

MASTER'S THESIS

Optimal Quantization for Sparse Reconstruction with Relaxed Belief Propagation

Ulugbek KAMILOV

Research Laboratory of Electronics (RLE)
Massachusetts Institute of Technology
Audiovisual Communications Laboratory (LCAV)
Ecole Polytechnique Fédérale de Lausanne

Supervisors

Prof. Vivek GOYAL
Prof. Martin VETTERLI

March 14, 2011

Abstract

Compressive sensing theory has demonstrated that sparse signals can be recovered from a small number of random linear measurements. However, for practical purposes, like storage, transmission, or processing with modern digital equipment, continuous-valued compressive sensing measurements need to be quantized. In this thesis we examine the topic of optimal quantization of compressive sensing measurements under reconstruction with message-passing algorithms by following the work on generalization of relaxed belief propagation (BP) for arbitrary measurement channels. Relaxed BP is an iterative reconstruction algorithm proposed for the task of estimation from random linear measurements. It was inspired by the traditional belief propagation algorithm widely used in decoding of low-density parity-check (LDPC) codes. One of the aspects that makes relaxed belief propagation so appealing is the state evolution framework, which predicts asymptotic error behavior of the algorithm. We utilize the predictive capability of the framework to design mean-square optimal scalar quantizers under relaxed BP signal reconstruction. We demonstrate that error performance of the reconstruction can be significantly improved by using state evolution optimized quantizers, compared to quantizers obtained via traditional design schemes. We finally propose relaxed BP as a practical algorithm for reconstruction from measurements digitized with binned quantizers, which further improve error performance of the reconstruction.

Acknowledgments

I would like to express my deepest gratitude to my advisor Prof. Vivek Goyal for making my work and stay at STIR group as stimulating and enjoyable as possible. With your continuous support, great ideas, and unceasing encouragement this project turned into a major learning experience, far beyond anything expected.

I would like to thank Prof. Martin Vetterli who agreed to supervise my work and made this all possible in the first place. I am extremely grateful for the opportunities I have been granted to work in your lab and it is my great honor to have been supervised by you. Your personality and enthusiasm are a great source of inspiration for me.

My stay at MIT would not have been as enriching without all of my colleagues at STIR. Special thanks to Vahid Montazerhodjat for his numerous recommendations and our endless lunch discussions over the “best sandwiches” at Subway, to John Sun for his uncountable words of wisdom and for introducing me to the world of poker, to Dr. Woohyuk Chang for continuously feeding me with great suggestions and for enjoyable football discussions in the office.

This work was done in collaboration with Prof. Sundeep Rangan from Polytechnic Institute of New York University, whom I would like to thank for the tremendous contributions he has made for this thesis. Thank you for your ideas, suggestions, and encouraging comments.

I would like to express my appreciation to Prof. Rüdiger Urbanke for allowing me to do a summer internship in his lab and to familiarize myself under his supervision with message-passing algorithms. Thank you for the enthusiasm and support you have always been giving me.

Very special thanks to Sylviane Dal Mas for never ending advice during the whole five years of my studies and for making many inherently complicated situations seem simple. You truly make I&C an exceptional department.

I would also like to thank my friends across continents, too numerous to name here, whose support and appreciation make me feel valued and happier

at all moments. I will thank you all in person with the nearest opportunity.

Finally, now more than ever I would like to thank my family and Yuji for their unconditional love, understanding, and support during all times. Your love provides me with an extra support to walk in this life and this thesis is dedicated to you.

Contents

1	Introduction	11
2	Background	13
2.1	Sparsity, Compression, Compressive Sensing	13
2.2	Belief Propagation, Relaxed Belief Propagation	21
2.3	Quantization	29
3	RBP for Quantized Measurements	35
3.1	Problem Setting	35
3.2	Relaxed BP	36
3.3	State Evolution for RBP	39
4	Optimal Quantization for RBP Estimation	45
4.1	Optimal Regular Quantization	45
4.2	Optimizing Binned Quantizers	49
5	Conclusions	57

List of Figures

2.1	DWT image compression	15
2.2	Generalized compressive sensing estimation problem	17
2.3	Illustration of signal space.	19
2.4	Geometry of compressive sensing recovery	20
2.5	Illustration of measurement space	20
2.6	Factor graphs for belief propagation	23
2.7	Belief propagation over factor graphs	25
2.8	Regular and binned quantizers	31
2.9	Illustration of quantized measurement space	32
3.1	Compressive sensing set up with quantized measurements	36
3.2	Relaxed belief propagation estimation	40
3.3	Reconstruction with relaxed belief propagation	41
3.4	Comparison of state evolution predictions against observations	44
4.1	State evolution MSE function	47
4.2	Performance comparison of quantizers under RBP reconstruction	50
4.3	Illustration of quantization boundaries	51
4.4	Comparison of optimal uniform and optimal quantizers	52
4.5	Comparison with other methods	53
4.6	Performance comparison between regular and binned quantizers	55

Chapter 1

Introduction

Compressive sensing (CS) is a framework enabling signal acquisition with fewer measurements than traditional sampling [4, 7]. Reduction in the number of measurements is achieved by exploiting the observation that the true dimension of many practical signals is far lower than their ambient dimension and by using some smart reconstruction scheme. Convex optimization based methods, like basis pursuit or LASSO [23], represent one family of standard CS reconstruction schemes providing near-optimal solutions in a variety of settings. These methods can be cast as linear programming (LP) problems and solved using standard algorithms in polynomial time. However, the search for computationally cheaper alternatives to LP-based methods has motivated much research into iterative reconstruction methods [14, 24]. One class of such methods were derived from belief propagation (BP), an estimation algorithm on bipartite graphs widely applied in the context of LDPC codes [19]. Although BP has been applied to CS reconstruction [5], its application in traditional form is computationally prohibitive, as the number of computations grow exponentially with density of the measurement matrix. However, it has been found that the mean-square optimality of BP can be achieved by its simplified version called relaxed BP [11, 16]. Furthermore, it has been shown that the asymptotic error behaviour of relaxed BP can be predicted via state evolution (SE) equations. Although, initially the convergence results for SE required large and sparse measurement matrices [16], recent theoretical work demonstrated that they hold asymptotically for matrices with i.i.d. Gaussian entries as well [1].

So far, most of the CS literature considered signal recovery directly from linear measurements. However, in most practical applications measurements have to be discretized to finite number of bits using some quantization scheme. Traditionally, the quality of such schemes are evaluated on their ability to minimize the error relative to some distortion metric like mean-

squared error (MSE) [10]. Previously the effect of quantized measurements on the performance of the CS reconstruction was studied in [3, 9]. In [6, 25] authors adapt CS reconstruction algorithms to mitigate quantization effects. By contrast in [22] high-resolution functional scalar quantization theory was used to design quantizers for LASSO reconstruction. The topic of binning quantizer output indices has been studied in [15].

In this thesis we utilize the recent work on generalization of relaxed BP to arbitrary output channels to design optimal scalar quantizers. Since CS measurements are used as the input to nonlinear relaxed BP, minimizing MSE between quantizer input and output is not necessarily equivalent to minimizing the MSE of the reconstruction. By modeling the quantizer as part of the measurement channel, we use the SE formalism to optimize the quantizer to asymptotically minimize distortions after the reconstruction by relaxed BP. We note that identical results can be developed for the approximate message passing (AMP) [8, 17] reconstruction algorithm closely related to relaxed BP, but we limit our study to relaxed BP.

This thesis is organized as follows: Chapter 2 gives background on topics considered in this work, including sparse approximations, compressive sensing, belief propagation, and quantization. Chapter 3 presents the exact algorithm for implementing relaxed BP reconstruction method for the CS with quantized measurements. Equations for state evolution to predict the performance of the reconstruction are found in the Chapter 3 as well. Finally in Chapter 4 we will present a scheme for designing asymptotically optimal quantizers for CS framework under relaxed BP reconstruction.

Chapter 2

Background

In this chapter we will develop key concepts related to this thesis work. We start by introducing the novel acquisition framework called compressive sensing. We will then present the highly successful iterative decoding algorithm called belief propagation, which in its simplified form has recently been applied to compressive sensing reconstruction. Finally, we will close this chapter by discussing the topic of scalar quantization. These topics represent active areas of research, and it is not possible to present the depth of these topics in one chapter of a thesis. However along the way we will be providing pointers to relevant publications covering the topics with more detail and rigor.

2.1 Sparsity, Compression, Compressive Sensing

Theory

Consider the problem of sensing or acquiring the signal $\mathbf{f} \in \mathbb{R}^n$ via m -linear measurements

$$z_a = \langle \mathbf{f}, \varphi_a \rangle, \quad a = 1, \dots, m, \quad (2.1)$$

where $\langle \cdot \rangle$ is the inner product between the signal of interest \mathbf{f} and measurement vectors $\varphi_a \in \mathbb{R}^n$. The equation (2.1) can be rewritten as a matrix product

$$\mathbf{z} = \Phi \mathbf{f}, \quad (2.2)$$

where $\mathbf{z} \in \mathbb{R}^m$ is called the measurement vector and $\Phi \in \mathbb{R}^{m \times n}$ is the measurement or sensing matrix. The matrix Φ simply consists of the vectors

$\varphi_1^*, \dots, \varphi_m^*$ stacked as rows¹. The fundamental theorem of linear algebra states that solving (2.2) requires taking as many measurements as there are unknowns. This suggests, that to be able to recover the signal \mathbf{f} from \mathbf{z} we need at least $m = n$ measurements as (2.1). It is likely that n is a very large number, for example, $n = 10^6$ for a digital image of 1 megapixels. However, we might find ourselves in a situation where taking each successive measurement a is expensive, or the total number of measurements m is limited due to the finite number of sensors available.

By extending the fundamental theorem, compressive sensing [4, 7] provides an unorthodox approach to signal acquisition. It relies on the observation that many types of signals can be *sparse* approximated in some convenient basis. This implies, that by finding a suitable orthonormal basis we can approximate \mathbf{f} with a small number of non-zero coefficients. Some well-studied bases, like wavelets, provide sparse expansions for many natural signals, the property widely used in standard compression techniques such as JPEG-2000. The compression in such techniques is partly obtained by storing only the largest coefficients in the sparseness inducing basis. For example, consider Figure 2.1, where we obtain the image (c) by keeping only largest 10% of all wavelet coefficients of the image (a). This illustrates how most of the image information is captured by a small number of large wavelet coefficients.

Using linear algebraic notations the basis expansion of $\mathbf{f} \in \mathbb{R}^n$ can be expressed as a linear combination of the basis vectors $\Psi = [\psi_1 \psi_2 \dots \psi_n]$

$$\mathbf{f} = \sum_{i=1}^n x_i \psi_i, \quad (2.3)$$

where $x_i = \langle \mathbf{f}, \psi_i \rangle$ are coefficients of \mathbf{f} in the basis. The formula (2.3) can be equivalently rewritten as a matrix product

$$\mathbf{f} = \Psi \mathbf{x}, \quad (2.4)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of coefficients, $\Psi \in \mathbb{R}^{n \times n}$ is the matrix with basis vectors ψ_1, \dots, ψ_n as columns. The k -sparse approximation of the signal \mathbf{f} is obtained by sorting the coefficients of \mathbf{x} in descending order and keeping only largest k , while setting the rest to zero. Denote by $\mathbf{x}_k \in \mathbb{R}^n$ the vector containing only largest k coefficients of \mathbf{x} , thus the approximation $\mathbf{f}_k \in \mathbb{R}^n$ of \mathbf{f} is obtained as

$$\mathbf{f}_k = \Psi \mathbf{x}_k. \quad (2.5)$$

¹ \mathbf{v}^* denotes the complex transpose of \mathbf{v} . In our case the vector φ_a is real, thus φ_a^* is simply its transpose.

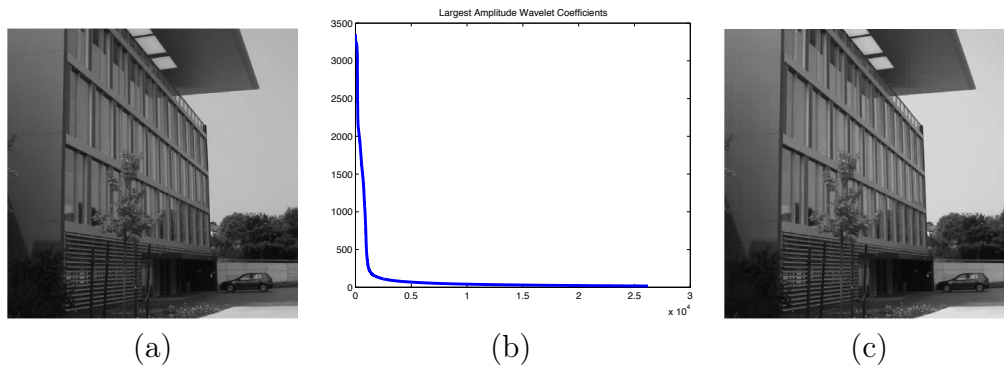


Figure 2.1: Image compression with discrete wavelet transform. (a) original image (b) largest 10% of wavelet coefficients sorted in descending order obtained by taking 4-level Haar wavelet transform of the image (c) image obtained by zeroing out all the wavelet coefficients but largest 10%. Pixel intensities of both images are thresholded to the range $[0, 255]$. As we see in (b) the amplitude of wavelet coefficients decay very rapidly. Thus (c) approximates quite well the image (a).

We call the signal *compressible* if the sorted magnitudes of x_i decay quickly. Typically compressible signals can be well approximated by a small number of coefficients, i.e. for $k \ll n$ the error $\|\mathbf{f} - \mathbf{f}_k\|_{\ell_2} = \|\mathbf{x} - \mathbf{x}_k\|_{\ell_2}$ is small. With reference to Figure 2.1 we can say that the image (a) is compressible in a wavelet basis. Then the compression strategy is quite clear, instead of storing the n elements of \mathbf{f} we can store $2k$ elements (k largest coefficients of \mathbf{x} and their respective positions). When reconstructing the image we simply obtain \mathbf{x}_k from stored coefficients and apply the equation (2.5). This compression strategy is called *adaptive* as the algorithm adapts to the knowledge of the signal \mathbf{f} . Exactly this strategy was utilized for obtaining the image (c) of the Figure 2.1. In this thesis we go one step further and assume that our signals are exactly k -sparse, hence we assume that the error $\|\mathbf{f} - \mathbf{f}_k\|_{\ell_2} = \|\mathbf{x} - \mathbf{x}_k\|_{\ell_2}$ is zero.

The adaptive strategy above is definitely appealing, nonetheless it has one major limitation: it assumes the knowledge of the signal \mathbf{f} . This implies that prior to being compressed the signal \mathbf{f} has to be acquired. In fact this is a standard setup in the modern image acquisition systems, where we first obtain the full image \mathbf{f} via CCD-like device and then compress it with an algorithm similar to JPEG-2000. Or put another way, the full information on the signal is obtained, most of which is thrown away at the compression stage.

Compressive sensing attempts to optimize the acquisition process. By

taking a small amount $m \ll n$ of linear and *non-adaptive*² measurements, as in (2.2), the signal is directly acquired in the undersampled form. Although, the equation (2.2) has infinitely many solutions, we can still solve it by assuming that the measured signal is sparse in some basis and that the measurement matrix satisfies some general conditions. We could recover the signal by searching for the sparsest vector $\tilde{\mathbf{x}} \in \mathbb{R}^n$ which is consistent with the measurements \mathbf{z} , i.e. by solving the following ℓ_0 -minimization problem³

$$\min \|\tilde{\mathbf{x}}\|_{\ell_0} \quad \text{subject to } z_k = \langle \varphi_k, \Psi \tilde{\mathbf{x}} \rangle, \quad k = 1, \dots, m. \quad (2.6)$$

Unfortunately it has been shown that this sparse reconstruction problem (2.6) is computationally intractable. A few practical alternatives to (2.6) have been proposed in the literature. One standard way is to solve the basis pursuit or ℓ_1 -minimization problem

$$\min \|\tilde{\mathbf{x}}\|_{\ell_1} \quad \text{subject to } z_k = \langle \varphi_k, \Psi \tilde{\mathbf{x}} \rangle, \quad k = 1, \dots, m, \quad (2.7)$$

which looks similar to (2.6), but can be recast as a linear program and solved using convex optimization techniques. It is important to note that by using (2.7) instead of (2.6) we trade algorithmic feasibility to potential sub-optimality of the result. However, due to the geometry of the ℓ_1 -norm, in many cases, (2.7) offers a good alternative to (2.6). *Restricted isometry property (RIP)* [2] was introduced as a sufficient condition for signal recovery via ℓ_1 -minimization. We can loosely say that Φ satisfies RIP if it approximately preserves the Euclidean length of k -sparse signals, i.e. $\|\Phi \mathbf{x}_k\|_{\ell_2}^2 \approx \|\mathbf{x}_k\|_{\ell_2}^2$. By choosing matrices that satisfy RIP we can obtain some guarantees on performance of ℓ_1 -minimization.

Finally there are two other considerations to take into account under the compressive sensing setup. First is the number of measurements $m < n$ required to be able to recover our signal \mathbf{f} from measurements \mathbf{z} . Much research has been done for deriving bounds for m and the results typically

²In the adaptive scheme one particular realization of \mathbf{f} influences the compression and reconstruction algorithms. In particular, the knowledge of the locations and amplitudes of largest non-zero coefficients are used. However, in the non-adaptive scheme no such knowledge is required, we use generic compression and reconstruction algorithms independent of one particular signal realization.

³Define $\|\mathbf{x}\|_{\ell_p} = (\sum_{k=1}^n |x_k|^p)^{\frac{1}{p}}$. For $p \in [1, \infty)$ this definition satisfies properties of the norm hence represents a valid vector norm over \mathbb{R}^n . However, for $p \in (0, 1)$ this definition is technically not a norm. The case $p = 0$ has a special interpretation. It is defined as $\|\mathbf{x}\|_{\ell_0} = \lim_{p \rightarrow 0} (|x_1|^p + |x_2|^p + \dots + |x_n|^p)$ and it simply counts the number of non-zero elements of the vector.

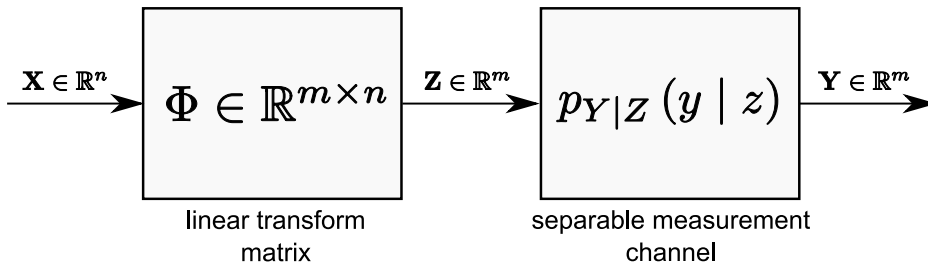


Figure 2.2: Generalized CS estimation problem. A random input vector \mathbf{X} with i.i.d. components is transformed by a matrix Φ before being corrupted by some noise. We characterize the noise as a separable measurement channel given by conditional distribution $p_{Y|Z}(y|z)$. Our goal is to estimate \mathbf{X} from \mathbf{Y} given the matrix Φ , the prior $p_X(x)$, and the noise $p_{Y|Z}(y|z)$. Note that without loss of generality we set $\Psi = \mathbb{I}_n$.

depend on the recovery algorithm used. For example, it has been shown that for (2.7) we can recover \mathbf{f} with high probability from $m \sim O(k \log \frac{n}{k})$ random measurements⁴.

Secondly, to make the results usable in the real world we should consider the effects of noise on our measurements, which arises due to imperfect equipment or limited precision. Consider the generalized setting in Figure 2.2 that we are going to study in this thesis. In this setting instead of being given the measurements \mathbf{z} we are given noisy data \mathbf{y} . As before given this data we would like to recover our signal \mathbf{f} (or equivalently \mathbf{x}). One common approach for this problem is to use ℓ_1 -minimization with relaxed constraints

$$\min \|\tilde{\mathbf{x}}\|_{\ell_1} \text{ subject to } \|A\tilde{\mathbf{x}} - \mathbf{y}\|_{\ell_2} \leq \epsilon, \quad (2.8)$$

where $A = \Phi\Psi \in \mathbb{R}^{m \times n}$ is the new measurement matrix obtained by combining Φ and Ψ , and ϵ bounds the amount of noise. Similarly to (2.7), (2.8) can also be solved efficiently using convex optimization techniques. For simplicity and without loss of generality in this thesis we will assume that $\Psi = \mathbb{I}_n$, hence $A = \Phi$.

However, ℓ_1 -minimization methods like (2.8) are not the only alternatives to (2.6). The search for computationally cheap methods for solving (2.2) has motivated much research into iterative methods for finding sparsest solutions. One class of these methods were inspired from the message passing algorithms

⁴One of the remarkable results in compressive sensing, due to RIP, is that all provably good measurement matrices designed so far are random. This randomness implies that there is always a chance that for the particular realization Φ the reconstruction might fail, but with sufficiently large m the probability of failure approaches zero.

on bipartite graphs, widely used in decoding error correction codes. In the next section we present one class of message passing algorithms called Belief Propagation, which, in its simplified form, has been successfully applied for sparse signal recovery.

Illustrative Example

Let us consider a small sample problem to get some more insight into geometry of signal recovery. Consider the setting where we have $n = 3$, $m = 2$, and $k = 1$. Suppose that we have signal $\mathbf{x} = [x_1, x_2, x_3]^*$. We consider that $\Psi = \mathbb{I}_n$, and we measure \mathbf{x} through the following measurement matrix

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \end{bmatrix}.$$

The measurements can be obtained by

$$\mathbf{z} = \Phi\mathbf{x} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}.$$

Now consider the problem of determining $\hat{\mathbf{x}} \in \mathbb{R}^3$, the estimate of \mathbf{x} , from the measurements \mathbf{z} and the measurement matrix Φ . One way to look at the random measurement geometry in CS is in signal space \mathbb{R}^n . In our case, Φ and \mathbf{z} define two planes in the signal space \mathbb{R}^3 each given by

$$\Phi_{a1}x_1 + \Phi_{a2}x_2 + \Phi_{a3}x_3 = z_a, \quad (2.9)$$

where $a = 1, 2$. If these two planes are not parallel to each other, then their intersection defines a line in \mathbb{R}^3 . Consider the line in the Figure 2.3, where to simplify the visualization we have selected the matrix Φ which constraints possible solution space to (x_1, x_2) plane. Without any prior information on the signal \mathbf{x} there are infinitely many potential $\hat{\mathbf{x}}$ that satisfy the constraints $\mathbf{z} = \Phi\hat{\mathbf{x}}$. Note that in Figure 2.3 the solution space defined by the line crosses two axes, which is undesirable for the recovery of the signal \mathbf{x} using (2.6).

Solving the ℓ_0 -minimization problem is a combinatorial task (Figure 2.4 depicts the process). We proceed by first considering 1-sparse vectors consistent with our measurements. If there would be no 1-sparse vectors satisfying our constraints, the algorithm would go on to consider $k = 2, 3, \dots$ sparse vectors. Similarly we could solve ℓ_1 -minimization problem (2.7), which to our greatest satisfaction recovers the correct result in Figure 2.4(b). Whereas typically for the sufficient number of measurements both ℓ_1 - and ℓ_0 - minimization succeed in recovering sparse vectors \mathbf{x} , ℓ_2 - minimization fails. This

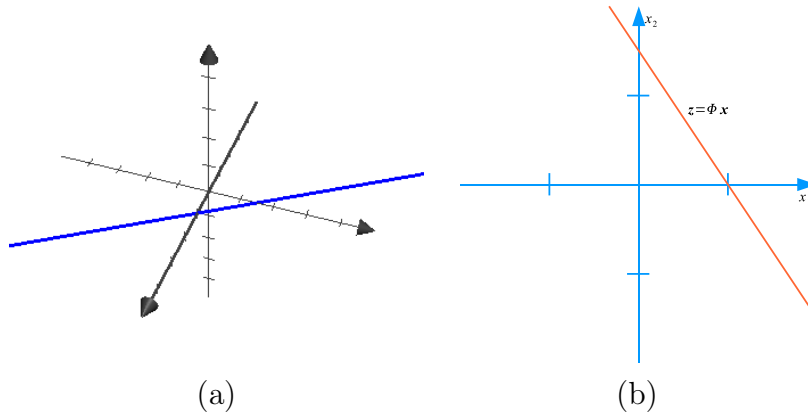


Figure 2.3: Toy problem illustration. Measurement matrix Φ together with measurement vector \mathbf{z} define two planes in signal space \mathbb{R}^3 . (a) intersection of these planes defines a line in \mathbb{R}^3 . (b) Solution space in (x_1, x_2) plane. As we see our matrix Φ is not a good compressive sensing matrix, as its solution space admits two 1-sparse solutions (intersection with axes).

is due to the geometry of the ℓ_2 -minimization, which in general does not induce sparse solutions (Figure 2.4(c)). In many scenarios any $p \in [0, 1]$ would work for recovering sparse vectors \mathbf{x} , however due to convexity and availability of fast implementations ℓ_1 -minimization is preferred.

Additionally the problem can be analyzed in the measurement space \mathbb{R}^m , where by knowing the sparseness k and the matrix Φ , we can define

$$\binom{n}{k}$$

k -dimensional subspaces where possible measurements lie. For example in the toy setting we will have 3 lines in the measurement space \mathbb{R}^2 given by

$$z_2 = \frac{\Phi_{21}}{\Phi_{11}} z_1, \quad z_2 = \frac{\Phi_{22}}{\Phi_{12}} z_1, \quad z_3 = \frac{\Phi_{23}}{\Phi_{13}} z_1. \quad (2.10)$$

Figure 2.5 depicts geometry of this. As long as these lines are separate we can recover our 1-sparse signal \mathbf{x} . However, in practice, our measurements are rarely exact, hence the mapping (2.2) might become un-invertible.

Summary of the section

- Some signals can be well approximated in some basis by keeping only a small number of large coefficients. We refer to such signals as *sparse* or *compressible*. For example, natural images tend to be sparse in wavelet domain.

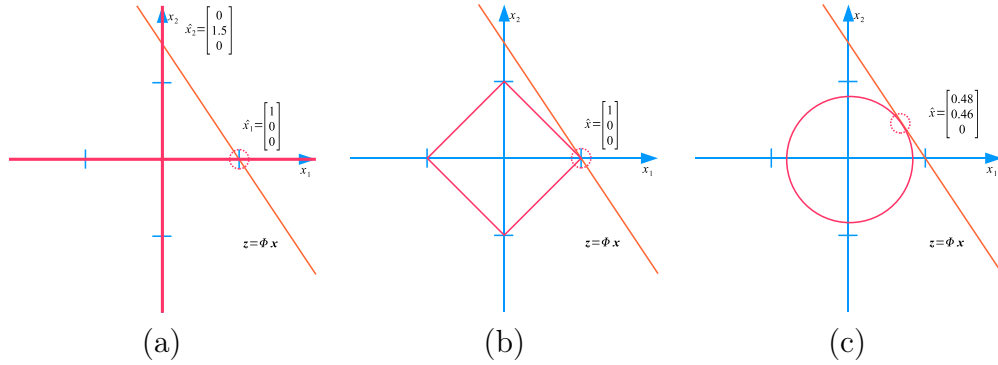


Figure 2.4: Geometry of ℓ_p -minimization recovery for (a) $p = 0$ (b) $p = 1$ (c) $p = 2$. The orange line is the constraint given by the measurements \mathbf{z} and by the measurement matrix Φ . The red lines represent the edges of the regions $(|x_1|^p + |x_2|^p)^{\frac{1}{p}} \leq r$ such that they touch the line. The point where two lines meet is the solution to the minimization problem.

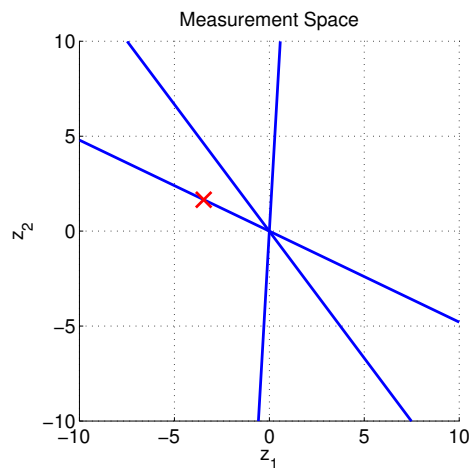


Figure 2.5: Illustration of measurement space. Blue lines define possible solutions to underdetermined system of linear equations with sparsity constraint $k = 1$. Red cross is the actual measurement (z_1, z_2) . As we can see in this case we can find an exact solution to our system.

- Lossy signal compression techniques rely on sparseness inducing transforms to achieve efficient compression with low distortion of the signal. As a rule, better compression can be achieved by choosing a sparser transform. For example, wavelet-based image compression method JPEG-2000 performs much better than cosine-based JPEG.
- *Compressive sensing* is a signal acquisition model. It promises that by taking a small amount of *linear* and *non-adaptive* measurements, we can still reconstruct our signal through practical algorithms.
- One aim of the compressive sensing theory is to design matrices to obtain suitable measurements. It has been showed that some *random* matrices are particularly good for acquisition, due to the property commonly referred to as *restricted isometry property*. For example, a matrix Φ with i.i.d. elements from the normal distribution with mean 0 and variance $\frac{1}{m}$ is a good sensing matrix with high probability provided that $m \sim O\left(k \log\left(\frac{n}{k}\right)\right)$.
- Finally, another important objective of compressive sensing is to determine efficient recovery algorithms. Many different algorithms have been proposed in recent years including convex optimization based *basis pursuit* and *lasso*, or iterative algorithms like *matching pursuit*, *approximate message passing*, and *relaxed belief propagation*.

2.2 Belief Propagation, Relaxed Belief Propagation

Belief Propagation

Consider the joint probability mass function $p_{\mathbf{X}}(\mathbf{x})$ of n discrete random variables $\mathbf{X} = [X_1, X_2, \dots, X_n]$. Let us suppose for the moment that each X_i takes value in some finite discrete set \mathcal{X} , hence $p_{\mathbf{X}}$ represents a joint probability distribution over \mathcal{X}^n . For example, if \mathbf{X} represents an image discretized to 8 bits, then $\mathcal{X} = \{0, 1, \dots, 255\}$. Suppose that we are interested in computing the *marginal* probabilities of $p_{\mathbf{X}}(\mathbf{x})$ with respect to each variable X_i , i.e. we are interested in computing

$$p_{X_i}(x_i) = \sum_{\{x_j\}, j \neq i} p_{\mathbf{X}}(\mathbf{x}), \quad i = 1, \dots, n, \quad (2.11)$$

where $p_X(\cdot)$ denotes the marginals of $p_{\mathbf{X}}$ and $\{x_j\}, j \neq i$ means that the summation is over all x_j except the element x_i . For example, for $n = 3$ and

$i = 1$ we can rewrite (2.11) as

$$p_{X_1}(x_1) = \sum_{x_2 \in \mathcal{X}} \sum_{x_3 \in \mathcal{X}} p_{\mathbf{X}}(x_1, x_2, x_3). \quad (2.12)$$

Determining $p_{X_i}(x_i)$ for each value of $x_i \in \mathcal{X}$ requires $O(|\mathcal{X}|^n)$ operations, which is in general very large and impossible to perform in practice. For instance, if we assume a 1 megapixel image with 8 bit values the cost of marginalization for the single pixel involves $256^{10^6} \approx 10^{2408240}$ operations⁵.

Now let us suppose that the joint distribution function $p_{\mathbf{X}}$ admits a factorization of the form

$$p_{\mathbf{X}}(\mathbf{x}) = \prod_{a=1}^m p_a(\mathbf{x}_a), \quad (2.13)$$

where m is the total number of factors and each factor $p_a(\mathbf{x}_a)$ is a function of a subset \mathbf{x}_a of variables that constitute \mathbf{x} . In practice factorization can be achieved by using some prior information on \mathbf{x} like interdependence or independence of variables among themselves. We can then exploit the factorization (2.13) of $p_{\mathbf{X}}$ to marginalize it efficiently. We do this by associating to the factorization a *factor* or *Tanner graph*, the type of graph often used in communications and information theory. Factor graphs are *undirected* and *bipartite*, consisting of two types of nodes or vertices (Figure 2.6). The first type of nodes are *variable nodes* and they are represented as circles. There is a variable node in the factor graph for each variable in \mathbf{x} . Second types of nodes are represented as squares and we call them *factor nodes*. We create a factor node for each factor p_a in the factorization (2.13). There is an edge between a variable node x_i and factor node p_a if and only if the corresponding variable appears in this factor. In the literature factor graphs are typically denoted as $G = (V, F, E)$ where $V = \{1, 2, \dots, n\}$ are variable nodes, $F = \{1, 2, \dots, m\}$ are factor nodes, and $E = \{(i, a) \in V \times F\}$ are edges. Given this graph, define the neighbor sets of the variable and factor nodes as

$$\mathcal{N}_V(i) = \{a \mid (i, a) \in E\}, \quad (2.14)$$

$$\mathcal{N}_F(a) = \{i \mid (i, a) \in E\}, \quad (2.15)$$

where $\mathcal{N}_V(i)$ is the set of neighbors of the variable node x_i , and $\mathcal{N}_F(a)$ is the set of neighbors of the factor node p_a . For example in Figure 2.6, $\mathcal{N}_V(2) = \{1\}$ and $\mathcal{N}_F(2) = \{1, 4, 6\}$.

⁵For the sake of comparison, the number of atoms in the observable universe is approximately 10^{80} .

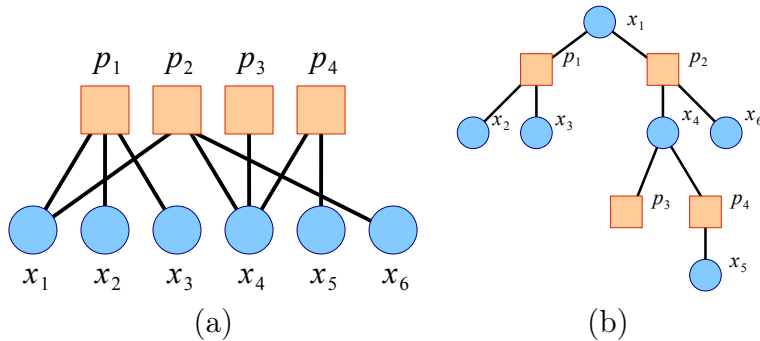


Figure 2.6: (a) Sample factorization of $p_{\mathbf{x}}$ (from [21]) represented as a factor graph. Circles in the graph are variable nodes and squares are factor nodes. We connect a variable node to a factor node by an edge if and only if the corresponding variable appears in this factor. (b) The same factor graph in (a) represented as a tree.

Let us assume for the moment that the factor graph constructed according to (2.13) is a tree⁶. We will generalize our discussion to graphs with cycles and continuous random variables later in this section. After constructing the factor graph F we can factorize $p_{\mathbf{x}}$ via a family of algorithms commonly referred to as *message passing algorithms*. In this thesis we are particularly interested in one of those algorithms, *belief propagation (BP)*, which has been extensively studied and applied across many applications. In communications literature the algorithm is commonly used for decoding and analyzing LDPC and turbo codes [13, 19, 21]. However, more recently the algorithm has also been successfully applied to number of other problems including sparse signal estimation in compressive sensing [5].

In a nutshell, BP is an iterative algorithm that works by passing messages, which represent probability mass functions over \mathcal{X} , along the edges of the factor graph. There are two ways to apply BP on a tree. The first method uses the representation in Figure 2.6(b) and passes messages starting from leaves of the tree towards the root. Exact equations for this method are presented in Algorithm 2.1. We denote messages from variable to factor nodes with $\mu(x)$ and from factor to variable nodes with $\hat{\mu}(x)$.

Although this method is practical for demonstrating the way BP computes the marginals, like $p_{X_1}(x_1)$, of the joint distribution $p_{\mathbf{x}}(\mathbf{x})$ in the Figure 2.6(b), it cannot be easily generalized to arbitrary graphs with loops. Hence a more practical method for our purpose uses the representation in Figure 2.6(a) and passes messages according to the Figure 2.7. At each itera-

⁶Tree is simply an undirected and connected graph without cycles.

Algorithm 2.1 BP rules on trees.

1. Initialization: At each leaf variable and leaf factor node send the following messages to the parent node:

$$\mu_{i \rightarrow a}(x_i) = 1, \quad (2.16)$$

$$\hat{\mu}_{a \rightarrow i}(x_i) = p_a(x_i), \quad (2.17)$$

where p_a is the factor of $p_{\mathbf{X}}$.

2. Processing: At the inner nodes, after reception of all the messages from children, process them according to the following equations and send the resulting messages to the parent nodes:

$$\mu_{i \rightarrow a}(x_i) \propto \prod_{b \in \mathcal{N}_V(i), b \neq a} \hat{\mu}_{b \rightarrow i}(x_i), \quad (2.18)$$

$$\hat{\mu}_{a \rightarrow i}(x_i) \propto \sum_{\mathbf{x}_a \setminus \{x_i\}} \left(p_a(\mathbf{x}_a) \prod_{j \in \mathcal{N}_F(a), j \neq i} \mu_{j \rightarrow a}(x_j) \right), \quad (2.19)$$

where the sum is over all the variables in \mathbf{x}_a except the variable x_i and the product is over all the neighbors except the parent node. The \propto sign means that the expression on the right hand side should be normalized to unity. Execute this step until the root is eventually reached.

3. Marginalization: Finally at the root node we obtain the marginal distribution of $p_{X_i}(x_i)$ according to the following rule:

$$p_{X_i}(x_i) \propto \prod_{b \in \mathcal{N}_V(i)} \hat{\mu}_{b \rightarrow i}(x_i), \quad (2.20)$$

here also the \propto sign means that the expression on the right hand side should be normalized to unity.

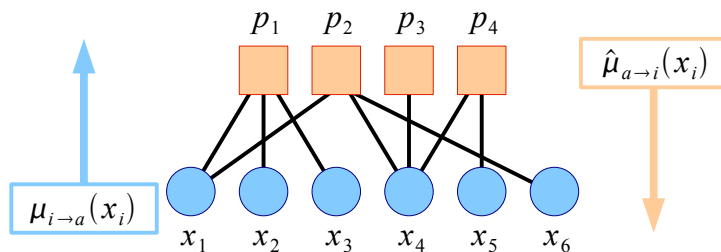


Figure 2.7: Notations used for messages passed along the edges of the factor graph. Messages are probability densities representing the “belief” for x_i to be of some value. For trees BP converges to consensus after several iterations of message passing returning the marginals for each x_i . For general graphs BP only approximates the marginals, however it has been empirically demonstrated to work well even in the presence of loops.

tion messages from variable nodes are passed to factor nodes, processed there and send back to variable nodes. Moreover at each iteration marginals of our distribution are estimated at the variable nodes. The complete algorithm is described in Algorithm 2.2. It can be showed that for tree factor graphs both methods converge to the same true marginals [13]. However, for graphs with cycles, BP only provides an approximation to the true marginals [12, 21].

We have assumed so far that the random variables X_i take values in some finite discrete set \mathcal{X} , however in many signal processing applications variables naturally take values in continuous spaces. In principle BP can easily be extended to such models by simply replacing summations in Algorithm 2.1 and Algorithm 2.2 by integrals. However, in practice such an approach is intractable as messages resulting from updates typically lack closed forms. One simple approach to avoid this problem is to discretize the probability densities, but for large number of edges even this approach quickly becomes computationally infeasible. Another possibility is to simplify the algorithm by making some assumptions about the probability densities exchanged in the factor graph. For example, if the messages exchanged are Gaussian then it is sufficient to track only the means and the variances across the updates.

As noted above when the factor graph contains loops BP provides merely estimates of the true marginals p_{X_i} . Despite that, in practice, loopy BP has performed exceptionally well, in particular its applications for LDPC and turbo code decoding proved to be effective [13]. Although, precise conditions under which loopy BP converges are still not well understood, several techniques were proposed for the analysis of the algorithm. *Density evolution (DE)* [19, 20] is one of the most widely used analysis tools and it works by tracking distribution of the messages exchanged via BP under the assump-

Algorithm 2.2 BP rules on general graphs.

1. Initialization: Start at $t = 0$ by sending:

$$\mu_{i \rightarrow a}^t(x_i) = 1, \quad (2.21)$$

along all edges $(i, a) \in E$.

2. Factor node update: For each edge $(i, a) \in E$ compute the marginal distributions:

$$\hat{\mu}_{a \rightarrow i}^t(x_i) \propto \sum_{\mathbf{x}_a \setminus \{x_i\}} \left(p_a(\mathbf{x}_a) \prod_{j \in \mathcal{N}_F(a), j \neq i} \mu_{j \rightarrow a}^t(x_j) \right), \quad (2.22)$$

where the sum is over all the variables in \mathbf{x}_a except the variable x_i and the product is over all the neighbors except the one the message is being sent to. The \propto sign means that the expression on the right hand side should be normalized to unity. Note that if a factor node p_a is connected to a single variable node x_i , then it simply sends the message $\hat{\mu}_{a \rightarrow i}^t(x_i) = p_a(x_i)$.

3. Variable node update: For each edge $(i, a) \in E$ compute the marginal distributions:

$$\mu_{i \rightarrow a}^{t+1}(x_i) \propto \prod_{b \in \mathcal{N}_V(i), b \neq a} \hat{\mu}_{b \rightarrow i}^t(x_i), \quad (2.23)$$

where \propto means normalization to unity and the product is over all the neighbors except the one the message is being sent to.

4. Marginalization: For each variable node i compute the current estimate of the marginal:

$$p_{X_i}^{t+1}(x_i) \propto \prod_{b \in \mathcal{N}_V(i)} \hat{\mu}_{b \rightarrow i}^t(x_i), \quad (2.24)$$

where \propto means normalization to unity. Go back to the step **2**. Execute steps **2**, **3**, **4** for fixed number of iterations or until convergence.

tion that $n \rightarrow \infty$. Tracking the density function through DE gives a very good picture of the actual behavior of the BP, moreover DE allows to predict the conditions under which BP decodes the messages successfully.

Illustrative Example

We now present a simple example from [21] illustrating potential computational savings achievable by using BP. Consider the joint probability mass function $p_{\mathbf{X}}$ of $n = 6$ random variables with the following factorization

$$p_{\mathbf{X}}(x_1, x_2, x_3, x_4, x_5, x_6) = p_1(x_1, x_2, x_3) p_2(x_1, x_4, x_6) p_3(x_4) p_4(x_4, x_5). \quad (2.25)$$

Suppose that we are interested in computing the marginal $p_{X_1}(x_1)$, then direct marginalization of $p_{\mathbf{X}}$ can be written as

$$p_{X_1}(x_1) = \sum_{x_2, x_3, x_4, x_5, x_6} p_{\mathbf{X}}(x_1, x_2, x_3, x_4, x_5, x_6). \quad (2.26)$$

Assuming that $X_i \in \mathcal{X}$ for all $i = 1, \dots, 6$, direct computation of (2.26) for all values of X_1 requires $O(|\mathcal{X}|^6)$ operations. However, we can do much better by using BP. The factor graph for $p_{\mathbf{X}}$ under (2.25) has already been presented in Figure 2.6 (b). We perform BP by following the Algorithm 2.1 and obtain the following factorization:

$$p_{X_1}(x_1) = \left(\sum_{x_2, x_3} p_1(x_1, x_2, x_3) \right) \left(\sum_{x_4} p_3(x_4) \left(\sum_{x_6} p_2(x_1, x_4, x_6) \right) \left(\sum_{x_5} p_4(x_4, x_5) \right) \right),$$

which is evaluated in favorable $O(|\mathcal{X}|^3)$ operations.

Bayesian CS via Message Passing

In this section we motivate the application of BP to compressive sensing estimation. Consider the problem of estimating a random vector $\mathbf{X} \in \mathbb{R}^n$ from noisy measurements $\mathbf{Y} \in \mathbb{R}^m$, depicted in Figure 2.2, where the noise is described by a measurement channel $p_{Y|Z}(y|z)$, which acts identically on each measurement z_a of the vector \mathbf{z} obtained via (2.2). Moreover suppose that elements in the vector \mathbf{X} are distributed i.i.d. according to $p_X(x)$. Then we can construct the following conditional probability distribution over random vector \mathbf{X} given the measurements \mathbf{Y} :

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \prod_{i=1}^n p_X(x_i) \prod_{a=1}^m p_{Y|Z}(y_a|z_a), \quad (2.27)$$

where Z is the normalization constant and $z_a = (\Phi x)_a$. By marginalizing this distribution it is possible to estimate each x_i . Although direct marginalization of $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y})$ is computationally intractable, we approximate marginals through BP. We apply BP following Algorithm 2.2, but replacing sums by integrals. This results in the following message passing rules

$$\mu_{i \rightarrow a}^{t+1}(x_i) \propto p_X(x_i) \prod_{b \neq a} \hat{\mu}_{b \rightarrow i}^t(x_i), \quad (2.28)$$

$$\hat{\mu}_{a \rightarrow i}^t(x_i) \propto \int p_{Y|Z}(y_a|z_a) \prod_{j \neq i} \mu_{j \rightarrow a}^t(x_j) dx, \quad (2.29)$$

where \propto means that the distribution is to be normalized so that it has unit integral. The integration is over the support of \mathbf{x} and performed over all the elements of \mathbf{x} except x_i . Initialization is performed by setting $\mu_{i \rightarrow a}^0(x_i) = p_X(x_i)$.

Earlier works on BP reconstruction have shown that it is asymptotically MSE optimal under certain verifiable conditions. These conditions involve simple single-dimensional recursive equations called *state evolution (SE)*, an equivalent of density evolution for dense matrices, which predicts that BP is optimal when corresponding SE admits unique fixed point [11, 16]. However, we have already noted that application of BP to continuous-valued vector estimation is impractical. In the CS framework the random vectors \mathbf{X} that we would like to estimate take values in \mathbb{R}^n , which makes it difficult to keep track of probability densities across iterations of BP. Specifically due to integrals the factor node updates might lack closed form expressions. However, as mentioned BP can be simplified through various Gaussian approximations. Recently two related algorithms have been studied in literature *Relaxed Belief Propagation (RBP)* [11, 16] and *Approximate Message Passing (AMP)* [8, 17]. Despite the fact that these algorithms are only approximations to the complete BP, they have been shown to asymptotically achieve error performance of BP.

Recent theoretical work and extensive simulations have demonstrated that the error performance of both RBP and AMP can also be accurately predicted by SE recursion [1, 16]. In this recursion, the MSE of the reconstruction represents the state and is tracked from iteration to iteration. Its change across iterations is modeled by a simple scalar mapping, which depends on underlying parameters of the problem like sparsity ratio, undersampling ratio, noise levels, etc. Besides providing reliable estimates of the reconstruction error, SE also predicts that when it has a unique fixed point the reconstruction algorithm will obtain minimum MSE estimates of the signal. In this thesis we will study the application of the RBP and SE

to the problem of CS estimation from quantized measurements, while keeping in mind that identical work can be done for AMP as well. In the next chapter we will present the equations for implementing RBP reconstruction from quantized measurements, together with corresponding SE equation. For more general and detailed analysis of RBP see [16].

Summary of the section

- *Belief propagation* (BP) is an iterative algorithm widely used for decoding and estimation in the context of LDPC codes. It works by passing messages representing probability densities along the edges of a bipartite *factor graph*.
- In the case where the graph is a tree BP results in exact marginal posterior probability distributions of the variables. However in reality factor graphs are rarely trees, hence BP provides only estimates of the posterior distributions.
- In principle it is possible to apply BP to CS reconstruction, however the resulting algorithm is not tractable due to the density of the measurement matrix and the fact that the variables to be estimated take values in continuous space. Nonetheless, by introducing some Gaussian approximations BP can be simplified to a more practical alternative – *relaxed belief propagation* (RBP).
- Error performance of RBP can be characterized and predicted by *state evolution* (SE) formalism. SE is a simple recursive equation tracking the MSE of the CS reconstruction under RBP. It is a powerful analysis and design tool that can be utilized to design certain parameters of the reconstruction. In this thesis we make use of this theory to design optimal quantizers for CS reconstruction under RBP.

2.3 Quantization

Theory

In this section we will present topics related to design of quantizers for a given stochastic source. More complete and elaborate discussion of quantizer design can be found in [10]. Quantization is the process of mapping random variables from some continuous space to some discrete set. More specifically

we define N -level scalar quantizer Q as a map

$$Q : \mathbb{R} \rightarrow \mathcal{C} = \{c_i; i = 1, \dots, N\}, \quad (2.30)$$

where we refer to \mathcal{C} as the *output set* and to c_i as *output levels*. The mapping is performed by partitioning the real line into N disjoint sub-set \mathcal{I}_i and associating an output level to each interval

$$\mathcal{I}_i = \{x \in \mathbb{R} : Q(x) = c_i\} \equiv Q^{-1}(c_i), \quad (2.31)$$

where we refer to $Q^{-1}(c_i)$ as the *inverse image* of the output level c_i under Q . Additionally we will be referring to quantizer as *regular* if the set \mathcal{I}_i can be described as a single interval $[b_{i-1}, b_i)$, i.e.

$$\mathcal{I}_i = [b_{i-1}, b_i) = \{x : b_{i-1} \leq x < b_i\}; i = 1, \dots, N, \quad (2.32)$$

where we refer to b_i as decision boundaries. For $i = 1$, if $b_0 = -\infty$ we replace the closed interval $[b_0, b_1)$ by an open interval (b_0, b_1) . We will additionally consider one type of non-regular quantizers that we will refer to as *binned* [15]. Binned quantizers reduce the rate by joining several disjoint intervals of a regular quantizer into a single bin, i.e.

$$\mathcal{B}_i = \bigcup_{l \in \mathcal{L}_i} \mathcal{I}_l = \bigcup_{l \in \mathcal{L}_i} [b_{l-1}, b_l); i = 1, \dots, M, \quad (2.33)$$

where \mathcal{L}_i are the disjoint subsets of $\{1, \dots, N\}$ and M is the total number of bins. Figure 2.8 illustrates the difference between regular and binned quantizers.

To demonstrate the functioning of the binned quantizer consider the encoder-decoder setting where the encoder uses the quantizer in the Figure 2.8(b) to digitize the random variable S . Assume that the decoder has access to the prior distribution $p_S(s)$. Suppose that at some moment s is such that the quantizer output $\hat{s} = Q(s) = 1$ is transmitted to the decoder. Then the decoder knows that $s \in [-3, -2) \cup [1, 2)$ and can combine this information with the prior $p_S(s)$ to estimate s . Consequently, binned quantizers can potentially offer better error performance for lower bitrate, when some auxiliary side information, like the prior distribution, is available at the decoder.

Typically regular quantizers are optimized by selecting decision boundaries and output levels in order to minimize the distortion between the random vector $\mathbf{S} \in \mathbb{R}^m$ and its quantized representation $\hat{\mathbf{S}} = Q(\mathbf{S})$. For example, for a given vector \mathbf{S} and the MSE distortion metric, optimization is performed by solving

$$Q^\# = \arg \min_Q \mathbb{E} \{ \|\mathbf{S} - Q(\mathbf{S})\|_{\ell_2}^2 \}, \quad (2.34)$$

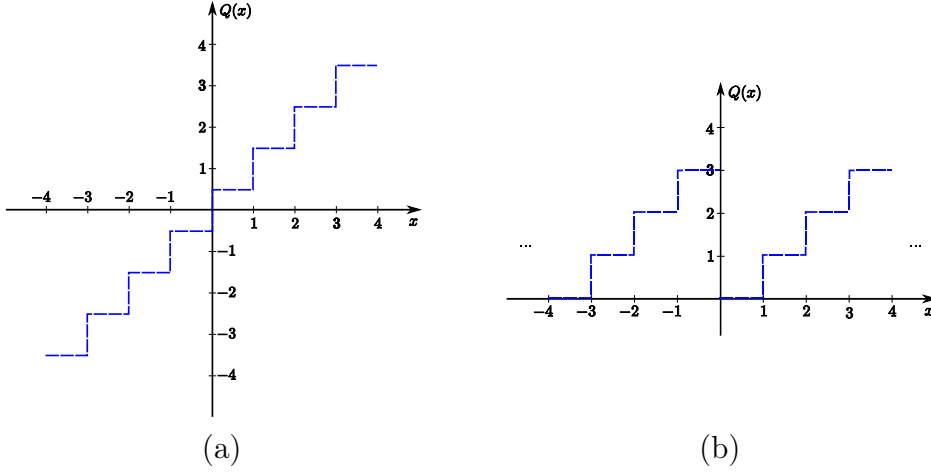


Figure 2.8: Example quantizer mappings. (a) Regular and (b) binned quantizers with uniformly spaced cells.

where the minimization is done over all N -level regular scalar quantizers. One standard way of optimizing Q is via the *Lloyd algorithm*, which iteratively updates the decision boundaries and output levels by applying necessary conditions for quantizer optimality [10].

However, for the CS framework finding the quantizer that minimizes MSE between \mathbf{S} and $\hat{\mathbf{S}}$ is not necessarily equivalent to minimizing the MSE between the sparse vector \mathbf{X} and its CS reconstruction from quantized measurements $\hat{\mathbf{X}}$ [22]. This is due to the nonlinear effect added by any particular CS reconstruction function. Hence, instead of solving (2.34), it is more interesting to solve

$$Q^* = \arg \min_Q \mathbb{E} \left\{ \left\| \mathbf{X} - \hat{\mathbf{X}} \right\|_{\ell_2}^2 \right\}, \quad (2.35)$$

where minimization is performed over all N -level scalar quantizers and $\hat{\mathbf{X}}$ is obtained through a CS reconstruction method like RBP or ℓ_1 -minimization. This is the approach we take in this work to obtain optimal quantizers for CS acquisition framework.

Illustrative Example

One way to illustrate the effect a quantizer has on compressive sensing reconstruction is to consider the problem in the measurement space \mathbb{R}^m . Consider again the case $n = 3$, $m = 2$, and $k = 1$, with measurements given by

$$\mathbf{y} = Q(\mathbf{z}) = \Phi \mathbf{x}, \quad (2.36)$$

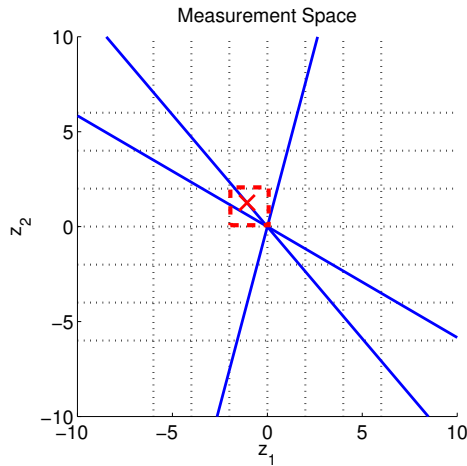


Figure 2.9: Illustration of measurement space under quantization. Blue lines define the possible 1-sparse solution sub-spaces. Black grid delimit quantizer boundaries. In this setting we do not have direct access to the red cross (z_1, z_2) , instead we have access to quantizer indices (y_1, y_2) , which define m -cube in \mathbb{R}^2 (red square).

where Q is a scalar quantizer. In this case, instead of pinpointing the exact sparse-subspace where the solution lies, measurements \mathbf{y} will merely provide us with some hypercube in \mathbb{R}^m the size of which depends on the quantizer. This is demonstrated in Figure 2.9. As we can see there are two possible sparse subspaces that lie within our quantization cell. Implication of this is that in this particular realization recovery if \mathbf{x} is not any more guaranteed. In the binned quantizer scenario we would associate one index with multiple quantizer outputs, resulting in several disjoint red squares in measurement space for one exact measurement.

Summary of the section

- *Quantization* is performed by mapping continuous-valued random variables into some finite discrete set. Quantizers that operate on scalars are commonly referred to as scalar quantizers.
- The most intuitive type of quantizers are *regular quantizers*, as they imitate the rounding process by assigning to some compact sub-interval of \mathbb{R} some discrete value.
- Alternatively, it is possible to bin disjoint cells of a regular quantizer to form a *binned quantizer*. This allows to reduce the bitrate of the

quantizer, while potentially preserving the distortion.

- Quantizers are evaluated and designed based on their error performance for a given rate. Traditional evaluation metric is the MSE between quantizer input and output.
- However, in compressive sensing framework minimizing the error between the quantizer input and output is not necessarily equivalent to minimizing the error of the reconstruction.

Chapter 3

RBP for Quantized Measurements

In this chapter we will present equations for implementing RBP for compressive sensing under quantized measurements. We will then work out the equations of the corresponding state evolution recursion, which will allow us to design good quantizers in the next chapter.

3.1 Problem Setting

Consider the Figure 3.1 where we depict classical CS acquisition setting, but where instead of having access to infinite precision measurements \mathbf{Z} the best decoder can have is the digitized measurement vector \mathbf{Y} . For simplicity we assume that $\Psi = \mathbb{I}_n$ and that our random vector $\mathbf{X} \in \mathbb{R}^n$ is distributed i.i.d. according to some sparse prior $p_X(x)$ ¹. This random vector is measured through the wide matrix Φ to result in $\mathbf{Z} \in \mathbb{R}^m$, which is further perturbed by some additive white Gaussian noise (AWGN). The resulting vector \mathbf{S} can be written as

$$\mathbf{S} = \mathbf{Z} + \eta = \Phi\mathbf{X} + \eta, \quad (3.1)$$

where the elements η_a of the vector $\eta \in \mathbb{R}^m$ are i.i.d. random variables distributed as $\mathcal{N}(0, \sigma^2)$. These noisy measurements are then quantized by N -level quantizer Q to give the CS measurements $\mathbf{Y} \in \mathbb{R}^m$ that can now be stored or transmitted. The goal of any CS reconstruction scheme is to provide an estimate of the signal \mathbf{X} from these measurements. In particular we can formulate this problem in the Bayesian framework as in (2.27) and apply RBP to do estimation given the matrix Φ , the signal prior $p_X(x)$, the noise variance $\sigma^2 > 0$, and the quantizer mapping Q .

¹A Few different sparseness inducing prior distributions have been considered in literature the most common ones being Laplace and Gauss-Bernoulli distributions.

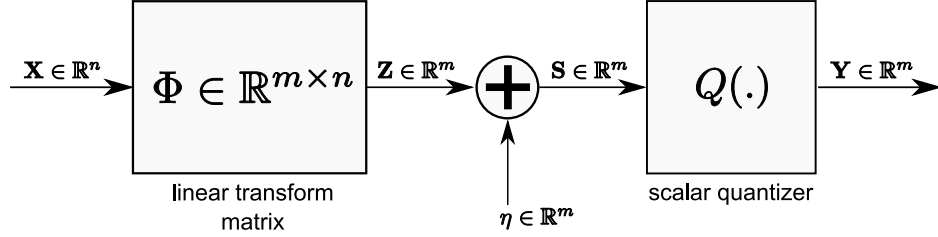


Figure 3.1: Compressive sensing set up with quantized measurements. We assume that prior to being estimated our measurements \mathbf{Z} are passed through some measurement channel including AWGN and scalar quantizer Q .

Note that under the model in Figure 3.1 each quantized measurement y_a indicates that $s_a \in Q^{-1}(y_a)$, hence our measurement channel can be characterized as

$$p_{Y|Z}(y_a | z_a) = \int_{Q^{-1}(y_a)} \phi(t - z_a; \sigma^2) dt, \quad (3.2)$$

for $a = 1, 2, \dots, m$ and where $\phi(\cdot)$ is Gaussian function

$$\phi(t; \nu) = \frac{1}{\sqrt{2\pi\nu}} \exp\left(-\frac{t^2}{2\nu}\right). \quad (3.3)$$

By knowing the channel characterization we can easily formulate RBP for estimating \mathbf{X} .

3.2 Relaxed BP

Using a central limit theorem argument RBP reduces the complexity of the standard BP by approximating the messages with Gaussian distributions. Then the Gaussianity of the messages allows us to reliably approximate the probability densities in (2.28) and (2.29) by two scalar parameters each. Algorithm 3.1 presents complete equations for the RBP update rules. We refer to messages $\{\hat{x}_{i \rightarrow a}, \hat{\tau}_{i \rightarrow a}\}_{(i,a) \in E}$ as variable updates and to messages $\{u_{a \rightarrow i}, \tau_{a \rightarrow i}\}_{(i,a) \in E}$ as measurement updates. One iteration of the algorithm consists of performing these two updates together with estimation of the input vector $\hat{\mathbf{X}}$.

The nonlinear functions F_{in} and \mathcal{E}_{in} in (3.8)-(3.9) are conditional means and variance

$$F_{\text{in}}(q, \nu) \equiv \mathbb{E}\{X | Q = q\}, \quad (3.11)$$

$$\mathcal{E}_{\text{in}}(q, \nu) \equiv \text{Var}\{X | Q = q\}, \quad (3.12)$$

Algorithm 3.1 Relaxed Belief Propagation for general channels.

1. Initialization: Start at $t = 0$ by sending

$$\hat{x}_{i \rightarrow a}^t \equiv \hat{x}_{\text{init}}, \quad (3.4)$$

$$\hat{\tau}_{i \rightarrow a}^t \equiv \hat{\tau}_{\text{init}}, \quad (3.5)$$

along all edges $(i, a) \in E$. \hat{x}_{init} and $\hat{\tau}_{\text{init}}$ are the mean and variance of the prior $p_X(x_i)$.

2. Measurement Update: For every $(i, a) \in E$ compute

$$u_{a \rightarrow i}^t \equiv -D_1 \left(y_a, \sum_{j \neq i} \Phi_{aj} \hat{x}_{j \rightarrow a}^t, \sum_{j \neq i} \Phi_{aj}^2 \hat{\tau}_{j \rightarrow a}^t + \sigma^2 \right), \quad (3.6)$$

$$\tau_{a \rightarrow i}^t \equiv D_2 \left(y_a, \sum_{j \neq i} \Phi_{aj} \hat{x}_{j \rightarrow a}^t, \sum_{j \neq i} \Phi_{aj}^2 \hat{\tau}_{j \rightarrow a}^t + \sigma^2 \right), \quad (3.7)$$

where $D_r(y, \hat{z}, \nu)$ are the functions defined in (3.13)-(3.14).

3. Variable Update: For every $(i, a) \in E$ compute

$$\hat{x}_{i \rightarrow a}^{t+1} \equiv F_{\text{in}} \left(\frac{\sum_{b \neq a} \Phi_{bi} u_{b \rightarrow i}^t}{\sum_{b \neq a} \Phi_{bi}^2 \tau_{b \rightarrow i}^t}, \frac{1}{\sum_{b \neq a} \Phi_{bi}^2 \tau_{b \rightarrow i}^t} \right), \quad (3.8)$$

$$\hat{\tau}_{i \rightarrow a}^{t+1} \equiv \mathcal{E}_{\text{in}} \left(\frac{\sum_{b \neq a} \Phi_{bi} u_{b \rightarrow i}^t}{\sum_{b \neq a} \Phi_{bi}^2 \tau_{b \rightarrow i}^t}, \frac{1}{\sum_{b \neq a} \Phi_{bi}^2 \tau_{b \rightarrow i}^t} \right), \quad (3.9)$$

where the functions F_{in} and \mathcal{E}_{in} are the conditional mean and variance given in (3.11)-(3.12).

4. Estimation: For every $i = 1, 2, \dots, n$ compute the estimate

$$\hat{x}_i^{t+1} \equiv F_{\text{in}} \left(\frac{\sum_{b=1}^m \Phi_{bi} u_{b \rightarrow i}^t}{\sum_{b=1}^m \Phi_{bi}^2 \tau_{b \rightarrow i}^t}, \frac{1}{\sum_{b=1}^m \Phi_{bi}^2 \tau_{b \rightarrow i}^t} \right), \quad (3.10)$$

where F_{in} is the conditional mean defined in (3.11). Go back to the step **2**. Execute the steps **2**, **3**, **4** for a fixed number of iterations or until convergence.

where $Q = X + V$, $X \sim p_X(x_i)$, and $V \sim \mathcal{N}(0, \nu)$. Note that in general these functions might not have closed form expressions, however they can be evaluated with one-dimensional integrals. Similarly the functions D_1 and D_2 in (3.6)-(3.7) can be computed via

$$D_1(y, \hat{z}, \nu) \equiv \frac{1}{\nu} (\hat{z} - F_{\text{out}}(y, \hat{z}, \nu)), \quad (3.13)$$

$$D_2(y, \hat{z}, \nu) \equiv \frac{1}{\nu} \left(1 - \frac{\mathcal{E}_{\text{out}}(y, \hat{z}, \nu)}{\nu} \right), \quad (3.14)$$

where F_{out} and \mathcal{E}_{out} are conditional mean and variance

$$F_{\text{out}}(y, \hat{z}, \nu) \equiv \mathbb{E} \{ Z \mid Z \in Q^{-1}(y) \}, \quad (3.15)$$

$$\mathcal{E}_{\text{out}}(y, \hat{z}, \nu) \equiv \text{Var} \{ Z \mid Z \in Q^{-1}(y) \}, \quad (3.16)$$

with random variable $Z \sim \mathcal{N}(\hat{z}, \nu)$. These functions admit closed-form expressions in terms of

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt. \quad (3.17)$$

The numerical complexity of the RBP implementation in Algorithm 3.1 can actually be further simplified by following suggestions in [16].

Figures 3.2 and 3.3 show the simulation results for the algorithm above. We consider sparse signals \mathbf{X} distributed i.i.d.} according to the Gauss-Bernoulli distribution

$$X_i \sim \begin{cases} \mathcal{N}(0, 1/\rho) & \text{with prob} = \rho, \\ 0 & \text{with prob} = 1 - \rho, \end{cases} \quad (3.18)$$

where $i = 1, \dots, n$. In the simulation the sparsity ratio is fixed to $\rho = 0.1$. Additionally we set the length of the signal to $n = 2000$ and the undersampling ratio to $\beta = n/m = 2$. The measurement matrix Φ is formed from i.i.d. Gaussian random variables $\Phi_{ai} \sim \mathcal{N}(0, 1/m)$. We assume that AWGN with variance $\sigma^2 = 10^{-3}$ perturbs our measurements prior to quantization and 32-level uniform quantizer Q with following boundaries

$$b_i = \begin{cases} -\infty & \text{for } i = 0, \\ -5 + \frac{i-1}{3} & \text{for } i = 1, \dots, 31, \\ +\infty & \text{for } i = 32. \end{cases} \quad (3.19)$$

digitizes our measurements.

In Figure 3.2(a) we validate the assumption of Gaussianity for noiseless-measurements \mathbf{Z} and plot the histogram of z_a against the normal density

$\mathcal{N}(0, \beta)$. Figure 3.2(b) plots MSE performance of the estimation for 15 iterations of the algorithm. As we can see RBP converges exponentially fast, reaching the estimation floor in approximately 8 iterations. Figures 3.3(a)-(c) demonstrate the reconstruction process across iterations of RBP, where the estimate at $t = 15$ is almost exact. Based on our extensive empirical experiments RBP obtains exceptionally good estimates from quantized measurements.

3.3 State Evolution for RBP

The equations in the Algorithm 3.1 are easy to implement, however they provide us no insight into the performance of the algorithm. The goal of SE equations is to describe the asymptotic behavior of RBP under large measurement matrices. The SE for RBP under the setting in the Figure 3.1 is given by the recursion

$$\bar{\tau}_{t+1} = \bar{\mathcal{E}}_{\text{in}} \left(\frac{1}{\bar{D}_2(\beta\bar{\tau}_t, \sigma^2)} \right), \quad (3.20)$$

where $t \geq 0$ is the iteration number, $\beta = n/m$ is a fixed number denoting the measurement ratio, and σ^2 the variance of the AWGN components is also fixed. We initialize the recursion by setting $\bar{\tau}_0 = \hat{\tau}_{\text{init}}$, where $\hat{\tau}_{\text{init}}$ is the variance of X_i according to the prior $p_X(x_i)$. We define the function $\bar{\mathcal{E}}_{\text{in}}$ as

$$\bar{\mathcal{E}}_{\text{in}}(\nu) = \mathbb{E} \{ \mathcal{E}_{\text{in}}(q, \nu) \}, \quad (3.21)$$

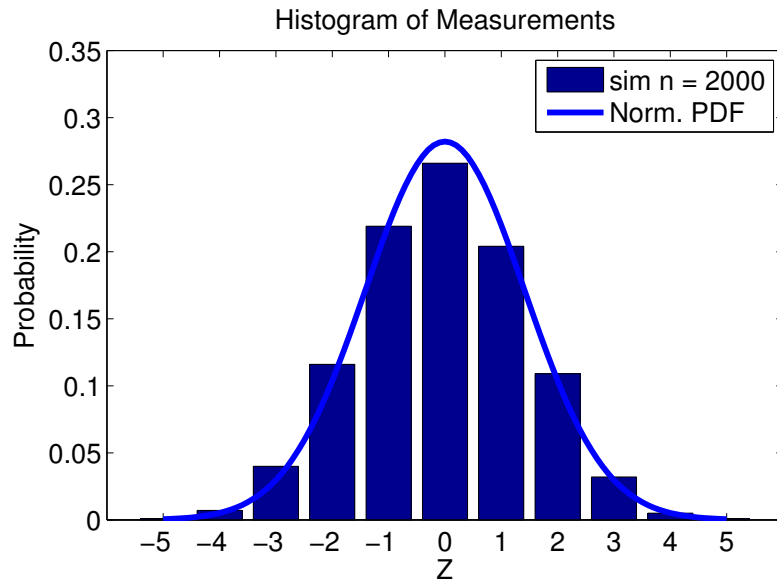
where the expectation is taken over the scalar random variable $Q = X + V$, with $X \sim p_X(x_i)$, and $V \sim \mathcal{N}(0, \nu)$. Similarly the function \bar{D}_2 is defined as

$$\bar{D}_2(\nu, \sigma^2) = \mathbb{E} \{ D_2(y, \hat{z}, \nu + \sigma^2) \}, \quad (3.22)$$

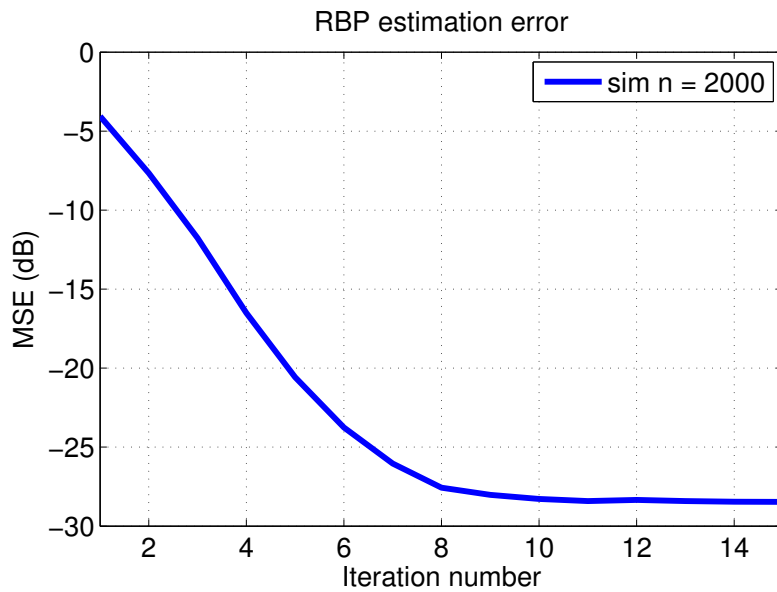
where D_2 is given by (3.14) and the expectation is taken over $p_{Y|Z}(y_a | z_a)$ and $(Z, \hat{Z}) \sim \mathcal{N}(0, P_z(\nu))$, with the covariance matrix

$$P_z(\nu) = \begin{pmatrix} \beta\hat{\tau}_{\text{init}} & \beta\hat{\tau}_{\text{init}} - \nu \\ \beta\hat{\tau}_{\text{init}} - \nu & \beta\hat{\tau}_{\text{init}} - \nu \end{pmatrix}. \quad (3.23)$$

One of the main results of [16], which we present below for completeness, was to demonstrate the convergence of the error performance of the RBP algorithm to the SE equations under large sparse measurement matrices. Denote by $d \leq m$ the number of nonzero elements per column of

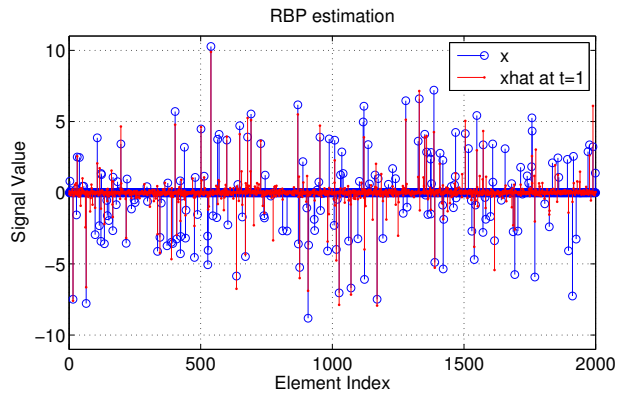


(a)

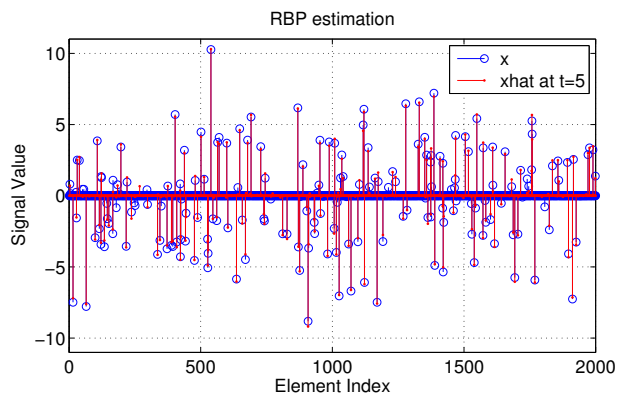


(b)

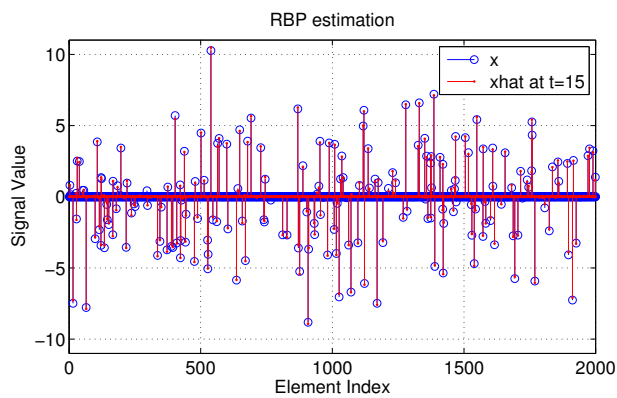
Figure 3.2: (a) Histogram of the measurements \mathbf{z} compared against normal distribution. From a central limit theorem argument for $n \rightarrow +\infty$ elements of \mathbf{z} will follow normal pdf. (b) Error performance of RBP in 15 iterations. We consider estimation of a sparse signal of length $n = 2000$ from quantized measurements. See text for details.



(a)



(b)



(c)

Figure 3.3: Reconstruction of the sparse signal using RBP from quantized measurements for iterations (a) $t = 1$, (b) $t = 5$, and (c) $t = 15$. As we see the reconstruction is almost exact.

Φ . In the large sparse limit analysis, first let $n \rightarrow +\infty$ with $m = \beta n$ and keeping d fixed. This enables the local-tree properties of the factor graph G . Then let $d \rightarrow +\infty$, which will enable the use of a Central Limit Theorem approximation.

Theorem 3.1. *Consider the RBP algorithm under the sparse limit model above with transform matrix Φ and index i satisfying the Assumption 1 of [16] for some fixed iteration number t . Then the error variances satisfy the limit*

$$\lim_{d \rightarrow +\infty} \lim_{n \rightarrow +\infty} \mathbb{E} \left\{ \left| X_i - \hat{X}_i \right|_{\ell_2}^2 \right\} = \bar{\tau}_t, \quad (3.24)$$

where $\bar{\tau}_t$ is the output of the SE equation (3.20).

Proof. See [16]. □

A second important result regarding the SE recursion demonstrates its convergence to some fixed point.

Theorem 3.2. *Consider the SE equations (3.20) initialized with $\bar{\tau}_0 = \hat{\tau}_{init}$. Suppose that $\bar{\mathcal{E}}_{in}(\nu)$ and $1/\bar{D}_2(\nu, \sigma^2)$ are continuous. Then, we have the limit:*

$$\lim_{t \rightarrow +\infty} \bar{\tau}_t = \tau, \quad (3.25)$$

where $\hat{\tau}_t$ decreases monotonically and τ satisfies the fixed point equation

$$\tau = \bar{\mathcal{E}}_{in} \left(\frac{1}{\bar{D}_2(\beta\tau, \sigma^2)} \right). \quad (3.26)$$

Proof. See [16]. □

The more complete analysis of RBP also demonstrates that it is asymptotically MSE optimal when the SE equation has a unique fixed point. Although in practice the measurement matrices are rarely sparse, simulations (see Figure 3.4) show that SE predicts well the behavior of relaxed BP. Moreover, recently more sophisticated techniques were used to demonstrate the convergence of approximate message passing algorithms to SE under large i.i.d. Gaussian matrices [1, 17].

Figure 3.4 considers predictions made by SE and their quantitative match with empirical observations. As discussed in Theorem 3.1 the asymptotic mean-square estimation error can be easily predicted via SE recursion (3.20), specifically the asymptotic MSE of reconstruction is the state variable that the recursion tracks. We compared the state evolution predictions with the actual reconstruction performance obtained by a Monte Carlo experiment.

We set (β, σ^2) to $(2, 0.1)$, $(3, 0.1)$, $(2, 0)$ and run 1000 problem instances for each $n \in \{100, 1000, 2000\}$. The signal prior used is the same as in (3.18) with $\rho = 0.1$ and the quantizer boundaries are given by (3.19). We ran each RBP reconstruction with a fixed number of iterations and averaged the results across the 1000 Monte Carlo realizations, producing empirical averages. In panels (b), (d), and (f) we illustrate the degree of variability of observations around the SE prediction by plotting the CDF of the reconstruction MSE over all Monte Carlo realizations. We see that there is a large variation for small n , which means that even if SE promises great error performance the actual reconstruction might perform well below it. However, as n increases variation decreases and MSE concentrates more tightly around the SE prediction.

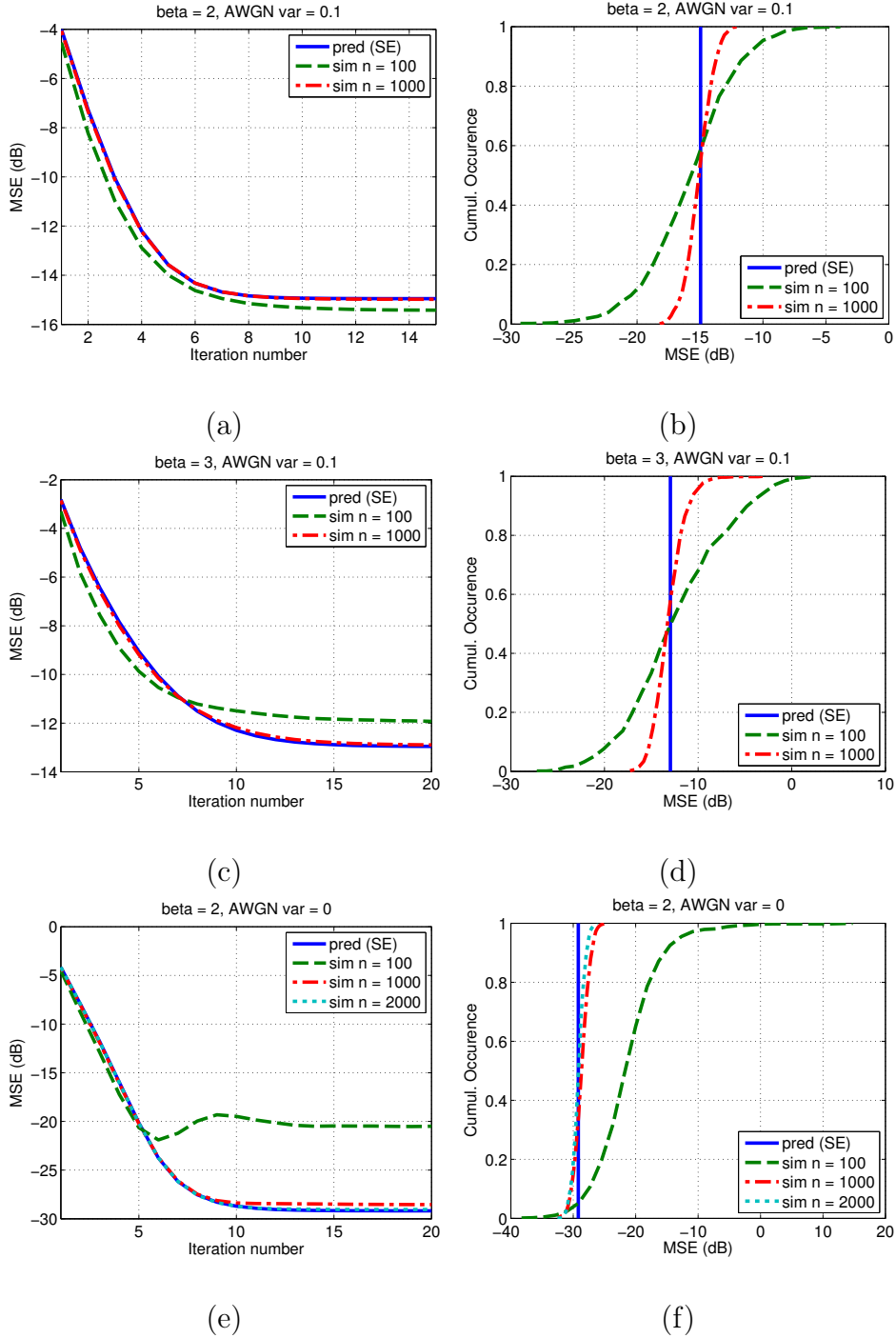


Figure 3.4: Comparison of SE predictions against observations. Panels (a), (c), (e) compare evolution of the MSE as iterations progress for various levels of AWGN and undersampling $\beta = n/m$. Blue solid curve plots the SE predicted MSE, while other curves plot mean MSE from empirical RBP reconstructions. Panels (b), (d), (f) demonstrate the variations of reconstruction MSE around SE prediction. As n grows empirical observations get tighter around SE prediction. We use the same uniform 32-level quantizer for all experiments. See text for details.

Chapter 4

Optimal Quantization for RBP Estimation

In this chapter we will present a framework to design asymptotically optimal quantizers for compressive sensing estimation via relaxed belief propagation. We say that the quantizer is optimal if it minimizes MSE of the estimation. We start by designing optimal regular quantizers and demonstrate their superior performance to other standard quantization schemes. We then dramatically improve the rate-distortion performance of quantization by binning disjoint quantization cells. As our optimization scheme is based on state evolution formalism, we note that in principle optimal quantizers for both approximate message passing (AMP) and RBP estimation can be obtained in parallel. However, in this thesis we concentrate on design for RBP while keeping in mind that identical work can be done for AMP as well.

4.1 Optimal Regular Quantization

We are now interested in solving (2.35) to design MSE-optimal quantizers under RBP. By modeling the quantizer as part of the channel and working out resulting equations for RBP and SE, we can make use of the prediction capability of SE to recast our optimization problem to

$$Q^{SE} = \arg \min_Q \left\{ \lim_{t \rightarrow \infty} \bar{\tau}_t \right\}, \quad (4.1)$$

where $\bar{\tau}_t$ is MSE prediction obtained by SE recursion (3.20). In practice either we run SE recursion for fixed amount of time, or we set some stopping criteria. For example, due to monotonic convergence of SE we could stop the

algorithm when the change in MSE is small from one iteration to the next

$$|\bar{\tau}_t - \bar{\tau}_{t-1}| < \text{tol}, \quad (4.2)$$

where tol is termination tolerance on MSE and $t > 0$ is the iteration number. Then by applying Theorem 3.1, we know that the asymptotic performance of Q^* of (2.35) will be identical to that of Q^{SE} .

The SE recursion behaves well under quantizer optimization (4.1). We can see this by looking at the equations (3.16), which is the only place modified during the optimization process. By inspecting this equation we can see that small changes in the quantizer boundaries result only in minor change in the recursion. Although, due to recursive nature of the SE, finding analytic or closed form expressions for derivatives of $\bar{\tau}_t$ for large t 's is hard, we can approximate them by using finite difference methods. Moreover, the recursion itself is fast to evaluate, which makes scheme (4.1) practically realizable under standard constrained optimization methods like sequential quadratic programming (SQP). Note that in principle $\bar{\tau}_t$ might admit many local minima, which might complicate the process of finding a global minimum. However, we found via numerous simulations that for regular quantizers there is a unique solution to (4.1).

Figure 4.1 depicts $\bar{\tau}_t$ at its fixed points as a function of quantization boundaries. We consider the signal model in (3.18) and set the AWGN variance to zero. In the panel (a) we consider 4-level regular quantizer with quantization boundaries

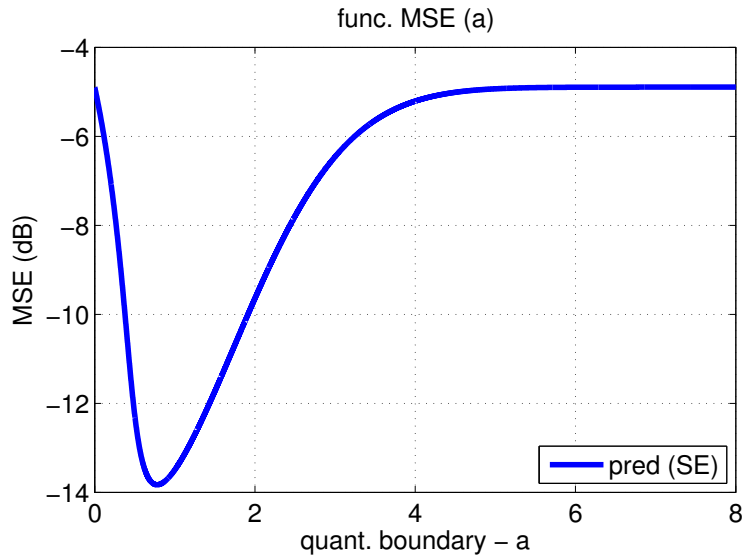
$$[-\infty, -a, 0, a, \infty], \quad (4.3)$$

where $0 < a < \infty$ is the parameter to optimize. We pick symmetric quantizer due to the symmetry of our problem (i.e. distribution of the measurements \mathbf{Z} is symmetric around 0). By optimizing the parameter a the MSE of the reconstruction can be substantially reduced. Panel (b) considers 6-level regular quantizer with the following boundaries

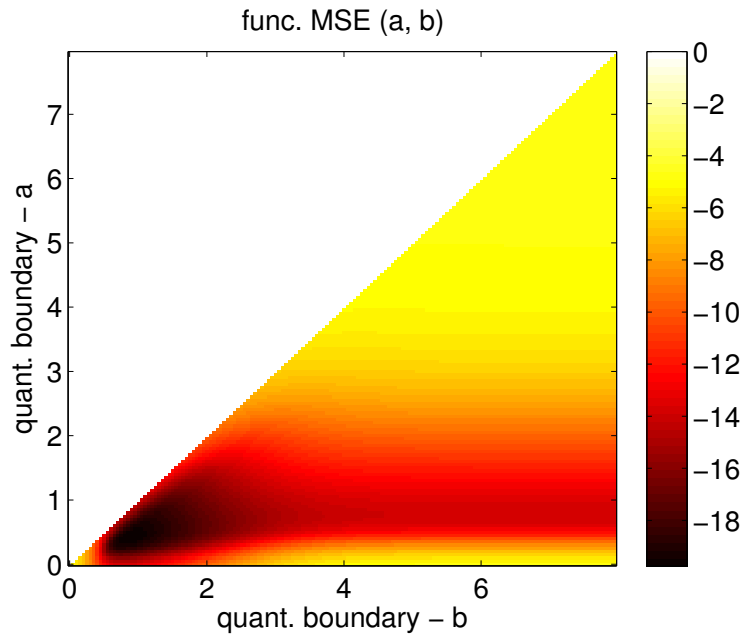
$$[-\infty, -b, -a, 0, a, b, \infty], \quad (4.4)$$

where $0 < a < b < \infty$ are the parameters to optimize. In both of the panels there exists a unique quantizer minimizing MSE. Simulations indicate that the uniqueness of the minimizer can be actually generalized to arbitrary N -level regular quantizers. Furthermore, when measurements are symmetric around the origin, optimal regular quantizers are also symmetric.

Figure 4.2 demonstrates distortions obtained by several types of quantizers under RBP estimation. In particular it compares quantizers obtained



(a)



(b)

Figure 4.1: Illustration of $\bar{\tau}_\infty$ for (a) 4-level quantizer and (b) 6-level quantizer as a function of quantization boundary values. Due to symmetry of \mathbf{S} and of SE recursion around zero, we have considered quantization boundaries to be symmetric as well. As we can see in the plot, for this case MSE is a smooth function of quantizer boundaries with a unique minimum. Other parameters used: $\sigma^2 = 0$, $\beta = 2$, $\rho = 0.1$.

via classical *Lloyd algorithm* [10] against quantizer obtained by (4.1). In this simulation, as before, we assume Gauss-Bernoulli distribution for X_i , with sparsity parameter $\rho = 0.1$. We fix quantizer boundary points $b_0 = -\infty$ and $b_N = +\infty$ for all the quantizers. We obtain *Uniform* quantizer via optimization of type (2.34), but with additional constraint of equally-spaced output levels. The quantizer *Lloyd* was obtained by iterating through necessary conditions of regular quantizer optimality. The measurements S_a for both Uniform and Lloyd quantizers were approximated as Gaussians by application of the central limit theorem. To determine *Optimal* quantizer, we perform (4.1) by using a standard SQP optimization algorithm for nonlinear continuous optimization. As we see the MSE of the quantizer optimized for the RBP reconstruction is much smaller, with more that 4 dB difference for many rates.

In [9] it has been shown that, compared to adaptive compression schemes, quantization of compressive sensing measurements incurs a significant penalty in terms of rate-distortion performance. In particular adaptive encoding of k -sparse signals with kR bits results in MSE that decreases exponentially as 2^{-2R} . As a consequence, in the adaptive encoding scenario approximately 6 dB/b decrease in distortion is achieved with the rate R . We have to renormalize this performance to be able to compare it to the results in Figure 4.2. Compressing a k -sparse signal with kR bits is equivalent to using

$$R_X = \frac{kR}{n} = \rho R, \quad (4.5)$$

bits per signal component, where $\rho = k/n$ is the sparsity ratio of the signal. This in turn is equivalent to using

$$R_S = \beta R_X = \beta \rho R, \quad (4.6)$$

bits per measurement, with $\beta = n/m$ the undersampling ratio. Equation (4.6) indicates that for a fixed $\rho = 0.1$, we can achieve 30 dB/b and 20 dB/b decrease in distortion with rate R_S for $\beta = 2$ and $\beta = 3$ respectively. This can be compared to respective 8.5 dB/b and 8.9 dB/b average decrease that we can observe for the optimal regular quantizers in Figure 4.2. Hence, even with optimal regular quantization, the penalty to be paid for using a non-adaptive or universal encoding is still significant.

Figure 4.3 demonstrates the quantization boundaries for quantizers obtained via simulation above. As we can see in comparison to Uniform and Lloyd quantizers Optimal quantizer has its boundaries clustered closer zero. This implies that high resolution approximations of the measurements in the range $[-1, 1]$ is preferred for better estimation with RBP. Another remarkable observation in Figure 4.3 is that quantization regions of the Optimal

quantizer have almost uniform spacing. This implies that we can potentially obtain comparable performance to Optimal quantizer by using uniform quantizers optimized via (4.1).

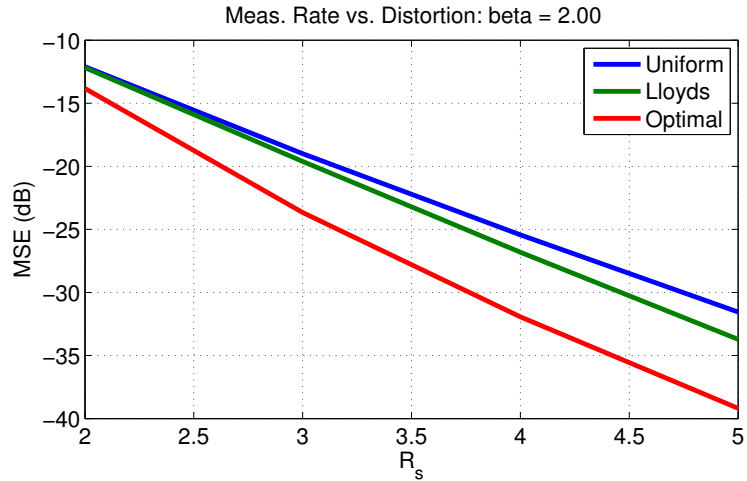
We verify the statement above in Figure 4.4. For the same problem setting as before we perform (4.1), but with additional constraint of equally spaced output levels. This results in a very simple and extremely fast optimization where we optimize over a single variable (similar to Figure 4.1(a)). As we can see in the figure, the resulting quantizer performs nearly as well as absolutely best regular quantizer with difference in MSE of only 0.2 dB.

Finally in Figure 4.5 we present MSE performance of RBP for various rates R_X (measured in bits/signal component). We vary the quantization rate from 1 to 2 bits per component of \mathbf{x} , and for each rate, we determine optimal undersampling β and quantizers based on methods discussed above. For comparison, Figure 4.5 also shows the performance of two other simple algorithms. The top curve is the MSE for optimal linear MMSE estimation, and the curve labeled LASSO is the MSE from the widely used LASSO algorithm from [23]. Performance of both algorithms is computed assuming a bounded uniform quantizer and the LASSO performance was predicted by state evolution equations in [17], with the thresholding parameter optimized by the iterative approach in [18]. It can be seen that RBP offers dramatically better performance with more than 10 dB of improvement. The figure also demonstrates that additional gains in performance can be attained by optimizing the quantizer.

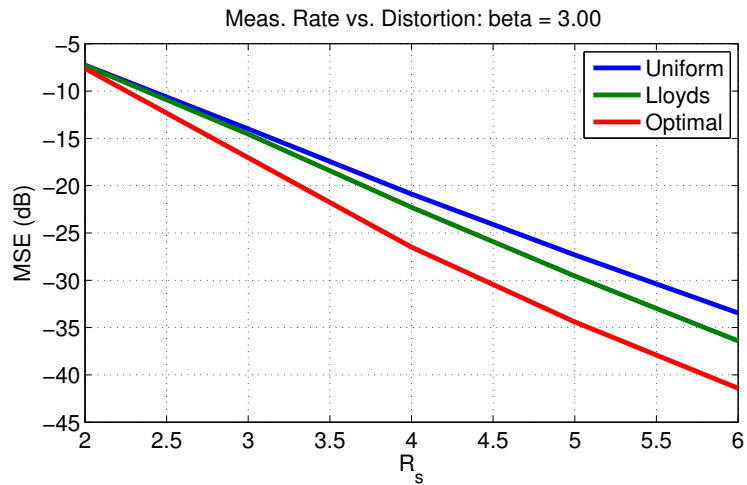
4.2 Optimizing Binned Quantizers

One way to further reduce the distortion of the RBP reconstruction while keeping rates constant is to use binned quantizers [15]. Binning strategy becomes viable in any scenario where auxiliary information about the signal being quantized is available at the decoder. In our problem at hand with RBP reconstruction, we have assumed that the algorithm has access to the sparse prior of the signal \mathbf{X} . We can then consider binning disjoint quantizer cells, and let the estimation algorithm infer which cell is the correct one during the reconstruction. Intuitive explanation why reductions are possible lie in the fact that in the measurement space \mathbb{R}^m , the k -sparse signal lies in only a small subset of dimension \mathbb{R}^k (Figure 2.9). Hence, by binning (or nesting) quantizers we can potentially obtain better rate-distortion performance.

Figure 4.6 presents comparison of optimized regular quantizers against optimized binned ones. We use binning strategy depicted in Figure 2.8(b) where quantization indices are taken modulo M . For example, if we have a

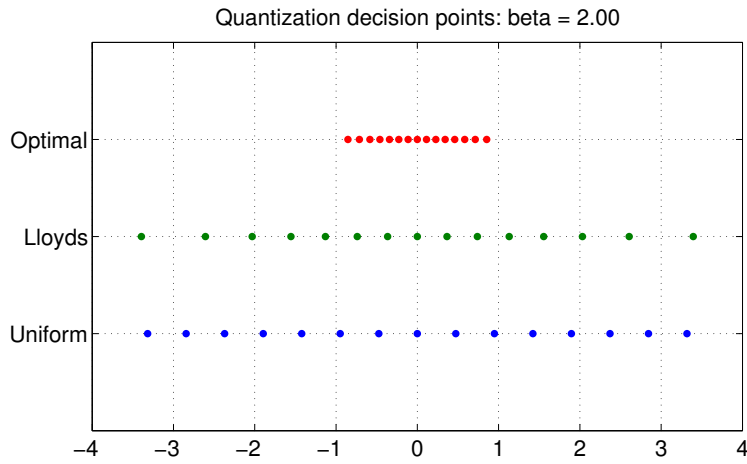


(a)

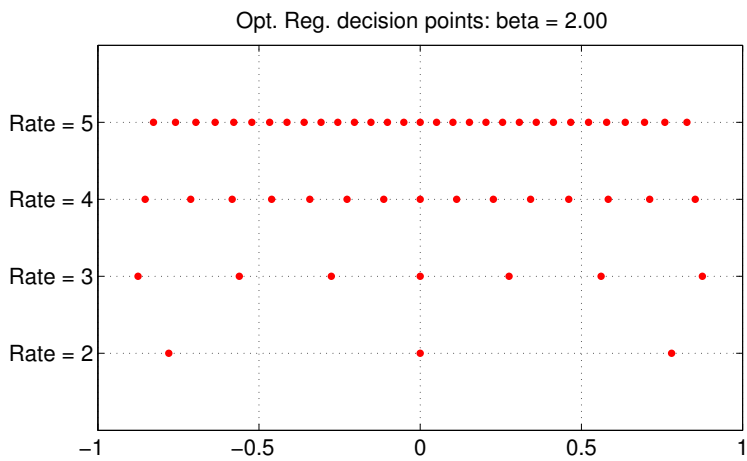


(b)

Figure 4.2: Performance comparison of quantizers under RBP reconstruction for (a) $\beta = 2$ and (b) $\beta = 3$. We plot MSE versus rate R_S in bits/measurement. As we see optimal quantizer obtained by (4.1) significantly outperforms quantizers optimized by minimizing MSE between quantizer input and output as in (2.34). Curves obtained via SE recursion with parameters $\rho = 0.1$ and $\sigma^2 = 0$.

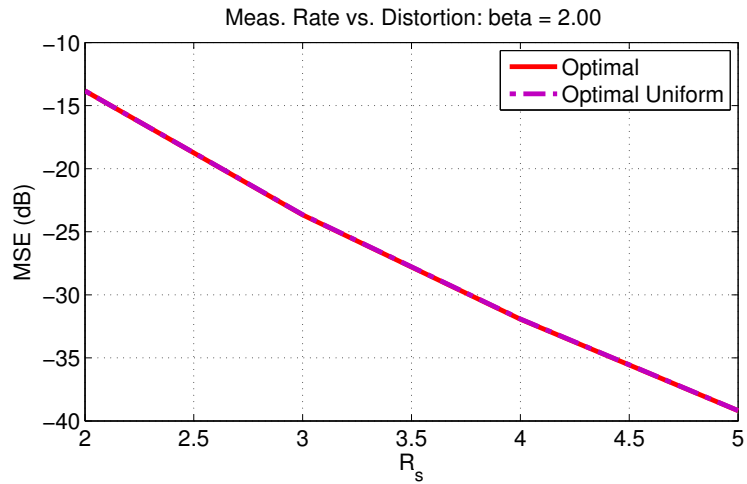


(a)

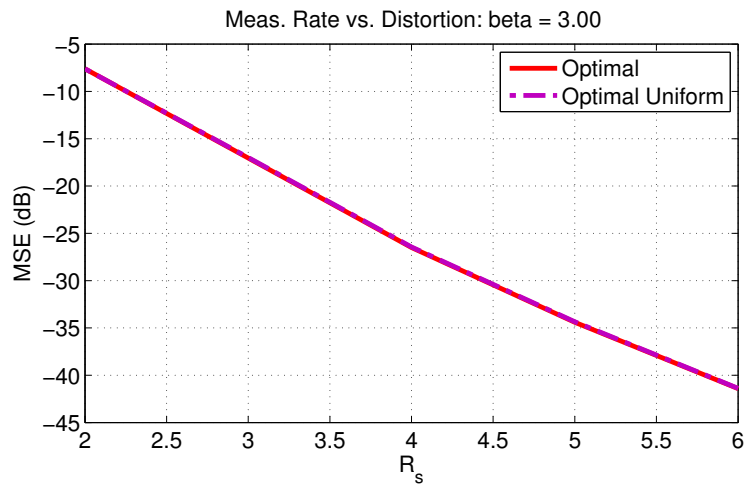


(b)

Figure 4.3: Illustration of quantization boundaries for the Figure 4.2 (a). Panel (a) compares quantization boundaries of three quantizers for the rate 4 bits/measurement. In panel (b) we illustrate the boundaries of the optimal regular quantizer for various rates. Note that optimal quantizer boundaries are clustered around zero and have nearly uniform spacing between each other.



(a)



(b)

Figure 4.4: Performance comparison of SE optimized uniform quantizer against optimal quantizer for (a) $\beta = 2$ and (b) $\beta = 3$. It is remarkable that RBP optimal quantizer has nearly uniform behaviour. In the plot the difference between the optimal quantizer and optimized uniform quantizer is within 0.2 dB of each other.

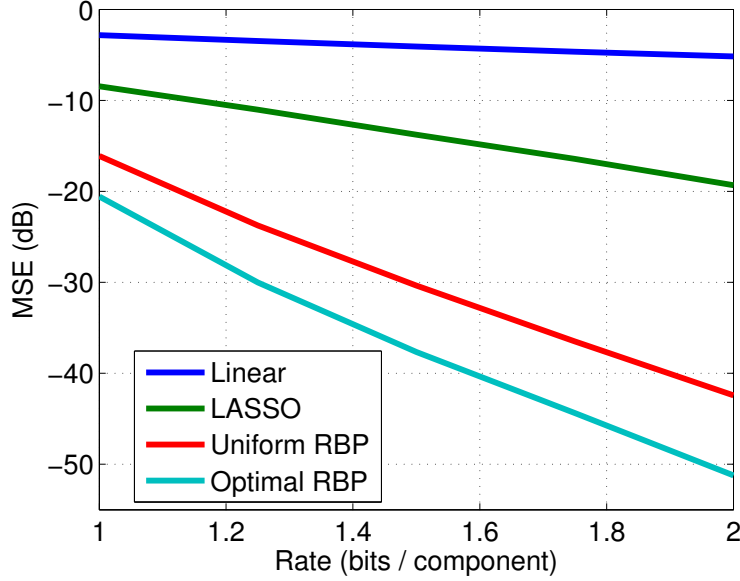


Figure 4.5: Comparison of quantized RBP performance with other standard sparse reconstruction methods. MSE performances versus the rate R_X bits/signal component for Uniform and Optimal quantizers for RBP. See text for details.

regular quantizer with $N = 8$ levels

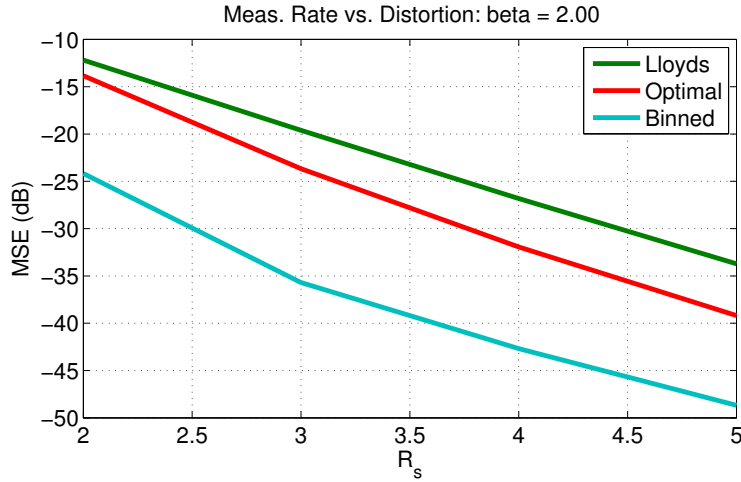
$$[1, 2, 3, 4, 5, 6, 7, 8], \quad (4.7)$$

we can bin $L = 2$ levels/bin to obtain $M = 4$ total bins

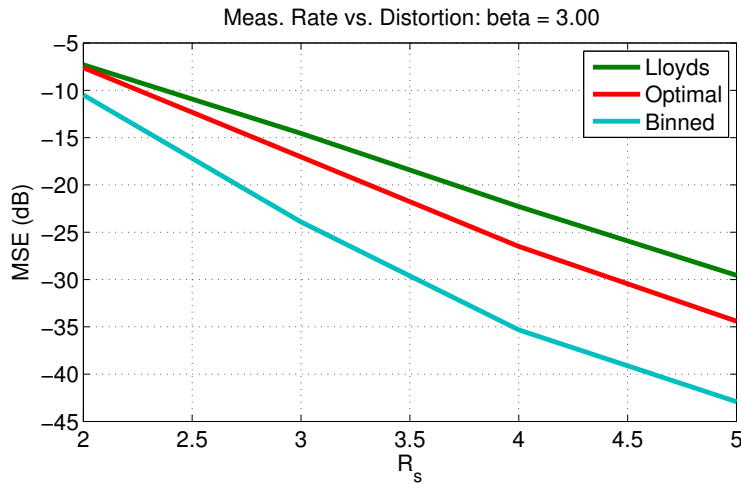
$$[1, 2, 3, 4, 1, 2, 3, 4]. \quad (4.8)$$

Optimization is then performed over quantization boundaries via (4.1). For the simulations presented in Figure 4.6 we set parameters the same way as for Figures 4.2 and 4.3 in the previous section. Additionally we optimize between two $L = 2$ and $L = 3$ and choose the one with the best MSE performance. As we can see significant performance improvements, up to 20 dB in the figures, can be obtained by binning quantizer outputs. However, optimization (4.1) for binned quantizers becomes a non-trivial task as the MSE function that we would like to minimize might develop few local minima. In our case for finding the global minimum we utilized standard functions in the Global Optimization Toolbox of the MATLAB software package. One important observation is that binning achieves approximately 9.3 dB/b and 11 dB/b decrease in MSE with rate R_S for $\beta = 2$ and $\beta = 3$ respectively. This

compares favorably with respective 8.5 dB/b and 8.9 dB/b obtained for optimal regular quantizers. Hence, binning effectively reduces the gap between the performances of the adaptive and non-adaptive encoding schemes.



(a)



(b)

Figure 4.6: Performance comparison of regular and binned quantizers under RBP for (a) $\beta = 2$ and (b) $\beta = 3$. Plotted are MSE of the reconstruction versus rate R_s in bits/measurements. The best error performance was achieved by binned quantizer with non-trivial gains across all rates. The curves are obtained for sparsity ratio $\rho = 0.1$ and AWGN variance $\sigma^2 = 0$.

Chapter 5

Conclusions

Quantization is a necessary component of any practical acquisition system, yet due to its nonlinear nature it does not lend itself to easy analysis. As a consequence the topic of quantized measurements is often ignored in compressive sensing literature. In this work we have studied the topic of quantizer design for compressive sensing under reconstruction by a family of algorithms commonly referred to as message-passing algorithms. In particular we followed the work on the relaxed belief propagation algorithm, however our results can be easily generalized to other similar algorithms like approximate message passing. Message passing algorithms, like relaxed belief propagation or approximate message passing, represent a novel, yet a very powerful class of methods for obtaining great sparse signal estimations with low computational costs. The error performance of these algorithms can be exactly characterized by a simple recursive equation called state evolution. By implementing and running state evolution we can accurately predict MSE or some other error metric of the reconstruction by message passing algorithms. Moreover, by analyzing fixed points of state evolution equation we can easily determine if the estimation is mean-square optimal or not.

Recent results for relaxed belief propagation have demonstrated that it can be generalized to arbitrary measurement channels, where noise of identical transition probability acts on each compressive sensing measurement. We utilized these results to incorporate quantization as part of the channel and worked out relaxed belief propagation and state evolution equations for reconstruction from quantized measurements. We then used the state evolution framework to optimize quantizer boundaries for optimal MSE performance. Our results demonstrate that with optimal quantization error performance of relaxed belief propagation reconstruction improves significantly. Moreover optimal regular quantizers look nearly uniform with quantization cells clustered close to zero. This indicates that for best estimation performance

relaxed belief propagation requires high precision measurements around zero, while allowing coarse quantization of measurements farther away from zero. Our optimization setting can as well be utilized for designing good binned quantizers, where disjoint quantization cells are binned together for reducing the bit rate. We experimentally demonstrate the superior error performance achieved by binned quantizers under relaxed belief propagation reconstruction.

Bibliography

- [1] M. Bayati and A. Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Trans. Inform. Theory*, 57(2):764–785, February 2011.
- [2] E. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.
- [3] E. J. Candès and J. Romberg. Encoding the ℓ_p ball from limited measurements. In *Proc. IEEE Data Compression Conf.*, pages 33–42, Snowbird, UT, March 2006.
- [4] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, February 2006.
- [5] S. Sarvotham D. Baron and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *Signal Processing, IEEE Transactions*, 58(1):269–280, 2008.
- [6] W. Dai, H. V. Pham, and O. Milenkovic. A comparative study of quantized compressive sensing schemes. In *Proc. IEEE International Symposium on Information Theory, ISIT'09*, pages 11–15, 2009.
- [7] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, April 2006.
- [8] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, November 2009.
- [9] V. K. Goyal, A. K. Fletcher, and S. Rangan. Compressive sampling and lossy compression. *IEEE Signal Processing Mag.*, 25(2):48–56, March 2008.

- [10] R. M. Gray and D. L. Neuhoff. Quantization. *IEEE Trans. Inform. Theory*, 44(6):2325–2383, October 1998.
- [11] D. Guo and C.-C. Wang. Asymptotic mean-square optimality of belief propagation for sparse linear systems. *Proc. Inform. Th. Workshop*, pages 194–198, 2006.
- [12] W. T. Freeman J. Yedidia and Y. Weiss. *Exploring Artificial Intelligence in the New Millenium*, chapter Understanding belief propagation and its generalizations, pages 239–269. Morgan Kaufmann, 2003.
- [13] David MacKay. *Information Theory, Inference, and Learning Algorithms*, chapter Exact Marginalization in Graphs. Cambridge University Press, 2005.
- [14] A. Maleki and D. L. Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE J. Sel. Top. Signal Process.*, 4(2):330–341, April 2010.
- [15] R. J. Pai. Nonadaptive lossy encoding of sparse signals. Master’s thesis, Massachusetts Inst. of Tech., Cambridge, MA, August 2006.
- [16] S. Rangan. Estimation with random linear mixing, belief propagation and compressed sensing. *Information Sciences and Systems (CISS), 2010 44th Annual Conference*, pages 1–6, 2010.
- [17] S. Rangan. Generalized approximate message passing for estimation with random linear mixing. arXiv:1010.5141v1 [cs.IT]., October 2010.
- [18] S. Rangan, A. Fletcher, and V. Goyal. Asymptotic analysis of map estimation via the replica method and applications to compressed sensing. <http://arxiv.org/abs/0906.3234>, 2009.
- [19] T. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, February 2001.
- [20] T. Richardson and R. Urbanke. Fixed points and stability of density evolution. *Communications in information and systems*, 4(1):103–116, 2004.
- [21] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*, chapter Factor Graphs. Cambridge University Press, New York, NY, 2008.

- [22] J. Z. Sun and V. K. Goyal. Optimal quantization of random measurements in compressed sensing. In *Proc. IEEE International Symposium on Information Theory*, ISIT'09, pages 6–10, 2009.
- [23] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. Ser. B*, 58(1):267–288, 1996.
- [24] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.
- [25] A. Zymnis, S. Boyd, and E. Candes. Compressed sensing with quantized measurements. *IEEE Signal Processing Letters*, 17:149–152, February 2010.