Approximate Message Passing Algorithms for Compressed Sensing

A Project Report

submitted by

HARIKUMAR KRISHNAMURTHY

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2019

THESIS CERTIFICATE

This is to certify that the thesis titled **Approximate Message Passing Algorithms for Compressed Sensing**, submitted by **Harikumar Krishnamurthy**, to the Indian Institute of Technology, Madras, for the award of the degree of **Dual Degree (Bachelor of Technology and Master of Technology)**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Andrew Thangaraj
Research Guide
Professor
Dept. of Electrical Engineering
IIT Madras, 600036

Place: Chennai

Date: 10th May 2019

ACKNOWLEDGEMENTS

Firstly, I would like to sincerely thank my guide Prof. Andrew Thangaraj for his constant support and motivation throughout the course of my project. His key insights, attention to details and suggestions were crucial to the progress of the project.

Besides my advisor, I would like to extend my thanks to Parikshit Hegde for working on this problem along with me and for all the stimulating discussions we had about the problem.

I am also thankful to all my friends for being a strong pillar of support and making my five years in this beautiful campus a very memorable experience.

Lastly, I would like to thank my parents and my brother for their constant encouragement, love and unwavering belief in me.

ABSTRACT

KEYWORDS: Approximate Message Passing, Compressed Sensing, Iterative Thresholding

Compressed sensing aims to undersample high-dimensional signals while still accurately reconstructing them by exploiting prior knowledge on the signal. Exact reconstruction is possible when the signal to be recovered is sufficiently sparse in a known basis. Having applications in wide areas of signal processing, machine learning and image processing, there has been extensive work in obtaining efficient algorithms to solve this problem. Convex optimization solutions are expensive for large systems and hence there has been considerable interest in fast iterative algorithms. Approximate Message Passing (AMP) algorithms provide the best of both worlds in terms of performance and speed. In this work we analyze the performance of AMP and provide a simple modification to AMP to improve its convergence.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

NP	Nondeterministic Polynomial Time
LP	Linear Programming
QP	Quadratic Programming
MP	Message Passing
MRI	Magnetic Resonance Imaging
DCT	Discrete Cosine Transform
IID	Independent and Identically Distributed
AMP	Approximate Message Passing
MSE	Mean Square Error
LDPC	Low Density Parity Check
GAMP	Generalized Approximate Message Passing
VAMP	Vector Approximate Message Passing
JPEG	Joint Photographic Experts Group

NOTATION

y	Output vector
A	Measurement matrix
x	Input vector
k	Number of non-zero elements in signal
n	Number of Measurements
N	Size of signal
δ	$^{n}/_{N}$
ρ	$^{k}/_{n}$
x_0	Original signal
x_i	i^{th} element of x
$\|\cdot\|_p$	p-norm
x^t	Estimate of signal at time t
z^t	Residue at time t
$\eta_t(.)$	Filter at time t
τ	Threshold
σ_t^2	Variance of the error at time t
γ^t	$A^* z^t + x^t$
α	Scaling factor in Scaled AMP

CHAPTER 1

INTRODUCTION

A major area of interest in signal processing deals with the reconstruction of a signal from a sequence of measurements. In general, it is not possible to uniquely reconstruct the signal as it can take arbitrary values at time instants when the signal is not measured. However, with some prior knowledge or certain assumptions on the signal, we can reconstruct the signal perfectly with a series of measurements. One such result is the Nyquist-Shannon sampling theorem which states that if the signal's maximum frequency is bounded by f_B, it is sufficient to sample the signal uniformly at time instants separated by $\frac{1}{2f_B}$, i.e., sample at a frequency of at least twice the signal's maximum frequency. The signal then can be reconstructed perfectly by means of sinc interpolation on these measurements. These assumptions and prior knowledge on the signal space vastly limits the solution space resulting in a unique solution.

We can reduce the required number of measurements even further when we have more prior knowledge on the signal. One common and realistic assumption is that of sparsity. Compressive sensing or sparse signal recovery allows us to perfectly reconstruct a signal using a small number of measurements of the signal provided the signal has a sparse representation in some transform domain [1, 2]. As most real life signals have compressible representation in certain domains, we can recover the signal with far fewer measurements than traditionally required by sampling theorems. However simple linear reconstruction protocols as in sampling theorems are no longer sufficient and we need to consider algorithms with non-linearity to efficiently recover the original signal. Few such applications include faster acquisition of Magnetic Resonance Imaging (MRI) signals [3, 4], computational photography, imaging using coded aperture and even network management [5].

Ideas from sparse signal estimation have been used extensively in various problems in signal processing, machine learning and image processing. Also use of sparsity-inspired models have resulted in state of the art results for a large set of application [6, 7, 8]. Sparse representation modelling also seems to have a strong connection with deep learning. [9]

1.1 Organization

In Chapter 2, we formalize the model under consideration and present algorithms to solve the compressed sensing problem. We will consider optimization and iterative approaches. We also define phase transition curves as a measure of performance of such algorithms. The main algorithm of interest would be the Approximate Message Passing algorithm which we describe in greater detail. We also compare its performance against that of optimization based algorithms through simulations

In Chapter 3, we explore the Universality of the AMP algorithm and evaluate its performance on sub-Exponential matrices.

In Chapter 4, we suggest an extension to AMP called scaled AMP to improve the convergence in the case of small systems. We show through simulations that scaled AMP also provides non-trivial performance gains for measurement matrices sampled according to heavy tailed distributions.

In Chapter 5, we conclude with the summary of the report along with some possible directions for future work.

CHAPTER 2

Efficient Algorithms for Sparse Signal Recovery

2.1 Compressed Sensing

The most basic model of sparse recovery involves a $N-$dimensional signal x which is measured through a known $n \times N$ sensing matrix A to obtain the measurement vector y of dimension n.

$$y = Ax$$

However the matrix A is fat, i.e., the number of rows is far lesser than the number of columns($n < N$). Hence the system of linear equations is underdetermined and can potentially have an infinite number of solutions. To combat this issue we assume that the signal x has a sparse support, i.e., the number of non-zero elements of x is limited.

One such problem setting is the acquisition of very large images using a single pixel camera. Here, rather than measuring the individual pixel values, we measure the inner product of the scene with a set of test functions far fewer than the number of pixels. When the image is compressible under JPEG, i.e., has a sparse Discrete Cosine Transform, we can reconstruct the image accurately. Here the signal under consideration would be the image in the DCT domain and the sensing matrix is the product of the measurement matrix and the DCT matrix. By carefully choosing our set of test functions, we can significantly reduce the number of measurements required [10].

In certain other problem settings, we may not have the freedom to design our sensing matrix. Provided the matrix A has certain properties, in most cases we can reconstruct the signal perfectly.

2.1.1 Model

Let $x_0 \in \mathbb{R}^N$ be the original signal of interest. Let A be a $n \times N$ sensing/measurement matrix with $n < N$. We obtain the measurement vector $y \in \mathbb{R}^n$ from x_0 as $y = Ax_0$.

Also k is the number of non zero entries in x_0. We wish to recover x_0 from knowledge of A and y.

As the system is underdetermined, recovery is not always possible. However under certain conditions on k, n, N, we can obtain algorithms which recover x_0 perfectly.

There are multiple canonical models for the signal x_0. Two such models are:

- $+$: The signal is non-negative and has at most k non-zero entries
- $\pm$: The signal has at most k non-zero entries which can have arbitrary signs.

In this report we consider the second case.

2.1.2 Phase transitions

For any algorithm, the trade-off between undersampling and sparsity can be most easily described in the large-system limit. We tend $k, n, N \to \infty$ such that the ratio of $n/N \to \delta$ and $k/n \to \rho$. Here, δ is a measure of undersampling while ρ is a measure of sparsity where smaller the value of ρ, larger the sparsity. We can associate every pair of $(\delta, \rho) \in (0,1)^2$ into one of two phases. The 'success' phase consists of all points for which an algorithm typically succeeds in recovering the signal perfectly and the 'failure' phase where it typically fails. The boundary separating these two phases forms the phase transition. These phase transition diagrams depend on the algorithm used for reconstruction, the canonical model for the signal and also on the ensemble of matrices under consideration. Some ensembles of random matrices of interest are :

- Gaussian : Random matrix A with entries sampled from i.i.d. $N(0,1)$
- Rademacher : Random matrix A with entries ± 1 with equal probability
- Bernoulli : Random matrix A with entries 0 or 1 with equal probability
- Partial Fourier : Sub-matrix of a Fourier matrix with random rows deleted

2.2 Optimization Based approach

2.2.1 Exact Solution

For the given measurement y we seek the solution x_0 to the equation $y = Ax$ such that among all solutions of the underdetermined system, x_0 has the least number of non-zero coefficients. This can be rewritten as a constrained optimization problem as

$$\hat{x_0} = \arg\min_{x \in \mathbb{R}^N} \|x\|_0 \quad \text{subject to } y = Ax$$

where $\|x\|_0 = |\{i : i \in \{1, 2, \ldots, N\}, x_i \neq 0\}|$ is the pseudo L_0 norm which counts the number of non-zero elements in x. This is however a NP-Hard problem and hence we do not have efficient algorithms which can solve it.

Even in the noisy case, i.e., when $y = Ax + w$, where w is the added noise vector, this can solve by relaxing the equality in the constraint to an inequality.

$$\hat{x_0} = \arg\min_{x \in \mathbb{R}^N} \|x\|_0 \quad \text{subject to } \|y - Ax\|_2^2 \leq \epsilon$$

which is equivalently rewritten in the Lagrangian form as

$$\hat{x_0} = \arg\min_{x \in \mathbb{R}^N} \lambda \|x\|_0 + \frac{1}{2} \|y - Ax\|_2^2$$

where λ is the regularization parameter.

In general even the noisy sparse recovery is intractable and belongs to the set of NP-Hard problems. Hence we explore approximate solutions to the equation above.

2.2.2 Convex Relaxation

One way to change the NP-Hard problem to a tractable one is by convex relaxation, i.e., by replacing the pseudo L_0 norm by a norm such as L_1 or L_2. Minimizing the L_2 norm results in the least square solution which is the solution with the least energy. This is simple to perform as it involves only multiplication by the pseudo inverse. However in the case of sparse signal recovery, it does not accurately obtain the sparse solution. On the other hand, minimzing the L_1 norm which is the sum of the absolute values, closely

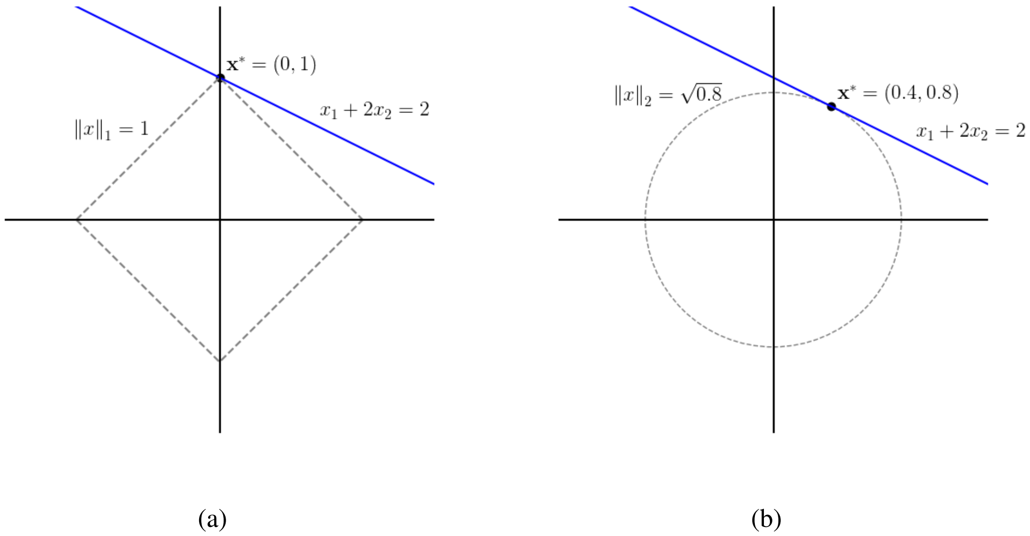


Figure 2.1: Finding sparse solution to the equation $x_1 + 2x_2 = 2$ using
(a) L_1 minimization which gives the correct result
(b) L_2 minimization which gives a low energy result which is not sparse

emulates the L_0 minimization by effectively pushing the non-zero coefficients to 0.

For the noiseless case this results in

$$\hat{x_0} = \underset{x \in \mathbb{R}^N}{\arg\min} \left\| x \right\|_1 \quad \text{subject to } y = Ax$$

known as basis pursuit.

For the noisy case it results in

$$\hat{x_0} = \underset{x \in \mathbb{R}^N}{\arg\min} \, \lambda \left\| x \right\|_1 + \frac{1}{2} \left\| y - Ax \right\|_2^2$$

which is known as basis pursuit denoising.

Under mild conditions on the sensing matrix A such as Mutual Coherence or Restricted Isometry Property[11] and on the sparsity level, it can be shown that the noiseless sparse recovery problem has a unique solution and that L_1 minimization or basis pursuit finds the correct solution. [12, 13]

We have now reduced the sparse recovery problem to a convex optimization problem. In fact it is a Linear Programming(LP) problem for the noiseless case and a Quadratic Programming(QP) problem in the noisy case. We can now use efficient LP

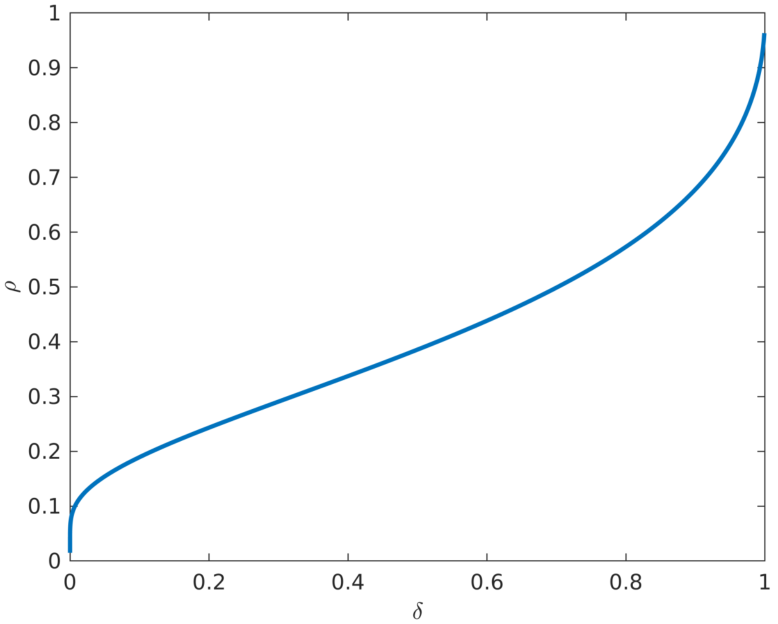


Figure 2.2: Asymptotic phase transition plot for Gaussian ensemble for ± input model

and QP methods to recover the sparse signal.

The asymptotic phase transition for L_1 minimization in the case of Gaussian ensemble can be theoretically obtained through an approach based on combinatorial geometry. [14] It is interesting to note that even for other matrix ensembles, experimentally obtained phase transitions closely match with that of the theoretical Gaussian phase transition.

Despite being more efficient than an L_0 minimization approach, the algorithm does not scale well with large number of variables. In imaging problems, the number of pixels could be as large as 10^6, resulting in a million variables and thousands of linear constraints in the LP problem. Recovering the original signal would be expensive both in terms of time and storage. Hence other approaches such as low complexity iterative thresholding algorithms are of significant interest.

2.3 Iterative Methods

We first convert the minimization problem into a problem of estimating the mean of a probability distribution. [15, 16] Consider the joint probability distribution over the variables $x_1, \ldots, x_N$ as

$$\mu(x) = \frac{1}{Z} \prod_{i=1}^{N} exp(-\beta\,|x_i|) \prod_{j=1}^{n} \delta_{\{y_j=(Ax)_j\}}$$

Here $\delta_{\{y_j=(Ax)_j\}}$ corresponds to the Dirac distribution which is 0 everywhere other than on $y_j = (Ax)_j$. Z is a normalization constant required to make this a probability distribution. As $\beta \to \infty$, the probability mass concentrates around the solutions of the equation having the least number of non-zero entries. If we have access to the marginals and if the solution is unique, we can solve for it using belief propagation.

The factor graph corresponding to this problem has N variable nodes (V) and n factor nodes (C). The edges of the graph correspond to the entries of A. This leads to the graph being a complete bipartite graph. Each edge in the graph is associated with belief propagation messages which in this case are probability measures over the real line. The messages from the variable nodes to the factor nodes are denoted by $\{\nu_{i\to j}\}_{i\in V, j\in C}$ and the messages from the factor nodes to the variable nodes are denoted by $\{\hat{\nu}_{j\to i}\}_{i\in V, j\in C}$.

In the large system limit, the messages to the variable nodes are approximately distributed as Gaussian and to the factor nodes are approximately distributed according to the product of Gaussian and Laplace distributions. Hence it is sufficient to track the distribution parameters(the mean and the variance).

Let us also define $\eta(x;\tau)$ to be the soft thresholding function given by

$$\eta(x;\tau) = \begin{cases} x - \tau & \text{if } x \geq \tau \\ x + \tau & \text{if } x \leq -\tau \\ 0 & \text{otherwise} \end{cases}$$

When also enforcing the large β limit we end up with the following equivalent

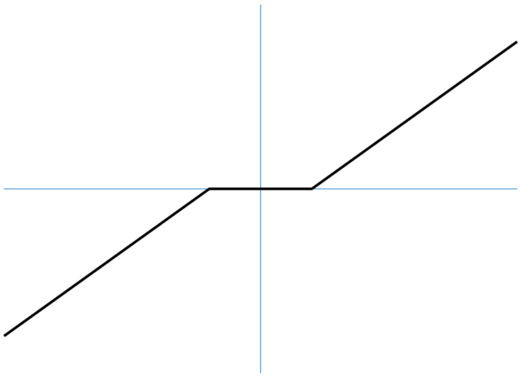


Figure 2.3: Soft Threshold

simpler form for the message passing algorithm.

$$x^{t+1}_{i\to j'} \equiv \eta\left(\sum_{j\neq j'} A_{ji} z^t_{j\to i}; \hat{\tau}^t\right) \tag{2.1}$$

$$z^t_{j\to i'} \equiv y_j - \sum_{i\neq i'} A_{ji} x^t_{i\to j} \tag{2.2}$$

$$\hat{\tau}^{t+1} \equiv \frac{\hat{\tau}^t}{N\delta}\sum_{i=1}^{N} \eta'\left(\sum_{j} A_{ji} z^t_{j\to i}; \hat{\tau}^t\right) \tag{2.3}$$

where $x^t_{i\to j}$ and $\hat{\tau}^t$ are the mean and variance of the message from the variable node $\nu_{i\to j}$ at time instant t.

The updates Equations (2.1) to (2.3) provide an easy way to implement the algorithm. However the number of messages passed is of the order of nN which can be quite large in applications such as imaging.

Notice that the message sent from the factor node j to a variable node i at time t, involves summing over all the messages received by j at time t except for the message from i itself. Hence the only difference between messages to various receivers is the exclusion of the message from the receiver. A similar situation is present in the messages from the variable nodes to the factor nodes. It is tempting to just disregard this exclusion and transmit to all the adjacent nodes the message obtained by adding all the incoming messages. By doing so we end up with the iterative thresholding scheme similar to

$$x^{t+1} = \eta_t\left(A^* z^t + x^t\right) \tag{2.4}$$

$$z^t = y - Ax^t \tag{2.5}$$

where A^* is the transpose of A and $\eta_t\left(.\right)$ are filters which depends on the iteration number.

Such algorithms are quite popular as they have very low per-iteration cost and scale well for large number of variables. Hence they can be used in applications where using standard LP solvers become very time intensive. However the downside to this simple approximation is that they have very poor sparsity undersampling trade-off, far from what standard LP solvers can achieve [17]. In particular, using the current approximated algorithm would result in the estimate diverging from the actual solution for a large set of (ρ, δ) pairs which would have otherwise been in the success phase of the LP algorithm.

2.4 Approximate Message Passing

Approximate Message Passing(AMP) [18] is an iterative approach to compressive sensing which provides sparsity undersampling trade-off similar to that of traditional LP based methods while having a significant speed up as each iteration involves simple matrix multiplications and element wise thresholding.

The algorithm starts with the initial estimate of x as $x^0 = 0$ and iteratively improves its estimate through the updates given below.

$$x^{t+1} = \eta_t\left(A^* z^t + x^t\right) \tag{2.6}$$

$$z^t = y - Ax^t + \frac{1}{\delta} z^{t-1} \langle \eta_t'(A^* z^t + z^t) \rangle \tag{2.7}$$

x^t is the current estimate of x while z^t is the current residue. $\eta_t\left(.\right)$ are scalar threshold functions applied element wise while $\eta_t'\left(a\right)$ is the differential of the filter $\eta_t(.)$ evaluated at a. $\langle s \rangle$ of the vector s is the average of the elements of s.

The difference between the previous iterative methods and AMP is the extra term in the computation of the residue, i.e.,$\frac{1}{\delta} z^{t-1} \langle \eta_t'(A^* z^t + z^t) \rangle$. This correction term is crucial to AMP and leads to performance similar to that of LP methods. Such correction terms are common in statistical physics and are referred to as Onsager Reaction terms.

2.4.1 Heuristic approach to AMP

We consider a heuristic approach to derive the AMP Algorithm starting from the Message Passing Algorithm defined in Equations (2.1) to (2.3). [19]

$$
\begin{aligned}
z^t_{j\to i'} &= y_j + A_{ji'}x^t_{i'\to j} - \sum_i A_{ji}x^t_{i\to j} \\
x^{t+1}_{i\to j'} &= \eta_t\left(-A_{j'i}z^t_{j'\to i} + \sum_j A_{ji}z^t_{j\to i}\right)
\end{aligned}
$$

Due to the choice of normalization of A such that the terms $A_{ji} \approx \frac{1}{\sqrt{n}}$, we can assume that the excluded terms $A_{ji'}x^t_{i'\to j}$ and $A_{j'i}z^t_{j'\to i}$ are $O\left(\frac{1}{\sqrt{n}}\right)$. With this assumption, we can infer that the messages are of the form $z^t_{j\to i} = z^t_j + O\left(\frac{1}{\sqrt{n}}\right)$ and $x^t_{i\to j} = x^t_i + O\left(\frac{1}{\sqrt{n}}\right)$ where x^t_i and z^t_j depend only on the sender and not on the receiver. The approximation in the previous section was obtained by neglecting these $O\left(\frac{1}{\sqrt{n}}\right)$ terms. However, this results in non converging solutions.

Denote these differences as $\delta z^t_{j\to i}$ and $\delta x^t_{i\to j}$. We obtain

$$
\begin{aligned}
z^t_j + \delta z^t_{j\to i'} &= y_j + A_{ji'}(x^t_{i'} + \delta x^t_{i'\to j}) - \sum_i A_{ji}(x^t_i + \delta x^t_{i\to j}) \\
x^{t+1}_i + \delta x^{t+1}_{i\to j'} &= \eta_t\left(-A_{j'i}(z^t_{j'} + \delta z^t_{j'\to i}) + \sum_j A_{ji}(z^t_j + \delta z^t_{j\to i})\right)
\end{aligned}
$$

Single terms of the form $A_{ji}\delta z^t_{j\to i}$ and $A_{ji}\delta x^t_{i\to j}$ are $O(\frac{1}{n})$ and can safely be neglected. We can also linearize η around $\sum_j A_{ji}(z^t_j + \delta z^t_{j\to i})$ to obtain

$$
\begin{aligned}
z^t_j + \delta z^t_{j\to i'} &= y_j + A_{ji'}x^t_{i'} - \sum_i A_{ji}(x^t_i + \delta x^t_{i\to j}) \\
x^{t+1}_i + \delta x^{t+1}_{i\to j'} &= \eta_t\left(\sum_j A_{ji}(z^t_j + \delta z^t_{j\to i})\right) - \eta'_t\left(\sum_j A_{ji}(z^t_j + \delta z^t_{j\to i})\right) A_{j'i}z^t_{j'}
\end{aligned}
$$

Comparing the terms on either side of the equation, we can infer

$$
\begin{aligned}
z^t_j &= y_j - \sum_i A_{ji}(x^t_i + \delta x^t_{i\to j}) \\
\delta z^t_{j\to i'} &= A_{ji}x^t_{i'}
\end{aligned}
$$

and

$$\begin{aligned} x_i^{t+1} &= \eta_t\left(\sum_j A_{ji}(z_j^t + \delta z_{j\to i}^t)\right) \\ \delta x_{i\to j'}^{t+1} &= -\eta_t'\left(\sum_j A_{ji}(z_j^t + \delta z_{j\to i}^t)\right) A_{j'i} z_{j'}^t \end{aligned}$$

Substituting the $\delta z_{j\to i}^t$ value back in the equation for x_i^t, we get

$$x_i^{t+1} = \eta_t\left(\sum_j A_{ji} z_j^t + \sum_j A_{ji}^2 x_i^t\right) \approx \eta_t\left(\sum_j A_{ji} z_j^t + x_i^t\right)$$

where the last approximation is due to the fact that $A_{ji} \approx \frac{1}{\sqrt{n}}$.

Similarly, for z_j^t, we get

$$\begin{aligned} z_j^t &= y_j - \sum_i A_{ji} x_i^t + \sum_i A_{ji}^2 z_j^t \eta_t'\left(\sum_{j'} A_{j'i} z_{j'}^t + x_i^t\right) \\ &\approx y_j - \sum_i A_{ji} x_i^t + \frac{1}{n}\sum_i z_j^t \eta_t'\left(\sum_{j'} A_{j'i} z_{j'}^t + x_i^t\right) \end{aligned}$$

We can rewrite these equations in terms of matrix operations to obtain the AMP update

$$\begin{aligned} x^{t+1} &= \eta_t\left(x^t + A^* z^t\right) \\ z^t &= y - Ax + \frac{1}{\delta}\langle \eta_t'\left(x^t + A^* z^t\right)\rangle \end{aligned}$$

2.4.2 Onsager term

The main idea behind iterative methods is to start with an initial estimate of the signal and through successive iterations, improve the estimate by moving closer towards the original signal. Looking at the distribution of $x_0 - x^t$ after every iteration sheds some light on the convergence of the algorithm.

Let us now consider the algorithm without the Onsager term so as to motivate its importance. Consider the matrix $H = A^*A - I$. We begin with our initial estimate of the signal as $x^0 = 0$ and get $z^0 = y$. Notice that $A^*y = x_0 + Hx_0$. When A is a Gaussian random matrix with variance $\frac{1}{\sqrt{n}}$ and x_0 is sparse, Hx_0 can be accurately modelled as i.i.d. Gaussian entries with variance $\frac{\|x_0\|^2}{n}$. So A^*z^0 is the original signal with some additive Gaussian noise. Soft thresholding this to obtain x^1 gives us a better estimate as soft thresholding with an appropriate choice of threshold reduces the mean-square error in sparse estimation problems. We can continue this for the next iteration where we have $A^*z^1 = x_0 + H(x_0 - x^1)$. Similar reasoning would suggest that the noise now is i.i.d. Gaussian with variance $\frac{\left\|x_0 - x^1\right\|^2}{n}$. Continuing this for all the iterations, we would be able to indirectly track the MSE across iterations. However this does not hold as the noise terms can no longer be approximated as i.i.d. Gaussian as H and $x_0 - x^t$ are no longer independent and are in fact largely correlated.

The inclusion of the Onsager term in the computation of the residue, cancels out a large part of this correlation improving convergence and allows us to efficiently track the MSE across iterations.

2.4.3 State Evolution

An important tool in analyzing the performance of message passing algorithm in decoding of LDPC codes is density evolution. Density evolution allows us to analytically track the probability of error across iterations and obtain the maximum noise at the channel input, the decoding algorithm can tolerate. If we are able to analyze AMP using density evolution, we can obtain analytical phase transition plots for ensembles of our choice. However one of the major assumptions of MP and density evolution is that the Factor Graph is sparse and tree like so as to have no small cycles. In our case the factor graph is far from sparse and is rather a complete bipartite graph.

Combining the intuition from the previous section and motivation through density evolution, [15] introduce state evolution which allows us to track not just the MSE, but any pseudo-Lipschitz function with some mild conditions.

We limit ourselves to the state evolution of MSE for AMP which is given by

$$\sigma_{t+1}^2 = \Psi(\sigma_t^2) \tag{2.8}$$

$$\Psi(\sigma_t^2) = \mathbb{E}\left[\left(\eta_t\left(X + \frac{\sigma_t}{\sqrt{\delta}}Z\right) - X\right)^2\right] \tag{2.9}$$

where σ_t^2 is the average MSE at iteration t. The expectation is over independent random variables X and Z where Z is zero mean Gaussian with variance 1 and X has distribution equal to the empirical distribution of the input signal.

2.4.4 Threshold Parameter

In the message passing formalism, we obtained a recursive definition of the threshold parameter $\hat{\tau}^t$. This makes the algorithm parameter free. Instead we could choose $\hat{\tau}^t$ as parameters and optimize over them to obtain better performance. A similar approach is chosen and we choose $\eta_t(x) = \eta(x; \lambda\sigma_t)$ where $\eta(.)$ is the soft thresholding function.

Define $\rho_{SE}(\delta, \lambda)$ as the maximum value of ρ for which AMP converges for the undersampling fraction δ with λ as the parameter. An interesting point to note is that the convergence region of AMP for the canonical model $\pm$ is independent of the input distribution.

For σ to tend to 0, we require that for σ close to 0, $\Psi(\sigma^2) < \sigma^2$, i.e., $\left.\frac{d\Psi(\sigma^2)}{d\sigma^2}\right|_{\sigma^2=0} < 1$. This is know as local stability and can be used to obtain bounds on ρ as follows

$$\rho_{LS}(\delta, \lambda) = \left.\frac{1 - (2/\delta)[(1+z^2)\Phi(-z) - z\phi(z)]}{1 + z^2 - 2[(1+z^2)\Phi(-z) - z\phi(z)]}\right|_{z=\lambda\sqrt{\delta}} \tag{2.10}$$

where $\phi(z)$ is the density of the standard normal distribution and $\Phi(z) = \int_{-\infty}^{z} \phi(x)dx$ is the Gauss error function.

The extra parameter λ can now be optimized as a function of δ such that ρ_{LS} is maximum, i.e., AMP converges to the correct solution for the least level of sparsity (maximum value of ρ).

$$\lambda(\delta) = \frac{1}{\sqrt{\delta}} \arg\max_{z\geq 0} \frac{1 - (2/\delta)[(1+z^2)\Phi(-z) - z\phi(z)]}{1 + z^2 - 2[(1+z^2)\Phi(-z) - z\phi(z)]} \tag{2.11}$$

The phase transition obtained through local stability for AMP closely matches the theoretical phase transition obtained for L_1 minimization based reconstruction. Hence AMP provides the same performance with significant time and complexity reductions.

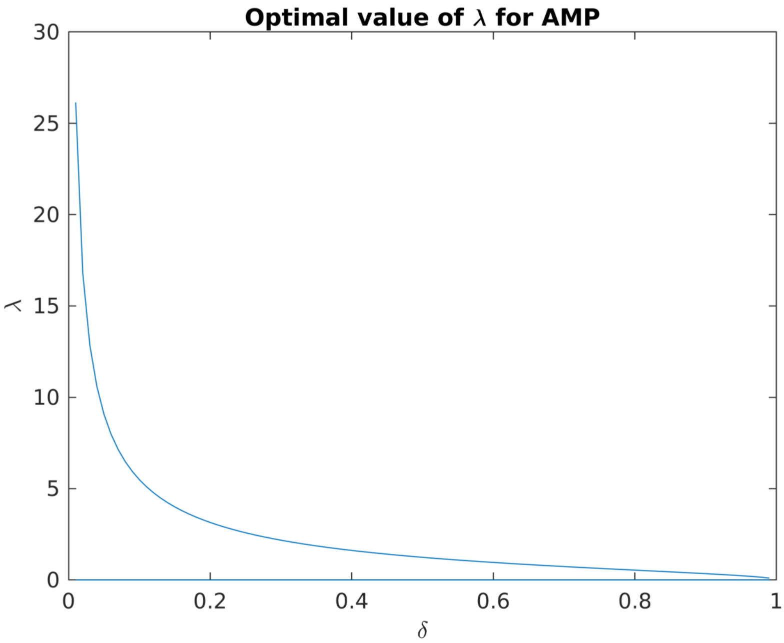


Figure 2.4: Optimal Value of λ for different δ obtained using local stability

2.5 Performance of AMP

As mentioned before, LP based methods provide very good sparsity undersampling trade-offs. However they are expensive in terms of both memory usage as well as time taken. AMP on the other hand has very low per iteration cost and requires negligible amount of extra memory while providing similar performance to that of L_1 minimization approaches.

2.5.1 Time complexity

We compare the time taken for these two algorithms for different problem sizes to see how they scale. We sample elements of A according to i.i.d. zero mean Gaussian with variance $\frac{1}{n}$. We choose $\delta = 0.2$ and $\rho = 0.2$ which is well within the asymptotic success phase of the LP problem as seen in Figure 2.2. We choose the input signal to have non-zero coefficients distributed according to the Rademacher distribution taking ± 1 with equal probability. To solve the L_1 minimization problem, we used CVX, a package for specifying and solving convex programs [20, 21]. We use CVX's default optimizer SeDuMi to obtain the reconstructed signal. Figure 2.5 shows the time required to reconstruct the original signal averaged over 25 problem instances.

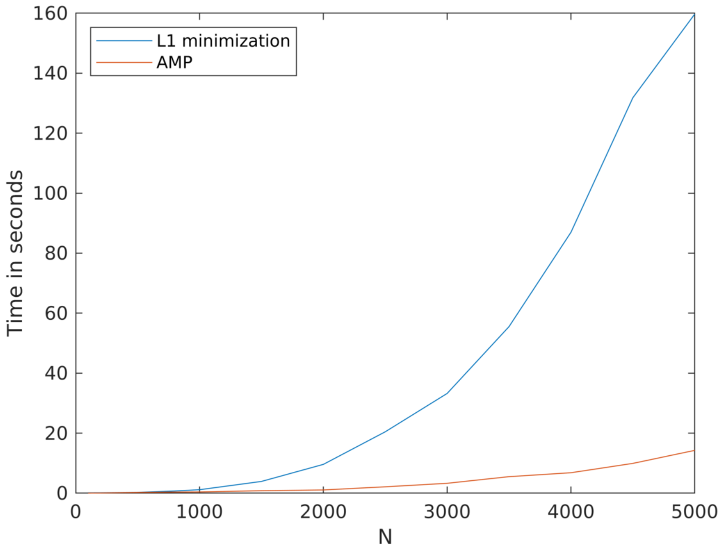


Figure 2.5: Average Time to reconstruct the signal

2.5.2 Sparsity undersampling trade-off

We obtain the experimental phase transition curves for AMP for the case of Gaussian random matrices sampled as before and compare it with that of the L_1 minimization based approach. For all the phase transition plots in our results, we choose δ values between 0.05 and 0.95 with step size of 0.02 and increment the values of ρ from 0.01 in steps of 0.01. We also choose the parameter λ for each δ as per Equation (2.11). We choose the distribution on the non-zero entries of input according to the Rademacher distribution.

For each value of δ and ρ, we create 20 instances of the problem and pass it to the AMP algorithm. We include the (δ, ρ) pair in the success phase if at least for 50% of the instances, AMP converges to the correct result. Similar plots can be obtained for higher percentage of success which we do not consider as in the large system limit, they converge to the same curve.

We notice that the gap between the theoretical phase transition for L_1 and that of AMP in Figure 2.6 is quite small. Another interesting observation is that though theoretical bounds on the performance of AMP were derived keeping the large system limit in mind, we see that even for small problems such as $N = 200$ we have the phase transition tending to that of the large system.

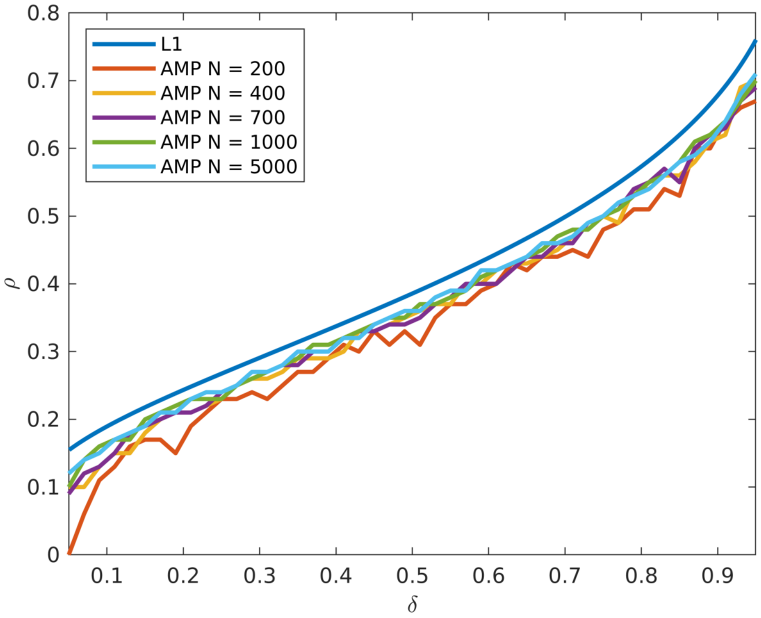


Figure 2.6: Phase Transition plots for various values of N

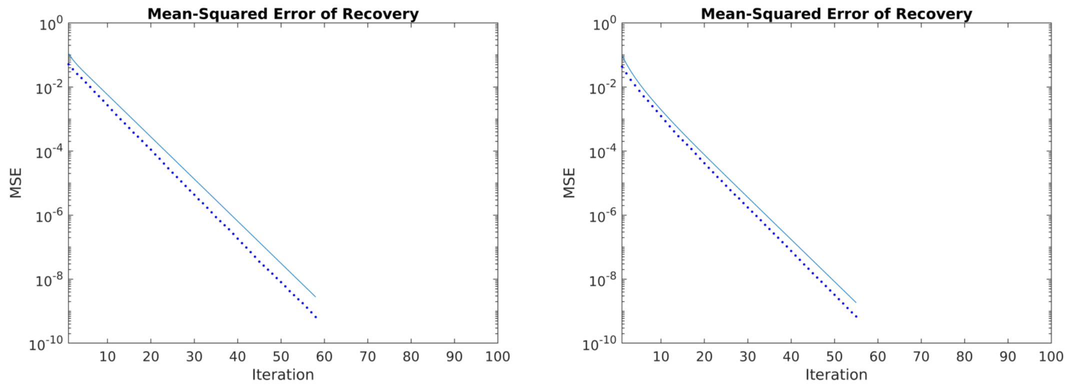
2.5.3 State Evolution vs actual MSE

We saw in Chapter 2 that AMP follows state evolution which can be used to track various quantities one of them being the MSE. We also motivated iterative thresholding methods by assuming that the difference between $A^*z^t + x^t$ (which we will refer to as γ^t from now) and x_0 is distributed according to an i.i.d. Gaussian.

We run simulations for two different input distributions namely the standard normal and the Rademacher distribution. We also consider two sets of (δ, ρ), one within the success phase $(0.4, 0.2)$ and one outside the success phase $(0.4, 0.6)$. We choose $N = 5000$ for all the cases. We plot the MSE at each time instant along with the MSE predicted by state evolution to compare.

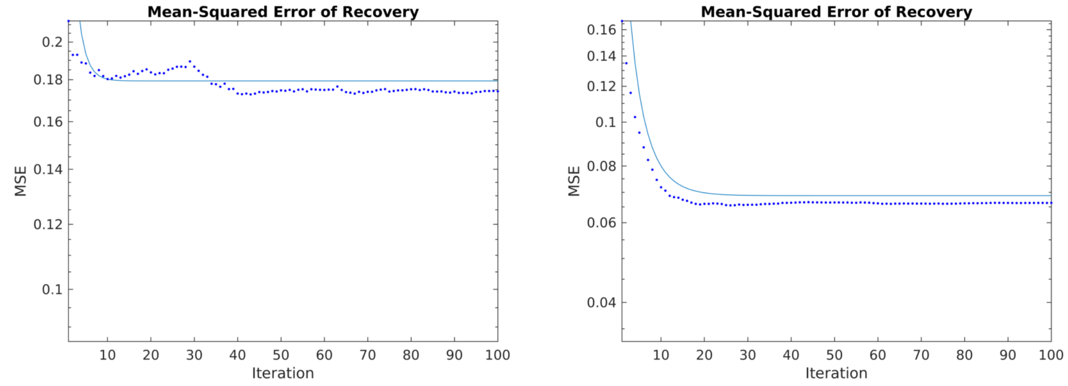
Notice that when (δ, ρ) pair are outside the success phase, AMP does not converge to the actual solution.

We plot the histogram of $\gamma^t - x_0$ at different instances of time t. Though this does not show the independence across the elements, it gives us a good idea about how the elements are distributed. We also plot the density of the zero mean Gaussian with standard deviation equal to the L_2 norm of $\gamma^t - x_0$ to compare. From here, we only plot the results for the Rademacher input distribution as the results for other distributions are similar.



(a) (b)

Figure 2.7: State evolution (line) and MSE (points) for Gaussian matrix with $\delta = 0.4$, $\rho = 0.2$ with input distribution (a) Rademacher (b) Gaussian



(a) (b)

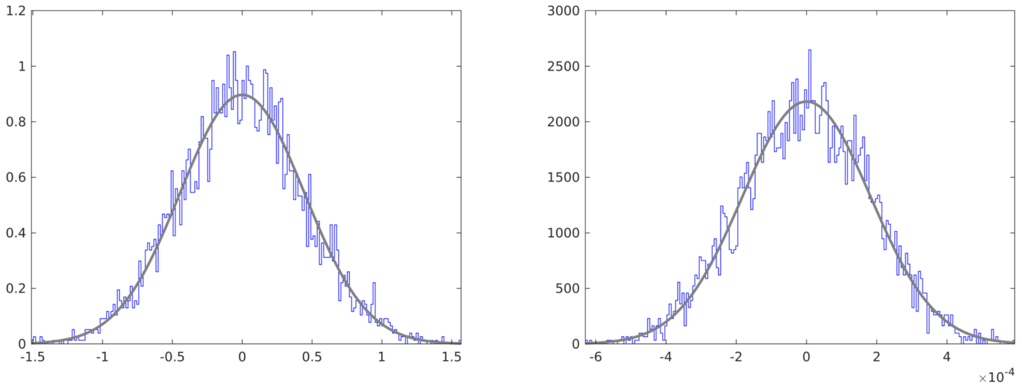
Figure 2.8: State evolution (line) and MSE (points) for Gaussian matrix with $\delta = 0.4$, $\rho = 0.6$ with input distribution (a) Rademacher (b) Gaussian

Though in the second case, AMP does not converge, the distribution of $\gamma^t - x_0$ is still Gaussian which confirms our previous assumption.

2.5.4 Alternate choice of threshold

In the current algorithm the threshold for the filter is the product of λ and current MSE. In general we cannot directly compute the MSE as we do not have access to the original signal. We instead estimate it and use the estimate to obtain the threshold.

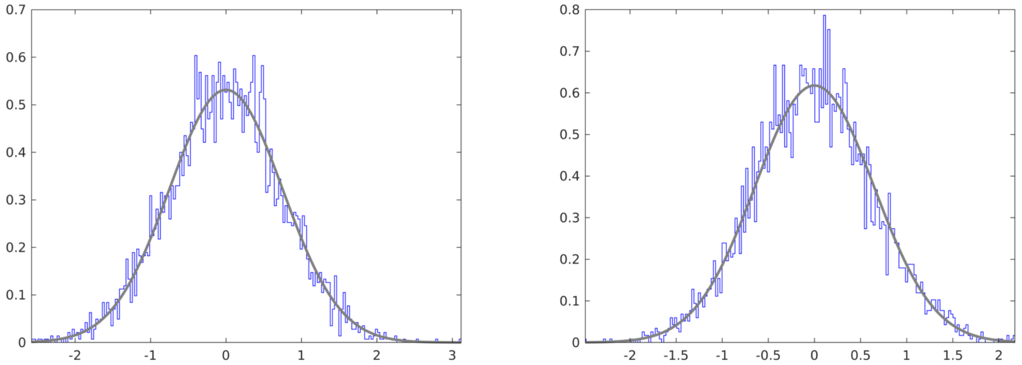
We can use a simpler alternative which works in practice while preserving the phase transition, MSE convergence and Gaussian like noise. We choose the threshold as the n^{th} largest absolute value of the vector γ^t. Using this as a threshold allows us to forgo the estimation of the MSE and makes the algorithm simpler.



(a)

(b)

Figure 2.9: Histogram of $\gamma^t - x_0$ for $\delta = 0.4$, $\rho = 0.2$ at (a) $t = 1$ and (b) $t = 50$



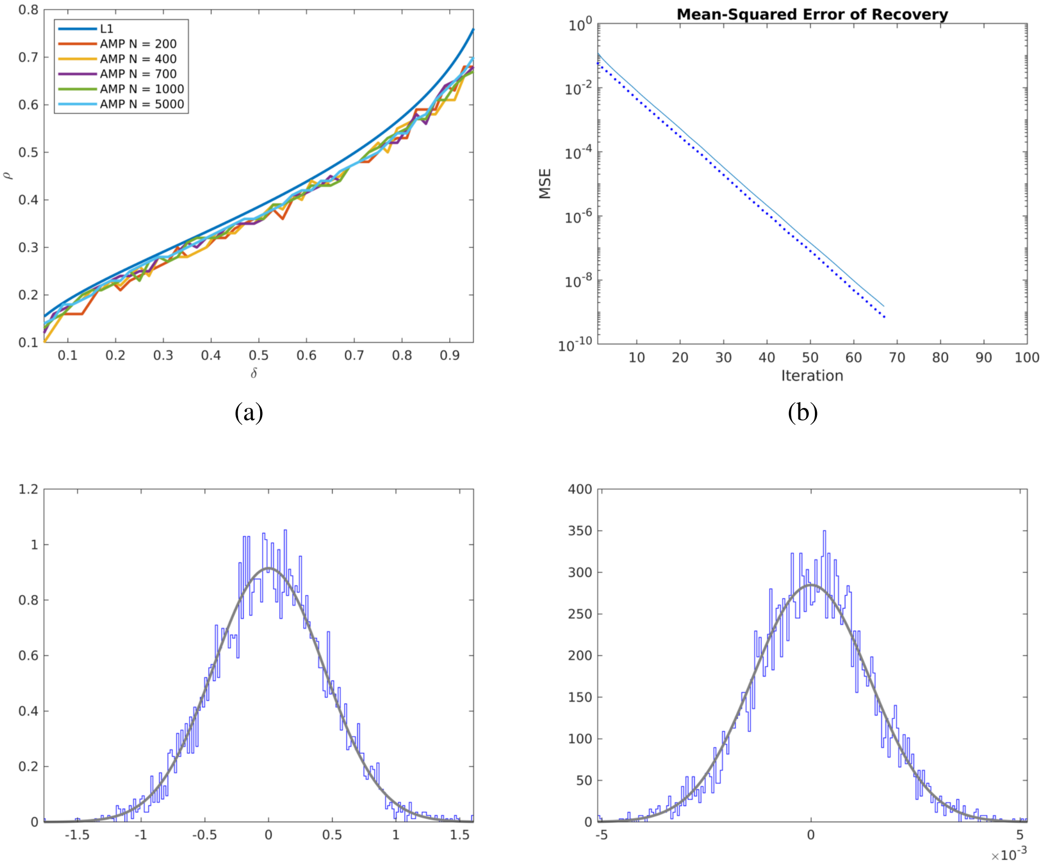
(a)

(b)

Figure 2.10: Histogram of $\gamma^t - x_0$ for $\delta = 0.4$, $\rho = 0.6$ at (a) $t = 1$ and (b) $t = 50$

For Figures 2.11b to 2.11d we chose the parameters $N = 5000$, $\delta = 0.4$ and $\rho = 0.2$

From here we use this simplified algorithm to obtain results for other random matrix ensembles.



(a)

(b)

(c)

(d)

Figure 2.11: (a) Phase Transition Plot
(b) State Evolution (line) and MSE (points)
Histogram of $\gamma^t - x_0$ at (c) $t = 1$ and (d) $t = 50$

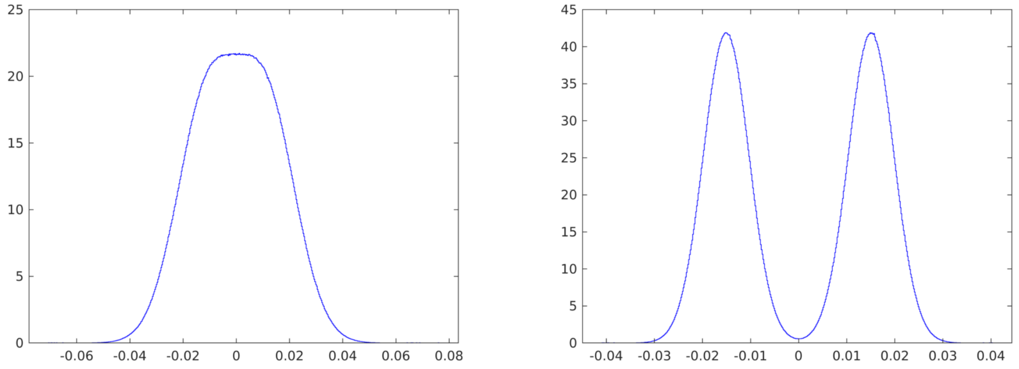
CHAPTER 3

Universality of AMP

Though the proof of AMP relies on the fact that the elements of the sensing matrices A are sampled according to i.i.d. Gaussian, in practice AMP seems to work for various ensembles of matrices such as Rademacher and partial Fourier matrices. In [22] they rigorously prove that AMP also works for matrices whose elements are drawn from a sub-Gaussian distribution with zero mean and variance $\frac{1}{n}$ with sub-Gaussian scale factor equal to $\frac{c}{n}$.

3.1 Sub-Gaussian Matrices

We obtain through simulation the phase transition curves for the Rademacher matrix and also the bimodal Gaussian for $N = 5000$. The bimodal Gaussian matrix is generated as the sum of a Rademacher matrix and a Gaussian matrix and is then normalized appropriately to ensure that the variance of the distribution is $\frac{1}{n}$. The ratio of the variance of the Gaussian distribution to that of the Rademacher distribution determines the separation between the two peaks. Smaller this ratio, further apart are the peaks of the Gaussian and for very large values there is no separation of peaks. We plot below the results for two cases, one for small separation, i.e., almost same as Gaussian and one for large separation.



(a)

(b)

Figure 3.1: Probability density of Bimodal Gaussians with
(a) small separation (Bimodal 1) and (b) large separation (Bimodal 2)

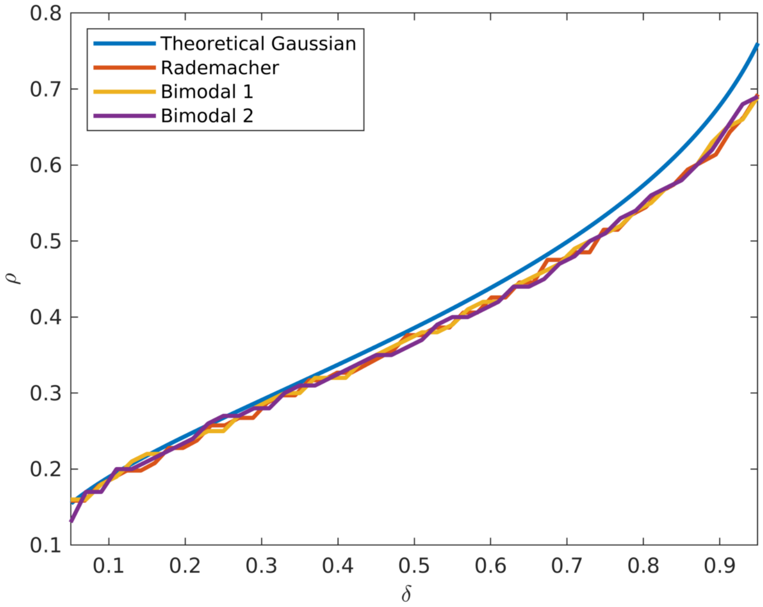


Figure 3.2: Phase Transition plots for Rademacher and Bimodal Gaussian

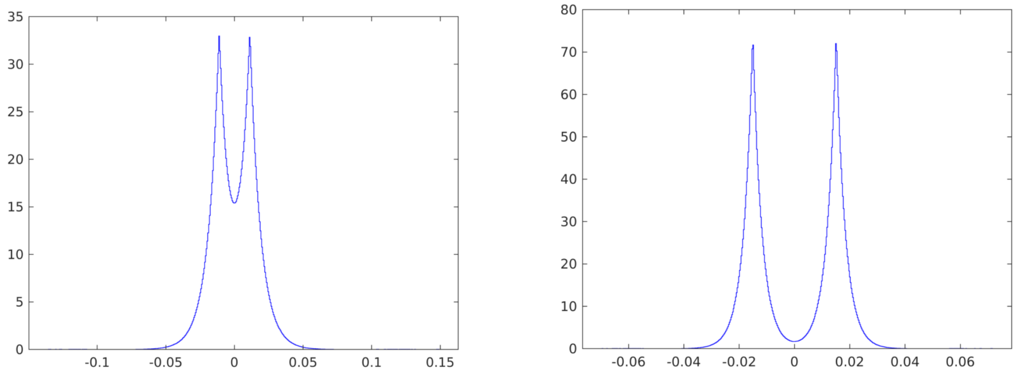
3.2 Sub-Exponential Matrices

We extend the convergence result from sub-Gaussian to other random matrices with heavier tails such as the Laplace distribution. The Laplace distribution given by the probability density $\mu(x) = \frac{\lambda}{2}e^{-\lambda|x|}$ is not sub-Gaussian as it has a heavier tail as compared to that of a Gaussian. Hence this distribution does not fulfil the sub-Gaussianity criteria stated in Section 3.1. However, when the elements of A are distributed according to the Laplace distribution, we can experimentally verify that AMP continues to work. We also consider the bimodal Laplacian defined similar to the bimodal Gaussian with the replacement of the Gaussian by the Laplacian. Even in this case, AMP recovers the original signal with similar sparsity undersampling trade-off. By this we can guess that the convergence is majorly a tail property and propose that AMP works for other sub-exponential matrices too.

To obtain the phase transition plots, we use $N = 5000$ and perform the simulation as described in Section 2.5.2. For the MSE and state evolution plot as well as for the histogram of $\gamma^t - x_0$ we use $N = 5000$, $\delta = 0.4$ and $\rho = 0.2$.

Notice that for all three choices of matrices, the phase transition curves match very closely with that of the Gaussian. We also note that the state evolution predicts the MSE with good accuracy for both the case when AMP converges to the original signal and when it does not. An important observation is that for the choice $\delta = 0.4$ and $\rho = 0.6$, the MSE does not converge to 0 but still settles away from 0 and does not diverge as the iterations increase. For both choices of ρ, the histogram of $\gamma^t - x_0$ is approximately Gaussian. Though for the larger value of ρ, the norm has not gone to 0, it still has the Gaussian shape.

Considering all these observations we can conclude that for sub-exponential matrices, the behaviour of AMP is similar to that of AMP on Gaussian matrices.



(a)

(b)

Figure 3.3: Probability density of Bimodal Laplace with
(a) small separation (Bimodal 1) and (b) large separation (Bimodal 2)

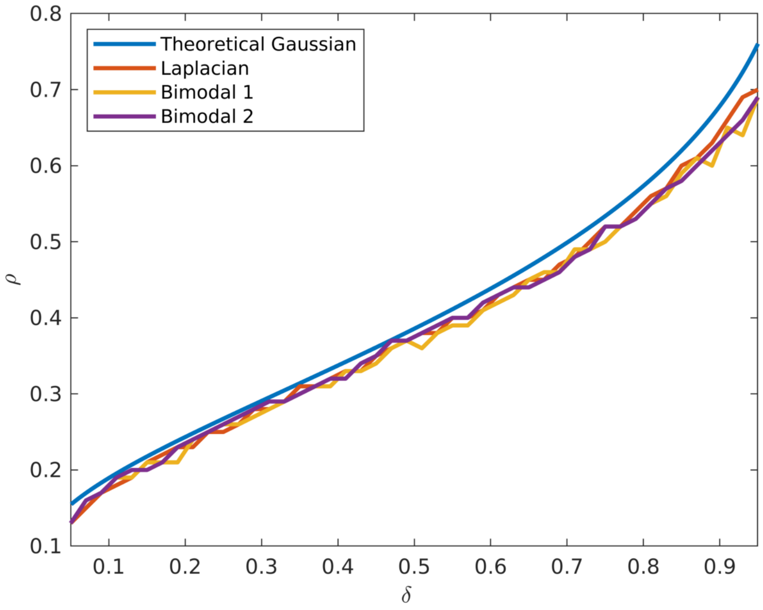
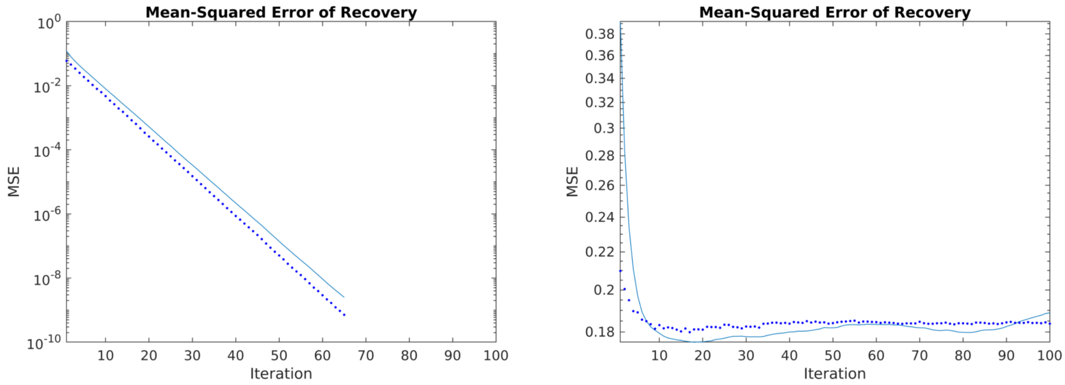
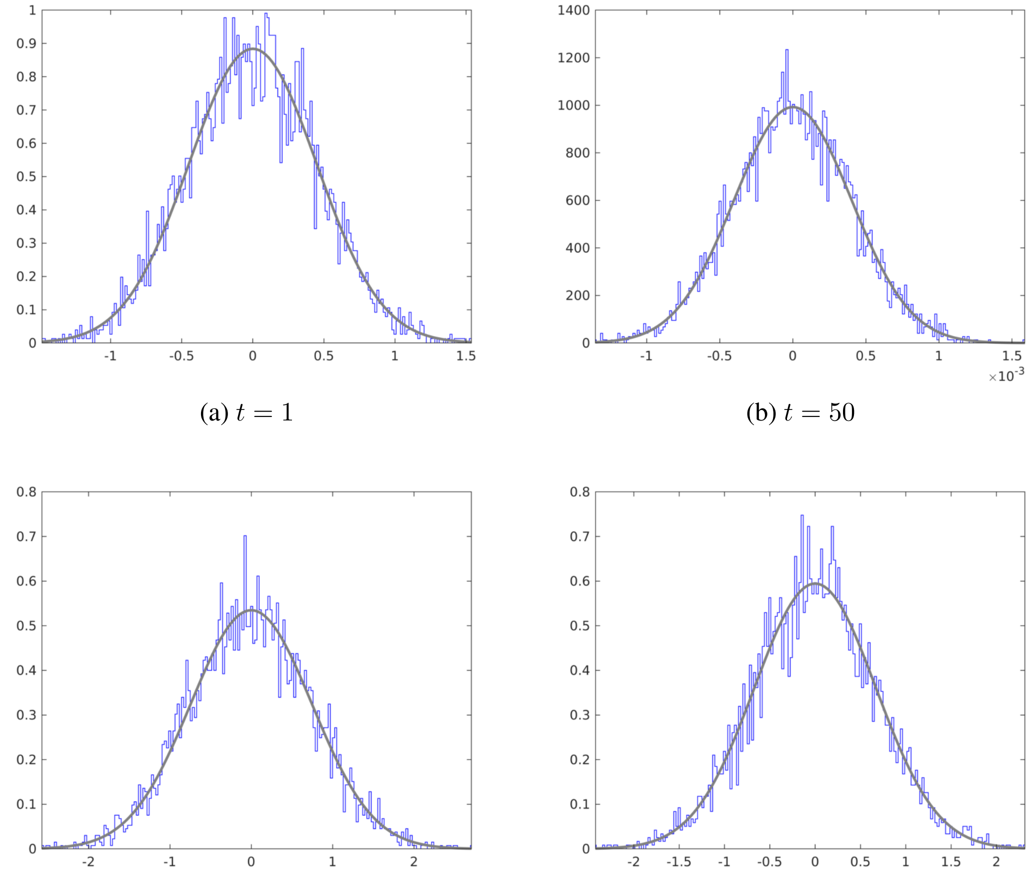


Figure 3.4: Phase Transition plots for Laplacian and Bimodal Laplacian



(a) (b)

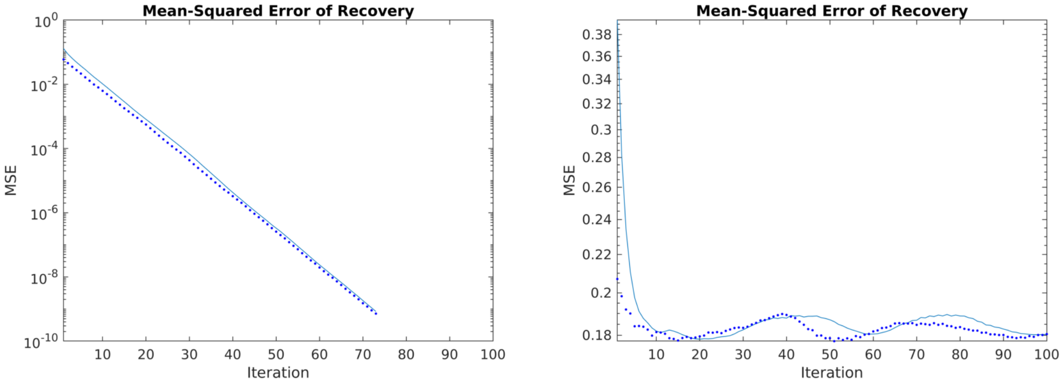
Figure 3.5: State evolution (line) and MSE (points) for Laplacian with (a) $\delta = 0.4$, $\rho = 0.2$ and (a) $\delta = 0.4$, $\rho = 0.6$



(a) $t = 1$ (b) $t = 50$

(c) $t = 1$ (d) $t = 50$

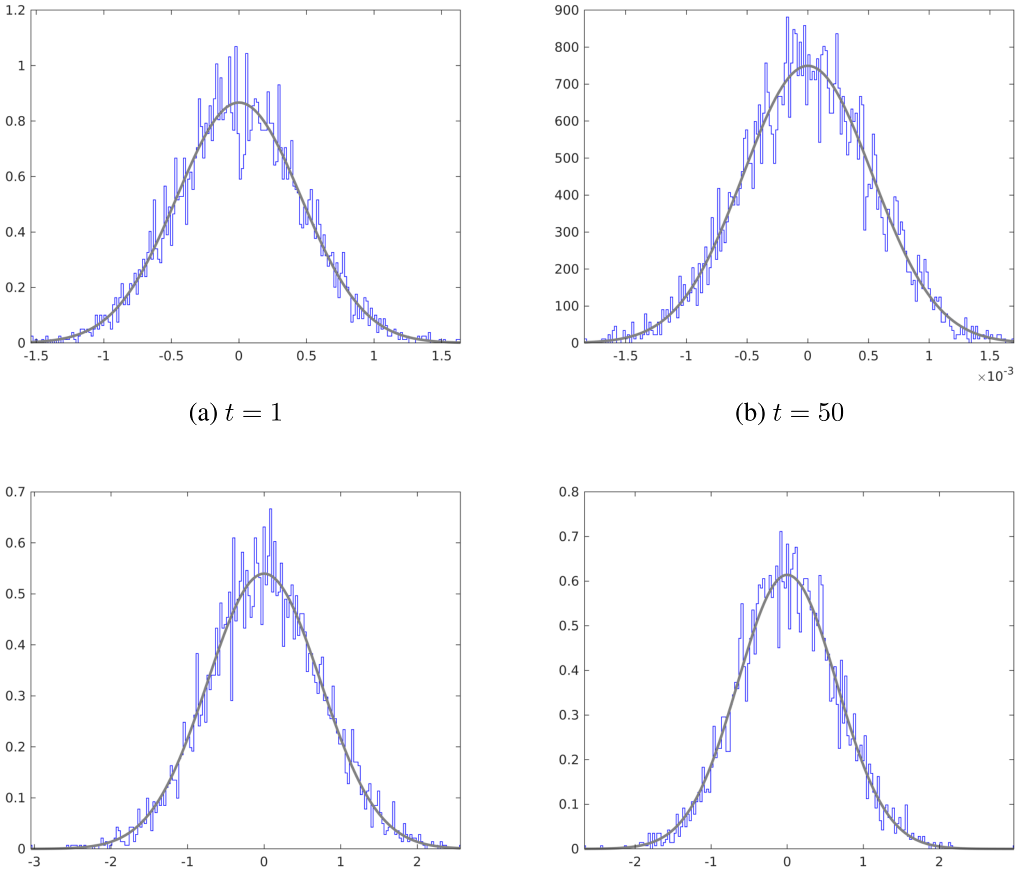
Figure 3.6: Histogram of $\gamma^t - x_0$ for Laplacian with (a)-(b) $\delta = 0.4$, $\rho = 0.2$ and (c)-(d) $\delta = 0.4$, $\rho = 0.6$



(a)

(b)

Figure 3.7: State evolution (line) and MSE (points) for Bimodal 1 with (a) $\delta = 0.4$, $\rho = 0.2$ and (a) $\delta = 0.4$, $\rho = 0.6$



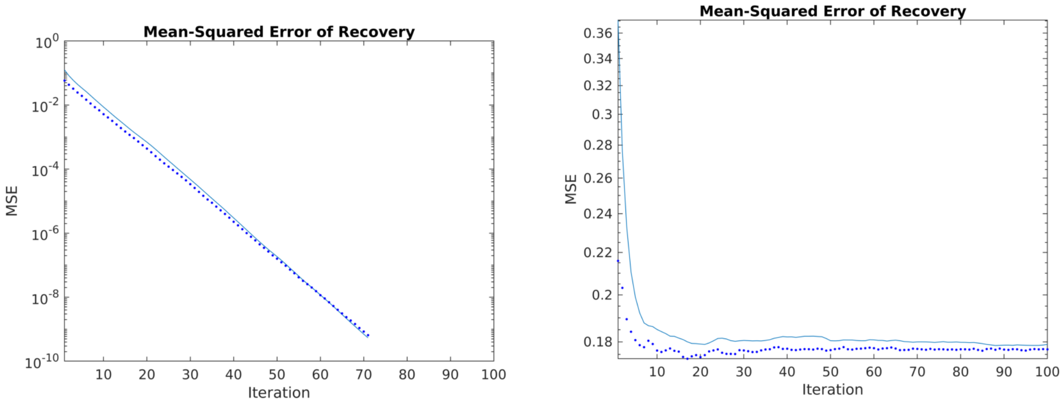
(a) $t = 1$

(b) $t = 50$

(c) $t = 1$

(d) $t = 50$

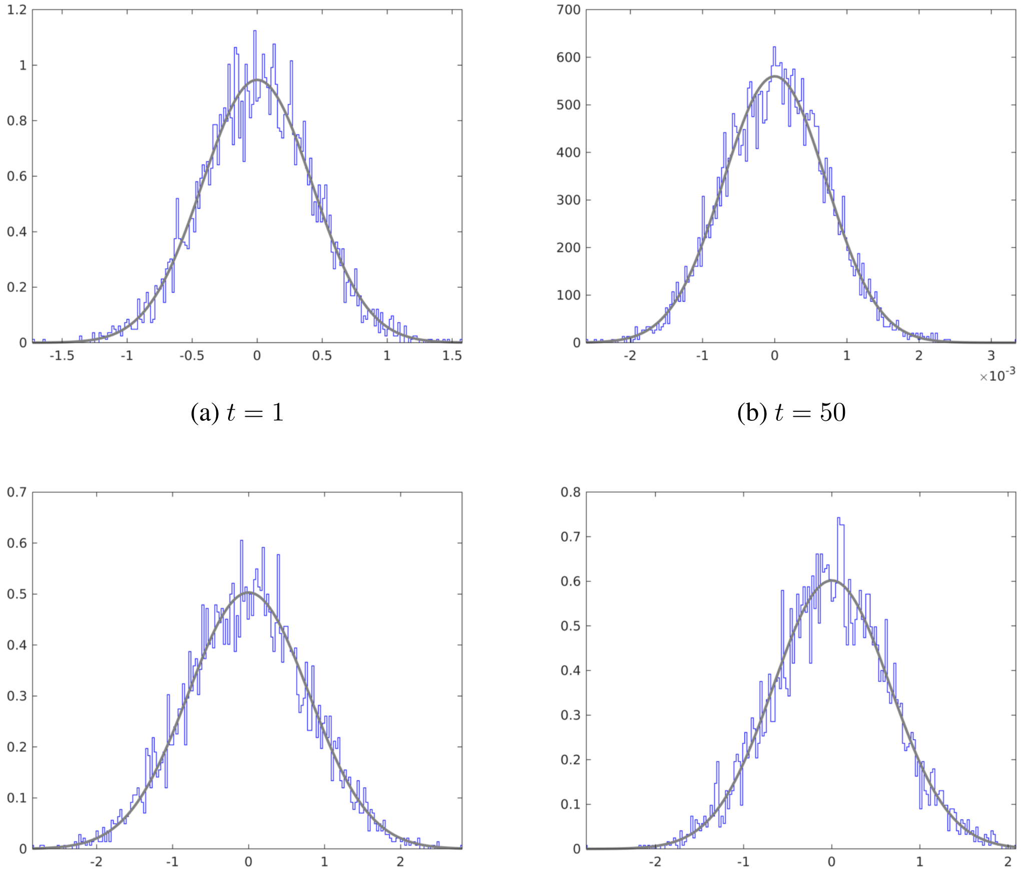
Figure 3.8: Histogram of $\gamma^t - x_0$ for Bimodal 1 with (a)-(b) $\delta = 0.4$, $\rho = 0.2$ and (c)-(d) $\delta = 0.4$, $\rho = 0.6$



(a)

(b)

Figure 3.9: State evolution (line) and MSE (points) for Bimodal 2 with (a) $\delta = 0.4$, $\rho = 0.2$ and (a) $\delta = 0.4$, $\rho = 0.6$



(a) $t = 1$

(b) $t = 50$

(c) $t = 1$

(d) $t = 50$

Figure 3.10: Histogram of $\gamma^t - x_0$ for Bimodal 2 with (a)-(b) $\delta = 0.4$, $\rho = 0.2$ and (c)-(d) $\delta = 0.4$, $\rho = 0.6$

CHAPTER 4

Extensions to AMP

Although AMP performs quite close to the LP based methods for a large set of matrices such as sub-Gaussian, sub-Exponential etc, it still excludes a large set of possible matrices. Since the introduction on AMP in [18], there have been many extensions proposed to AMP to improve the convergence and also include a more general setting.

4.1 Existing extensions of AMP

4.1.1 Generalized AMP

Generalized AMP [23] allows us to work with a more generalized model. An input vector $q \in Q^N$ has elements from the set Q and this generates an unknown signal $x \in \mathbb{R}^N$ though an element wise channel having the conditional distribution as $p_{X|Q}(x_j|q_j)$. We obtain $z \in \mathbb{R}^n$ from x through a linear transform $A \in \mathbb{R}^{n \times N}$, i.e., $z = Ax$. Now the elements of z are passed through a second channel with conditional distribution $p_{Y|Z}(y_i)$. The problem now is to estimate x given the input vector q and output vector y.

This can be easily reduced to the sparse estimation problem by choosing

- Q to have a single element, i.e., $q_j = q_{j'} \quad \forall j, j'$
- $p_{X|Q}(x_j|q_j)$ as a distribution with $X = 0$ occuring with a probability of $(1 - \rho)\delta$
- $p_{Y|Z}(y_j|z_j) \propto 1[y_j = z_j]$

GAMP provides many advantages as opposed to traditional AMP.

- Due to the choice of input vector q, elements of x are no longer required to be identically distributed but still have to be independently distributed.
- We no longer require sparsity. We can tolerate arbitrary distributions on x
- We can also model non-linearity and noise at the output through appropriate choice of $p_{Y|Z}$

4.1.2 Vector AMP

Another extension to traditional AMP is Vector AMP [24]. VAMP has a rigorous state evolution that holds for a much broader class of random matrices namely right orthogonal matrices. The key difference in the algorithm involves performing an SVD on the matrix A to split the matrix into U and SV^* and introduce an intermediate variable $w = SV^*x$. We now iteratively estimate both variables to reconstruct the original signal x. The per iteration cost is similar to that of AMP. However for very large problems performing SVD becomes computationally intensive.

4.1.3 Damped AMP

In this version of AMP, the updates to the estimate are damped by a factor α to improve convergence. A weighted average of the previous estimate x^t and the output of the soft threshold gives us the new estimate x^{t+1}. The update rules now are given by

$$\begin{aligned} x^{t+1} &= \alpha x^t + (1-\alpha)\eta_t\left(A^* z^t + x^t\right) \\ z^t &= y - Ax^t + \frac{1}{\delta} z^{t-1} \langle \eta_t'(A^* z^t + z^t) \rangle \end{aligned}$$

where α controls the damping and is typically chosen between 0 and 0.05. [25]

4.2 Scaled AMP

We introduce a simple modification to the original AMP algorithm to improve the performance of AMP on Non-Gaussian matrices. The change is similar but not identical to that of Damped AMP.

Noting that the update rules are not linear in A, we scale the matrix by α to obtain $A' = \alpha A$ and get $y' = \alpha A = A'x$. The estimated signal is not scaled and remains the same. We now use these new matrix A' and new output y' to estimate the input x. We experimentally verify that by appropriate choice of α, we can increase the success phase of heavy tailed distributions from that of AMP.

4.2.1 Laplacian Matrix

In Section 3.2, we saw that Laplacian matrices also follow state evolution of AMP and result in the same phase transition curve. However for small values of N like 1000, the phase transition for small δ occurs at a much smaller value of ρ than for that of Gaussian. By running scaled AMP even with a small change of $\alpha = 0.9$ from 1, we obtain phase transition similar to that of Gaussian.

Looking at Figure 4.1, we notice that for α close to 1, performance is similar to that of AMP. We see an improvement in the performance for small ρ as we decrease the value of α. However when α becomes very small such as 0.5, we see a dip in performance for values of δ close to 1.

From Figure 4.3, we notice that state evolution as defined for AMP overestimates the MSE and smaller the value of α, larger is the over-estimate. We also notice that the elements of $\gamma^t - x_0$ are no longer distributed according to a Gaussian. Hence arguments used in justifying AMP do not completely hold for the scaled version. Again in the failure region the MSE is shifted away from 0 but does not diverge.

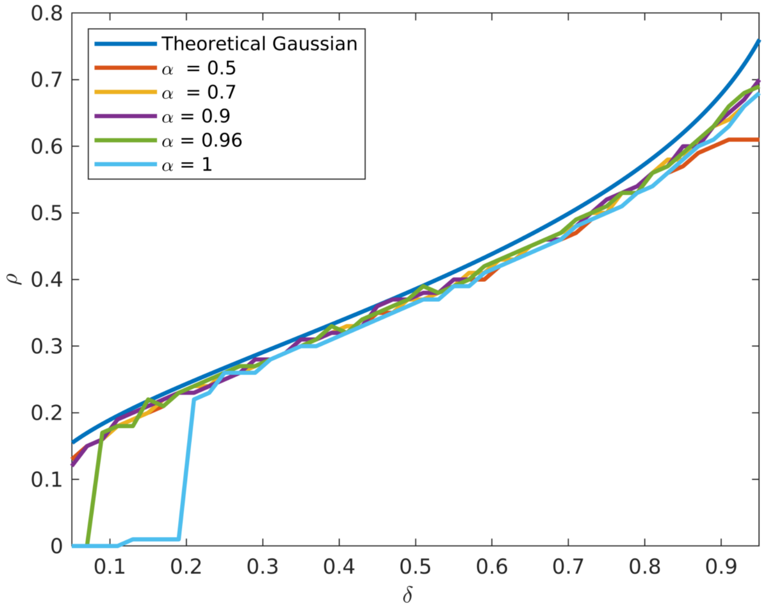
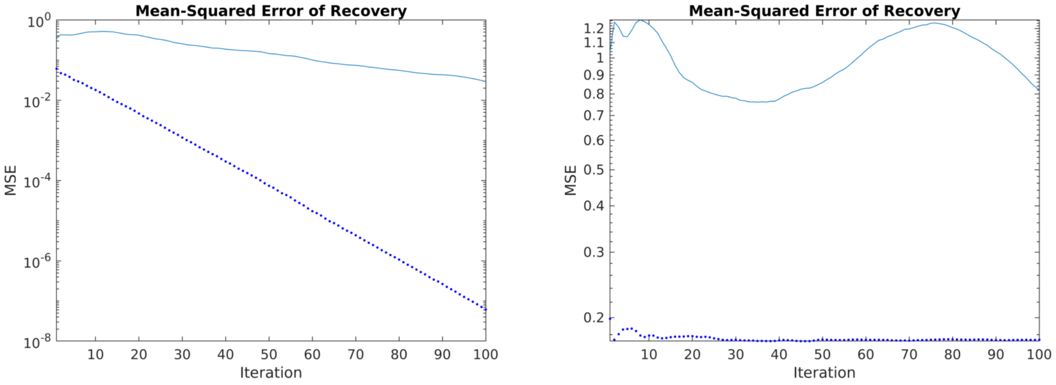
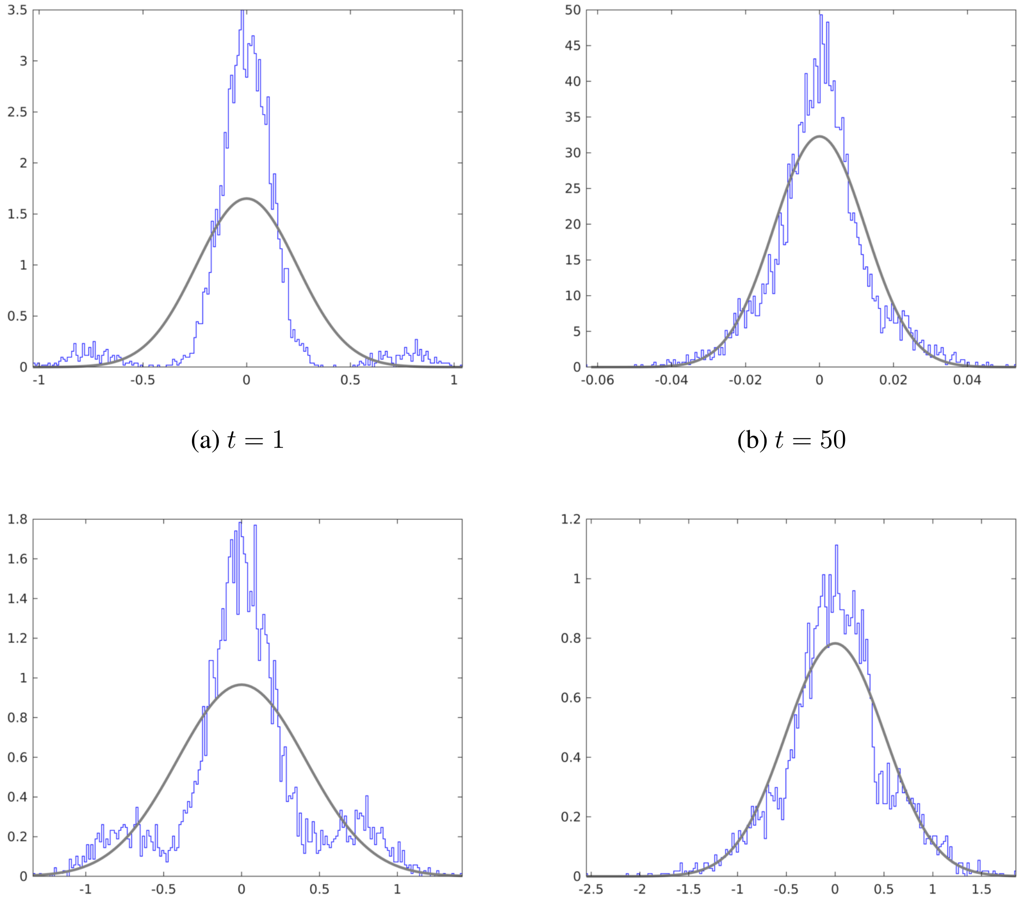


Figure 4.1: Phase Transition plots for Laplacian with scaling factor α



(a) (b)

Figure 4.2: State evolution (line) and MSE (points) for Laplacian ($\alpha = 0.5$) with (a) $\delta = 0.4$, $\rho = 0.2$ and (a) $\delta = 0.4$, $\rho = 0.6$



(a) $t = 1$

(b) $t = 50$

(c) $t = 1$

(d) $t = 50$

Figure 4.3: Histogram of $\gamma^t - x_0$ for Laplacian ($\alpha = 0.5$) with (a)-(b) $\delta = 0.4$, $\rho = 0.2$ and (c)-(d) $\delta = 0.4$, $\rho = 0.6$

4.2.2 Heavy Tailed Distributions

We will consider the set of distributions parametrized by d having the following probability density function

$$p(x) = \begin{cases} \frac{d-1}{2|x|^d} & \text{if } |x| \geq 1 \\ 0 & \text{if } |x| < 1 \end{cases}$$

We sample elements of the matrix A according to $p(x)$ and scale them appropriately so as to have variance equal to $\frac{1}{n}$. Clearly, $d > 3$ is required for the existence of variance. Smaller the value of d, heavier the tail. So, we are justified to assume that as d increases AMP performs better.

From Figure 4.4, we see that AMP does not recover the signal at all for any value of δ. So we scale the λ value by different values to try and obtain a better curve. Here we plot for λ which is 10 times the optimal value for Gaussian. We also run scaled AMP to obtain the phase transition. Similar to the case of Laplacian, as α decreases, initially the performance improves and then reaches a maximum and again starts to deteriorate for higher values of δ. For $\alpha = 0.5$, the phase transition comes very close to that of the Gaussian case.

When using Scaled AMP for this distribution, similar to the case of Laplacian, the histogram of $\gamma^t - x_0$ is not distributed according to a Gaussian. The algorithm still converges to the optimal solution for $\delta = 0.4$ and $\rho = 0.2$. For the case $\delta = 0.4$ and $\rho = 0.6$, the algorithm does not converge. Yet the MSE does not diverge.

The same is not true for the case when we use AMP with $\lambda = 10\lambda_{opt}$. As the algorithm progresses, the estimate diverges rapidly from the optimal value. Looking at the histogram of $\gamma^t - x_0$ sheds some light on this issue. We notice that due to the heavy tailed nature of $p(x)$, few elements of $\gamma^t - x_0$ have large absolute values (notice the shift in the plot). When performing soft threshold, these values are reduced only by a small value. Hence in the next iteration they contribute to large errors in the estimate and the cycle repeats.

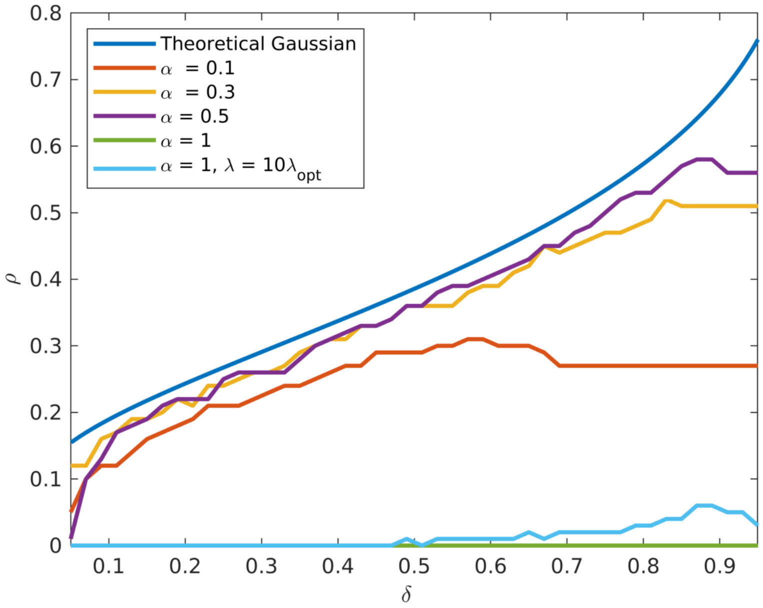
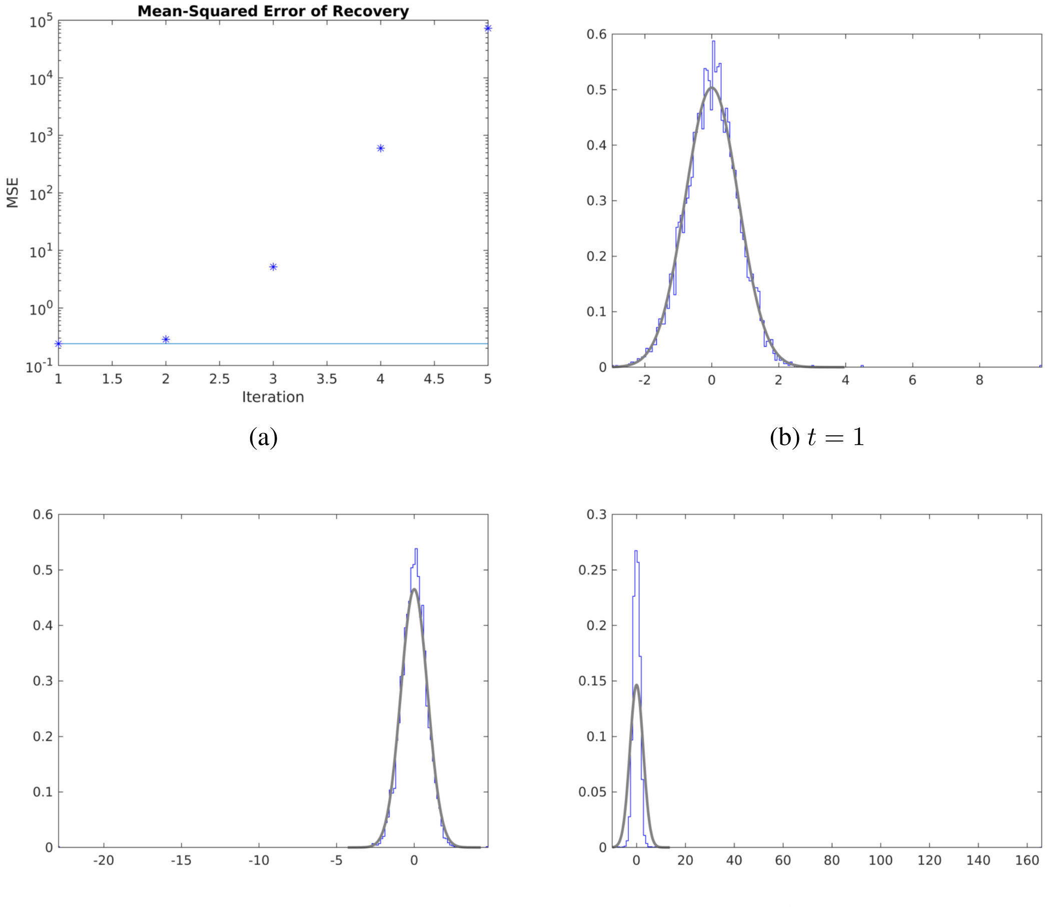


Figure 4.4: Phase Transition plots for d = 4 with scaling factor α



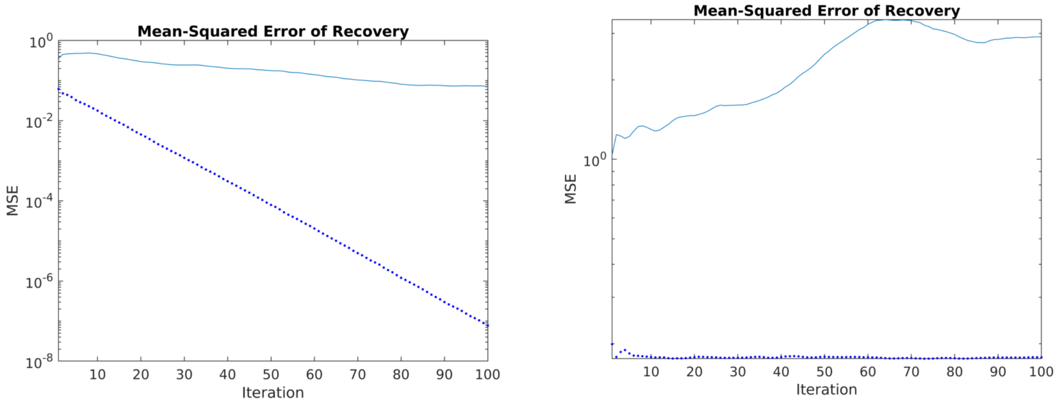
(a)

(b) $t = 1$

(c) $t = 2$

(d) $t = 3$

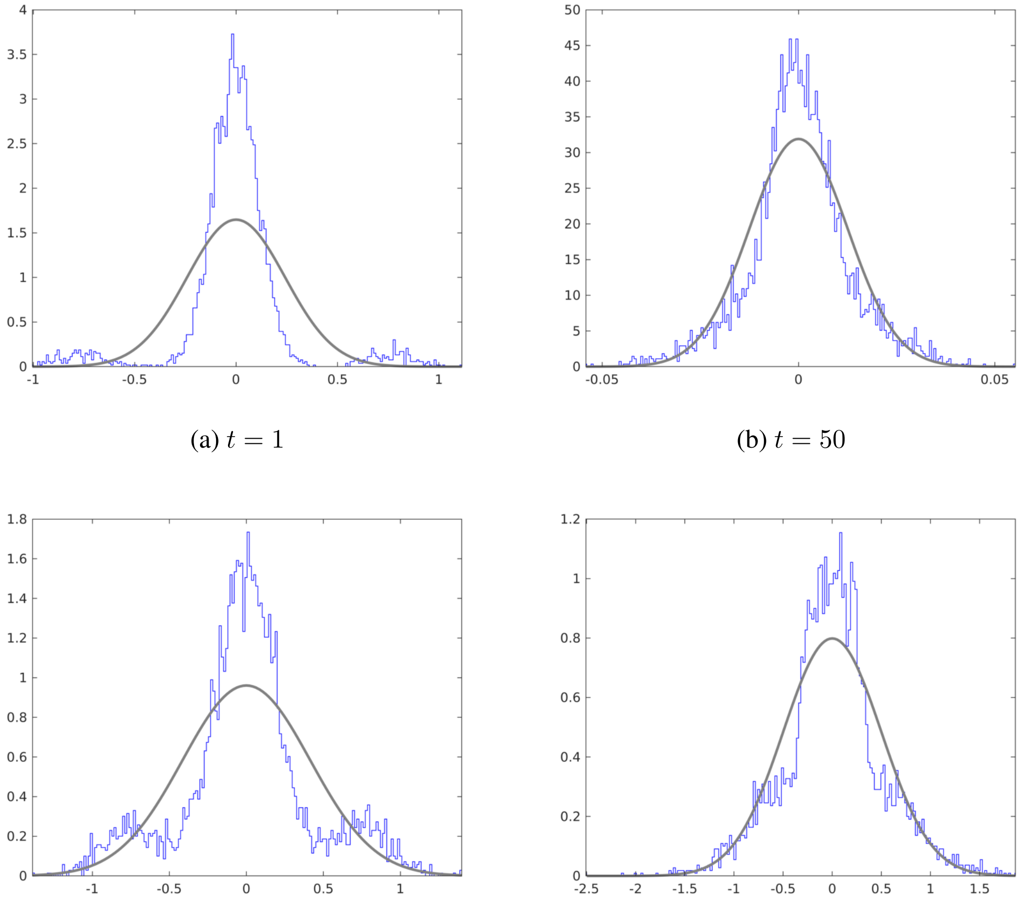
Figure 4.5: (a)State evolution (line) and MSE (points) (b)- (d)Histogram of $\gamma^t - x_0$ for d = 4, $\alpha = 1$, $\lambda = 10\lambda_{opt}$, $\delta = 0.4$ and $\rho = 0.6$



(a)

(b)

Figure 4.6: State evolution (line) and MSE (points) for d = 4 and $\alpha = 0.5$ with (a) $\delta = 0.4$, $\rho = 0.2$ and (a) $\delta = 0.4$, $\rho = 0.6$



(a) $t = 1$

(b) $t = 50$

(c) $t = 1$

(d) $t = 50$

Figure 4.7: Histogram of $\gamma^t - x_0$ for d = 4 and $\alpha = 0.5$ (a)-(b) $\delta = 0.4$, $\rho = 0.2$ and (c)-(d) $\delta = 0.4$, $\rho = 0.6$

CHAPTER 5

Conclusions and Future Work

This report primarily deals with the performance of Approximate Message Passing and a simple modification of AMP. A summary of the work done and some future directions are given below.

5.1 Summary

In Chapter 4 we verify that AMP has similar undersampling sparsity trade-off for random matrices such as Laplacian and Bimodal Laplacian, is similar to that of AMP for Gaussian i.i.d. matrices. We also verified through simulations that state evolution continues to hold and gives a good estimate of the MSE across iterations and that the noise, i.e., $\gamma^t - x_0$ is distributed like a Gaussian for both cases when AMP converges to the accurate solution and when it does not.

In Chapter 5 we present a simple modification to AMP which is the scaled AMP. For the case of Laplacian random matrix, and small system size, we noticed that the phase transition of AMP for small undersampling fractions is well below that of Gaussian. By choosing an appropriate value of α, scaled AMP closes the gap between the Gaussian and Laplacian cases for small systems. We considered heavier tailed distributions on the random matrix A and verified that AMP has poor performance in this setting. We also noted that in the failure region, the estimate diverges rapidly due to some very large values on some indices. By choosing an appropriate scaling factor α for scaled AMP, we see that the performance improves by a great extent.

5.2 Future Directions

We verified that the performance of AMP for sub-Exponential matrices is similar to that for Gaussian matrices in the sparse estimation setting. One future direction would be to theoretical justify this result by using similar techniques used to prove the performance guarantees for sub-Gaussian matrices.

Another interesting direction would be to obtain an equivalent state evolution rule for scaled AMP as the state evolution result used in AMP overestimates the MSE when used with scaled AMP. The optimal scaling parameter α for a particular distribution was found through simulations and was kept constant for all the undersampling ratios. Obtaining an analytical optimal value as a function of the matrix A and parameters δ, ρ would be interesting as this would allow us to achieve better performance when using scaled AMP.

We also notice that scaled AMP is more resilient than AMP to nonzero mean measurement matrices. A more detailed study and simulations to confirm this claim are required to be carried out.

REFERENCES

[1] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, April 2006.

[2] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] M. Lustig, D. Donoho, and J. M. Pauly, “Sparse mri: The application of compressed sensing for rapid mr imaging,” *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[4] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, “Compressed sensing mri,” *IEEE Signal Processing Magazine*, vol. 25, pp. 72–82, March 2008.

[5] M. H. Firooz and S. Roy, “Network tomography via compressed sensing,” in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–5, Dec 2010.

[6] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma, “Applications of sparse representation and compressive sensing [scanning the issue],” *Proceedings of the IEEE*, vol. 98, pp. 906–909, June 2010.

[7] M. Elad, M. A. T. Figueiredo, and Y. Ma, “On the role of sparse and redundant representations in image processing,” *Proceedings of the IEEE*, vol. 98, pp. 972–982, June 2010.

[8] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse representations in audio and music: From coding to source separation,” *Proceedings of the IEEE*, vol. 98, pp. 995–1005, June 2010.

[9] V. Papyan, Y. Romano, and M. Elad, “Convolutional neural networks analyzed via convolutional sparse coding,” *J. Mach. Learn. Res.*, vol. 18, pp. 2887–2938, Jan. 2017.

[10] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, pp. 83–91, March 2008.

[11] E. J. CandÃĺs, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589 – 592, 2008.

[12] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[13] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal îŢŃ1-norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[14] D. L. Donoho and J. Tanner, “Precise undersampling theorems,” *Proceedings of the IEEE*, vol. 98, pp. 913–924, June 2010.

[15] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: I. motivation and construction,” in *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pp. 1–5, Jan 2010.

[16] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: Ii. analysis and validation,” in *2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo)*, pp. 1–5, Jan 2010.

[17] A. Maleki and D. L. Donoho, “Optimally tuned iterative reconstruction algorithms for compressed sensing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 330–341, April 2010.

[18] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[19] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inf. Theor.*, vol. 57, pp. 764–785, Feb. 2011.

[20] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1.” `http://cvxr.com/cvx`, Mar. 2014.

[21] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[22] M. Bayati, M. Lelarge, and A. Montanari, “Universality in polytope phase transitions and message passing algorithms,” *Ann. Appl. Probab.*, vol. 25, pp. 753–822, 04 2015.

[23] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *2011 IEEE International Symposium on Information Theory Proceedings*, pp. 2168–2172, July 2011.

[24] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 1588–1592, June 2017.

[25] S. Rangan, P. Schniter, and A. Fletcher, “On the convergence of approximate message passing with arbitrary matrices,” in *2014 IEEE International Symposium on Information Theory*, pp. 236–240, June 2014.