Using Prior Support Information in Compressed Sensing

by

Navid Ghadermarzy

B.Sc., Sharif University of Technology, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2013

© Navid Ghadermarzy 2013

Abstract

Compressed sensing is a data acquisition technique that entails recovering estimates of sparse and compressible signals from n linear measurements, significantly fewer than the signal ambient dimension N. In this thesis we show how we can reduce the required number of measurements even further if we incorporate prior information about the signal into the reconstruction algorithm. Specifically, we study certain weighted nonconvex ℓ_p minimization algorithms and a weighted approximate message passing algorithm.

In Chapter 1 we describe compressed sensing as a practicable signal acquisition method in application and introduce the generic sparse approximation problem. Then we review some of the algorithms used in compressed sensing literature and briefly introduce the method we used to incorporate prior support information into these problems.

In Chapter 2 we derive sufficient conditions for stable and robust recovery using weighted ℓ_p minimization and show that these conditions are better than those for recovery by regular ℓ_p and weighted ℓ_1. We present extensive numerical experiments, both on synthetic examples and on audio, and seismic signals.

In Chapter 3 we derive weighted AMP algorithm which iteratively solves the weighted ℓ_1 minimization. We also introduce a reweighting scheme for weighted AMP algorithms which enhances the recovery performance of weighted AMP. We also apply these algorithms on synthetic experiments and on real audio signals.

Preface

Chapter 1 is an introduction and motivation for the problems studied in this thesis. Sections 1.1 and 1.2 reviews well-known results in the field, no originality claimed. Figures 1.4 and 1.7 has been adopted from [2] and [4], respectively, by permission.

Chapter 2 is a joint work with Hassan Mansour and Özgür Yılmaz. I conducted all the experiments. A version of this chapter will be published by the title "Recovering Compressively Sampled Signals by non-convex optimization and using partial support information". This chapter is based on works conducted by M. P. Friedlander, H. Mansour, R. Saab, and O. Yılmaz [1] and R. Saab and O.Yılmaz [2].

Chapter 3 incorporates prior support information into the algorithm introduced by D. L. Donoho, A. Maleki, and A. Montanari [3] and is based on their work in [34]. all the lemmas and theorems in [34] has been modified by the author except for Lemma 3.1 which has been restated. A version of this chapter will be published by the title "Weighted and reweighted approximate message passing".

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank my supervisor, Özgür Yılmaz, for his support and guidance throughout these last two years. His patience with me and his insightful suggestions are much appreciated. I would also like to thank Hassan Mansour for all his help during my graduate studies and all the meetings where he patiently shared his ideas with me. I would also want to thank Felix Herrmann and all the other members of SLIM group for their support. Last but not least I would want to thank my family who I owe where I am right now to them.

Chapter 1

Introduction and overview

Compressed sensing is a data acquisition technique for efficiently acquiring and recovering sparse or approximately sparse signals from seemingly incomplete and noisy linear measurements. We say that a signal is sparse when the signal admits a representation in some transform domain where most of its coefficients are zero. There are many applications where the target signals admit a nearly sparse representation in some transform domain. In other words, most of the underlying information of the signal can be obtained by relatively fewer coefficients. For example, natural images are nearly sparse in discrete cosine transform domain (DCT) and in the wavelet domain which is crucial in applications like JPEG and JPEG2000. Similarly audio signals are approximately sparse in short time DCT domain and short time Fourier domain which allows MP3 compression.

Exploiting this observation and using appropriate reconstruction schemes, compressive sensing enables signal acquisition with fewer measurements than traditional sampling. Compressed sensing is especially promising in applications where taking measurements is costly (e.g., hyperspectral imaging [6]) as well as in applications where the ambient dimension of the underlying signal is very large, i.e., medical [7] and seismic imaging [8, 9] (seismic images are approximately sparse in the curvelet domain).

Compressed sensing utilizes the sparsity information of the signal to reduce the necessary number of measurements in order to capture all (or most) the information of the signal. On the other hand, several studies, e.g., [10–12] show that the necessary number of measurements can be reduced even more if we use prior information about the signal to be recovered. Prior information about the signal can be obtained by considering the properties of the signal which we want to recover. For example, we know that speech signals have large low-frequency coefficients. Other than this, signals such as video and audio exhibit correlation over temporal frames that can be exploited to estimate a portion of the support using previously decoded frames. As an other example correlations between locations of significant coefficients of different partitions of seismic data, e.g., shot records, common-offset gathers, or frequency slices of the acquired data can be used as a prior information to interpolate seismic data sets.

In this thesis we introduce modified versions of two well-known recovery algorithms when additional partial support information is available and examine the performance of these modified algorithms. Chapter 2 focuses on *weighted* non-convex optimization for signal reconstruction and Chapter 3 investigates the *weighted* approximate message passing (AMP) algorithm. Before summarizing our main contribution, we first provide the necessary context. Section 1.1 describes compressed sensing as a practicable signal acquisition method in applications and introduces the generic sparse approximation problem. Section 1.2 reviews some of the algorithms used in compressed sensing literature. Section 1.3 clarifies the main idea of using prior information to enhance the recovery performance and summarizes our main contributions, weighted ℓ_p minimization, and weighted AMP.

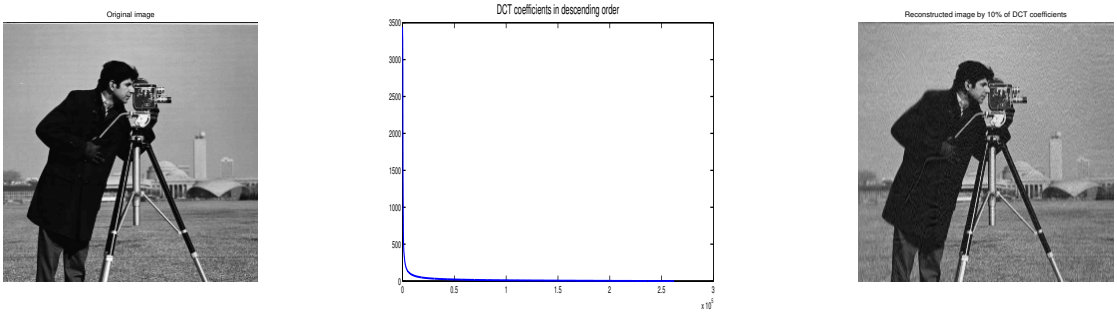
1.1 Compressed sensing and the sparse approximation problem

Role of sparsity

Consider the problem of acquiring a signal $f \in \mathbb{R}^N$ via n linear measurements:

$$y = \Phi f. \tag{1.1}$$

Here Φ is called the sensing matrix and y is the measurement vector. Recalling fundamental theorem of linear algebra, if we want to reconstruct f from y in (1.1), the number of measurements, n, should be at least equal to the number of unknowns N. In many applications, the signal size N, is very big and it is very costly and time consuming to take that many measurements. On the other hand many signals can be approximated by a sparse signal in some convenient basis. In other words most of the information of the signal can be captured by a relatively few nonzero coefficients if the signal is represented in an appropriate basis. Figures 1.1 illustrate this observation: In Figure 1.1.a we show a grayscale image. Figure 1.1.b shows the magnitude of the DCT coefficients of this image in descending order. Notice how fast the DCT coefficients tend to zero. Figure 1.1.c is a reconstruction of the original image in 1.1.a from only the largest 10% of its DCT coefficients. As we can see in Figure 1.1.b, the smallest 90% coefficients of the image in DCT basis are very close to zero and therefore, we do not lose much information when we throw them out.



(a) (b) (c)

Figure 1.1: Image compression with discrete cosine transform (DCT). (a) Original image. (b) DCT coefficients sorted in descending order. (c) Image, reconstructed by zeroing out all the DCT coefficients but largest 10%.

Compressed sensing

Similar to the observation we made in Figure 1.1, for many classes of signals, such as natural images, audio, video, seismic data and images, an appropriate basis can be found such that signal is approximately sparse in that basis.
Assume that the signal f is sparse (or approximately sparse) in the basis $\Psi = \{\psi_1, \psi_2, \ldots, \psi_N\}$. Then f can be written as a linear combination of the basis vectors ψ_1, ψ_2, ..., ψ_N as

$$f = \sum_{i=1}^{N} \psi_i x_i = \Psi x, \tag{1.2}$$

such that $x \in \mathbb{R}^N$ is sparse. Notice that with slight abuse of notation we use Ψ to refer to both the matrix $[\psi_1, \psi_2, \ldots, \psi_N]$ and the basis generated by the columns of this matrix. Substituting (1.2) in (1.1), we get $y = \Phi\Psi x$. Defining $A := \Phi\Psi$, we have $y = Ax$ with the additional information that x is sparse. Exploiting this sparsity lets us reconstruct x and hence f, by much fewer measurements than the number of unknowns N. The sampling paradigm that is based on this observation and hence allows us to acquire sparse high dimensional signals by making a small number of non-adaptive linear measurements is called *compressed sensing* [13, 14]. Of course, our discussion above is sketchy in that we have not specified how to recover the original sparse vector, i.e., how to solve the "sparse recovery problem".

Sparse recovery problem

As mentioned above, sparse recovery problem is at the foundation of compressed sensing. First we quantify what we mean by sparsity. Let $k < N$ be two positive integers. We say that $x \in \mathbb{R}^N$ is k-sparse if $x \in \Sigma_k^N := \{u \in \mathbb{R}^N : \|u\|_0 \leq k\}$ ($\|u\|_0$, denotes the number of non-zero coefficients of u).
Let $x \in \Sigma_k^N$ and assume that $y \in \mathbb{R}^n$, the vector of n linear and potentially noisy measurements of x is acquired via $y := Ax + e$. Here A is an $n \times N$ measurement matrix with $n \ll N$ and e denotes the noise in our measurements with $\|e\|_2 \leq \varepsilon$. We wish to recover x from y by solving the *sparse recovery problem*. This entails finding the sparsest vector, say z, that is feasible, i.e., $\|Az - y\| \leq \epsilon$. In the noise-free case, i.e., when $\epsilon = 0$, the decoder $\triangle_0 : \mathbb{R}^{n \times N} \times \mathbb{R}^n \mapsto \mathbb{R}^N$ is defined as

$$\triangle_0(A, y) := \underset{z \in \mathbb{R}^N}{\operatorname{argmin}} \ ||z||_0 \quad \text{subject to} \ \ Az = y. \tag{1.3}$$

It has been proven, e.g., in [15], that if $n > 2k$, then (1.3) recovers $x \in \Sigma_k^N$ perfectly when A is in general position. That is, in this case $\triangle_0(A, y) = x$. However, (1.3) is a combinatorial problem which becomes intractable as the dimensions of the problem increase. Therefore, one seeks to modify the optimization problem so that it can be solved (at least approximately) with methods that are more tractable than combinatorial search. In the next section we describe some of the algorithms that are used to find the solution of (1.3).

1.2 Alternative algorithms and recovery guarantees

The optimization problem (1.3) is a combinatorial problem. Hence alternative algorithms has been introduced to find its solution (or approximate it) in certain cases that can be identified quantitatively. In this section we describe three of these algorithms.

1.2.1 Convex relaxation

The most common approach to solve (1.3) is to replace the ℓ_0 norm in (1.3) by a convex ℓ_1 norm. More precisely, $\triangle_1 : \mathbb{R}^{n \times N} \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^N$ is defined as

$$\triangle_1(A, y, \epsilon) := \underset{z \in \mathbb{R}^N}{\operatorname{argmin}} \ ||z||_1 \quad \text{subject to} \ \ ||Az - y||_2 \leq \epsilon. \tag{1.4}$$

Figure 1.2 illustrates the idea of using ℓ_1 norm instead of ℓ_0 norm with a simple 2-D example. The line shows a constraint determined by the measurements. The point indicated by the

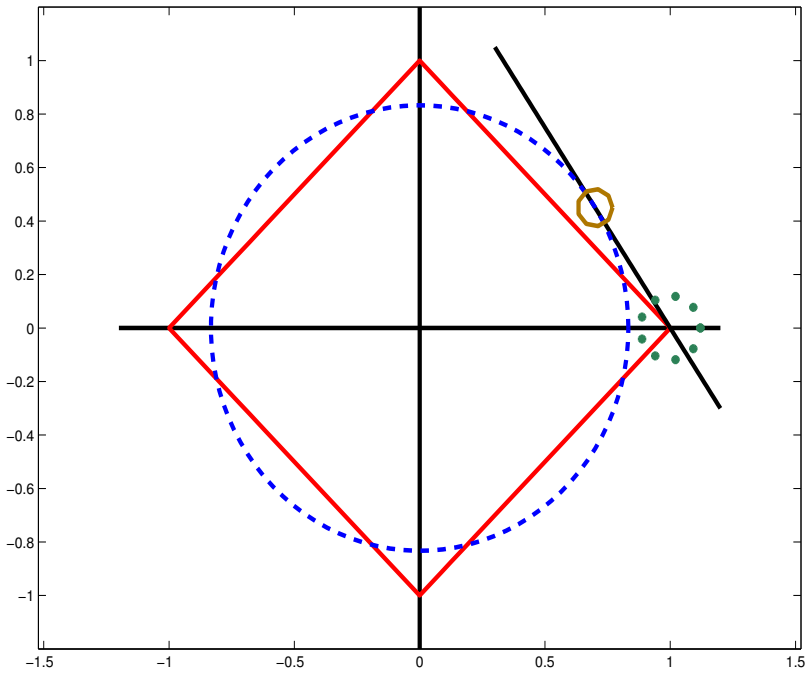


Figure 1.2: ℓ_0, ℓ_1 and ℓ_2 recovery for a simple 2-D example. The line shows the constraint determined by the measurements. The square shows the 1-norm ball and the dashed circle shows the 2-norm ball.

dotted circle shows one of the solutions of the ℓ_0 minimization problem which is the intersection of the constraint line with the x-axis. The square shows the smallest 1-norm ball that touches the constraint line which shows that one of the solutions to the ℓ_0 problem coincides with the unique solution of the ℓ_1 problem. On the other hand, this figure shows that an ℓ_2 minimization problem does not find the sparsest solution on the constraint line in this example. The solution to ℓ_2 minimization is determined by the solid circle which indicates the intersection of the constraint line and the dashed 2-norm ball.

The ℓ_1 minimization problem (1.4) is a convex relaxation of the ℓ_0 problem and hence can be solved in polynomial cost. However, the computational tractability of ℓ_1 minimization comes at the cost of increasing the number of measures taken. Several works has been done to close the gap in the required number of measurements for recovery by ℓ_0 and ℓ_1 minimization problem, including solving a non-convex ℓ_p problem when $0 < p < 1$ which is explained in the

next section.

1.2.2 Nonconvex optimization

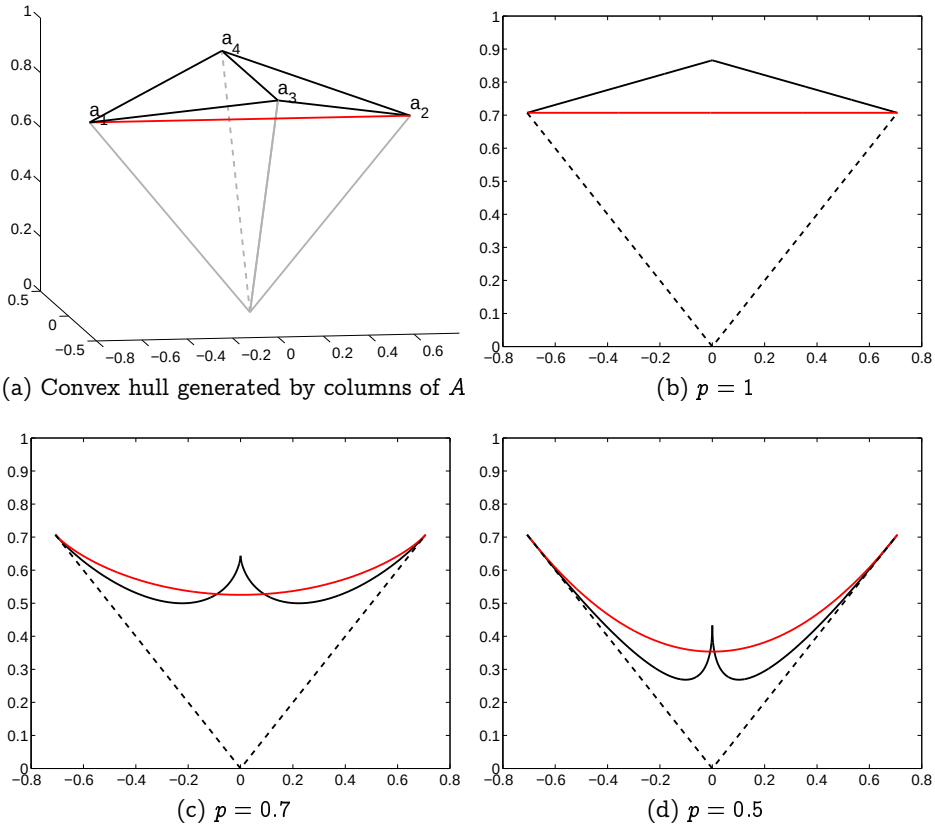
As mentioned in the previous section, the ℓ_1 minimization problem in (1.4) can be formulated as a linear program, which can be solved in polynomial time using standard algorithms (in contrast to combinatorial complexity of the ℓ_0 problem (1.3)). However, this advantage comes with the cost of increasing the number of measurements that has to be taken. One idea to close this gap is to solve a nonconvex ℓ_p minimization problem with $0 < p < 1$ instead of the ℓ_1 minimization.

Chartrand [16] and Saab, and Yılmaz [2], cf. [17], considered the ℓ_p minimization based sparse recovery method when $0 < p < 1$. They have shown that the ℓ_p minimization problem enjoys better recovery guarantees compared to the ℓ_1 problem. Here the ℓ_1 norm in (1.4) is replaced by the ℓ_p norm. Notice that when $0 < p < 1$, this is not a norm, but a quasi-norm. However, in what follows, we shall abuse notation and refer to the ℓ_p quasi-norm as the ℓ_p norm. The decoder $\triangle_p : \mathbb{R}^{n\times N} \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^N$ is defined as

$$\triangle_p(A, y, \epsilon) := \underset{z\in\mathbb{R}^N}{\operatorname{argmin}} \ ||z||_p \quad \text{subject to} \ \ ||Az - y||_2 \leq \epsilon, \tag{1.5}$$

where $0 < p < 1$. Although this method involves solving a nonconvex minimization problem, it can be solved, at least locally, much faster than (1.3). The algorithms used to approximate the solution of this problem will be explained in detail in Chapter 2.

We illustrate the benefits of using such an optimization by a simple example, shown in Figure 1.3. Assume that $a_1, ..., a_4$ denotes the columns of a matrix $A \in \mathbb{R}^{3\times 4}$. Figure 1.3.a shows the convex hull generated by the columns of A. For simplicity, we ignore $-a_1, ..., -a_4$ in the figure; note that including these does not change the outcome of our example. Assume that $x \in \Sigma_2^4$ and the nonzero components of x correspond to the columns a_1 and a_2 with $x_1 > 0$ and $x_2 > 0$. Hence, y is placed on the solid line segment connecting a_1 and a_2 (for simplicity, we name this line segment a_{12}). Assume B_p^N is the unit p-norm ball in $\mathbb{R}^N$; then Figures 1.3.b–d show a cross-section of AB_p^N for $p = 1,\ 0.7,\ 0.5$ respectively. Recall that the ℓ_p minimization problem can recover x if the support of x corresponds to a point on the face of AB_p^N. As we can see in this figure when $p = 1$ the line a_{12} is completely inside the polytope which means that ℓ_1 fails to recover x. As we decrease p, points on a_{12} gradually become points on some face of AB_p^N, making x recoverable. This example illustrates how using ℓ_p minimization with $0 < p < 1$ can be advantageous.



(a) Convex hull generated by columns of A

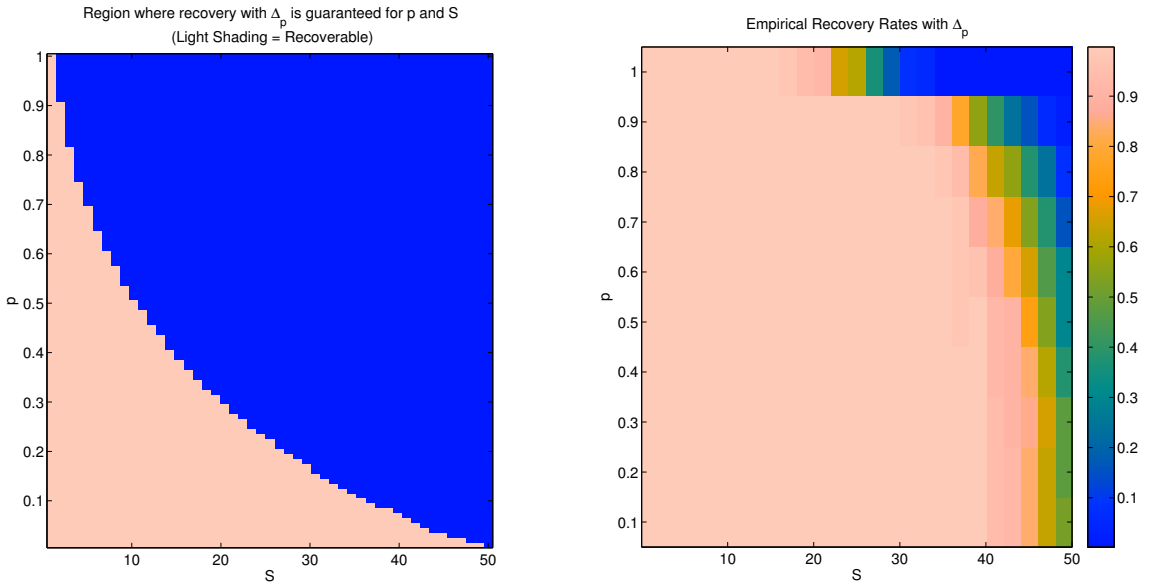
(b) $p = 1$

(c) $p = 0.7$

(d) $p = 0.5$

Figure 1.3: Convex hull generated by columns of A and the polytopes generated by different p's. The ℓ_p minimization can recover x if Ax corresponds to a point on the face of AB_p^N.

Figure 1.4, taken from [2], illustrates the advantage of using $\triangle_p$ for sparse recovery. This diagram shows the success rate of recovering S-sparse signals using ℓ_p minimization, when the measurement matrix is a Gaussian matrix $A \in \mathbb{R}^{100\times 300}$. In Figure 1.4.a the light-shaded areas show the pairs (p,S) that we have guaranteed recovery using ℓ_p minimization to recover an S-sparse signal. In Figure 1.4.b Saab and Yılmaz [2] show the empirical results. Again light-shaded regions correspond to higher rates of successful recovery. This figure illustrates how using ℓ_p minimization improves the recovery conditions. For example, empirically with 90% probability, 20-sparse signals can be recovered by ℓ_1 minimization whereas, ℓ_p minimization with $p = \frac{1}{2}$, recovers 40-sparse signal with probability greater than 90%.



(a) Theoretical results

(b) Empirical results

Figure 1.4: Theoritical and empirical recovery guarantees using ℓ_p minimization to recover an S-sparse signal, when the the measurement matrix is a Gaussian matrix $A \in \mathbb{R}^{100\times 300}$. Adapted from [2] with permission.

1.2.3 Approximate message passing

Applicability of simple linear programming algorithms to the basis pursuit problem (1.4) has made it an appealing choice for compressed sensing problems. However, in large applications, high computational complexity of these algorithms is a problem. Therefore, significant effort has been made to find fast first order methods to recover signals from compressed measurements in applications ranging from biology to imaging where very large dimensional signals are involved. Iterative thresholding algorithms is one family of algorithms which has attracted much attention in the literature [18–20]. However, until recently the iterative thresholding algorithms introduced, had worse recovery conditions than those of convex optimization. In other words, the required number of measurements which had to be taken for recovery by IT algorithms were more than those needed for convex relaxation. Inspired by ideas from theory of graphical models, message passing algorithms and statistical physics, Donoho, Maleki, and Montanari [21] proposed a new iterative thresholding algorithm for sparse recovery, which is referred to as approximate message passing (AMP), which was shown to have empirical perfor-

mance matching BP while still enjoying the low complexity nature of iterative algorithms. The performance of AMP has been justified in the large system limit for Gaussian measurement matrices. The AMP algorithm goes as

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t) \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}) \rangle, \end{aligned} \tag{1.6}$$

where $\delta = \frac{n}{N}$ is the undersampling ratio, η is a scalar thresholding function which acts component-wise on the vector inside, e.g., a soft thresholding function defined as $\eta(a;b) = sign(a)(a-b)_+$, and $\hat{\tau}^t$ is a small threshold number which dictates the sparsity of the signal. Furthermore, $\eta'(x;b) = \frac{\partial}{\partial x}\eta(x;b)$ and $\langle \cdot \rangle$ is the arithmetic mean. x^t is the current estimate of the sparse signal x, z^t is the current residual and A^* is the Hermitian transpose of the measurement matrix, A.

1.3 Using prior information in the recovery algorithm

The algorithms explained in the previous section do not use any prior information on the signal to be recovered—though for successful recovery the signal needs to be sparse. However, in many applications it is possible to use prior information on the signal to reduce the number of measurements even further.

For example we know that speech signals have large low-frequency coefficients. Figure 1.5 shows the amplitude of coefficients of an audio signal in discrete cosine transform domain. Notice how the largest coefficients are concentrated around low frequencies. This information about the audio signals can be used to enhance the recovery conditions. As another example we consider the problem of interpolating irregularly sampled and incomplete seismic data. Assume that we have N_s sources located on earth surface which send sound waves into the earth and N_r receivers record the reflection in N_t time samples. Hence the seismic data is organized in a 3-D seismic line with N_s sources, N_r receivers, and N_t time samples. Rearranging the seismic line, we have a signal $f \in \mathbb{R}^N$, where $N = N_s N_r N_t$. Figure 1.6.a shows an example of a shotgather from a seismic line from the Gulf of Suez, which is extracted from the data using one source and different receivers and times. Figure 1.6.b shows an example of a timeslice from the same seismic line from the Gulf of Suez, which is extracted from the data using one time slice and different sources and receivers. Seismic data is approximately sparse in curvelet domain [22,23] and hence we can formulate the seismic data interpolation problem as an instance of recovery from compressive samples. Assume $x = Sf$ where x is the sparse representation of f in curvelet

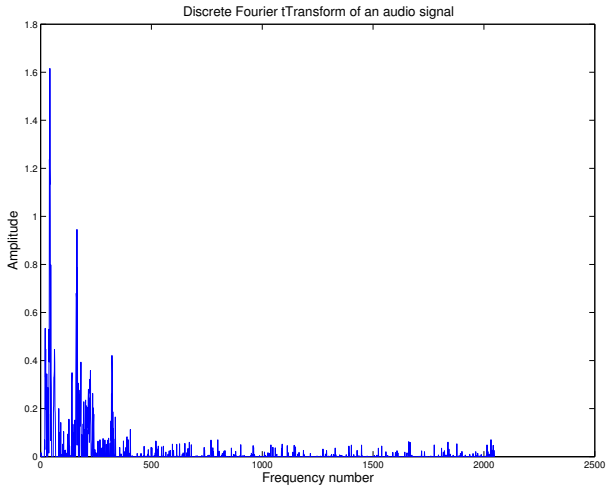
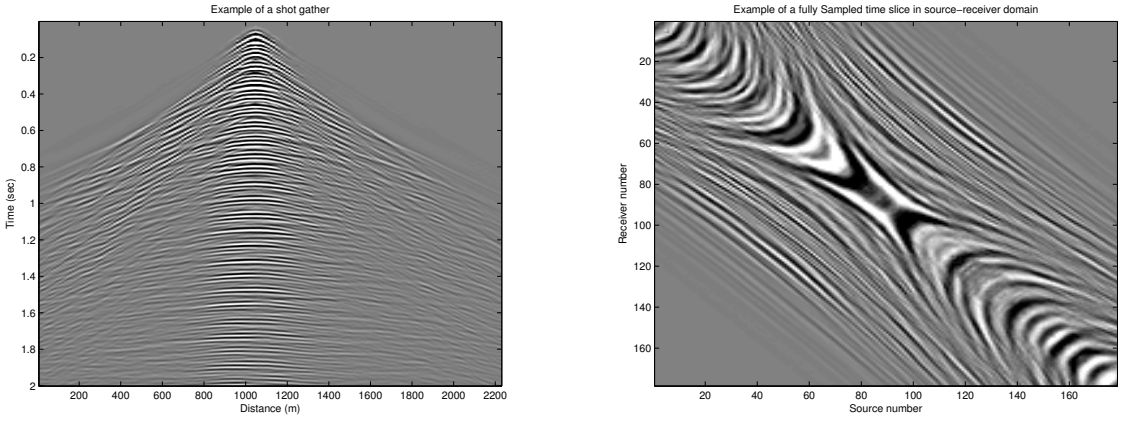


Figure 1.5: Amplitude of frequency coefficients of an audio signal in DCT domain. Most of the large coefficients correspond to low frequencies.

domain. We want to recover a very high dimensional seismic data volume x by interpolating between a smaller number of measurements $b = RMS^*$, where RM is a sampling operator composed of the product of a restriction matrix R, which specifies the sample locations that have data in them and a measurement basis matrix M, which represents the basis in which the measurements are taken [4]. To overcome the problem of high dimensionality, recovery is performed by first partitioning the seismic data volumes into frequency slices, or into common offset-azimuth gathers and then solving a sequence of individual subproblems [4]. Doing this helps us in two ways. It reduces the size of the problem and we can also use the support set of each partition as an estimate of the support set of the next partition. Figure 1.7 taken from [4] is an example of the fraction of overlapping support when adjacent frequencies is used to predict the support of the curvelet coefficients in a seismic line.
Next, we provide a brief introduction to incorporating prior support information into the algorithms introduced in the previous section.

1.3.1 Recovery by weighted ℓ_1 minimization

Suppose that x is an arbitrary vector in $\mathbb{R}^N$ and let x_k be the best k-term approximation of x, i.e., set T_0 to be the set of indices of k largest coefficients of x in magnitude and set $x_k(j) = x(j)$ for $j \in T_0$ and $x_k(j) = 0$ for $j \notin T_0$. Also assume that using prior information



(a) (b)

Figure 1.6: (a) Example of a shot gather from a seismic line from the Gulf of Suez. (b) Example of a high resolution time slice in the source-receiver (SR) domain from a seismic line from the Gulf of Suez.

about the signal we can estimate the locations of the largest coefficients by $\tilde{T}$, i.e., $\tilde{T}$ is an estimate for T_0. It was noted in [1] that one can improve the recovery performance if there is such prior information about the signal that is sufficiently accurate. Specifically, in this case, the required number of measurements for exact or approximate recovery can be reduced by incorporating the prior support information into the ℓ_1 minimization based recovery algorithm. In particular [1] proposes the weighted ℓ_1 decoder $\triangle_{1,w} : \mathbb{R}^{n\times N} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^N \mapsto \mathbb{R}^N$ defined as

$$\triangle_{1,w}(A, y, \epsilon, \mathrm{w}) := \underset{z\in\mathbb{R}^N}{\operatorname{argmin}} \ ||z||_{1,\mathrm{w}} \quad \text{subject to} \ \ ||Az - y||_2 \leq \epsilon, \tag{1.7}$$

where $\mathrm{w} \in \{\omega, 1\}^N$ is the weight vector and $\|z\|_{1,\mathrm{w}} := \Sigma_i \mathrm{w}_i|z_i|$ is the weighted ℓ_1 norm. The optimization problem in (1.7) uses bigger weights for coefficients which are not in the support estimate and smaller weights for the ones in the support estimate. Therefore, the coefficients in the support estimate are more likely to remain nonzero. Details of this algorithm has been presented in Chapter 2.

1.3.2 Recovery by weighted ℓ_p minimization

As mentioned in Section 1.2.1, solving the ℓ_1 minimization problem (1.4) gives us the benefits of solving a convex optimization problem but comes with the cost of increasing number of

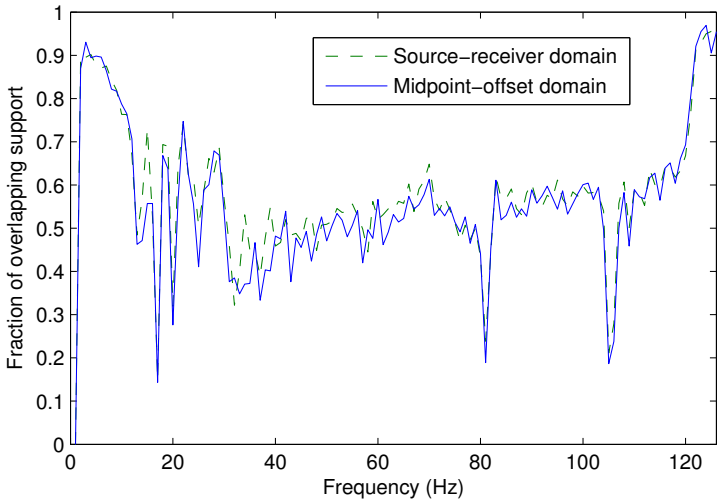


Figure 1.7: Example of the accuracy achieved from using adjacent frequencies to predict the support of the curvelet coefficients in a seismic line. Adapted from [4] with permission.

measurements. So far we have reviewed two algorithms which attempted to close the gap between ℓ_0 and ℓ_1, namely, ℓ_p minimization with $p < 1$ and weighted ℓ_1 minimization. In Section 2 we propose combining these two algorithms to reduce the required number of measurements even further. In particular when we have a support estimate $\widetilde{T}$, we propose to approximate x by $\triangle_{p,w}(A, y, \epsilon, \mathrm{w})$ where $\triangle_{p,w} : \mathbb{R}^{n\times N} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^N \mapsto \mathbb{R}^N$ is defined as

$$\triangle_{p,w}(A, y, \epsilon, \mathrm{w}) := \underset{z\in\mathbb{R}^N}{\operatorname{argmin}} \ \|z\|_{p,\mathrm{w}} \quad \text{subject to} \ \ \|Az - y\|_2 \le \epsilon \ \ \textit{with} \ \ \mathrm{w}_i = \begin{cases} 1, & \text{if } i \in \widetilde{T}^c \\ \omega, & \text{if } i \in \widetilde{T} \end{cases}. \tag{1.8}$$

Here $\mathrm{w} \in \{\omega, 1\}^N$ is the weight vector and $\|z\|_{p,\mathrm{w}} := (\Sigma_i \mathrm{w}_i^p |z_i|^p)^{\frac{1}{p}}$ is the weighted ℓ_p norm. If the support estimate is sufficiently accurate, The sufficient recovery conditions for this optimization problem are weaker than those of weighted ℓ_1—see Theorem 2.12—and also this algorithm outperforms ℓ_p minimization if the support estimate is accurate enough. A thorough discussion including theoretical guarantees associated with weighted ℓ_p minimization, algorithmic issues, and numerical experiments on synthetic data, audio and seismic signals is given in Chapter 2.

1.3.3 Recovery by weighted AMP

Incorporating prior information into ℓ_1 and ℓ_p minimization improves the recovery conditions of sparse and approximately sparse signals. This observation motivated us to derive a weighted AMP algorithm which benefits from the low complexity nature of the AMP algorithm and uses the prior support information about the signal. Let $x \in \mathbb{R}^N$ be a sparse vector. As before we try to recover x from $n < N$ linear measurements acquired via $y = Ax + e$ with $\|e\| \leq \epsilon$. Here A is an $n \times N$ matrix whose coefficients are drawn from a sub-Gaussian distribution. Assume that we have a support estimate $\tilde{T}$ of the nonzero coefficients of the signal x. We incorporate this information into the AMP algorithm by the following iterative algorithm: At $t = 0$ start from $x^0 = 0$ and $\hat{\tau}^0 = 1$ and $z^0 = y$. Assume $\mathrm{w} = [\mathrm{w}_1,\ \mathrm{w}_2,\ \ldots,\ \mathrm{w}_N]^T$ is the weight vector where $\mathrm{w}_i = \omega < 1$ for $i \in \tilde{T}$ and $\mathrm{w}_i = 1$ for $i \in \tilde{T}^c$. Then the weighted AMP algorithm proceeds as

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathrm{w}) \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle \\ \hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle, \end{aligned} \tag{1.9}$$

where η is the soft thresholding function defined in (1.6).
In Chapter 3 we explain the derivation of this algorithms and show numerical results which compare this algorithm with regular AMP and weighted ℓ_1.

Chapter 2

Recovering compressively sampled signals by non-convex optimization and using partial support information

2.1 Introduction and overview

In this section we review the ℓ_1, ℓ_p and weighted ℓ_1 minimization algorithms introduced in the previous chapter. Our setting is as before: We have a sparse signal $x \in \Sigma_k^N$ and we want to recover this signal from $y \in \mathbb{R}^n$, n linear and potentially noisy measurements of x which are acquired via $y := Ax + e$. As before A is an $n \times N$ measurement matrix with $n \ll N$ and e denotes the noise in our measurements with $\|e\|_2 \leq \varepsilon$. This entails finding the sparsest vector x that is feasible, i.e., $\|Ax - y\| \leq \varepsilon$. Restating (1.3), in the noise-free case, the decoder $\triangle_0 : \mathbb{R}^{n \times N} \times \mathbb{R}^n \mapsto \mathbb{R}^N$ is defined as

$$\triangle_0(A, y) := \underset{z \in \mathbb{R}^N}{\operatorname{argmin}} \ ||z||_0 \quad \text{subject to} \ \ Az = y, \tag{2.1}$$

and $\triangle_0(A, y)$ recovers x if $n > 2k$ and A is in general position (see, e.g., [15]). As mentioned earlier, generally (2.1) is a combinatorial problem which becomes intractable as the dimensions of the problem increase. Therefore, one seeks to modify the optimization problem so that it can be solved (at least approximately) with methods that are more tractable than combinatorial search. We introduced some of these algorithms in Chapter 1, namely, ℓ_1, ℓ_p and weighted ℓ_1 minimization. In this section we review these algorithms more carefully and provide the stability and robustness theorems of each.

In all these algorithms the restricted isometry constants, defined in [24], play a central role.

Definition 1. *A matrix A satisfies restricted isometry property (RIP) of order k with con-*

stant δ_k if for all k-sparse vectors $z \in \Sigma_k^N$,

$$(1-\delta_k)||z||_2^2 \leq ||Az||_2^2 \leq (1+\delta_k)||z||_2^2. \tag{2.2}$$

2.1.1 Recovery by ℓ_1 minimization

Donoho [13] and Candés, Romberg, and Tao [14] showed that if A obeys a certain restricted isometry property, then solving (1.4) can stably and robustly recover x from seemingly incomplete and inaccurate measurements $y = Ax + e$. The following theorem, proved in [14], shows the error guarantees of finding the best k-sparse solution with (1.4):

Theorem 2.1. *(Candés, Romberg and Tao [14]) Let k, N be positive integers with $k < N$. Suppose that x is an arbitrary vector in $\mathbb{R}^N$ and let x_k be the best k-term approximation of x. Let $y = Ax + e$ with $\|e\|_2 \leq \epsilon$. If A satisfies the restricted isometry property (RIP) with $\delta_{ak} + a\delta_{(a+1)k} < a - 1$, then*

$$||\triangle_1(A, y, \epsilon) - x||_{\ell_2} \leq C_1^{\ell_1} \cdot \epsilon + C_2^{\ell_1} \cdot \frac{||x - x_k||_{\ell_1}}{\sqrt{k}}.$$

Remark 2.2. The constants $C_1^{\ell_1}$ and $C_2^{\ell_1}$ in Theorem 2.1 are given explicitly by

$$\begin{aligned} C_1^{\ell_1} &= \frac{2(1 + a^{-\frac{1}{2}})}{\sqrt{1 - \delta_{(a+1)k}} - a^{-\frac{1}{2}}\sqrt{1 + \delta_{ak}}}, \\ C_2^{\ell_1} &= \frac{2a^{-\frac{1}{2}}(\sqrt{1 - \delta_{(a+1)k}} + \sqrt{1 + \delta_{ak}})}{\sqrt{1 - \delta_{(a+1)k}} - a^{-\frac{1}{2}}\sqrt{1 + \delta_{ak}}}. \end{aligned} \tag{2.3}$$

Remark 2.3. Theorem 2.1 states that if $\delta_{ak} + a\delta_{(a+1)k} < a - 1$, then (1.4) can recover any k-sparse vector x with an error proportional to the 2-norm of the measurement noise. For example in this case if we set $a = 3$, $\| \triangle_1 (A, y, \epsilon) - x\|_{\ell_2} \leq C_1^{\ell_1}\epsilon$, and for reasonable values of δ_{4k}, $C_1^{\ell_1}$ is well behaved, e.g., $C_1^{\ell_1} < 12.04$ for $\delta_{4k} = \frac{1}{5}$.

Remark 2.4. It is worth noting that calculating the RIP constants for an arbitrary matrix A is computationally expensive. On the other hand it was proved in [25] that if columns of A are independent, identically distributed (i.i.d.) random vectors with any sub-Gaussian distribution, then A satisfies RIP property (2.2) for any $0 < \delta < 1$ with probability greater than $1-2e^{-c_2 k}$, whenever $k \leq c_1 \frac{n}{log\frac{N}{n}}$; here c_1, and c_2 are positive constants that only depend on δ and the distribution of A.

As mentioned in Chapter 1, the ℓ_1 minimization problem (1.4) is a convex optimization problem and can be solved in polynomial time. However, the computational tractability of ℓ_1 minimization comes at the cost of increasing the number of measurements taken. For example if the columns of the measurement matrix A are independent, identically distributed random vectors with any sub-Gaussian distribution, Theorem 2.1 together with Remark 2.4 implies that $\triangle_1$ can recover any k-sparse vector x when $n \gtrsim k \ \ log(\frac{N}{k})$ rather than the $n > 2k$ property which is sufficient for recovery by $\triangle_0$. Therefore, as mentioned earlier, several works has been done to close the gap in the required number of measurements for recovery via ℓ_0 and ℓ_1 minimization problems, including solving a non-convex ℓ_p problem when $p < 1$ [2, 16] and using prior knowledge about the signal to be recovered [1].

2.1.2 Recovery by ℓ_p minimization

It was shown in [2,16,17,26] that recovery by ℓ_p minimization is stable and robust under weaker sufficient conditions than the analogous ℓ_1 minimization conditions. Therefore, the global minimizer of (1.5) is an approximate solution to (2.1) for a more general class of measurement matrices A than the minimizer of (1.4). This result is made explicit by the following theorem from [2].

Theorem 2.5. *(Saab and Yılmaz [2]) Suppose that x is an arbitrary vector in $\mathbb{R}^N$ and let x_k be the best k-term approximation of x and let $y = Ax + e$ with $\|e\|_2 \leq \epsilon$. If A satisfies $\delta_{ak} + a^{\frac{2}{p}-1}\delta_{(a+1)k} < a^{\frac{2}{p}-1} - 1$, for $a \in \frac{1}{k}\mathbb{N}$ and $k > 1$ and $0 < p < 1$ then*

$$||\triangle_p(A, y, \epsilon) - x||_2^p \leq C_1^{\ell_p} \cdot \epsilon^p + C_2^{\ell_p} \cdot \frac{||x - x_k||_p^p}{k^{1-p/2}}.$$

Remark 2.6. It is sufficient that A satisfies

$$\delta_{(a+1)k} < \hat{\delta}^{\ell_p} := \frac{a^{\frac{2}{p}-1} - 1}{a^{\frac{2}{p}-1} + 1} \tag{2.4}$$

for Theorem 2.5 to hold, i.e., to guarantee stable and robust recovery described in the theorem with the same constants $C_1^{\ell_p}$ and $C_2^{\ell_p}$.

Remark 2.7. Constants $C_1^{\ell_p}$ and $C_2^{\ell_p}$ in Theorem 2.5 are given explicitly by:

$$\begin{aligned} C_1^{\ell_p} &= \frac{2^p\left(1+\frac{1}{\left(a^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)}{(1-\delta_{(a+1)k})^{\frac{p}{2}}-(1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1}} \\ C_2^{\ell_p} &= \frac{2\left((1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1}+\frac{(1-\delta_{(a+1)k})^{\frac{p}{2}}}{\left(a^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)}{(1-\delta_{(a+1)k})^{\frac{p}{2}}-(1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1}}. \end{aligned} \tag{2.5}$$

As stated in [2] the constants $C_1^{\ell_p}$ and $C_2^{\ell_p}$ are well behaved. When $p=1$ these constants reduce to those of Theorem 2.1. If $p=\frac{1}{2}$ and $\delta_{4k}=\frac{1}{5}$, then $C_1^{\ell_p}<3.87$ and $C_2^{\ell_p}<3.18$. It was shown in [26] that smaller bounds are achievable in the noise-free case. Moreover, the result in [2] says that given a matrix A that satisfies $\delta_{(a+1)k}<\frac{a-1}{a+1}$ and suppose that k_1 is the largest k for which the inequality holds, then Theorem 2.8 guarantees that (1.4) can recover all k_1-sparse signals whereas Theorem 2.5 guarantees that (1.5) can recover all k_p-sparse vectors where $k_p=\left\lfloor\frac{a+1}{a^{\frac{p}{2-p}}+1}k_1\right\rfloor$ which is bigger than k_1 when $p<1$.

2.1.3 Recovery by weighted ℓ_1 minimization

As stated also in Chapter 1, the ℓ_1 problem (1.4) does not use any prior information about the signal (though for successful recovery the signal needs to be sparse). However, in many applications it is possible to estimate the support of the signal which we want to recover. It was noted in [1] that one can improve the recovery conditions if there is prior information about the signal. This result is demonstrated by the following theorem from [1].

Theorem 2.8. *(FMSY [1]) Suppose that x is an arbitrary vector in $\mathbb{R}^N$ and $y=Ax+e$ with $\|e\|_2\leq\epsilon$. Let x_k be the best k-term approximation of x with $supp\{x_k\}=T_0$. Let $\tilde{T}$ be an arbitrary subset of $\{1,2,...,N\}$ and define ρ and α such that $|\tilde{T}|=\rho k$ and $|T_0\cap\tilde{T}|=\alpha\rho k$. (Here $|\tilde{T}|$ denotes the cardinality of $\tilde{T}$.) Suppose there exists an $a\in\frac{1}{k}\mathbb{Z}$ with $a\geq(1-\alpha)\rho$ and $a>1$ and the measurement matrix A has RIP with*

$$\delta_{ak}+\frac{a}{(\omega+(1-\omega)\sqrt{1+\rho-2\alpha\rho})^2}\delta_{(a+1)k}<\frac{a}{(\omega+(1-\omega)\sqrt{1+\rho-2\alpha\rho})^2}-1$$

for some $0 \leq \omega \leq 1$. *Then*

$$||\triangle_{1,w}(A, y, \epsilon, \mathrm{w}) - x||_2 \leq C_1^{w\ell_1}\epsilon + C_2^{w\ell_1} k^{\frac{-1}{2}}(\omega\|x - x_k\|_1 + (1-\omega)\|x_{\widetilde{T}^c \cap T_0^c}\|_1).$$

Remark 2.9. Constants $C_1^{w\ell_1}$ and $C_2^{w\ell_1}$ in Theorem 2.8 are given explicitly by:

$$\begin{aligned} C_1^{w\ell_1} &= \frac{2(1 + \frac{w+(1-w)\sqrt{1+\rho-2\alpha\rho}}{\sqrt{a}})}{\sqrt{1-\delta_{(a+1)k}} - \frac{w+(1-w)\sqrt{1+\rho-2\alpha\rho}}{\sqrt{a}}\sqrt{1+\delta_{ak}}} \\ C_2^{w\ell_1} &= \frac{2a^{\frac{-1}{2}}(\sqrt{1-\delta_{(a+1)k}} + \sqrt{1+\delta_{ak}})}{\sqrt{1-\delta_{(a+1)k}} - \frac{w+(1-w)\sqrt{1+\rho-2\alpha\rho}}{\sqrt{a}}\sqrt{1+\delta_{ak}}}. \end{aligned} \tag{2.6}$$

Remark 2.10. It is sufficient that A satisfies

$$\delta_{(a+1)k} < \hat{\delta}^{w\ell_1} := \frac{a - (\omega + (1-\omega)\sqrt{1+\rho-2\alpha\rho})^2}{a + (\omega + (1-\omega)\sqrt{1+\rho-2\alpha\rho})^2} \tag{2.7}$$

for Theorem 2.8 to hold, i.e., to guarantee stable and robust recovery described in the theorem with the same constants $C_1^{w\ell_1}$ and $C_2^{w\ell_1}$.

Notice that ρ determines the ratio of the size of the estimated support to the size of the actual support of x_k (or the support of x if x is k-sparse) and α determines the accuracy of the support estimate which is the ratio of the size of $\widetilde{T} \cap T$, to the the size of our estimate $\widetilde{T}$.

Remark 2.11. Theorem 2.8 shows that if $\alpha > 0.5$ the sufficient recovery conditions of weighted ℓ_1 become weaker than those of regular ℓ_1. Moreover, the error bound constants become smaller. The result also indicates that when the support estimate is accurate, i.e., $\alpha > 0.5$, a choice of the weight $\omega = 0$ results in the weakest sufficient conditions for recovery and smallest error bounds, whereas choosing $\omega = 1$ achieves the weakest sufficient conditions for recovery and smallest error bounds when the support accuracy is low ($\alpha < 0.5$). However, numerical experiments conducted in [1] have shown that weighted ℓ_1 minimization achieves the best recovery when intermediate values of ω are used for different accuracy levels. We discuss this issue more extensively in the next section.

Combining these two ideas, in this chapter we study the sparse recovery guarantees of the weighted ℓ_p minimization problem (1.8). Specifically, we show that the weighted ℓ_p minimization algorithm outperforms both ℓ_p minimization and weighted ℓ_1 minimization under certain circumstances.

In Section 2.2, we describe the proposed weighted ℓ_p minimization algorithm, derive stability

and robustness guarantees for this algorithm and compare it with regular ℓ_p and weighted ℓ_1. Specifically, we prove that the recovery guarantees of the weighted ℓ_p algorithm with $0 < p < 1$ are better than those of weighted ℓ_1 and regular ℓ_p when we have a prior support estimate with accuracy better than 50%. In Section 2.3, we explain the algorithmic issues that come with solving the proposed non-convex optimization problem and the approach we take to empirically overcome them. In Section 2.4, we present numerical experiments where we apply the weighted ℓ_p method to recover sparse and compressible signals. In Section 2.5, we show the result of applying these algorithms to audio and seismic signals. In Section 2.6, we provide the proof for our main theorem.

2.2 Main results

In this section we explain the recovery by weighted ℓ_p minimization. In particular when we have a prior support estimate $\widetilde{T}$, we approximate x by $\triangle_{p,w}(A, y, \epsilon, \mathrm{w})$ where $\triangle_{p,w} : \mathbb{R}^{n\times N} \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^N \mapsto \mathbb{R}^N$ is defined as

$$\triangle_{p,w}(A, y, \epsilon, \mathrm{w}) := \underset{z\in\mathbb{R}^N}{\operatorname{argmin}} \ \|z\|_{p,\mathrm{w}} \quad \text{subject to} \ \ \|Az - y\|_2 \leq \epsilon \ \ \textit{with} \ \ \mathrm{w}_i = \begin{cases} 1, & \text{if } i \in \widetilde{T}^c \\ \omega, & \text{if } i \in \widetilde{T} \end{cases}. \tag{2.8}$$

Here $\mathrm{w} \in \{\omega, 1\}^N$ is the weight vector and $\|z\|_{p,\mathrm{w}} := (\Sigma_i \mathrm{w}_i^p |z_i|^p)^{\frac{1}{p}}$ is the weighted ℓ_p norm. Next we provide the stable and robust recovery conditions of this algorithm and compare it with weighted ℓ_1 and ℓ_p in terms of sufficient conditions for stable and robust recovery and associated constants.

2.2.1 Weighted ℓ_p minimization with estimated support

As mentioned in the previous section, one can improve the recovery guarantees of $\triangle_1$ by using $\triangle_p$ and by incorporating prior support information into the the optimization problem. In this section we provide the recovery conditions when we combine both these approaches. The following theorem states the main result.

Theorem 2.12. *Suppose that x is an arbitrary vector in $\mathbb{R}^N$ and $y = Ax + e$ with $\|e\|_2 \leq \epsilon$. Let x_k be the best k-term approximation of x with supp$\{x_k\} = T_0$. Let $\widetilde{T}$ be an arbitrary subset of $\{1, 2, ..., N\}$ and define ρ and α such that $|\widetilde{T}| = \rho k$ and $|T_0 \cap \widetilde{T}| = \alpha\rho k$. Suppose there exist an $a \in \frac{1}{k}\mathbb{Z}$, with $a \geq (1-\alpha)\rho$ and $a > 1$ and the measurement matrix A has*

RIP with

$$\delta_{ak} + \frac{a^{\frac{2}{p}-1}}{(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}}\delta_{(a+1)k} < \frac{a^{\frac{2}{p}-1}}{(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}} - 1, \tag{2.9}$$

for some $0 \leq \omega \leq 1$ *and* $0 < p < 1$. *Then*

$$\| \triangle_{p,w}(A, y, \epsilon, \mathrm{w}) - x\|_2^p \leq C_1\epsilon^p + C_2 k^{\frac{p}{2}-1}(\omega^p\|x - x_k\|_p^p + (1-\omega^p)\|x_{\widetilde{T}^c \cap T_0^c}\|_p^p). \tag{2.10}$$

Remark 2.13. Note that ρ and α have the same definitions as in Theorem 2.8, i.e., ρ denotes the ratio of the size of the estimated support to the size of the actual support of x_k and α denotes the accuracy of our estimate which is the ratio of the size of $\widetilde{T} \cap T_0$, to the the size of our estimate $\widetilde{T}$.

Remark 2.14. The constants C_1 and C_2 are explicitly given by:

$$C_1 = \frac{2^p\left(1 + \frac{\left(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}}\right)}{\left(a^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)}{(1-\delta_{(a+1)k})^{\frac{p}{2}} - (1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1}\left(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}}\right)}, \tag{2.11}$$

$$C_2 = \frac{2\left((1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1} + \frac{(1-\delta_{(a+1)k})^{\frac{p}{2}}}{\left(a^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)}{(1-\delta_{(a+1)k})^{\frac{p}{2}} - (1+\delta_{ak})^{\frac{p}{2}}a^{\frac{p}{2}-1}\left(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}}\right)}. \tag{2.12}$$

Remark 2.15. Theorem 2.12 is consistent with the analogous theorem for ℓ_p [2], i.e., Theorem 2.5 and for weighted ℓ_1 [1], i.e., Theorem 2.8. That is, if we set $\omega = 1$, Theorem 2.12 reduces to Theorem 2.5 and setting $p = 1$, Theorem 2.12 yields Theorem 2.8 as a special case.

Remark 2.16. It is sufficient that A satisfies

$$\delta_{(a+1)k} < \hat{\delta}^{w\ell_p} := \frac{a^{\frac{2}{p}-1} - (\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}}{a^{\frac{2}{p}-1} + (\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}} \tag{2.13}$$

for Theorem 2.12 to hold, i.e., to guarantee stable and robust recovery described in the theorem with the same constants C_1 and C_2. Setting $\omega = 1$ gives us sufficient conditions for recovery by $\triangle_p$ and setting $p = 1$ yields sufficient conditions for recovery by $\triangle_{1,w}$. Notice that these

conditions are in terms of bounds on RIP constants. In Section 2.2.2 we compare these bounds.

Remark 2.17. In [27] Candés used a slightly different approach to guarantee the stable and robust recovery when $a = 1$. Candés proved that if $\delta_{2k} < (1+\sqrt{2})^{-1}$, then $\triangle_1$ achieves stable and robust recovery. Using the same approach as in [27] we get the condition

$$\delta_{2k} < \frac{1}{1+\sqrt{2}(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})} \tag{2.14}$$

which gives us an alternative stable and robust recovery condition for $\triangle_{p,w}$. When $\alpha > 0.5$ this condition provides weaker conditions on the measurement matrices compared to $\triangle_p$. We omit the derivation of this condition as it closely mimics the respective calculations in [27].

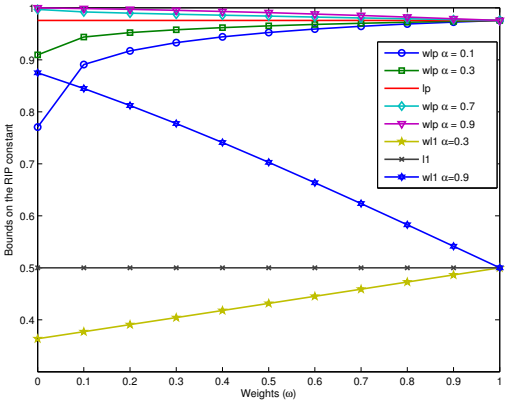
2.2.2 Comparison to weighted ℓ_1 recovery

In this section we compare the conditions for which Theorem 2.12 holds with the corresponding conditions for Theorem 2.8. The following observations are easy to verify.

Proposition 2.18. *Let C_1, C_2, $C_1^{w\ell_1}$ and $C_2^{w\ell_1}$ be as defined above. If $p = 1$ then $C_1 = C_1^{w\ell_1}$ and $C_2 = C_2^{w\ell_1}$ and the sufficient condition for Theorem 2.12 would be identical to Theorem 2.8.*

Remark 2.19. If $\alpha = 0.5$, then the sufficient condition for stable and robust recovery with (2.8) would be $\delta_{(a+1)k} < \frac{a^{\frac{2}{p}-1}-1}{a^{\frac{2}{p}-1}+1}$ and the analogous condition for stable and robust recovery with (1.7) would be $\delta_{(a+1)k} < \frac{a-1}{a+1}$. Since $a > 1$ for $0 < p < 1$, $a^{\frac{2}{p}-1} > a$. Therefore, the weighted ℓ_p conditions would be much weaker than the weighted ℓ_1 conditions, for example, when $a = 3$ and $p = 0.5$, then the sufficient condition for weighted ℓ_p is $\delta_{4k} < \frac{26}{28}$ which is much weaker than the sufficient condition for weighted ℓ_1 which is $\delta_{4k} < 0.5$.

Figure 2.1 illustrates how the sufficient conditions on the RIP constants vary with α and ω in the case of weighted ℓ_1 and weighted ℓ_p. In particular these sufficient conditions are introduced in Theorem 2.8 and Theorem 2.12, i.e., $\hat{\delta}^{w\ell_1}$ defined in (2.7) and $\hat{\delta}^{w\ell_p}$ defined in (2.13) which determine bounds on the RIP constants. In Figure 2.1, we plot $\hat{\delta}^{w\ell_p}$ versus ω for $p = 1$ and $p = \frac{2}{5}$, with $a = 3$ in both cases and different values of α. The bounds on RIP constants get larger as α increases which means that a wider range of measurement matrices satisfy the sufficient condition when the support estimate is more accurate. Note that when $\alpha = 0.5$ the sufficient conditions for recovery by weighted ℓ_p would be identical to sufficient conditions for recovery by standard ℓ_p for $0 < p < 1$. Comparing these results with recovery by weighted ℓ_1,



(a) $\hat{\delta}^w$ vs ω

Figure 2.1: Comparison of the sufficient conditions for recovery with weighted ℓ_p reconstruction with various α. In all the Figures, we set a $= 3$ and $\rho = 1$ and $p = \frac{2}{5}$.

we see that in recovery by weighted ℓ_p the measurement matrix A has to satisfy much weaker conditions than the analogous conditions in recovery by weighted ℓ_1 even when we do not have a good support estimate. For example with $a = 3$ and $\omega = 0.2$, when the support estimate is 70% accurate, we get the sufficient condition $\hat{\delta}^{wlp} < 0.9897$ which is so close to the best that we can get ($\hat{\delta} < 1$) and it is weaker than the sufficient recovery condition for weighted ℓ_1 which is $\hat{\delta}^{w\ell_1} < 0.763$ and regular ℓ_1 which is $\hat{\delta}^1 < 0.5$.

It is worth considering the sufficient recovery conditions for the special case of zero weight. As seen in Figure 2.1 setting $\omega = 0$ is beneficial when $\alpha > 0.5$, however, the sufficient recovery condition is stricter when $\alpha < 0.5$. This suggests that setting the weight $\omega = 0$ is beneficial in the case where the support estimate is highly accurate.

Figure 2.2 compares the recovery guarantees we obtain in the zero-weight case with conditions that guarantee recovery via weighted ℓ_p and weighted ℓ_1 minimization. To this end, we present the phase diagrams of measurement matrices A with Gaussian entries that satisfy the conditions on the restricted isometry constants $\delta_{(a+1)k}$ given in (2.7) and (2.13) with $\omega = 0$, $\rho = 1$, and $\alpha = 0.3, 0.6$, and 0.8. The phase diagrams are calculated using the upper bounds on the RIP constants derived in [5] and reflect the sparsity levels for which the three theorems guarantee exact signal recovery as a function of the aspect ratio of the measurement matrix A.

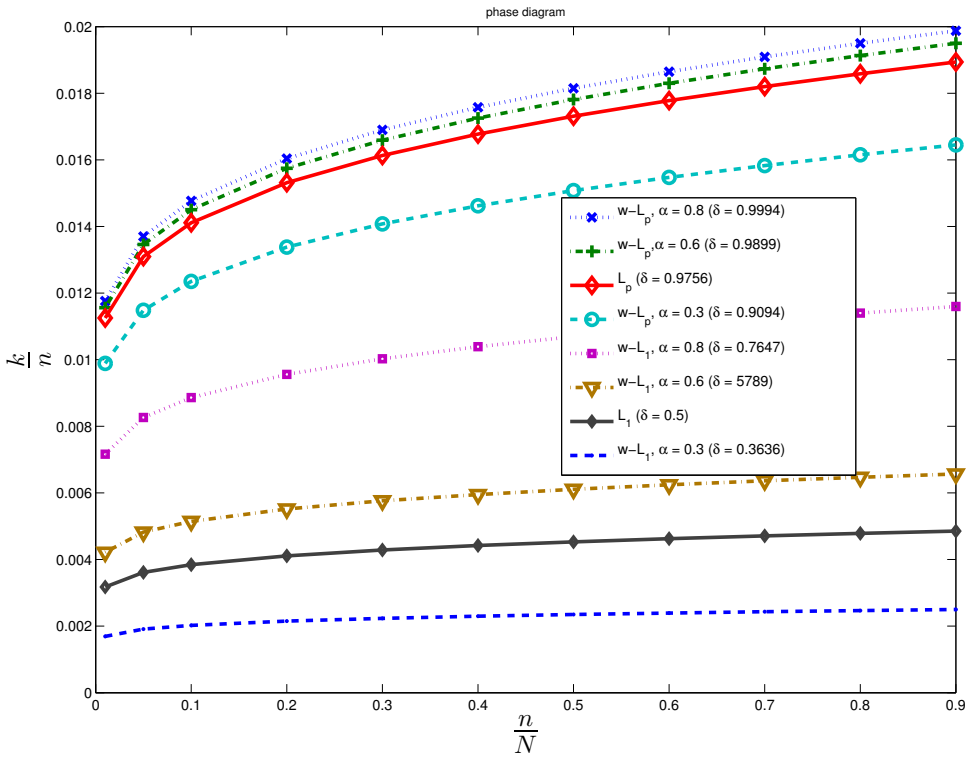
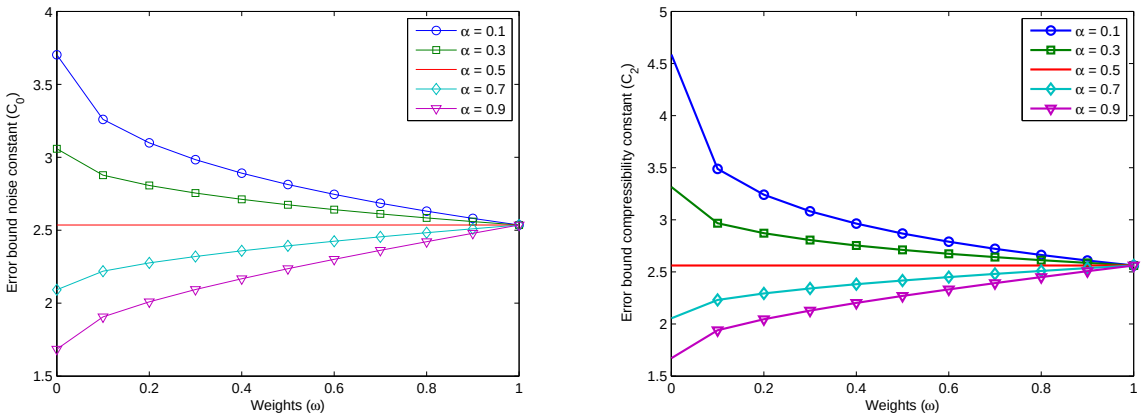


Figure 2.2: Comparison between the phase diagrams of measurement matrices with Gaussian entries satisfying the sufficient recovery conditions of weighted ℓ_p minimization and weighted ℓ_1 minimization with $\omega = 0$ and $\alpha = 0.3$, 0.6, and 0.8 and standard ℓ_1. The plots are calculated using the upper bounds on the restricted isometry constants derived in [5]. Points below each curve determine the sparsity-undersampling ratios that satisfy the sufficient bounds on the RIP constants introduced in (2.7) and (2.13).

2.2.3 Comparison to ℓ_p recovery

In this section we compare the sufficient conditions of Theorem 2.5 and Theorem 2.12. The following observations are easy to verify.

Proposition 2.20. *Let C_1, C_2, $C_1^{\ell_p}$ and $C_2^{\ell_p}$ be as defined above .*
(i) If $\omega = 1$ then $C_1 = C_1^{\ell_p}$ and $C_2 = C_2^{\ell_p}$ and the sufficient condition for Theorem 2.12 and Theorem 2.5 would be identical.
(ii) If $\alpha = 0.5$ then again $C_1 = C_1^{\ell_p}$ and $C_2 = C_2^{\ell_p}$ and the sufficient condition for Theorem 2.12 would be identical to Theorem 2.5.
(iii) Suppose $0 \leq \omega < 1$ then $C_1 < C_1^{\ell_p}$ and $C_2 < C_2^{\ell_p}$ if and only if $\alpha > 0.5$.



(a) C_1 vs ω (b) C_2 vs ω

Figure 2.3: Comparison of the recovery constants for weighted ℓ_p reconstruction with various α. In all the figures, we set a $= 3$ and $\rho = 1$ and $p = \frac{2}{5}$.

Proposition 2.20 reflects the results shown in Figure 2.3. Figures 2.3.a and 2.3.b show how constants C_1 and C_2 in (2.10) change with ω for different values of α. Notice that when we increase α we get smaller error constants.

When $\alpha < 0.5$, i.e., when our estimate is less than 50% accurate, using bigger weights would result in more robust recovery which is useful when we do not know the estimate accuracy exactly. However, numerical results—Section 2.4—show that using intermediate weights usually results in better recovery not only when $\alpha < 0.5$ but also in some cases of $\alpha > 0.5$. When $\omega = 1$, the conditions of Theorems 2.12 and 2.5 become identical. For all values of $\omega < 1$, having a support estimate accuracy $\alpha > 0.5$ results in a weaker condition on the RIP constant and smaller error bound constants compared with the conditions of standard ℓ_p. On the other hand, if $\alpha < 0.5$, i.e., the support estimate has low accuracy, then standard ℓ_p has weaker sufficient recovery conditions and smaller error bound constants compared to weighted ℓ_p. This behavior is similar to that derived for weighted ℓ_1 minimization in [1].

2.3 Algorithms

In this section, we study the algorithm that we have used for the weighted ℓ_p minimization problem (2.8).

2.3.1 Algorithmic issues

In this section, we study the performance of three different algorithms that attempt to solve the ℓ_p minimization problem. There are a few algorithms which are commonly used to solve this minimization problem including the projected gradient method [16], the iterative reweighted ℓ_1 method [28], and the iterative reweighted least squares method [29]. Since the ℓ_p minimization problem is non-convex and several local minima exist, these algorithms attempt to converge to local minima that are close to the global minimizer of the problem. To that end, the only proofs of global convergence that currently exist assume that the global minimizer can be found if a feasible point can be found. However, numerical experiments show that these algorithms perform well when the measurement matrix has i.i.d. Gaussian random entries.
We describe these algorithms in the remainder of this section.

2.3.2 Projected gradient method

An approach based on the projected gradient method has been proposed in [16] and also used, for example, in [2] to compute approximate solutions of (1.5).
An iteration of this algorithm minimizes a smoothed ℓ_p objective instead of the ℓ_p norm. The smoothed ℓ_p cost function is given by $(\sum_i (x_i^2 + \sigma)^{p/2})^{1/p}$. The smoothing parameter σ is initialized with a large value, say 10. After taking a projected gradient step in every iteration, the value of σ is reduced. In every iteration, the new iterant is projected onto the affine space defined by $Ax = b$.

Adopting to the weighted ℓ_p minimization

This method can be adopted to solve the weighted ℓ_p problem by replacing the ℓ_p-norm, $\|\cdot\|_p$, with the weighted ℓ_p norm, $\|\cdot\|_{p,w}$, and making corresponding changes to the algorithm to work for the weighted version. Algorithm 1 explains the details of this algorithm. Here $\nabla(f_x)_i = p \times \mathrm{w}_i^p \times (x_i^t \times (x_i^t)^* + \sigma^2)^{p/2-1} \times x_i^t$.

2.3.3 Iterative reweighted ℓ_p

The iterative reweighted ℓ_p method, proposed in [28], is a modification of the iterative reweighted ℓ_1 (IRL1) algorithm of [30]. This method uses a modified version of IRL1 to solve the following penalized likelihood signal restoration problem:

$$\min_{x \in \mathbb{R}^N} \quad \|Ax - b\|_2^2 + \lambda \|x\|_p^p. \tag{2.15}$$

Algorithm 1 Weighted projected gradient method

Input $b = Ax + e$, p, A, $\omega \in [0, 1]$, Λ
Output $x^{(t)}$
Initialize $\sigma = 10$, $t = 0$, $x^0 = A^* b$, $[M \quad N] = size(A)$, $Q = A^\dagger \times A$, $\mathrm{w}_i = \begin{cases} \omega, & i \in \Lambda \\ 1, & i \in \Lambda^c \end{cases}$
loop
  $f_x = \sum_i (\mathrm{w}_i^2 \times ((x_i^t)^2 + \sigma))^{p/2}$
  $d = -\nabla(f_x)$
  $pd = d - Q \times d$
  $t = t + 1$
  line search
  $x^t = x^{t-1} + l \times pd$
  Indicator=$\frac{\sqrt{1-p} \times x^t}{1-\sqrt{p}}$
  Idx=find(Indicator $<$ w $\times\, \sigma$)
  $\sigma = \min(0.98 \times \sigma, \max(\text{Indicator(Idx)})$
end loop

Here $0 < p < 1$, λ is a positive penalty parameter, and $\|x\|_p^p = \sum_i |x_i|^p$. To attempt to solve (2.15) the algorithm starts from an initial guess x^1 and at each iteration k, it defines the following weights:

$$\mathrm{w}_i^k = \frac{p}{(|x_i^k| + \sigma)^{1-p}}, \tag{2.16}$$

and solves the following ℓ_1 minimization:

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^N} \ \|Ax - b\|_2^2 + \lambda \|x\|_{1, \mathrm{w}^k}. \tag{2.17}$$

2.3.4 Iterative reweighted least squares

Using iterative reweighted least squares (IRLS) to solve the ℓ_p minimization problem has been studied in [31] and [29]. We have implemented the algorithm proposed in [29]. At each iteration k, of this method we solve the following least squares problem:

$$\min_{u \in \mathbb{R}^N} \ \sum_{i=1}^{i=N} \mathrm{w}_i u_i^2 \ \text{ subject to } \ Au = b, \tag{2.18}$$

where $\mathrm{w}_i = ((u_i^{(k-1)})^2 + \sigma)^{\frac{p}{2}-1}$ and σ is a regularization parameter which takes care of the problem of having $u_i = 0$. As stated in [29], at each iteration k, we can solve (2.18) explicitly by solving the Euler-Lagrange equation of (2.18) and using the constraint to solve for the

Lagrange multipliers. This yields

$$u^{(k)} = Q_k A^T (A Q_k A^T)^{-1} b, \tag{2.19}$$

where $Q_k = diag(\frac{1}{w_i})$ for w as defined above. The solution at each iteration is used to generate the weights for the next iteration. After some iterations we get close to the optimal solution and then we make σ smaller until we get to the desired "convergence level", i.e., when the norm of the difference between consequent recovered signals is smaller than a predefined threshold. Our implementation of the weighted versions of these three algorithms show that the the projected gradient method is more compatible with the way we do weighting and the performance of the modified projected gradient method is better than the modified versions of the other two. Therefore, our numerical results, presented in the next section, has been produced by using modified projected gradient method described in Algorithm 1.

2.4 Numerical examples

In this section, we provide numerical results to show how $\triangle_{p,w}$ improves the recovery conditions of sparse and approximately sparse signals compared to $\triangle_p$ and $\triangle_{1,w}$. We show the results for sparse signals and compressible signals. In these examples, we use the projected gradient algorithm that is described in Section 2.3.2 to solve the weighted ℓ_p minimization problem (2.8).

2.4.1 The sparse case

In this section, we generate signals $x \in \mathbb{R}^N$ where $N = 500$ and with fixed sparsity $k = 40$. We compute the (noisy) compressed measurements of x using a Gaussian random measurement matrix A with dimensions $n \times N$ where n varies between 80 and 200 with an increment of 20. In the case of noisy measurements, we have assumed 5% noise, i.e., signal-to-noise ratio of 26 db.

Figure 2.4 shows the reconstruction signal-to-noise ratio (SNR) averaged over 10 experiments, using weighted ℓ_p and weighted ℓ_1 minimization, versus the number of measurements, i.e., n. In Figures 2.4.a–c, we show result in the noise-free case when $\alpha = 0.3,\ 0.5,\ 0.7$ respectively. In figures 2.4.d–f we repeat the above experiment in the noisy case. In all the experiments, $p = 0.5$ and the recovery was done with the projected gradient algorithm described in Section

2.3.2. Here the SNR is measured in db and is given by

$$\mathrm{SNR}(x, \hat{x}) = 10 \log_{10}\left(\frac{\|x\|_2^2}{\|x - \hat{x}\|_2^2}\right). \tag{2.20}$$

Recall that when $\omega = 1$ weighted ℓ_p is equivalent to regular ℓ_p and weighted ℓ_1 is equivalent to regular ℓ_1. Figures 2.4.a–c illustrate that in the noise-free case, the experimental results are consistent with the theoretical results derived in Theorem 2.12. More precisely when $\alpha > 0.5$, the best recovery is achieved when the weights are set equal to zero and as our estimate gets worse (α decreases), the best recovery is achieved when larger weights are used. Also we observe that weighted ℓ_p recovers significantly better than weighted ℓ_1, especially when we have few measurements, which is consistent with our analysis in Section 2.2.

Remark 2.21. In Figures 2.1 and 2.3 we can see that when $\alpha < 0.5$ both the sufficient recovery conditions and error bound constants point towards using $\omega = 1$ for better recovery. However, Figure 2.4 shows that this is not always true. We attribute this behavior to the best k-term approximation term in the error bound of Theorem 2.12. Consider the noise-free case where the error bound becomes:

$$\| \triangle_{p,\mathrm{w}}(A, y, 0, \mathrm{w}) - x\|_2^p \leq C_2 k^{\frac{p}{2}-1}(\omega^p \|x - x_k\|_p^p + (1 - \omega^p)\|x_{\widetilde{T}^c \cap T_0^c}\|_p^p).$$

As we can we can see in Figure 2.3, C_2 decreases when we use bigger weights. Notice that on T_0^c, $x_k = 0$ so we have $\|x_{\widetilde{T}^c \cap T_0^c}\| = \|(x - x_k)_{\widetilde{T}^c \cap T_0^c}\|$ which means that $\|x_{\widetilde{T}^c \cap T_0^c}\|_p^p$ is always less than $\|x - x_k\|_p^p$. When we use bigger weights $\|x - x_k\|$ would have more impact on the error which results in bigger error when we use bigger weights. Hence using bigger weights decreases the constant C_2 and increases the factor $\omega^p \|x - x_k\|_p^p + (1 - \omega^p)\|x_{\widetilde{T}^c \cap T_0^c}\|_p^p$. Consequently when the algorithm cannot recover the full support of x, i.e., when $\|x - x_k\| > 0$, an intermediate value of ω in $[0, 1]$ may result in the smallest recovery error. A full mathematical analysis of this observation needs to consider all the parameters in Theorem 2.12 including $\omega,\ k,\ \alpha$ which is beyond the scope of this thesis.

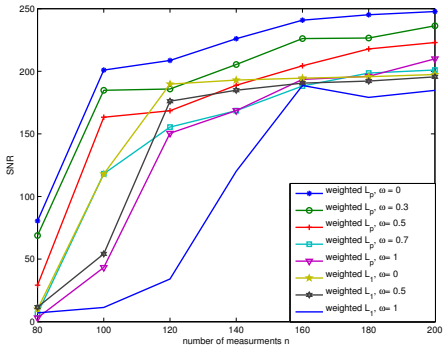
Figures 2.4.d–f show results for the noisy case. As we can see using intermediate weights results in best recovery and we can see that weighted ℓ_p is outperforming weighted ℓ_1 especially when the number of measurements is small. It is also evident that when we have a better support estimate, we reach the best possible recovery with fewer number of measurements n. Next, we investigate the effect of the choice of weights and p on the approximation error, when we have support estimates with various accuracy levels. Figure 2.5 shows the average SNR

over 10 experiments with $\rho = 1$ and $n = 100$ for different values of p, ω, and α when we have 5% noise and when we have no noise. Figures 2.5.a–c show the no-noise case. As we can see using smaller p is beneficial when we do not have a good support estimate. However, Notice that using smaller p results in slower recovery and dealing with the non-convexity of ℓ_p norm is more challenging when p is small. It is also worth noting that we again observe better performance with intermediate weights when $\alpha < 0.5$.
Figures 2.5.d, 2.5.e, and 2.5.f show the recovery for different values of p when we have 5% noise. As we observe generally using smaller p results in better recovery and again intermediate weights are recovering better. One important thing to observe here is how using smaller p improves the results when we do not have a good support estimate, which is a result of weaker sufficient recovery conditions when p is smaller.
Figure 2.6 shows the averaged SNR for recovering a 40-sparse signal using weighted ℓ_p with different support estimate sizes. Figures 2.6.a–c show the results when we do not have any noise and Figures 2.6.d–f show the results when we have 5% noise. We can see that generally using weighted solvers is useful especially in the non-convex case where we get better results even when $\alpha = 0.3$. Note that having a larger support estimate size usually results in better recovery, but the results are more sensitive to the accuracy of the support estimate than its size. Another important point here is that when we are in the noisy case or when we do not have a good support estimate, using intermediate weights results in better recovery which is very important for applications. Also this figure emphasizes the benefit of using weighted ℓ_p when we do not have a good support estimate.

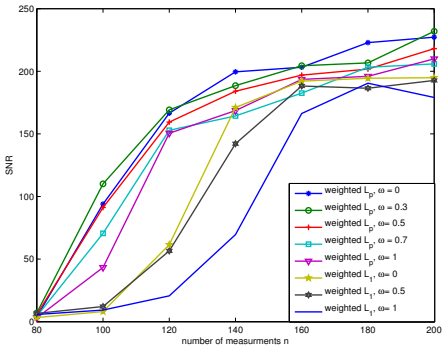
2.4.2 The compressible case

In this section we generate signals $x \in \mathbb{R}^N$, sorted coefficients of which decay like j^{-d} where $j \in \{1, 2, ..., N\}$ and $d > 1$. Figure 2.7 illustrates the results when $d = 1.1$. Here, we attempt to estimate the support of the best k-term approximation of the signal when $n = 100$. Accordingly, we find α with respect to the best 40-term approximation. Figures 2.7.a–c show the no-noise case where Figures 2.7.d–f have 5% noise. As we can see when we have a good support estimate, having a larger support estimate ends in better recovery and like the sparse case when we do not have a good support estimate, i.e., α<0.5, using intermediate weights results in better reconstruction. When α=0.7 using zero weights is giving us the best recovery. Generally in this figure we see that unlike the sparse case using weighted ℓ_p for recovering compressible signals does not give us much better results compared to weighted ℓ_1, specifically

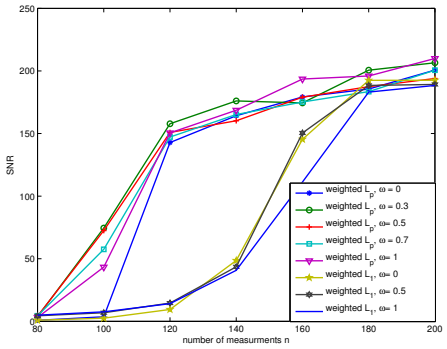
in Figure 2.7.d we see that weighted ℓ_1 with zero weight is recovering better than weighted ℓ_p. We believe that this is an artifact of the algorithm we are using. As we said before we do not have any proof for global convergence of the algorithm and the projected gradient algorithm handles the local minima by a smoothing parameter σ. In the noisy compressible case we have a large number of these local minima which may be a reason that in some of the compressible noisy cases we see that weighted ℓ_1 is recovering better than weighted ℓ_p. However, an exact explanation of why this happens is beyond the scope of this thesis.
Figure 2.8 shows the case when $d = 1.5$. Here the decay of coefficients of the signal is much faster so we have used fewer coefficients as our support. As we can see the results are similar to the case with $d = 1.1$ and generally when we have larger support estimate we get better recovery. Here we find α using the best 20-sparse approximation. Again in the noisy compressible case we see that weighted ℓ_1 is recovering better than weighted ℓ_p when we have a good support estimate.



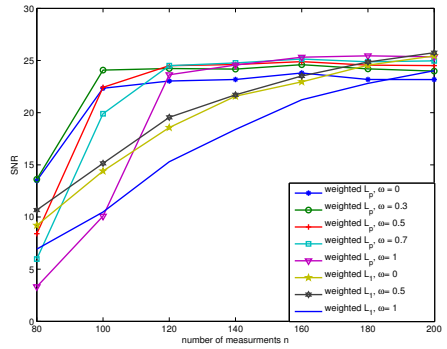
(a) $\alpha = 0.7$ with no noise



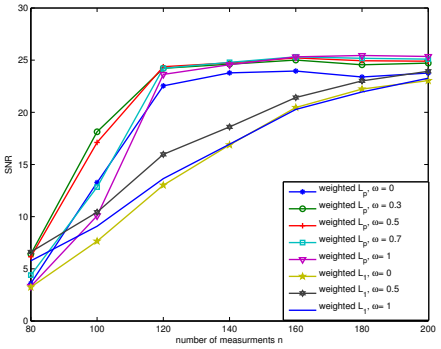
(b) $\alpha = 0.5$ with no noise



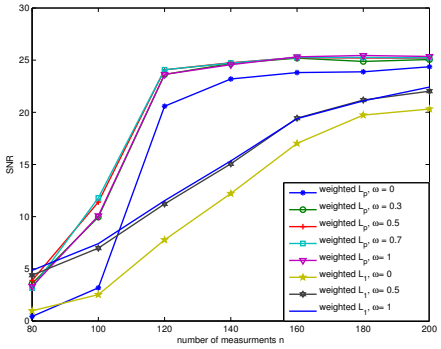
(c) $\alpha = 0.3$ with no noise



(d) $\alpha = 0.7$ with 5% noise

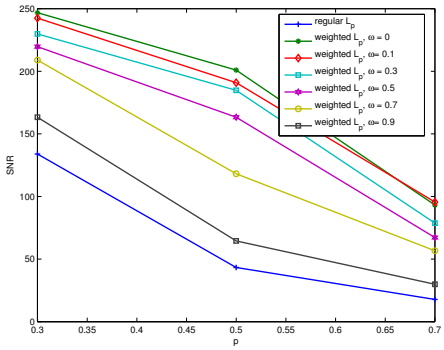


(e) $\alpha = 0.5$ with 5% noise

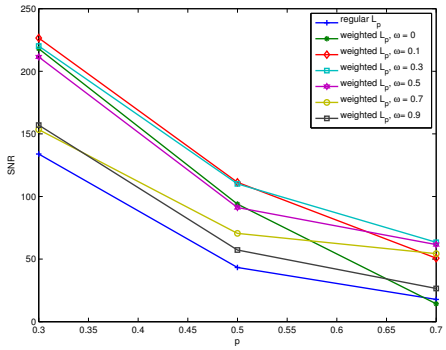


(f) $\alpha = 0.3$ with 5% noise

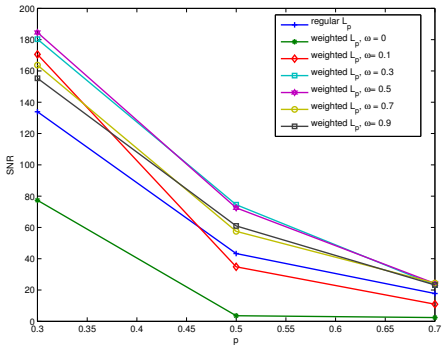
Figure 2.4: Comparison of performance of weighted ℓ_p and weighted ℓ_1 recovery in terms of SNR averaged over 10 experiments for sparse signals with variable weights and measurements and $\rho = 1$ and $p = 0.5$.



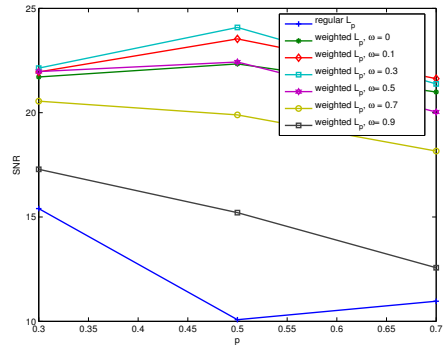
(a) $\alpha = 0.7$ with no noise



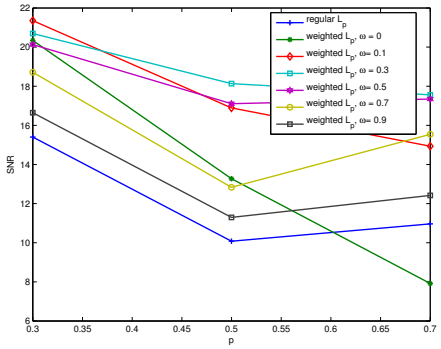
(b) $\alpha = 0.5$ with no noise



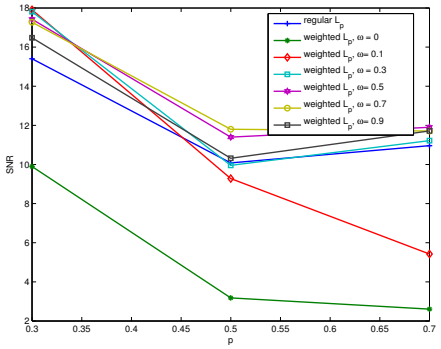
(c) $\alpha = 0.3$ with no noise



(d) $\alpha = 0.7$ with 5% noise

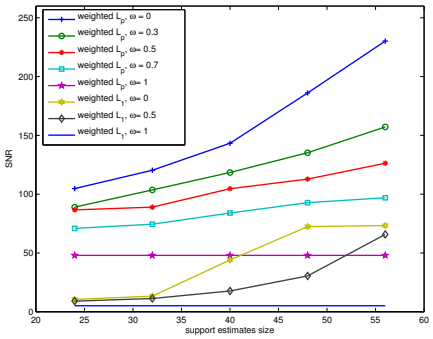


(e) $\alpha = 0.5$ with 5% noise

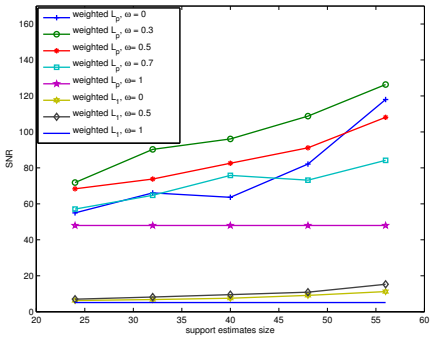


(f) $\alpha = 0.3$ with 5% noise

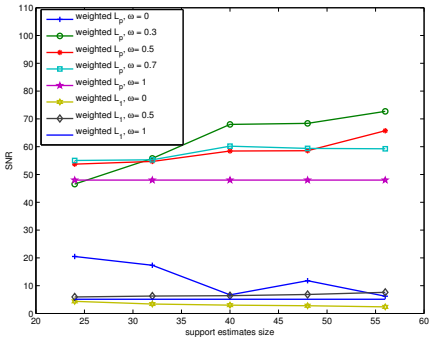
Figure 2.5: Comparison of SNR for variable weights and p for $\rho = 1$, $k = 40$ and $n = 100$.



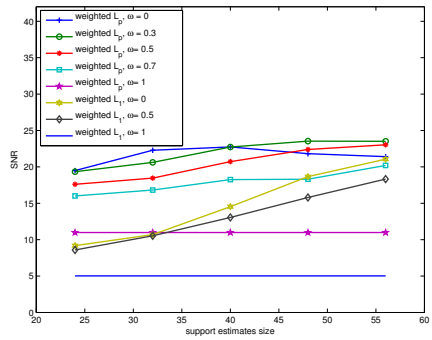
(a) $\alpha = 0.7$ with no noise



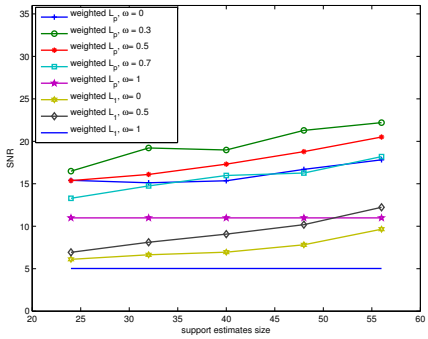
(b) $\alpha = 0.5$ with no noise



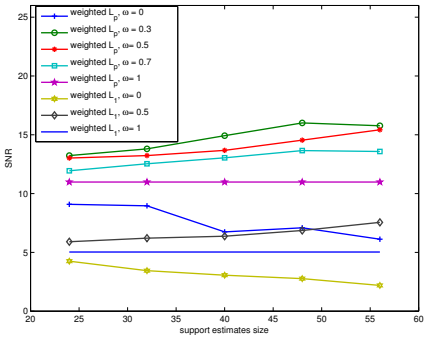
(c) $\alpha = 0.3$ with no noise



(d) $\alpha = 0.7$ with 5% noise

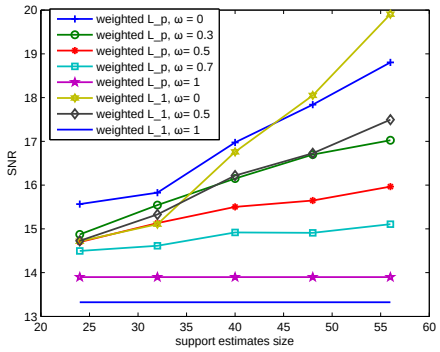


(e) $\alpha = 0.5$ with 5% noise

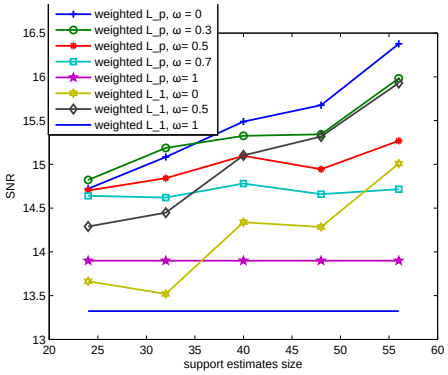


(f) $\alpha = 0.3$ with 5% noise

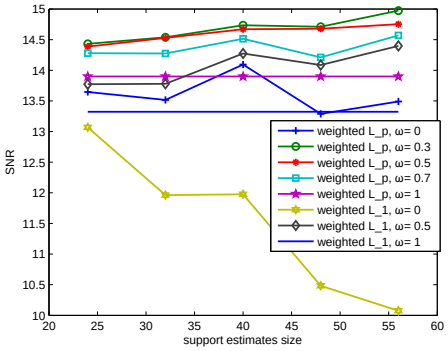
Figure 2.6: Comparison of performance of weighted ℓ_p and weighted ℓ_1 recovery in terms of SNR averaged over 20 experiments for sparse signals x with $n = 100, N = 500$ with variable support size and variable α and ω



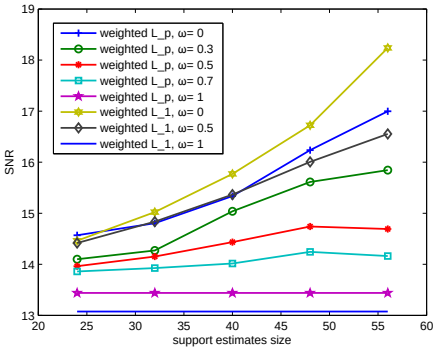
(a) $\alpha = 0.7$ with no noise



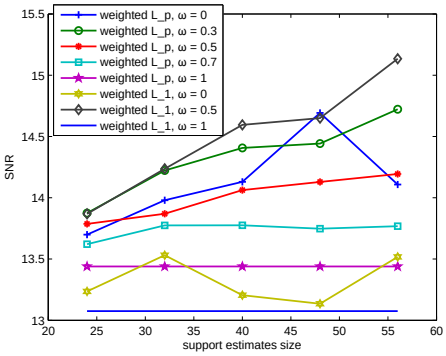
(b) $\alpha = 0.5$ with no noise



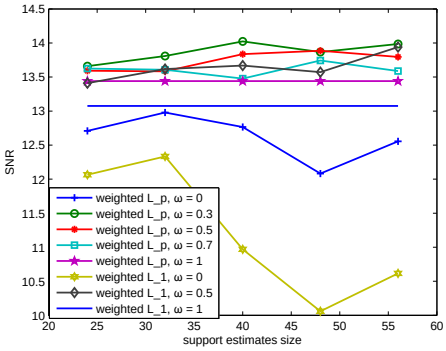
(c) $\alpha = 0.3$ with no noise



(d) $\alpha = 0.7$ with 5% noise

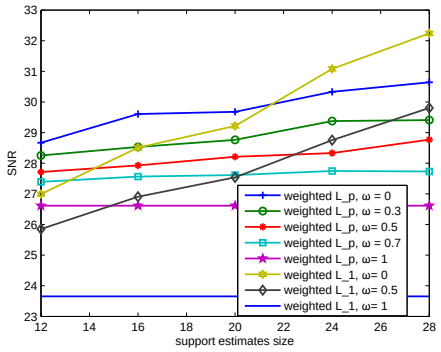


(e) $\alpha = 0.5$ with 5% noise

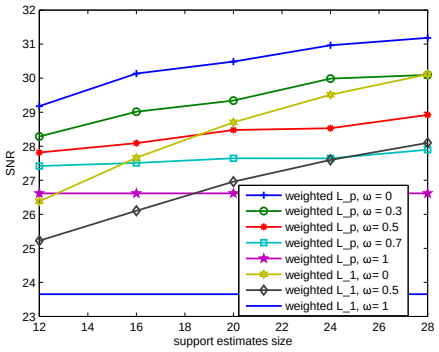


(f) $\alpha = 0.3$ with 5% noise

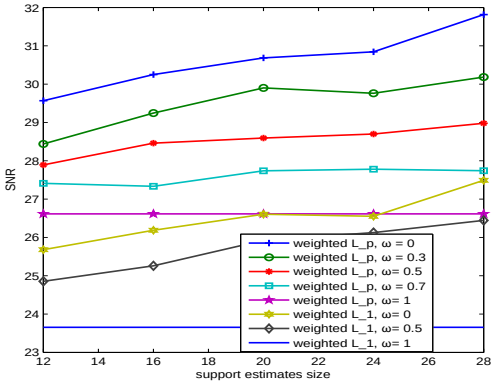
Figure 2.7: Comparison of performance of weighted ℓ_p and weighted ℓ_1 recovery in terms of SNR averaged over 20 experiments for compressible signals x with $n = 100, N = 500$. The coefficients decay with a power $d = 1.1$. The accuracy of the support estimate α is calculated with respect to the best $k = 40$ term approximation.



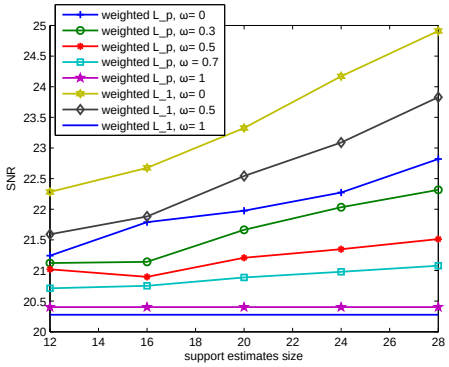
(a) $\alpha = 0.7$ with no noise



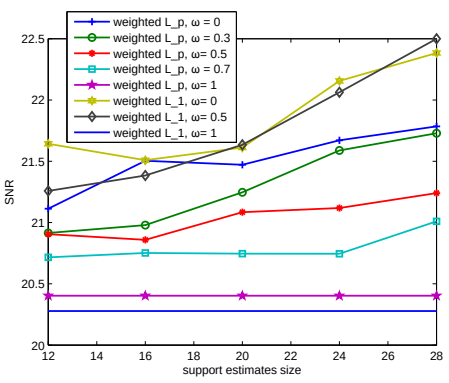
(b) $\alpha = 0.5$ with no noise



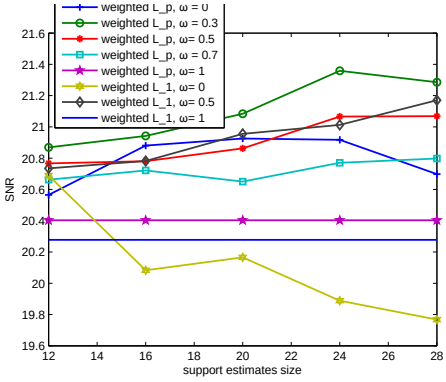
(c) $\alpha = 0.3$ with no noise



(d) $\alpha = 0.7$ with 5% noise



(e) $\alpha = 0.5$ with 5% noise



(f) $\alpha = 0.3$ with 5% noise

Figure 2.8: Comparison of performance of weighted ℓ_p and weighted ℓ_1 recovery in terms of SNR averaged over 20 experiments for compressible signals x with $n = 100, N = 500$. The coefficients decay with a power $d = 1.5$. The accuracy of the support estimate α is calculated with respect to the best $k = 20$ term approximation.

2.5 Stylized application

In this section, we apply standard and weighted ℓ_p minimization to recover real audio and seismic signals that are compressively sampled.

2.5.1 Audio signals

In this section we examine the performance of weighted ℓ_p minimization for the recovery of compressed sensing measurements of speech signals. Here the speech signals are sampled at 44.1 kHz and we randomly choose only $\frac{1}{4}$th of the samples (their indices chosen randomly from uniform distribution). Assuming that s is the speech signal we have the measurements $y = Rs$ where R is a restriction operator. We divide our measurements y into 21 blocks, i.e., $y = [y_1^T,\ y_2^T,\ \ldots,\ y_{21}^T]^T$. Assuming the speech signal to be compressible in DCT domain (for example a version of standard DCT is used to compress audio signals in standard MP3), we try to recover the speech signal using each block measurement.
This helps us in two ways:

- It reduces the size of the problem
- Considering the fact that the support set corresponding to the largest coefficients does not change much from one block to another, we can use the indices of the largest coefficients of each block as a support estimate for the next one.

So for each block, we find the speech signal by solving $y_j = R_j s_j$, where $R_j \in R^{n_j \times N}$ is the associated restriction matrix. We also know that speech signals have large low-frequency coefficients, so we use this fact and the recovered signal at previous block to build our support estimate and find the speech signal at each block by weighted ℓ_p. We choose the support estimate to be $\tilde{T} = \tilde{T}^1 \cup \tilde{T}^2$. Here $\tilde{T}^1$ is the set corresponding to frequencies up to 4 kHz and $\tilde{T}^2$ is the set corresponding to the largest $\frac{n_j}{16}$ recovered coefficients of the previous block (for the first block $\tilde{T}^2$ is empty). The results of using weighted ℓ_p and weighted ℓ_1 for reconstruction of two audio signals (one male and one female) are illustrated in Figure 2.9. Here $N = 2048$, and $\omega \in \{0,\ \frac{1}{6},\ \frac{2}{6},\ \ldots,\ 1\}$. Weighted ℓ_p gives about 1 db improvement in reconstruction.

2.5.2 Seismic signals

The problem of interpolating irregularly sampled and incomplete seismic data to a regular periodic grid often occurs in 2D and 3D seismic settings [4].

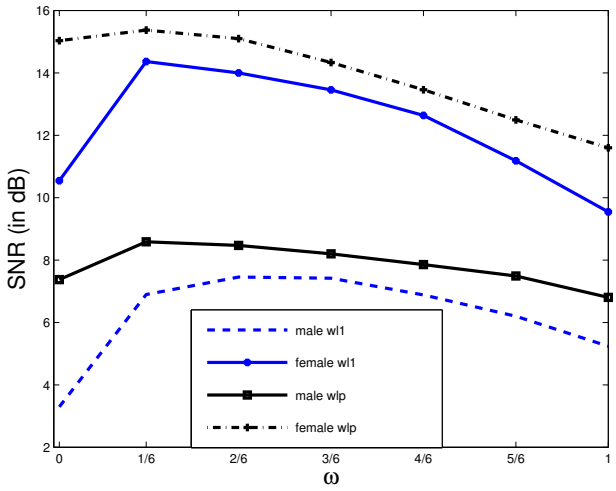


Figure 2.9: SNRs of reconstructed signal from compressed sensing measurements plotted against ω. An intermediate value of ω yields the best performance.

Our setting is as explained in Section 1.3.1. We have a seismic line with N_s sources, N_r receivers, and N_t time samples. We are dealing with a signal $f \in \mathbb{R}^N$, where $N = N_s N_r N_t$. We want to recover a very high dimensional seismic data volume $f = S^* x$ by interpolating between a smaller number of measurements $b = RMS^* x$, where R is a restriction matrix, M represents the basis in which the measurements are taken, and S is the 2D curvelet transform. Seismic data is approximately sparse in curvelet domain and hence we can formulate the seismic data interpolation problem as an instance of recovery from compressive samples [22, 23]. We partition the seismic data volume into frequency slices and approximate $x^{(1)}$ by $\triangle_p(R^{(1)}MS^*, b^{(1)}, \epsilon)$ where ϵ is a small number (estimate of the noise level) and $R^{(1)}$ is the subsampling operator restricted to the first partition and $b^{(1)}$ is the subsampled measurements of the data $f^{(1)}$ in the first partition. After this we use the support of each recovered partition as a support estimate for next partition. In particular for $j \geq 1$ we approximate $x^{(j+1)}$ by $\triangle_{p,\mathrm{w}}(R^{(j)}MS^H, b^{(j)}, \epsilon, \mathrm{w})$ where w is the weight vector which puts smaller weights on the coefficients that correspond to the support of the previous recovered partition. In [4] the performance of weighted ℓ_1 minimization has been tested for recovering a seismic line using 50% randomly subsampled receivers. Exploiting the ideas in [4] we test the weighted ℓ_p minimization algorithm to recover a test seismic problem when we subsample 50% of the the receivers using the mask shown in Figure 2.10.b. We omit the details of this algorithm as it mimics the steps taken in [4] when weighted ℓ_1 is replaced by weighted ℓ_p.

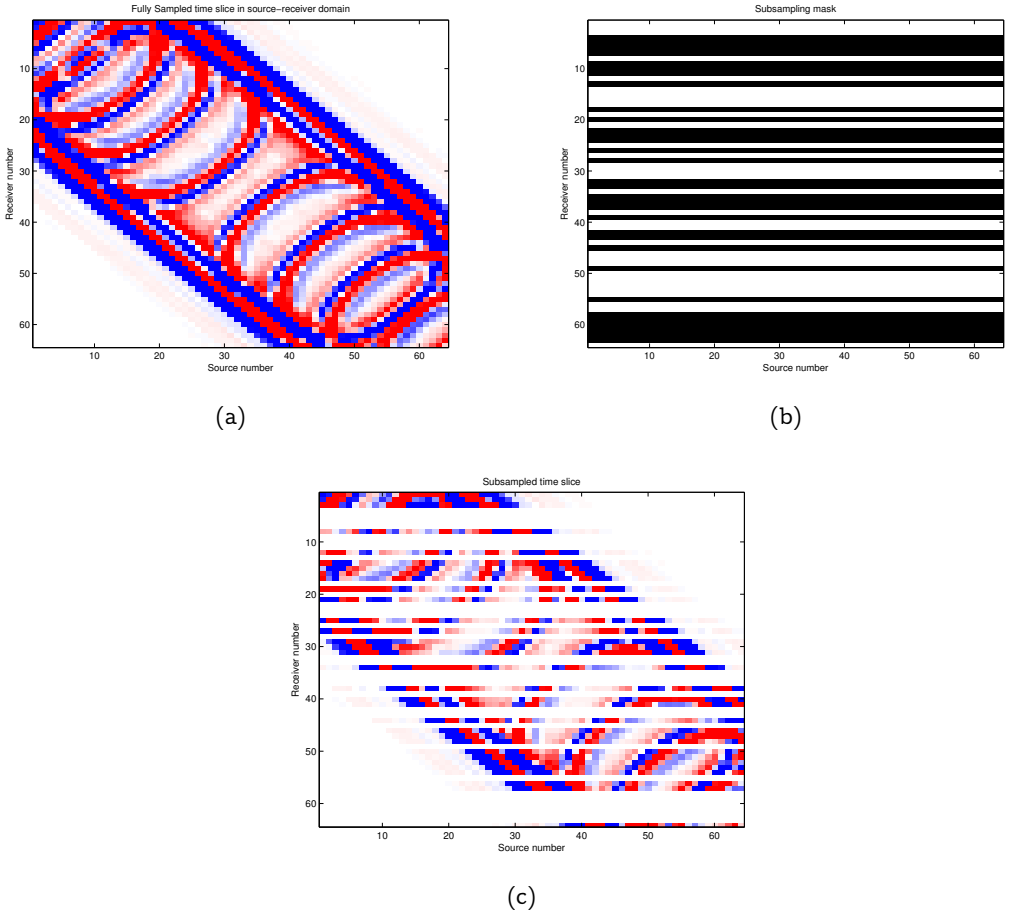
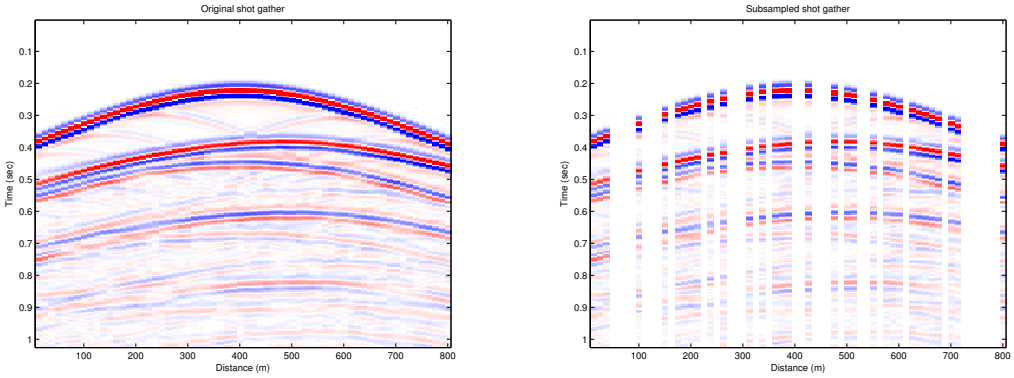


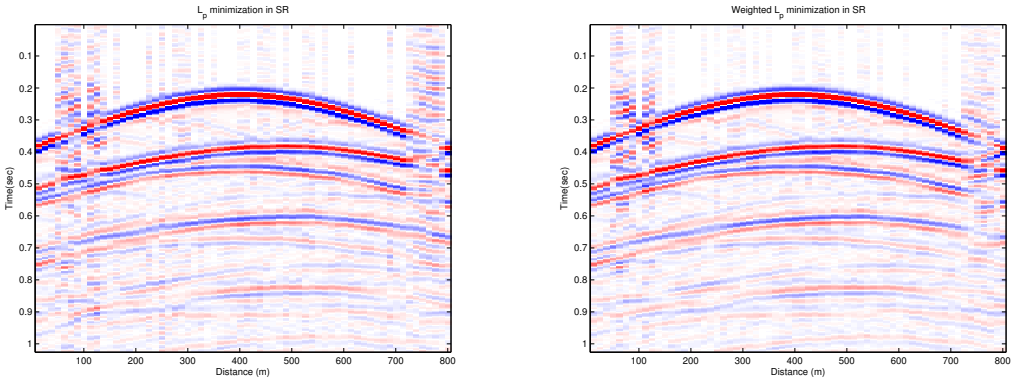
Figure 2.10: (a) Example of a high resolution time slice at $t = 0.32$ s in the source-receiver (SR) domain, (b) the random subsampling mask where the black lines correspond to the locations of inactive receivers, and (c) the subsampled time slice. The subsampling ratio is 50%.

The Seismic line at full resolution has $N_s = 64$ sources, $N_r = 64$ receivers with a sample distance of 12.5 meters, and $N_t = 256$ time samples acquired with a sampling interval of 4 milliseconds. Consequently, the seismic line contains samples collected in a 1s temporal window with a maximum frequency of 125 Hz. To access frequency slices, we take the one dimensional discrete Fourier transform (DFT) of the data along the time axis. We solve the ℓ_p and weighted ℓ_p minimization problems. In each of the weighted ℓ_p problems, the support estimate set of a partition is derived from the analysis coefficients of the previously recovered partition. Moreover, p is set to be 0.5 and the weight is set to 0.3.



(a) (b)

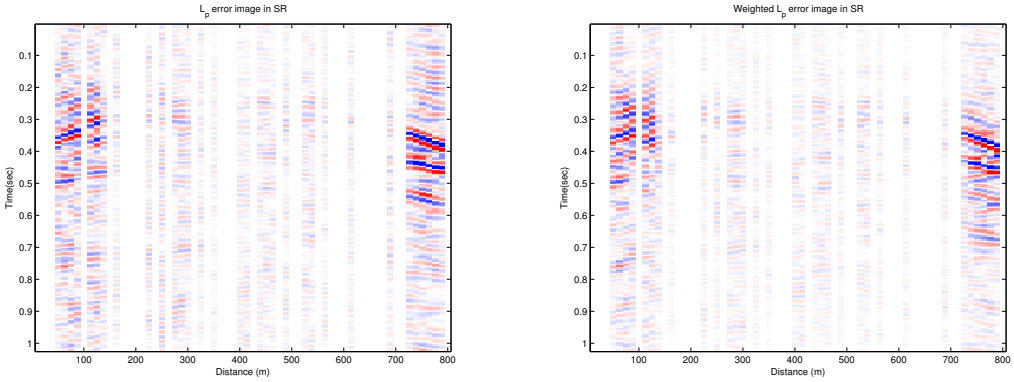
Figure 2.11: (a) Shot gather number 32 from a seismic line from the Gulf of Suez. (b) Subsampled shot gather using column 32 from the mask in Figure 2.10.b.



(a) (b)

Figure 2.12: (a) Recovered shot gather using ℓ_p minimization in the SR domain. (b) Recovered shot gather using weighted ℓ_p minimization in the SR domain.

Figures 2.11.a and 2.11.b show a fully sampled and the corresponding subsampled shot gather, respectively. The shot gather corresponds to shot number 32 of the seismic line. Figures 2.12.a and 2.12.b show the reconstructed shot gathers using ℓ_p minimization and weighted ℓ_p minimization, respectively. The error plots of both reconstructions are shown in Figures 2.13.a and 2.13.b. The error plots show that the magnitude of the reconstruction error of weighted ℓ_p minimization is smaller than that of standard ℓ_p. Figure 2.14 shows the SNRs of all shot gathers recovered by using regular and weighted ℓ_p and ℓ_1 minimization problems.



(a) (b)

Figure 2.13: (a) Error plots showing the difference between the original shot gather and the reconstruction from ℓ_p minimization in the source-receiver domain. (b) Error plots showing the difference between the original shot gather and the reconstruction from weighted ℓ_p minimization in the SR domain.

The plots demonstrate that recovery by weighted ℓ_p in the frequency-source-receiver domain is always better than recovery by regular ℓ_p. In this plot we also see that although recovery by weighted ℓ_p minimization is better than regular ℓ_1 minimization but recovery by weighted ℓ_1 minimization is still about 1 db better than recovery by weighted ℓ_p minimization. We believe that similar to the case we see in the noisy compressible case this is an artifact of the algorithm we are using.

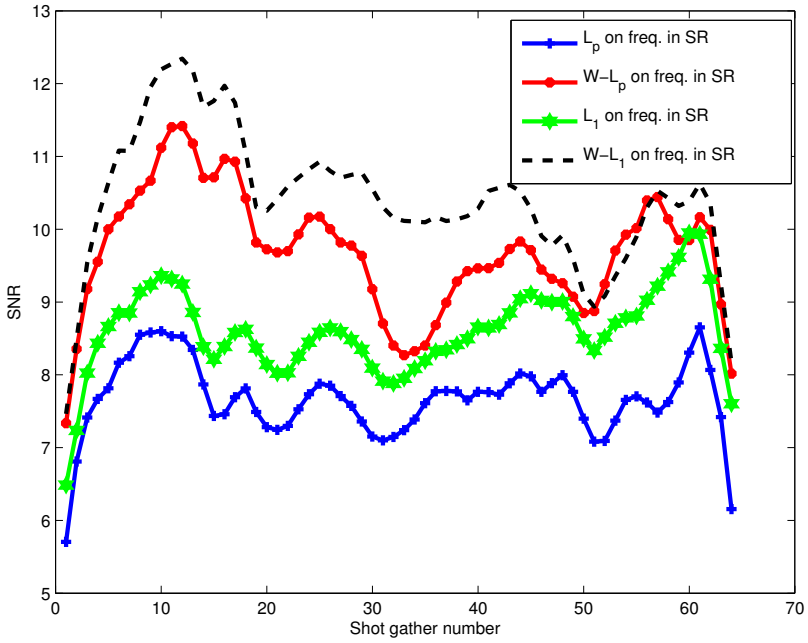


Figure 2.14: Comparison of the SNRs achieved by ℓ_1, ℓ_p, weighted ℓ_1, and weighted ℓ_p minimization in recovering shot gathers applied to source-receiver domain

2.6 Proof of Theorem 2.12

Recall that $\tilde{T}$, an arbitrary subset of $\{1, 2, ..., N\}$, is of size ρk where $0 \leq \rho \leq a$ and a is some number larger than 1. Let the set $\tilde{T}_\alpha = T_0 \cap \tilde{T}$ and $\tilde{T}_\beta = T_0^c \cap \tilde{T}$ where, $|\tilde{T}_\alpha| = \alpha\tilde{T} = \alpha\rho k$ and $\alpha + \beta = 1$.

Let $x^* = x + h$ be a minimizer of the weighted ℓ_p problem. Then:

$$\|x + h\|_{p,\mathrm{w}} \leq \|x\|_{p,\mathrm{w}} \Rightarrow \|x + h\|_{p,\mathrm{w}}^p \leq \|x\|_{p,\mathrm{w}}^p.$$

Using the weights we have:

$$\omega^p \|x_{\tilde{T}} + h_{\tilde{T}}\|_p^p + \|x_{\tilde{T}^c} + h_{\tilde{T}^c}\|_p^p \leq \omega^p \|x_{\tilde{T}}\|_p^p + \|x_{\tilde{T}^c}\|_p^p.$$

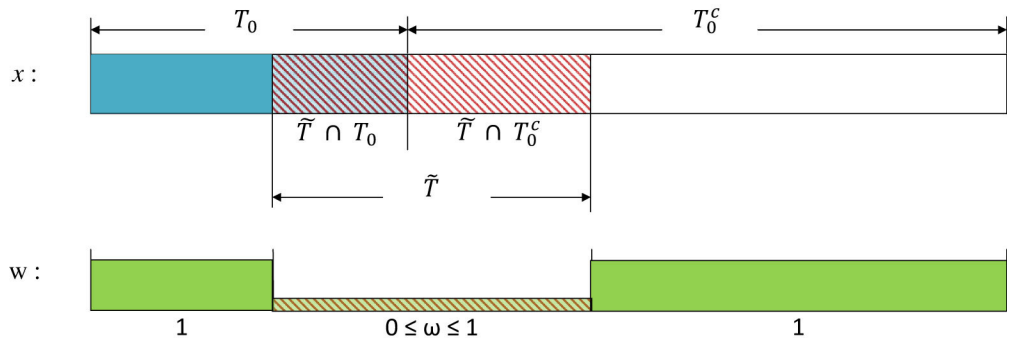


Figure 2.15: Illustration of the signal x and weight vector w emphasizing the relationship between the sets T_0 and $\tilde{T}$.

Consequently,

$$\omega^p \|x_{\tilde{T}\cap T_0} + h_{\tilde{T}\cap T_0}\|_p^p + \omega^p \|x_{\tilde{T}\cap T_0^c} + h_{\tilde{T}\cap T_0^c}\|_p^p + \|x_{\tilde{T}^c\cap T_0} + h_{\tilde{T}^c\cap T_0}\|_p^p + \|x_{\tilde{T}^c\cap T_0^c} + h_{\tilde{T}^c\cap T_0^c}\|_p^p$$

$$\le \omega^p \|x_{\tilde{T}\cap T_0}\|_p^p + \omega^p \|x_{\tilde{T}\cap T_0^c}\|_p^p + \|x_{\tilde{T}^c\cap T_0}\|_p^p + \|x_{\tilde{T}^c\cap T_0^c}\|_p^p.$$

We use the forward and reverse triangle inequalities to get:

$$\omega^p \|h_{\tilde{T}\cap T_0^c}\|_p^p + \|h_{\tilde{T}^c\cap T_0^c}\|_p^p \le \omega^p \|h_{\tilde{T}\cap T_0}\|_p^p + \|h_{\tilde{T}^c\cap T_0}\|_p^p + 2(\omega^p \|x_{\tilde{T}\cap T_0^c}\|_p^p + \|x_{\tilde{T}^c\cap T_0^c}\|_p^p).$$

Adding and subtracting $\omega^p \|h_{\tilde{T}^c\cap T_0^c}\|_p^p$ to the left hand side and adding and subtracting $\omega^p \|h_{\tilde{T}^c\cap T_0}\|_p^p + \omega^p \|x_{\tilde{T}^c\cap T_0^c}\|_p^p$ to the right hand side we get:

$$\omega^p \|h_{\tilde{T}\cap T_0^c}\|_p^p + \omega^p \|h_{\tilde{T}^c\cap T_0^c}\|_p^p + \|h_{\tilde{T}^c\cap T_0^c}\|_p^p - \omega^p \|h_{\tilde{T}^c\cap T_0^c}\|_p^p$$

$$\le \omega^p \|h_{\tilde{T}\cap T_0}\|_p^p + \omega^p \|h_{\tilde{T}^c\cap T_0}\|_p^p + \|h_{\tilde{T}^c\cap T_0}\|_p^p - \omega^p \|h_{\tilde{T}^c\cap T_0}\|_p^p$$

$$+2(\omega^p \|x_{\tilde{T}\cap T_0^c}\|_p^p + \omega^p \|x_{\tilde{T}^c\cap T_0^c}\|_p^p + \|x_{\tilde{T}^c\cap T_0^c}\|_p^p - \omega^p \|x_{\tilde{T}^c\cap T_0^c}\|_p^p).$$

Since $\|h_{T_o^c}\|_p^p = \|h_{\tilde{T}\cap T_0^c}\|_p^p + \|h_{\tilde{T}^c\cap T_0^c}\|_p^p$ we get:

$$\omega^p \|h_{T_0^c}\|_p^p + (1-\omega^p)\|h_{\tilde{T}^c\cap T_0^c}\|_p^p \le \omega^p \|h_{T_0}\| + 2(\omega^p \|x_{T_0^c}\|_p^p + (1-\omega^p)\|x_{\tilde{T}^c\cap T_0^c}\|_p^p). \tag{2.21}$$

We also have $\|h_{T_o^c}\|_p^p = \omega^p\|h_{T_o^c}\|_p^p + (1-\omega^p)\|h_{\widetilde{T}\cap T_0^c}\|_p^p + (1-\omega^p)\|h_{\widetilde{T}^c\cap T_0^c}\|_p^p$. Combining this with (2.21) we get:

$$\begin{aligned}\|h_{T_0^c}\|_p^p \leq \omega^p\|h_{T_0}\|_p^p + (1-\omega^p)(\|h_{\widetilde{T}^c\cap T_0}\|_p^p + \|h_{\widetilde{T}\cap T_0^c}\|_p^p)\\ +2(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)(\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p).\end{aligned} \tag{2.22}$$

$$\widetilde{T}_\alpha = T_0 \cap \widetilde{T} \Rightarrow \|h_{\widetilde{T}^c\cap T_0}\|_p^p + \|h_{\widetilde{T}\cap T_0^c}\|_p^p = \|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_p^p.$$

$$\|h_{T_0^c}\|_p^p \leq \omega^p\|h_{T_0}\|_p^p + (1-\omega^p)\|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_p^p + 2(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)(\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p). \tag{2.23}$$

Now partition T_0^c into sets of $T_1, T_2, ..., |T_j| = ak$ for $j \geq 1$, such that T_1 is the set of indices of the ak largest (in magnitude) coefficients of $h_{T_0^c}$ and so on. Finally let $T_{01} := T_0 \cup T_1$. Now we can find a lower bound for $\|Ah\|_2^p$ using the RIP constants of the matrix A. We have:

$$\begin{aligned}\|Ah\|_2^p = \|Ah_{T_{01}} + \sum_{j\geq 2} Ah_{T_j}\|_2^p \geq \|Ah_{T_{01}}\|_2^p - \sum_{j\geq 2}\|Ah_{T_j}\|_2^p\\ \geq (1-\delta_{ak+|T_0|})^{\frac{p}{2}}\|h_{T_{01}}\|_2^p - (1+\delta_{ak})^{\frac{p}{2}}\sum_{j\geq 2}\|h_{T_j}\|_2^p.\end{aligned} \tag{2.24}$$

Here we also use the fact that $\|.\|_2^p$ satisfies the triangle inequality for $0 < p < 1$.
Now we should note that $|h_{T_{j+1}}(l)|^p \leq |h_{T_j}(l')|^p$ for all $l \in T_{j+1}$ and $l' \in T_j$, and thus $|h_{T_{j+1}}(l)|^p \leq \frac{\|h_{T_j}\|_p^p}{ak}$. It follows that $\|h_{T_j}\|_2^2 \leq (ak)^{1-\frac{2}{p}}\|h_{T_j}\|_p^2$ and consequently :

$$\sum_{j\geq 2}\|h_{T_j}\|_2^p \leq (ak)^{\frac{p}{2}-1}\sum_{j\geq 1}\|h_{T_j}\|_p^p = (ak)^{\frac{p}{2}-1}\|h_{T_0^c}\|_p^p. \tag{2.25}$$

Using (2.25)in (2.24) we get:

$$\|Ah\|_2^p \geq (1-\delta_{ak+|T_0|})^{\frac{p}{2}}\|h_{T_{01}}\|_2^p - (1+\delta_{ak})^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\|h_{T_0^c}\|_p^p. \tag{2.26}$$

Next, consider the feasibility of x^* and x. Both vectors are feasible, so we have $\|Ah\|_2 \leq 2\varepsilon$. Also note that $|T_0 \cup \widetilde{T} \setminus \widetilde{T}_\alpha| = (1+\rho-2\alpha\rho)k$ and $\|h_{T_0}\|_p^p \leq |T_0|^{1-\frac{p}{2}}\|h_{T_0}\|_2^p$. Using these and

(2.23) in (2.26) we get:

$$(1-\delta_{ak+|T_0|})^{\frac{p}{2}}\|h_{T_{01}}\|_2^p \le (2\varepsilon)^p + 2(1+\delta_{ak})^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p\right)$$
$$+(1+\delta_{ak})^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p|T_0|^{1-\frac{p}{2}}\|h_{T_0}\|_2^p + (1-\omega^p)\left((1+\rho-2\alpha\rho)k\right)^{1-\frac{p}{2}}\|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_2^p\right). \tag{2.27}$$

T_1 contains the largest ak coefficients of $h_{T_0^c}$ with $a>1$ so $|\widetilde{T}\setminus\widetilde{T}_\alpha| = (1-\alpha)\rho k \le ak$ then $\|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_2 \le \|h_{T_{01}}\|_2$, also we have $\|h_{T_0}\|_2 \le \|h_{T_{01}}\|_2$ so we get:

$$\|h_{T_{01}}\|_2^p \le \frac{(2\varepsilon)^p + 2(1+\delta_{ak})^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p\right)}{(1-\delta_{ak+|T_0|})^{\frac{p}{2}} - (1+\delta_{ak})^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p|T_0|^{1-\frac{p}{2}} + (1-\omega^p)\left((1+\rho-2\alpha\rho)k\right)^{1-\frac{p}{2}}\right)}. \tag{2.28}$$

To complete the proof denote by $h_{T_0^c}[m]$ the m-th largest coefficient of $h_{T_0^c}$ and observe that $|h_{T_0^c}[m]|^p \le \frac{\|h_{T_0^c}\|_p^p}{m}$. As $h_{T_{01}^c}[m] = h_{T_0^c}[m+ak]$ we have:

$$\|h_{T_{01}^c}\|_2^2 = \sum_{m\ge ak+1}|h_{T_0^c}[m]|^2 \le \sum_{m\ge ak+1}\left(\frac{\|h_{T_0^c}\|_p^p}{m}\right)^{\frac{2}{p}} \le \frac{\|h_{T_0^c}\|_p^2}{(ak)^{\frac{2}{p}-1}(\frac{2}{p}-1)}. \tag{2.29}$$

Here the last inequality follows because for $0<p<1$:

$$\sum_{m\ge ak+1} m^{-\frac{2}{p}} \le \int_{ak}^{\infty} t^{-\frac{2}{p}}dt = \frac{1}{(ak)^{\frac{2}{p}-1}(\frac{2}{p}-1)}.$$

Combining (2.29) with (2.23) we get:

$$\|h_{T_{01}^c}\|_2^p \le \left((ak)^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{-\frac{p}{2}} *$$
$$\left(\omega^p\|h_{T_0}\|_p^p + (1-\omega^p)\|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_p^p + 2\left(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)(\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p)\right)\right). \tag{2.30}$$

We showed that $\|h_{T_0\cup\widetilde{T}\setminus\widetilde{T}_\alpha}\|_2 \le \|h_{T_{01}}\|_2$ and $\|h_{T_0}\|_2 \le \|h_{T_{01}}\|_2$.
Using these in (2.30) we get:

$$\|h_{T_{01}^c}\|_2^p \le \left((ak)^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{-\frac{p}{2}} *$$
$$\left(\left(\omega^p|T_0|^{1-\frac{p}{2}} + (1-\omega^p)\left((1+\rho-2\alpha\rho)k\right)^{1-\frac{p}{2}}\right)\|h_{T_{01}}\|_2^p + 2\left(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)(\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p)\right)\right). \tag{2.31}$$

Now we can find a bound for $\|h\|_2$ using (2.28) and (2.31):

$$\|h\|_2^2 = (\|h_{T_{01}}\|_2^p)^{\frac{2}{p}} + (\|h_{T_{01}^c}\|_2^p)^{\frac{2}{p}} \leq \left(\|h_{T_{01}}\|_2^p + \|h_{T_{01}^c}\|_2^p\right)^{\frac{2}{p}}. \tag{2.32}$$

$$\begin{aligned}\|h\|_2^p \leq & \frac{\left(1 + \frac{\left(\omega^p |T_0|^{1-\frac{p}{2}} + (1-\omega^p)((1+\rho-2\alpha\rho)k)^{1-\frac{p}{2}}\right)}{\left((ak)^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)(2\varepsilon)^p}{\left(1-\delta_{ak+|T_0|}\right)^{\frac{p}{2}} - \left(1+\delta_{ak}\right)^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p|T_0|^{1-\frac{p}{2}} + (1-\omega^p)\left((1+\rho-2\alpha\rho)k\right)^{1-\frac{p}{2}}\right)} \\ & + \frac{2\left(\left(1+\delta_a\right)^{\frac{p}{2}}a^{\frac{p}{2}-1} + \frac{\left(1-\delta_{(a+1)k}\right)^{\frac{p}{2}}}{\left(a^{\frac{2}{p}-1}(\frac{2}{p}-1)\right)^{\frac{p}{2}}}\right)\left(\omega^p\|x_{T_0^c}\|_p^p + (1-\omega^p)\|x_{\widetilde{T}^c\cap T_0^c}\|_p^p\right)}{\left(1-\delta_{ak+|T_0|}\right)^{\frac{p}{2}} - \left(1+\delta_{ak}\right)^{\frac{p}{2}}(ak)^{\frac{p}{2}-1}\left(\omega^p|T_0|^{1-\frac{p}{2}} + (1-\omega^p)\left((1+\rho-2\alpha\rho)k\right)^{1-\frac{p}{2}}\right)},\end{aligned} \tag{2.33}$$

with the condition that the denominator is positive, equivalently:

$$\delta_{ak} + \frac{a^{\frac{2}{p}-1}}{(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}}\delta_{(a+1)k} < \frac{a^{\frac{2}{p}-1}}{(\omega^p + (1-\omega^p)(1+\rho-2\alpha\rho)^{1-\frac{p}{2}})^{\frac{2}{p}}} - 1. \tag{2.34}$$

Chapter 3

Weighted AMP

3.1 Introduction

Significant effort has been made recently to find fast algorithms for recovering sparse signals from a small number of linear measurements. Our setting is as before: Let $x \in \mathbb{R}^N$ be a sparse vector. We try to recover x from $n < N$ linear and potentially noisy measurements acquired via $y = Ax + e$. Here A is an $n \times N$ matrix whose coefficients are drawn from a sub-Gaussian distribution and $\|e\| \leq \epsilon$. As mentioned before the BP problem (1.4) is perhaps the most common approach to recover x and can be solved by linear programming algorithms. Relatively high computational complexity of these algorithms has made them difficult to use in applications where the signals are very high dimensional. On the other hand, the low computational complexity of iterative algorithms has made them an appealing alternative for BP [18–20]. A general form of these algorithms is as

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t, \hat{\tau}^t) \\ z^t &= y - Ax^t, \end{aligned} \tag{3.1}$$

where as before x^t is the current estimate of x, z^t is the current residual, A^* is the Hermitian of the measurement matrix and η is a non-linear thresholding function which acts component-wise on its vector-valued argument. Two popular examples are the soft thresholding function $\eta(a;b) = sign(a)(a-b)_+$, and the hard thresholding function where $\eta_H(a;b) = a\mathbf{I}(|a| \geq b)_+$. More precisely these functions act on the coefficients of the signal a and zero out any value which is less than the scalar b in magnitude.

Recently an extensive numerical study [32] found that even under optimal tunings, the recovery conditions achieved by IT is worse than those of BP. As mentioned in the introduction Donoho, Maleki and Montanari proposed a new iterative thresholding algorithm which is referred to as approximate message passing [21] that was shown to enjoy both the low complexity of IT algorithms and the superior recovery conditions of BP. As stated also in Section 1.2.3, the AMP algorithm starts from an initial x^0 and an initial threshold $\hat{\tau}^0 = 1$ and iteratively goes

by

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \tau^t), \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \tau^{t-1}) \rangle. \end{aligned} \tag{3.2}$$

Notice that the only difference between AMP (3.2) and generic IT algorithm (3.1) is the extra term $\delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \tau^{t-1}) \rangle$ in the calculation of the residual. This term has been derived in [21] using the theory of belief propagation in graphical models and has been empirically shown to improve the recovery conditions. Statistical physicists call this term the Onsager reaction term [33].

In this section we design a weighted approximate message passing algorithm for recovering sparse signals when there exists prior information about the support of the signal. We build up the weighted AMP algorithm by following [34] step by step and empirically show that when the support estimate is accurate enough, weighting results in faster recovery and better sparsity-undersampling tradeoff. In particular we derive a "*weighted AMP*" algorithm to solve the weighted ℓ_1 minimization

$$\underset{s \in \mathbb{R}^N}{\text{argmin}} \quad ||s||_{1,\mathrm{w}} \quad \text{subject to} \quad y = As, \tag{3.3}$$

where $\mathrm{w} \in \{\omega, 1\}^N$ is the weight vector and $\|s\|_{1,\mathrm{w}} := \Sigma_i \mathrm{w}_i |s_i|$ is the weighted ℓ_1 norm. Note that in (3.3) we have restated the weighted ℓ_1 minimization (1.7) when $\epsilon = 0$. As before given a support estimate $\widetilde{T} \subseteq \{1, ..., N\}$, $\mathrm{w}_j = \omega < 1$ for $j \in \widetilde{T}$ and $\mathrm{w_j} = 1$ for $j \notin \widetilde{T}$.

3.2 Construction of the graphical model for weighted BP

Estimating marginals

Assume $[\mathrm{w}_1, \mathrm{w}_2, ..., \mathrm{w_N}]^{\mathrm{T}}$ are the weights we use for the coefficients of the signal s. Consider the following distribution over variables $s_1,\ s_2,\ \ldots,\ s_N$:

$$\mu(ds) = \frac{1}{Z} \prod_{i=1}^{N} exp(-\beta \mathrm{w}_i |s_i|) \prod_{a=1}^{n} \delta_{\{y_a = (As)_a\}}, \tag{3.4}$$

where $\delta_{\{y_a=(As)_a\}}$ denotes a Dirac distribution on the hyperplane $y_a = (As)_a$. As $\beta \to \infty$ the mass of this distribution concentrates around the solutions of $y = As$ which has more zero coefficients and hence the solution of (3.3)—the maximum of $exp(-|t|)$ is achieved when $t = 0$. If the solution of (3.3) is unique, then finding the marginals of (3.4) gives us the solution.

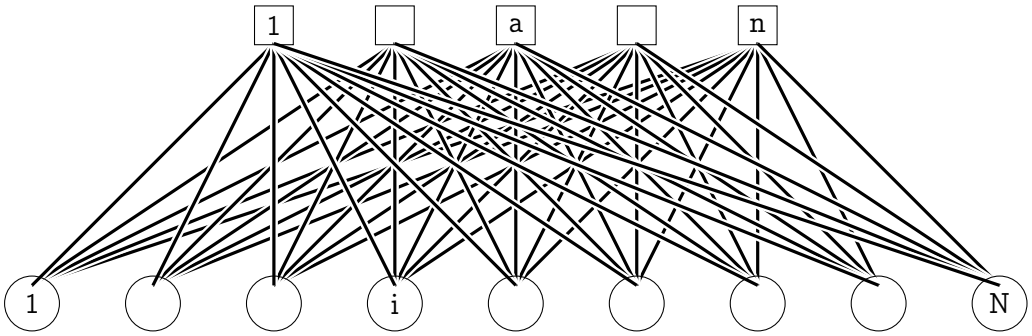


Figure 3.1: Factor graph associated to the probability distribution 3.4. Circles correspond to variables s_i, $i \in [n]$ and squares correspond to measurements y_a, $a \in [m]$.

Belief propagation

Belief propagation provides a low complexity tool to estimate the marginals of (3.4). This method has been introduced in [34]. In this section we review this method and make the corresponding changes for the weighted version. For any $d \in \mathbb{N}$ define $[d] := \{1,\ 2,\ \ldots,\ d\}$. Consider the bipartite factor graph $G = (V, F, E)$, shown in Figure 3.1, which includes a variable node $i \in V = [N]$ for each variable s_i and a factor node $a \in F = [n]$ for each term $\delta_{\{y_a=(As)_a\}}$. $E = [N] \times [n] = \{(i,a) : i \in [N], a \in [n]\}$ where variable i and factor a are connected by an edge if $\delta_{\{y_a=(As)_a\}}$ depends non-trivially on s_i, i.e., if $A_{ai} \neq 0$. As in our case A is a dense matrix, G is a complete bipartite matrix [34]. The state variables of this belief propagation are the messages $\{\nu_{i\to a}\}_{i\in V, a\in F}$ and $\{\hat{\nu}_{a\to i}\}_{i\in V, a\in F}$ which are associated to each edge of the factor graph G. The update rules for these densities are:

$$\begin{aligned} \nu^{t+1}_{i\to a}(s_i) &\cong e^{-\beta \mathrm{w}_i |s_i|} \prod_{b\neq a} \hat{\nu}^t_{b\to i}(s_i), \\ \hat{\nu}^t_{a\to i}(s_i) &\cong \int \prod_{j\neq i} \nu^t_{j\to a}(s_i)\ \ \delta_{\{y_a=(As)_a\}} ds. \end{aligned} \tag{3.5}$$

Here and below the subscripts denote the iteration number and $\cong$ means identity between distributions up to a normalization constant.

This message passing has two challenges in implementation. First the messages are density functions over the real line and keeping track of them is difficult unless they have some structure. Furthermore at each iteration, $2nN$ messages should be calculated which is computationally expensive. In Sections 3.3 and 3.4 we show that in the large system limit and as $\beta \to \infty$ this message passing algorithm is equivalent to the following simple iterative algo-

rithm:
As mentioned in (3.2) at $t = 0$ we start from $x^0 = 0$ and $\hat{\tau}^0 = 1$ and $z^0 = y$. Assume $\mathrm{w} = [\mathrm{w}_1,\ \mathrm{w}_2,\ \ldots,\ \mathrm{w}_N]^T$ is the weight vector. Then the algorithm proceeds as follows:

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathrm{w}) \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w})\rangle \\ \hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w})\rangle. \end{aligned} \tag{3.6}$$

In each iteration, x_i^t is the mean value of the message $\nu_{i\to a}^t$, $\hat{\tau}^t$ is the variance of $\nu_{i\to a}^t$, and the residual z_a^t corresponds to the mean of $\hat{\nu}_{a\to i}^t$. An analysis of these terms—following the steps in Section 3 of [34]—is given in the next section.

3.3 Large system limit

In this section we explain the derivation of (3.6) which is a straight-forward adaptation to the weighted case of the derivation of Section 3 of Donoho et al. in [34]. We restate some of lemmas and theorems in that paper without providing the proofs and justify the simple changes which should be done to solve the weighted ℓ_1 problem (1.7) by weighted AMP.
The following lemma in [34, Lemma 3.1] approximates $\hat{\nu}_{a\to i}^t$ by a Gaussian distribution. We restate this lemma without making any changes.

Lemma 3.1. *Let $x_{j\to a}^t$ and $\frac{\tau_{j\to a}^t}{\beta}$ be, respectively, the mean and variance of the distribution $\hat{\nu}_{j\to a}^t$. Further assume $\int |s-j|^3 d\nu_{j\to a}^t(s_j) \le C_t$ uniformly in N and n. Then there exist a constant C_t' such that*

$$\begin{aligned} ||\hat{\nu}_{a\to i}^t - \hat{\phi}_{a\to i}^t||_K &\le \frac{C_t'}{N^{\frac{1}{2}}(\hat{\tau}_{a\to i}^t)^{\frac{3}{2}}}, \\ \hat{\phi}_{a\to i}^t(ds_i) &:= \sqrt{\frac{\beta A_{ai}^2}{2\pi\hat{\tau}_{a\to i}^t}} exp\{\frac{\beta}{2\hat{\tau}_{a\to i}^t}(A_{ai}s_i - z_{a\to i}^t)^2\}ds_i, \end{aligned} \tag{3.7}$$

where the distribution parameters are given by

$$\begin{aligned} z_{a\to i}^t &:= y_a - \sum_{j\neq i} A_{aj} x_{j\to a}^t, \\ \hat{\tau}_{a\to i}^t &:= \sum_{j\neq i} A_{aj}^2 \tau_{j\to a}^t. \end{aligned} \tag{3.8}$$

Here $||\cdot||_K$ is Kolmogorov distance, which for two distributions μ_1 and μ_2 is defined as

$$||\mu_1 - \mu_2||_K := \sup_{a\in\mathbb{R}} |\int_{-\infty}^{a} \mu_1(dx) - \int_{-\infty}^{a} \mu_1(dx)|. \tag{3.9}$$

Notice that $\hat{\tau}^t_{a\to i} = \sum_{j\neq i} A^2_{aj}\tau^t_{j\to a} = C_a - A^2_{ai}\tau^t_{i\to a}$, where $C_a = \sum_j A^2_{aj}\tau^t_{j\to a}$. Assuming that matrix A is drawn from a random Gaussian distribution with mean 0 and variance $\frac{1}{n}$ we can approximate $\hat{\tau}^t_{a\to i}$ by an edge independent quantity $\hat{\tau}^t$.
Motivated by this lemma we find the mean and variance of the messages $\tau^{t+1}_{i\to a}(s_i)$. Consider the following family of densities introduced in [34]

$$f_\beta(s; x, b) := \frac{1}{z_\beta(x,b)} exp\{-\beta|s| - \frac{\beta}{2b}(s-x)^2\}, \tag{3.10}$$

where $z_\beta(x,b)$ is a normalization constant.
Also denote the mean and variance of these distributions as

$$F_\beta(x;b) := \mathbb{E}_{f_\beta(\cdot;x,b)}(Z), \quad G_\beta(x;b) := \mathrm{Var}_{f_\beta(\cdot;x,b)}(Z), \tag{3.11}$$

where Z has density $f_\beta(\cdot; x, b)$ [34]. Simple modification of the Lemma 3.2 in [34] gives us the mean and variance of $\tau^{t+1}_{i\to a}(s_i)$. Notice that in our case, f_β is replaced by $f_{\beta \mathrm{w}_i}$ and similarly we replace F_β and G_β by $F_{\beta \mathrm{w}_i}$ and $G_{\beta \mathrm{w}_i}$ respectively.

Lemma 3.2. *Suppose at iteration t, the messages from factor nodes to the variable nodes are set to be $\hat{\tau}^t_{a\to i}(s_i) = \hat{\phi}^t_{a\to i}(s_i)$, with $\hat{\phi}^t_{a\to i}$ as defined in (3.7) with parameters $z^t_{a\to i}$ and $\hat{\tau}^t_{a\to i} = \hat{\tau}^t$. Then at the next iteration we have*

$$\nu^{t+1}_{i\to a}(s_i) = \phi^{t+1}_{i\to a}(s_i)\{1 + O(\frac{s_i^2}{n})\}, \qquad \phi^{t+1}_{i\to a}(s_i) = f_{\beta \mathrm{w}_i}(s_i; \sum_{b\neq a} A_{bi} z^t_{b\to i}, \hat{\tau}^t). \tag{3.12}$$

Combining this lemma with Lemma 3.1, the mean and variance is given by

$$x^{t+1}_{i\to a} = F_{\beta \mathrm{w}_i}(\sum_{b\neq a} A_{bi} z^t_{b\to i}; \hat{\tau}^t), \qquad \tau^t_{i\to a} = \beta G_{\beta \mathrm{w}_i}(\sum_{b\neq a} A_{bi} z^t_{b\to i}; \hat{\tau}^t). \tag{3.13}$$

Proof. Combining (3.7) and (3.5) we have:

$$\begin{aligned}
\nu_{i\to a}^{t+1}(s_i) \quad &\cong \quad e^{-\beta \mathrm{w}_i|s_i|}\prod_{b\neq a}\hat{\nu}_{b\to i}^{t}(s_i) = exp\{-\beta \mathrm{w}-\mathrm{i}|\mathrm{s_i}|-\sum_{\mathrm{b}\neq \mathrm{a}}\frac{\beta}{2\hat{\tau}^{\mathrm{t}}}(\mathrm{A_{ai}s_i}-\mathrm{z_{b\to i}^t})^2\}\cong \\
&exp\{-\beta \mathrm{w}_i|s_i|-\frac{\beta}{2\hat{\tau}^t}(\frac{n-1}{n}s_i^2-2s_i\sum_{b\neq a}A_{bi}z_{b\to i}^t)^2\} \\
&\cong exp\{-(\beta \mathrm{w}_i)|s_i|-\frac{(\beta \mathrm{w}_i)}{2(\mathrm{w}_i\hat{\tau}^t)}(\frac{n-1}{n}s_i^2-2s_i\sum_{b\neq a}A_{bi}z_{b\to i}^t)^2\},
\end{aligned} \tag{3.14}$$

which coincides with $\phi_{i\to a}^{t+1}(s_i)$ up to terms of $\frac{s_i^2}{n}$. Notice that here we have used the fact that $A_{ai}^2 \sim \frac{1}{\sqrt{n}}$ and $\cong$ means identity between distributions up to a normalization constant. And finally the formulae for $x_{i\to a}^{t+1}$ and $\tau_{i\to a}^t$ follows from (3.11).

$$\begin{aligned}
x_{i\to a}^{t+1} &= F_{\beta \mathrm{w}_i}(\sum_{b\neq a}A_{bi}z_{b\to i}^t;\mathrm{w}_i\hat{\tau}^t), \\
z_{a\to i}^t &= y_a-\sum_{j\neq i}A_{aj}z_{j\to a}^t, \\
\hat{\tau}_i^{t+1} &= \frac{\beta}{n}G_{\beta \mathrm{w}_i}(\sum_{i=1}^{N}A_{bi}z_{b\to i}^t;\mathrm{w}_i\hat{\tau}^t).
\end{aligned} \tag{3.15}$$

3.4 Large β limit

As explained in Section 3.2 we are interested in the case where $\beta\to\infty$. In this section we simplify the belief propagation formulas (3.15). In Section 3.3 of [34] the functions F_β and G_β have been studied in the large β limit. Here we follow the same argument. Consider the soft thresholding function:

$$\eta(x;b)=\begin{cases} x-b, & \text{if } x>b \\ 0, & \text{if } -b\le x\le b \\ x+b, & \text{if } x<-b \end{cases}. \tag{3.16}$$

We can easily confirm that

$$\eta(x;\mathrm{w}_ib)=\underset{s\in\mathbb{R}}{argmin}\{|s|+\frac{1}{2\mathrm{w}_ib}(s-x)^2\}=\underset{s\in\mathbb{R}}{argmin}\{\mathrm{w}_i|s|+\frac{1}{2b}(s-x)^2\}. \tag{3.17}$$

In the $\beta\to\infty$ limit, the integral that defines $F_{\beta \mathrm{w}_i}(x;\mathrm{w}_ib)$ is dominated by its maximum value which is $s^*=\eta(x;\mathrm{w}_ib)$. Therefore $F_{\beta \mathrm{w}_i}(x;\mathrm{w}_ib)\to\eta(x;\mathrm{w}_ib)$. The variance can be

estimated by approximating $f_{\beta \mathrm{w}_i}(s; x, \mathrm{w}_i b)$ near s^*. If $s^* = 0$ then $f_{\beta \mathrm{w}_i}(s; x, \mathrm{w}_i b)$ can be approximated by a Laplace distribution which leads to $G_{\beta \mathrm{w}_i}(x; \mathrm{w}_i b) = \Theta(\frac{1}{(\beta \mathrm{w}_i)^2})$ and if $s^* \neq 0$ then $f_{\beta \mathrm{w}_i}(s; x, \mathrm{w}_i b)$ can be approximated by a Gaussian distribution distribution which leads to $G_{\beta \mathrm{w}_i}(x; \mathrm{w}_i b) = \Theta(\frac{1}{\beta \mathrm{w}_i})$. Hence using Lemma 3.3 of [34] we get the following lemma:

Lemma 3.3.

$$\begin{aligned} &\lim_{\beta \mathrm{w}_i \to \infty} F_{\beta \mathrm{w}_i}(x; \mathrm{w}_i b) = \eta(x; \mathrm{w}_i b), \\ &\lim_{\beta \mathrm{w}_i \to \infty} \beta G_{\beta \mathrm{w}_i}(x; \mathrm{w}_i b) = \frac{\lim_{\beta \mathrm{w}_i \to \infty} \beta \mathrm{w}_i G_{\beta \mathrm{w}_i}(x; \mathrm{w}_i b)}{\mathrm{w}_i} = \frac{\mathrm{w}_i b \quad \eta'(x; \mathrm{w}_i b)}{\mathrm{w}_i} = b \eta'(x; \mathrm{w}_i b). \end{aligned} \tag{3.18}$$

Accordingly, we are led to the following simplified message passing algorithm:

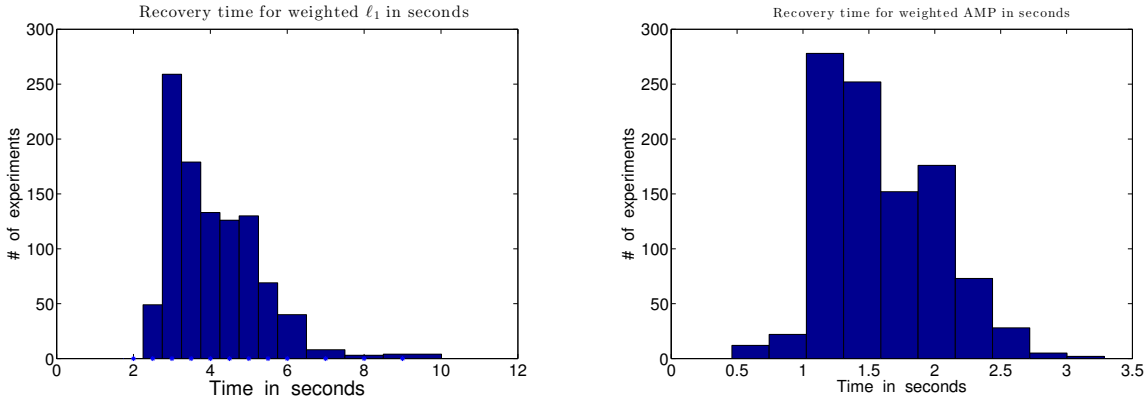
$$\begin{aligned} x_{i \to a}^{t+1} &= \eta(\sum_{b \neq a} A_{bi} z_{b \to i}^t; \mathrm{w}_i \hat{\tau}^t), \\ z_{a \to i}^t &= y_a - \sum_{j \neq i} A_{aj} z_{j \to a}^t, \\ \hat{\tau}^{t+1} &= \frac{\hat{\tau}^t}{N\delta} \sum_{i=1}^{N} \eta'(\sum_b A_{bi} z_{b \to i}^t; \mathrm{w}_i \hat{\tau}^t). \end{aligned} \tag{3.19}$$

3.5 From message passing to AMP

The simplified message passing algorithm (3.19), still needs $2nN$ updates at each iteration which is computationally intractable for large problems. In Section 3.4 of [34] the regular message passing algorithm has been approximated by AMP with approximations $x_{i \to a}^t = x_i^t + \delta x_{i \to a}^t + O(\frac{1}{N})$ and $z_{i \to a}^t = z_i^t + \delta z_{i \to a}^t + O(\frac{1}{N})$ which leads to the algorithm (3.2). An identical approach can be applied to (3.19) which approximates this message passing algorithm by the following algorithm. As calculations are the same as calculations in Section 3.4 of [34] once we include the weighting, we omit them.

$$\begin{aligned} x^{t+1} &= \eta(x^t + A^* z^t; \hat{\tau}^t \mathrm{w}) \\ z^t &= y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathrm{w}) \rangle \\ \hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1} \mathrm{w}) \rangle. \end{aligned} \tag{3.20}$$

As mentioned earlier the main advantage of weighted AMP over weighted ℓ_1 minimization is the low computational complexity of this algorithm. Figures 3.2 and Figure 3.3 illustrate this



(a) Weighted ℓ_1 (b) Weighted AMP

Figure 3.2: Histograms of the recovery time of weighted ℓ_1 and weighted AMP. Plots show the times it takes for (a) weighted ℓ_1, and (b) weighted AMP to recover 1000 k-sparse signals $x \in \mathbb{R}^{4000}$ with $k = 400$ and $n = 1500$ measurements when we have a support estimate with 50% accuracy.

advantage. Here we use weighted AMP and weighted ℓ_1 to recover 1000 400-sparse signals $x \in \mathbb{R}^{4000}$ by $n = 1500$ linear measurements obtained via $y = Ax$ where $A \in \mathbb{R}^{1500\times 4000}$ is a Gaussian random measurement matrix. For each instance of the experiment, assuming that the signal x is supported on the set T—with $|T| = 400$—we have a support estimate $\tilde{T}$ with 50% accuracy, i.e., $\frac{|T\cap\tilde{T}|}{|T|} = 0.5$. Our goal is to compare the recovery time of weighted AMP and weighted ℓ_1. To do this we compute the time it takes for each algorithm to find an approximation x^* such that $\|x - x^*\| \leq 10^{-8}$, i.e., 80-db SNR. Figure 3.2.a shows the histogram of the time it takes for weighted ℓ_1 to recover the signals and Figure 3.2.b shows the histogram for weighted AMP. Notice the difference in the horizontal axes. As we can see in this figure, more than 98% of the signals was recovered with weighted AMP in less than 2.5 seconds, whereas, the fastest recovery with weighted ℓ_1 minimization is 2.5 seconds.

Figure 3.3 shows the recovery times of each one of the 1000 signals sorted with respect to the recovery time of weighted ℓ_1. In this figure we can see that weighted AMP is always—and in some cases, up to 10 times—faster than weighted ℓ_1.

In Section 3.6 we introduce a reweighted AMP algorithm to recover sparse signals and in Section 3.7 we present extensive numerical results comparing weighted ℓ_1, weighted AMP, and reweighted AMP.

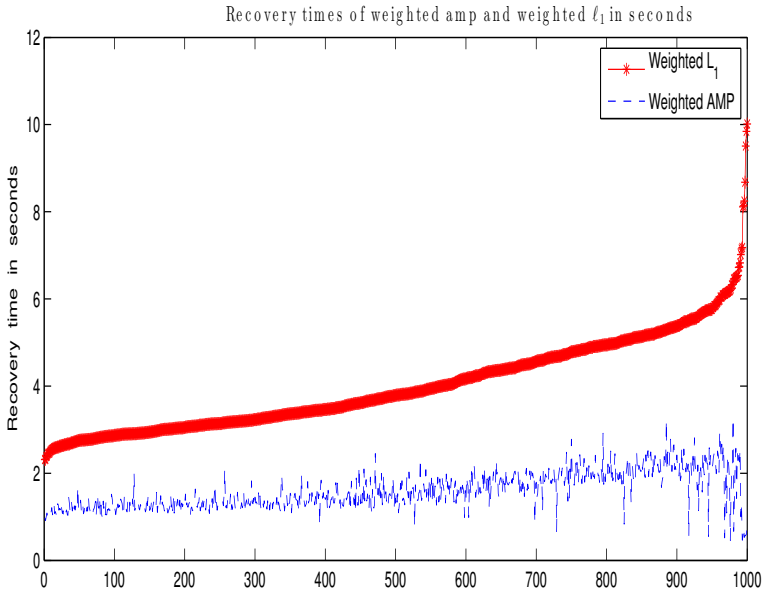


Figure 3.3: Comparison of recovery time of each one of the 1000 signals by weighted ℓ_1 and weighted AMP. The experiments are sorted with respect to the time it takes for weighted ℓ_1 to recover the signal. Specifically, the first experiments is one that weighted ℓ_1 minimization recovers it faster than all the others and experiment number 1000 is the one which is recovered slowest by weighted ℓ_1.

3.6 Reweighted AMP and reweighted W-AMP

In this section we introduce the reweighted AMP and reweighted W-AMP algorithms for recovering sparse and compressible signals and show preliminary results to justify the advantages of reweighting.

3.6.1 Reweighted AMP

Figure 3.4 shows the percentage of the true support recovered versus the iteration number when using regular AMP to recover a 100-sparse signal in $\mathbb{R}^{1000}$ and taking 250, 300, 350 and 400 measurements. The measurement matrix is a Gaussian normalized random matrix.
As we see in this figure even when the number of measurements is not enough (when AMP does not reach full recovery), more than 50% of the support is recovered after a few iterations. This observation motivates us to design a reweighted AMP algorithm which finds the support of the current estimate every 20 iterations and uses that as an approximation for the true

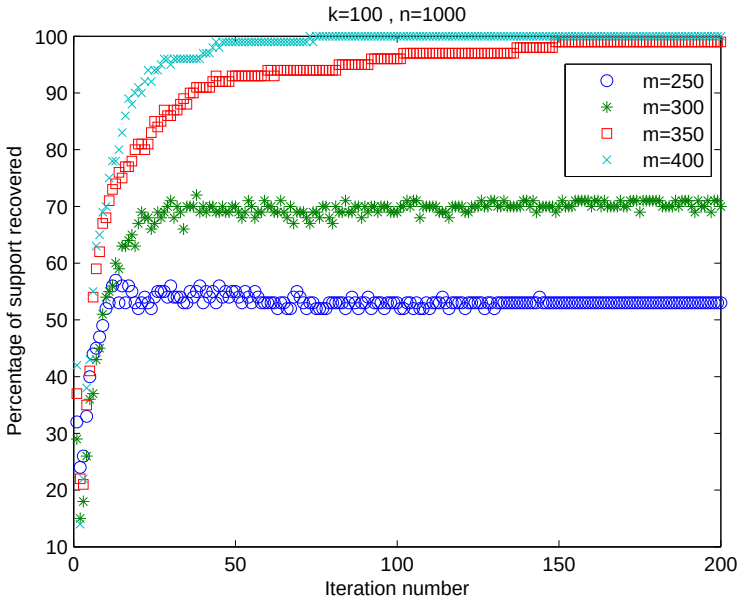


Figure 3.4: Percentage of the support recovered by AMP versus iteration number when the signal $x \in \mathbb{R}^{1000}$ is 100-sparse.

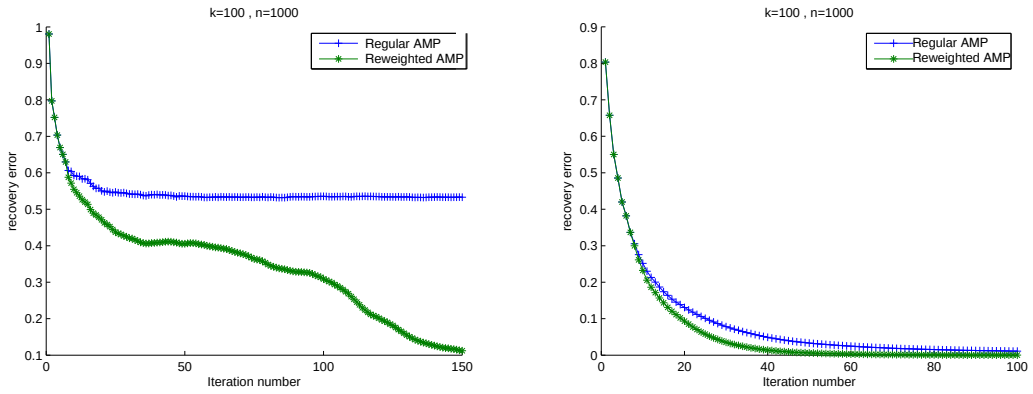
support and recovers the signal by solving a weighted AMP after that. Algorithm 2 explains the details of this algorithm.

Reweighted AMP increases the convergence speed and gives us better sparsity-undersampling trade-off. In other words there are cases that the number of measurements are not enough to recover the signal using regular AMP (and hence ℓ_1) but reweighted AMP can recover the signal using the same number of measurements. Figure 3.5 compares reweighted and regular AMP. Here we recover a 100-sparse signal in $\mathbb{R}^{1000}$ using an 250×1000 Gaussian matrix (left) and an 350×1000 Gaussian matrix (right). Both figures show the misfit error $\frac{\|x^t_{rec} - x\|}{\|x\|}$ (t is the current iteration) versus the iteration number. The left figure shows the case that the number of measurements is not enough in order to get full recovery by regular AMP and the right figure shows the case that regular AMP achieves full recovery.

To test our reweighted AMP algorithm, we compare the recovery performance of this algorithm with two well known algorithms that use reweighting to enhance the recovery conditions, i.e., iterative reweighted ℓ_1 minimization (IRL1) [30], and support driven reweighted ℓ_1 minimization (SDRL1) [35] in recovering signals $x \in \mathbb{R}^N$ with $N = 1000$. First we compare the results when we use these algorithms to recover 100 k-sparse signals using compressed measurements obtained by Gaussian matrices $A \in \mathbb{R}^{n \times N}$ where $n \in \{\frac{N}{10}, \frac{N}{4}, \frac{N}{2}\}$. In Figure 3.6 we compare

Algorithm 2 Reweighted AMP

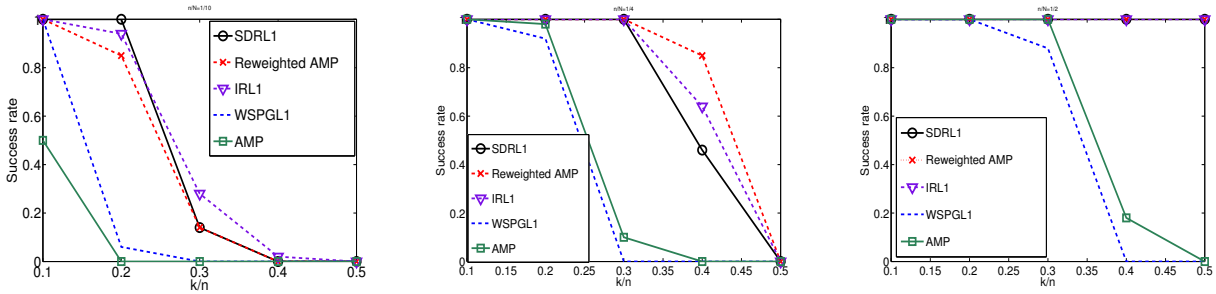
Input $b = Ax$, $t_{\max}$, ω
Output $x^{(t)}$
Initialize $\hat{p} = 0.98$, $\hat{k} = \frac{nlog(\frac{N}{n})}{2}$, $\mathrm{w}_i = 1$ for all $i \in \{1, \ldots, N\}$, $\delta = \frac{n}{N}$, $\Omega = \emptyset$, $t = 0$, $x^0 = 0$, $z^0 = y$, $\hat{\tau}^0 = 1$
while $t < t_{\max}$ **do**
    t=t+1
    $x^t = \eta(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w})$
    $z^t = y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle$
    $\hat{\tau}^t = \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle$
    if $\mathrm{mod}(t, 20) = 10$ **then**
        $l = \min_{\Lambda} |\Lambda|$ s.t. $\|x^t_\Lambda\| \geq \hat{p}\|x^t\|$
        $s = \min(l, \hat{k})$
        $\Omega = \mathrm{supp}(x^t|_s)$
        Set the weights equal to $\mathrm{w}_i = \begin{cases} 1, & \text{if } i \in \Omega^c \\ \omega, & \text{if } i \in \Omega \end{cases}$
    end if
end while



(a) $m = 250$ (b) $m = 350$

Figure 3.5: Comparison between regular AMP and reweighted AMP when m=250 and m=350, recovering a 100-sparse signal in $\mathbb{R}^{1000}$.

full recovery success rate when $\frac{k}{n} \in \{\frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10}, \frac{5}{10}\}$. For the sake of comparison we also include the recovery results of regular AMP and WSPGL1 algorithm introduced in [36]. The figure shows that our proposed reweighted AMP has comparable performance with IRL1 and SDRL1 (in some cases reweighted AMP outperforms the other two). Notice that both



(a) $\frac{n}{N} = \frac{1}{10}$ (b) $\frac{n}{N} = \frac{1}{4}$ (c) $\frac{n}{N} = \frac{1}{2}$

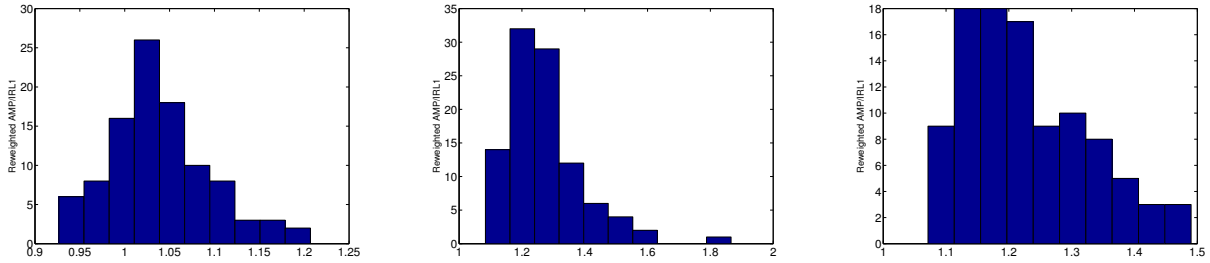
Figure 3.6: Comparison of percentage of exact recovery by reweighted AMP, IRL1, SDRL1, WSPGL1, and regular AMP recovering 100 sparse signals $x \in \mathbb{R}^{1000}$ with different number of measurement, n, and different levels of sparsity.

IRL1 and SDRL1 algorithms solve a series of weighted ℓ_1 minimizations to recover the signal and our proposed reweighted AMP is much faster than both these algorithms. On the other hand the recovery results of reweighted AMP are significantly better than algorithms which has comparable complexities, i.e., regular AMP and WSPGL1.

Next, we generate compressible signals $x \in \mathbb{R}^N$, sorted coefficients of which decay like j^{-p} where $j \in \{1,\ 2,\ \ldots,\ N\}$ with dimension $N = 1000$. We consider the case where $\frac{n}{N} = 0.3$ and decay power $p \in \{1.1,\ 1.5,\ 2\}$. Figure 3.7 shows the histogram of ratio of the reconstruction error of reweighted AMP over that of IRL1 for 100 experiments. Notice that a ratio smaller than one means that reweighted AMP has smaller reconstruction error than IRL1 and a ratio greater than one that the the reconstruction error of IRL1 is smaller than that of reweighted AMP.

3.6.2 Reweighted W-AMP

Applying the same reweighting idea, explained in Section 3.6.1, to the weighted AMP algorithm (3.20), we derive the reweighted W-AMP algorithm when we have prior support information about the signal which is explained in Algorithm 3. Notice that here the only difference is in the way we do reweighting, i.e., as before we use small weights for the coefficients which are in the support estimates and also every 20 iterations we estimate the support set of the current estimate and use smaller weights for those coefficients.



(a) $p = 1.1$, $MSE = 0.0964$ (b) $p = 1.5$, $MSE = 0.0173$ (c) $p = 2$, $MSE = 0.0018$

Figure 3.7: Histogram of ratio of mean squared error (MSE) between reweighted AMP and IRL1 for recovering compressible signals $x \in \mathbb{R}^{1000}$, sorted coefficients of which decay like j^{-p} with $j \in \{1,\ 2,\ \ldots,\ 100\}$ and with different decay rates p.

Algorithm 3 Reweighted W-AMP

Input $b = Ax$, $t_{\max}$, $\tilde{T}$, ω
Output $x^{(t)}$
Initialize $\hat{p} = 0.98$, $\hat{k} = \frac{nlog(\frac{N}{n})}{2}$, $\mathrm{w}_i = 1$ for $i \in \tilde{T}^c$ and $\mathrm{w}_i = \omega$ for $i \in \tilde{T}$, $\delta = \frac{n}{N}$, $\Omega = \emptyset$, $t = 0$, $x^0 = 0$, $z^0 = y$, $\hat{\tau}^0 = 1$
while $t < t_{\max}$ **do**
    t=t+1
    $x^t = \eta(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w})$
    $z^t = y - Ax^t + \delta^{-1} z^{t-1} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle$
    $\hat{\tau}^t = \frac{\hat{\tau}^{t-1}}{\delta} \langle \eta'(x^{t-1} + A^* z^{t-1}; \hat{\tau}^{t-1}\mathrm{w}) \rangle$
    if $\mathrm{mod}(t, 20) = 10$ **then**
        $l = \min_{\Lambda} |\Lambda|$ s.t. $\|x^t_\Lambda\| \geq \hat{p}\|x^t\|$
        $s = \min(l, \hat{k})$
        $\Omega = \mathrm{supp}(x^t|_s)$
        Set the weights equal to $\mathrm{w}_i = \begin{cases} 1, & \text{if } i \in \tilde{T}^c \cap \Omega^c \\ \omega, & \text{if } i \in \tilde{T} \cap \Omega^c \\ \frac{\omega}{5}, & \text{if } i \in \Omega \end{cases}$
    end if
end while

3.7 Numerical results

In this section we provide numerical examples to compare regular, weighted and reweighted AMP with ℓ_1 and weighted ℓ_1 algorithms (we use the SPGL1 software to solve the weighted ℓ_1 minimization problem [37]).

Table 3.1 compares the time it takes for AMP, reweighted AMP, and ℓ_1 minimization to recover a 100-sparse signal in $\mathbb{R}^{1000}$ with maximum SNR of 40 db. Here we take the average recovery time over 20 experiments when the number of measurements changes from 300 to 500 with an increment of 50. ∞ means that the method fails full recovery in at least one of the examples. Notice that in Table 1 we see that reweighted AMP can recover 100-sparse signals with fewer measurements compared to regular AMP and ℓ_1 minimization.

To examine this observation more carefully we consider the case that we have prior support information (Notice that in these experiments we also include the cases that the recovery algorithm does not use any prior support information). We generate signals $x \in \mathbb{R}^N$ where $N = 500$ and with fixed sparsity $k = 40$. We compute the (noisy) compressed measurements of x using a Gaussian random measurement matrix A with dimensions $n \times N$ where n varies between 80 and 200 with an increment of 20. In the case of noisy measurements, we have assumed 5% noise, i.e., 26 db SNR. Figure 3.8 shows the reconstruction signal-to-noise ratio averaged over 20 experiments, using weighted AMP and reweighted W-AMP to recover sparse signals $x \in \mathbb{R}^N$ in the noise-free case and for $\alpha = 0.3, 0.5, 0.7$ (α determines the accuracy of the support estimate). The SNR is measured in dB, as defined in (2.20). In these figures we also include the results of weighted ℓ_1, IRL1, SDRL1, and WSPGL1. Notice that when $\omega = 1$, weighted AMP reduces to regular AMP, reweighted W-AMP reduces to reweighted AMP, and weighted ℓ_1 reduces to regular ℓ_1. Figures 3.8.a, 3.8.c, and 3.8.e show the results for weighted AMP and weighted ℓ_1 when $\alpha = 0.7,\ 0.5$, and 0.3 respectively. Notice that although weighted AMP is much faster than weighted ℓ_1,its recovery performance is slightly better than weighted ℓ_1. Similarly Figures 3.8.b, 3.8.d, and 3.8.f show the results for reweighted W-AMP, IRL1, SDRL1, and WSPGL1.

Remark 3.1. As α increases—support estimate is more accurate—both weighted and reweighted W-AMP achieve full recovery with fewer measurements. Also notice that intermediate value of ω gives us the best recovery when $\alpha > 0.5$. This behavior is similar to what we see with weighted ℓ_1 and weighted ℓ_p minimization.

Remark 3.2. We can use Figure 3.8 to compare weighted AMP and weighted ℓ_1 minimization with reweighted W-AMP empirically. In all the cases, reweighted W-AMP outperforms

method\m	300	350	400	450	500
$\triangle_1$	∞	∞	0.3367	0.2071	0.1538
regular AMP	∞	∞	0.1111	0.0823	0.0727
reweighted AMP	2.1029	0.2481	0.1102	0.0783	0.0724

Table 3.1: Comparison of recovery time of AMP, reweighted AMP, and ℓ_1 minimization in seconds.

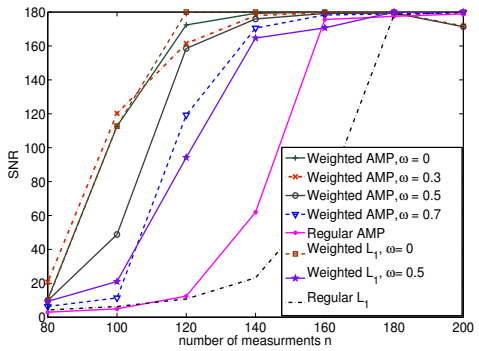
weighted AMP and weighted ℓ_1 significantly. For example when $\alpha = 0.5$, $n = 100$, and $\omega = 0.5$ reweighted W-AMP reaches full recovery for all the 40-sparse signals (average SNR of 180 db), whereas both weighted AMP and weighted ℓ_1 recover with average SNR of less than 10 db. As another example when $\alpha = 0.3$ reweighted W-AMP achieves full recovery with 120 measurements whereas weighted AMP needs 160 measurement for full recovery.

Remark 3.3. In all the experiments reweighted W-AMP outperforms IRL1 and SDRL1 (while it is still much faster than both). For example when $n = 100$ reweighted W-AMP recovers the signals with an average SNR of 75 db, whereas both IRL1 and SDRL1 recover the signals with an average SNR of less than 15 db. Also notice that in this case when $\omega = 0.3$ or 0.5 we reach full recovery for all the signals when we have a 50% accurate support estimate.

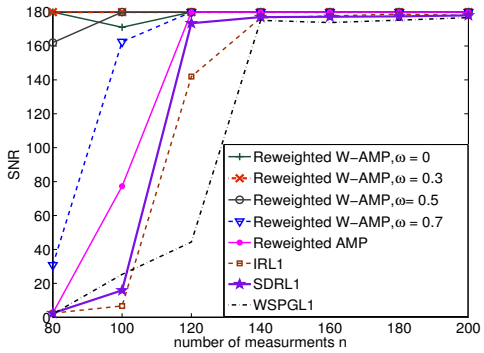
Remark 3.4. Notice the results of reweighted W-AMP when we have a 70% accurate support estimate (Figure 3.8.b). In this experiment when $\omega = 0.3$ reweighted W-AMP has recovered all the signals when the number of measurements is as small as twice the number of non-zero coefficients of the signal.

Next we repeat the above experiment except this time the measurements have 5% noise, i.e., 26 db SNR. The results are shown in Figure 3.9. Figures 3.9.a, 3.9.c, and 3.9.e show the weighted AMP and weighted ℓ_1 reconstruction signal-to-noise ratio averaged over 20 experiments and Figures 3.9.b, 3.9.d, 3.9.f show the results when we use reweighted W-AMP, IRL1, SDRL1, and WSPGL1. Reweighted W-AMP outperforms weighted AMP again. As mentioned earlier when $\omega = 1$ weighted AMP reduces to regular AMP and reweighted W-AMP reduces to reweighted AMP. Similar to the noise-free case reweighted W-AMP always outperforms all the other algorithms. Notice that using intermediate weights is always beneficial in the noisy case even when the support estimate is inaccurate, i.e., $\alpha < 0.5$. For example when $\alpha = 0.3$ and $n = 120$ with weighted AMP and $\omega = 0$ we get SNR$\approx$6 db and with regular AMP we get SNR$\approx$10 db whereas with weighted AMP and $\omega = 0.5$ we get SNR$\approx$12 db.

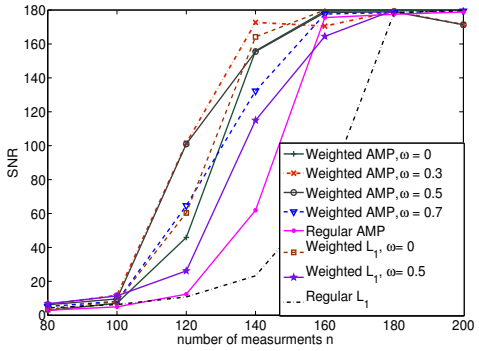
Remark 3.5. Notice the improvement in recovery performance of weighted W-AMP when we have an accurate prior support information ($\alpha = 0.7$). For example when $n = 80$, on average,



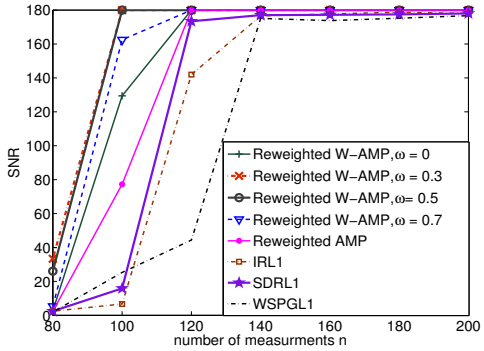
(a) Weighted AMP, $\alpha = 0.7$



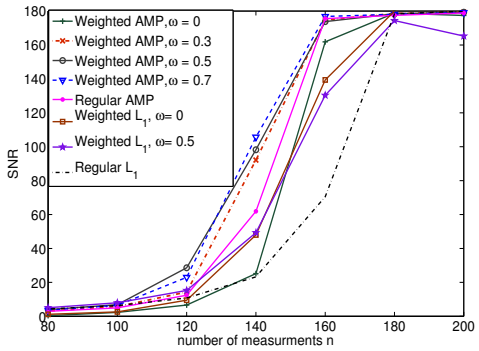
(b) Reweighted W-AMP, $\alpha = 0.7$



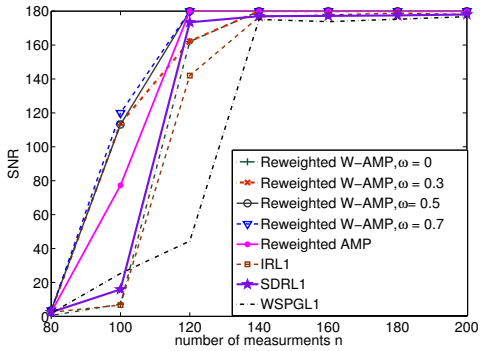
(c) Weighted AMP, $\alpha = 0.5$



(d) Reweighted W-AMP, $\alpha = 0.5$



(e) Weighted AMP, $\alpha = 0.3$

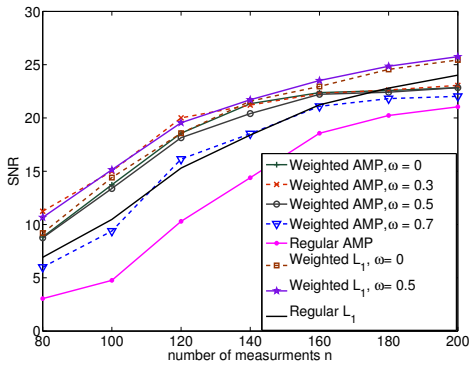


(f) Reweighted W-AMP, $\alpha = 0.3$

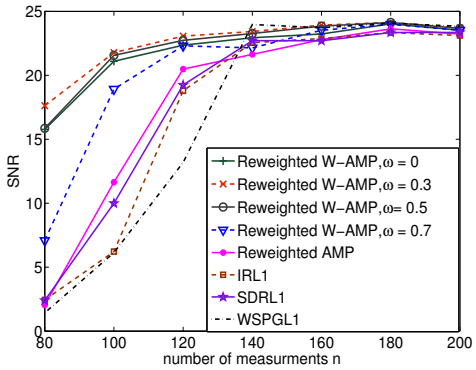
Figure 3.8: (Noise-free case) Comparison of performance of (a, c, e) weighted AMP and weighted ℓ_1 and (b, d, f) reweighted W-AMP, IRL1, SDRL1, and WSPGL1 in terms of SNR averaged over 20 experiments for sparse signals with variable weights and measurements when there is no noise.

using reweighted W-AMP with $\omega = 0.3$ improves the recovery SNR of reweighted AMP by about 15 db.

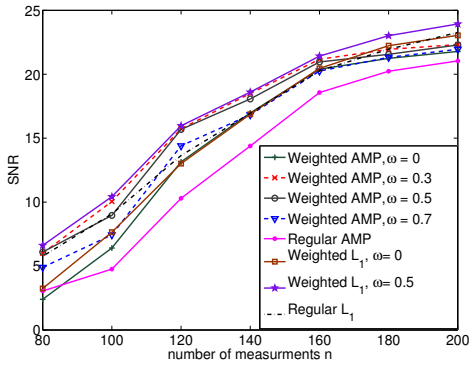
The interesting advantage of reweighted AMP over regular AMP and ℓ_1 minimization is that the sufficient number of measurements for recovery by reweighted AMP seems to be less than those of AMP and ℓ_1 minimization. Figure 3.10 empirically compares the recovery success percentage of these algorithms recovering signals $x \in \mathbb{R}^{200}$ with different levels of sparsity and different number of measurements. Horizontal axis determines the undersampling fraction, $\delta = \frac{m}{n}$, and the vertical axis represents the sparsity fraction, $\rho = \frac{k}{m}$. Figures 3.10.a–c show the results for recovery by $\triangle_1$, regular AMP, and reweighted AMP respectively. In this figures each point indicates the fraction of realizations with successful recovery, when we use that method to recover signals with corresponding δ and ρ. This figure empirically shows the advantage of reweighted AMP over the other two, for example, when m=100, $\triangle_1$ and AMP has recovered all the 25-sparse signals, where reweighted AMP has recovered all 30-sparse signals.
Figure 3.11 summarizes the empirical results of Figure 3.10. At the points below each curve, we empirically get more than 80% successful recovery and at the points above the curve, we get less than 80% successful recovery when the corresponding method is used to recover signals in $\mathbb{R}^{200}$. Notice that if we use 100% successful recovery the results of $\triangle_1$ and AMP are essentially similar to each other. However in our experiments we see that in the cases that these algorithms do not recover all the signals, AMP reaches full recovery for more signals compared to $\triangle_1$. Therefore in this experiment we show the result when we use 80% full recovery threshold.



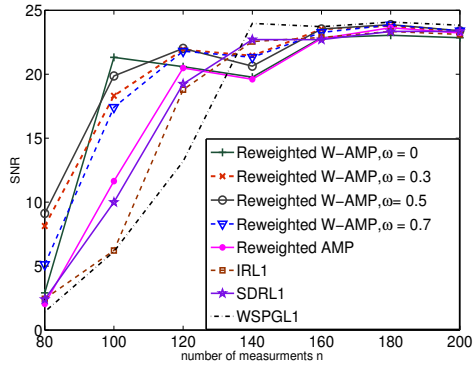
(a) Weighted AMP, $\alpha = 0.7$



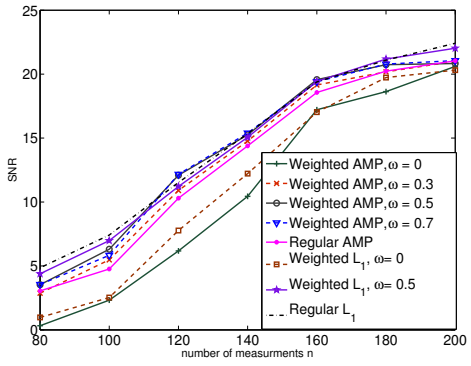
(b) Reweighted W-AMP, $\alpha = 0.7$



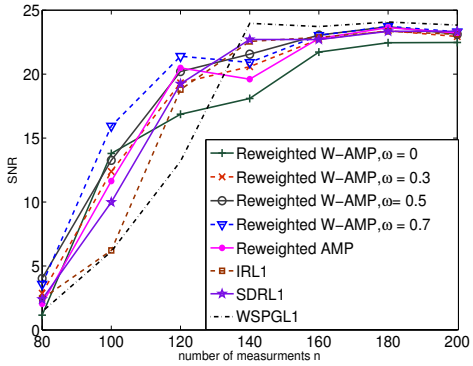
(c) Weighted AMP, $\alpha = 0.5$



(d) Reweighted W-AMP, $\alpha = 0.5$

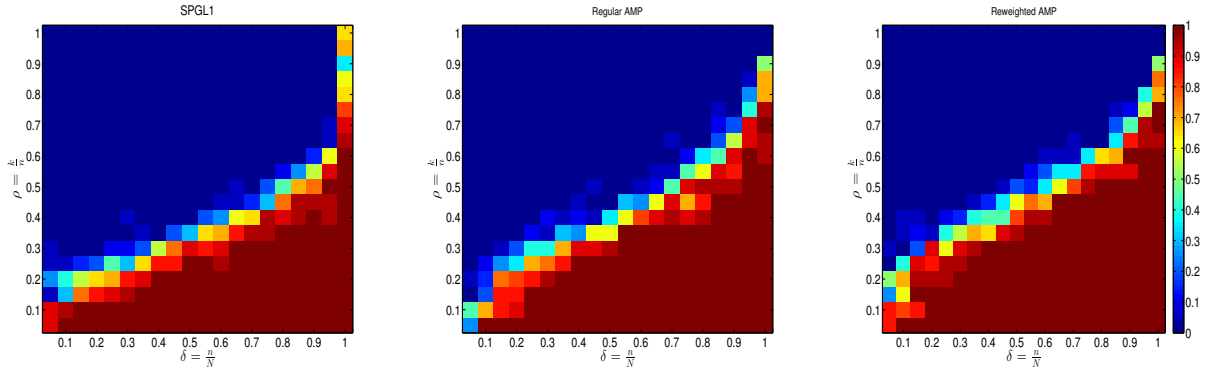


(e) Weighted AMP, $\alpha = 0.3$



(f) Reweighted W-AMP, $\alpha = 0.3$

Figure 3.9: (Noisy case) Comparison of performance of (a, c, e) weighted AMP and weighted ℓ_1 and (b, d, f) reweighted W-AMP, IRL1, SDRL1, and WSPGL1 in terms of SNR averaged over 20 experiments for sparse signals with variable weights and measurements when there is 5% noise.



(a) $\triangle_1$ (b) Regular AMP (c) Reweighted AMP

Figure 3.10: Empirical phase transition of $\triangle_1$, regular AMP and reweighted AMP recovering sparse signals in $\mathbb{R}^{200}$. Figures show the percentage of successful recovery over 20 experiments when the measurement matrix is Gaussian.

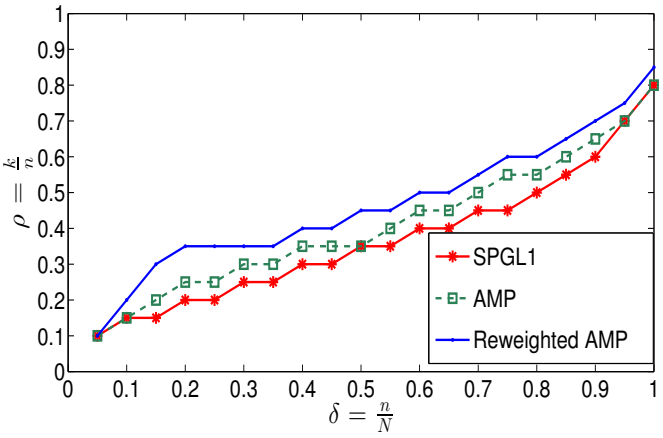


Figure 3.11: Illustration of the phasediagrams of AMP, reweighted AMP, and $\triangle_1$ presented in Figure 3.10.

3.8 Stylized applications

In this section, we apply weighted AMP and reweighted W-AMP to recover real speech signals that are compressively sampled. The speech signals are the same speech signals used in Chapter 2 and we choose the weights the same way. We break the measurements into 21 blocks and use smaller weights on the low frequencies. We also use the large coefficients in each block as an approximation for the large coefficients in the next block. For the reweighted AMP we

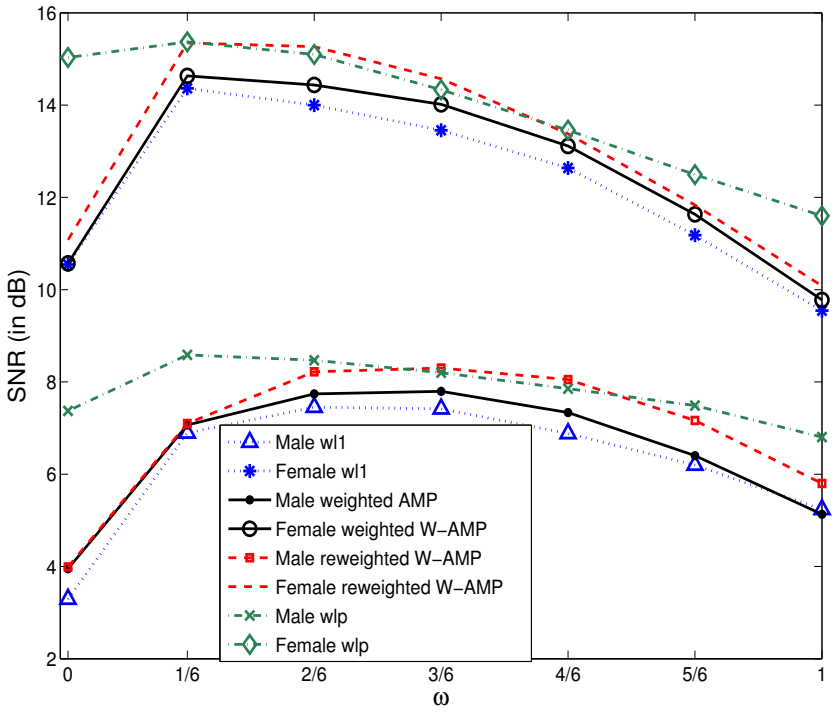


Figure 3.12: SNRs of reconstructed audio signals from compressed sensing measurements plotted against ω. Figure shows the result for reconstruction via weighted ℓ_1 minimization, weighted ℓ_p minimization, weighted AMP, and reweighted W-AMP.

start with with these weights for each block and renew the weights every 20 iterations.

Figure 3.12 shows the results when we apply weighted AMP, reweighted W-AMP, weighted ℓ_1, and weighted ℓ_p to recover two compressively sampled speech signals (one male and one female). The results are presented for $\omega \in [0, \frac{1}{6}, \frac{2}{6}, \ldots, 1]$. Generally, using intermediate weights results in better recovery and the results of reweighted W-AMP is slightly better than the results of weighted AMP and weighted ℓ_1 minimization.

Bibliography

[1] M. P. Friedlander, H. Mansour, R. Saab, and Özgür Yılmaz, "Recovering compressively sampled signals using partial support information," *Arxiv preprint arXiv:1010.4612v2*, 2011.

[2] R. Saab and Özgür Yılmaz, "Sparse recovery by non-convex optimization -instance optimality," *Applied and Computational Harmonic Analysis*, vol. 29, no. 1, pp. 30–48, 2010.

[3] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," *Proc. ITW, Cairo,Egypt*, 2010.

[4] H. Mansour, F. Herrmann, and O. Yılmaz, "Improved wavefield reconstruction from randomized sampling via weighted one-norm minimization," *to appear in Geophysics, GEO-2012-0383, 2012.*

[5] B. Bah and J. Tanner, "Improved bounds on restricted isometry constants for gaussian matrices," *CoRR*, 2010. [Online]. Available: arXiv:1003.3299v2

[6] Q. Du and J. E. Fowler, "Hyperspectral image compression using jpeg2000 and principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 201–205, April 2007.

[7] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging," *Preprint*, 2007.

[8] G. Hennenfent and F. Herrmann, "Application of stable signal recovery to seismic interpolation," 2006. [Online]. Available: http://slim.eos.ubc.ca/Publications/Private/Conferences/SEG/hennenfent06seg.pdf

[9] G. Hennenfent and F. J. Herrmann, "Seismic denoising with non-uniformly sampled curvelets," *IEEE Comp. in Sci. and Eng.*, vol. 8, no. 3, pp. 16–25, 2006.

[10] R. von Borries, C. Miosso, and C. Potes, "Compressed sensing using prior information," in *2nd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMPSAP 2007.*, 12-14 2007, pp. 121 – 124.

[11] N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," *IEEE Trans. on Signal Processing*, vol. 58, no. 9, pp. 4595 – 4607, September 2010.

[12] M. Amin Khajehnejad, W. Xu, A. Salman Avestimehr, and B. Hassibi, "Weighted l1 minimization for sparse recovery with prior information," in *IEEE International Symposium on Information Theory, ISIT 2009*, June 2009, pp. 483 – 487.

[13] D. Donoho, "Compressed sensing." *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] E. J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, pp. 1207–1223, 2006.

[15] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 5, pp. 2197–2202, 2003.

[16] R. Chartrand, "Exact reconstructions of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[17] S. Foucart and M. Lai, "Sparsest solutions of underdetermined linear systems via ℓ^q-minimization for $0 < q \leq 1$," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 395–407, 2009.

[18] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *Fourier Anal. Appl.*, vol. 14(5), pp. 629–654, 2008.

[19] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, p. 265?274, 2009.

[20] K. K. Herrity, A. C. Gilbert, , and J. A. Tropp, "Sparse approximation via iterative thresholding," *in Proceedings of the Int. Conf. on Acoustics, Speech and Signal Processing,*, 2006.

[21] D. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106(45), pp. 18 914–18 919, 2009.

[22] L. Demanet and E. J. Candés, "The curvelet representation of wave propagators is optimally sparse," vol. 58, 2005, pp. 1472–1528.

[23] F. J. Herrmann, P. P. Moghaddam, and C. C. Stolk, "Sparsity- and continuity- promoting seismic imaging with curvelet frames," *Journal of Applied and Computational Harmonic Analysis*, vol. 24, pp. 150–173, 2008.

[24] E. J. Candès and T. Tao, "Decoding by linear programming." *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 489–509, 2005.

[25] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[26] R. Saab, R. Chartrand, and O. Yilmaz, "Stable sparse approximations via nonconvex optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 3885–3888.

[27] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes rendus-Mathématique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[28] X. Chen and W. Zhou, "Convergence of reweighted ℓ_1 minimization algorithms and unique solution of truncated ℓ_p minimization."

[29] R. Chartrand and W. Tin, "Iteratively reweighted algorithms for compressive sensing," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2008.*, pp. 3869–3872, 31 2008-April 4 2008.

[30] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[31] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, pp. 187–200, 1999.

[32] A. Maleki and D. L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *IEEE J. Sel. Topics in Signal Processing*, vol. 4, no. 2, pp. 330–341, 2010.

[33] T. DJ, A. PW, and P. RG, "Solution of 'solvable model of a spin glass'," *Philos Mag*, vol. 35, pp. 593–601, 1977.

[34] D. L. Donoho, A. Maleki, , and A. Montanari, "Constructing message passing algorithms for compressed sensing," *submitted to IEEE Trans. Inf. Theory*.

[35] H. Mansour and O. Yılmaz, "Support driven reweighted 1-norm minimization," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[36] H. Mansour, F. Herrmann, and O. Yılmaz, "Beyond l1 minimization for sparse signal recovery," *Proc. of the IEEE Statistical Signal Processing Workshop (SSP), Ann Arbor, Michigan, August 2012.*

[37] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, http://www.cs.ubc.ca/labs/scl/spgl1.