exp(-s_t N \Delta_q)}{|\mathscr{R}_F|^N} = \frac{\exp(-s_t N \Delta_q)}{|\mathscr{S}|}.$$

Using the alternative definition of the ergodic coefficient (C.1),

$$\begin{aligned}\xi\left(\mathbf{P}_{(t)}\right) &= 1 - \min_{\mathbf{w}',\mathbf{w}''\in\mathscr{S}} \sum_{\tilde{\mathbf{w}}\in\mathscr{S}} \min(\mathbb{P}_{(t)}(\tilde{\mathbf{w}}|\mathbf{w}'), \mathbb{P}_{(t)}(\tilde{\mathbf{w}}|\mathbf{w}'')) \\ &\le 1 - |\mathscr{S}| \frac{\exp(-s_t N \Delta_q)}{|\mathscr{S}|} = 1 - \exp(-s_t N \Delta_q),\end{aligned}$$

proving the lemma.