Compressed Sensing for Jointly Sparse Signals

by

Alireza Makhzani

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

Compressed Sensing for Jointly Sparse Signals

Alireza Makhzani
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2012

Compressed sensing is an emerging field, which proposes that a small collection of linear projections of a sparse signal contains enough information for perfect reconstruction of the signal. In this thesis, we study the general problem of modeling and reconstructing spatially or temporally correlated sparse signals in a distributed scenario. The correlation among signals provides an additional information, which could be captured by joint sparsity models. After modeling the correlation, we propose two different reconstruction algorithms that are able to successfully exploit this additional information. The first algorithm is a very fast greedy algorithm, which is suitable for large scale problems and can exploit spatial correlation. The second algorithm is based on a thresholding algorithm and can exploit both the temporal and spatial correlation. We also generalize the standard joint sparsity model and propose a new model for capturing the correlation in the sensor networks.

Dedication

To my beloved mother,
Nasrin

Acknowledgements

I would like to thank all the people who have helped and inspired me during the pursuit of my Masters degree.

I wish to express my sincerest thanks to my supervisor, Prof. Shahrokh Valaee, whose support made this work possible. His knowledge, guidance, and encouragement has been of invaluable help to me in producing this work.

I would also like to thank my colleagues at Wireless and Internet Research Laboratory (WIRLab) for their constructive comments.

I take this opportunity to specially thank Nasrin Tehrani and Hamid Emami. I thank them for their very kind helps and supports during the past years.

My deepest gratitude and love, of course, belong to my parents, Hassan and Nasrin, and my sister, Parastesh, for their unconditional love and support throughout my life. To them I owe all that I am and all that I have ever accomplished.

Contents

List of Figures

Chapter 1

Introduction

The conventional method of sampling signals is based on the work of Nyquist and Shannon. Their results suggest that the signals have to be sampled at twice their highest frequency for perfect reconstruction. Using this theorem, signal processing world has moved from the analog to digital domain. Unfortunately, in many important and emerging applications, the resulting Nyquist rate is so high that we end up with far too many samples or it may simply be physically impossible, to build devices capable of acquiring samples at the Nyquist rate.

Compressed sensing has recently emerged as a new framework for signal acquisition. It is based on the fact that the twice bandwidth Nyquist rate is a worst case bound for reconstructing any signal, however, in practice we are just dealing with the natural signals. All natural signals are compressible and can be sparsely represented in a predefined basis. In other words, we can sparsify natural signals such as images, speech, etc, in a wavelet, Fourier or DCT basis. Compressed Sensing [1] uses this structure of natural signals to reduce the number of measurements. According to this theory if a signal has K nonezero coefficients in a fixed basis, roughly cK incoherent measurements suffice to perfectly reconstruct the signal $[c \approx \log_2(1 + \frac{N}{K}) \approx 3 \; or \; 4]$. Two conditions should be satisfied by our measurement matrix for perfect reconstruction. These Conditions are called *Restricted Isometry Property* (RIP) and *Incoherency* [2]. Interestingly, it can be shown that a blind random projection by random matrices whose entries are i.i.d. Gaussian or Bernoulli random variables satisfies both RIP and incoherence condition. Many algorithms such as *Basis Pursuit* (BP) and *Orthogonal Matching Pursuit* (OMP) have been proposed for the reconstruction of sparse signals from their incoherent measurements.

1.1 Compressed Sensing for Jointly Sparse Signals

The CS literature has mostly focused on problems involving single sensors; however, many important applications that hold significant promise for CS involve signals that are multidimensional. The coordinates of these signals may span several temporal or spatial dimensions leading to a correlation among signals. We will model the correlation between signals with a joint sparsity model.

A practical situation well-modeled by this joint sparsity model is where multiple sensors acquire replicas of the same Fourier-sparse signal. Each signal experiences different attenuation and multipath effect which cause different amplitudes and phase shifts. But we expect the location of non-zero coefficients to be roughly the same. Under the right conditions, a decoder at the fusion centre can reconstruct all signals perfectly.

We propose different algorithms for joint recovery of jointly sparse signals from incoherent projections, and characterize theoretically and empirically the number of measurements per sensor required for perfect reconstruction.

1.2 Scope and Objectives

The work presented in this thesis has two main goals. The first goal is to develop and study new models for sparsity and compressibility of correlated signals. The second goal is to propose reconstruction algorithms for jointly sparse signals that outperforms the state of the art algorithms in terms of reconstruction rate and computation time. This thesis is organized as follows.

In Chapter 2, the relevant background material is presented. First, we discuss the basic ideas of sparsity and compressive sensing theory. Then, we cover some algorithms for sparse reconstruction such as ℓ_1 minimization, orthogonal matching pursuit and hard thresholding algorithms. Next, the concept of signals being jointly sparse is discussed and the state of the art reconstruction algorithms for jointly sparse signals such as Simultaneous Orthogonal Matching Pursuit (SOMP), 2-Thresholding, ReMBO algorithm, Mixed norm approach and Distributed SOMP.

In Chapter 3, a new technique for reconstructing jointly sparse signals based on hard thresholding algorithms is presented. More precisely, we extend the iterative hard thresholding algorithm for single sensor case to jointly sparse signals and propose a very low computational complexity reconstruction algorithm, which we call it Simultaneous Iterative Hard Thresholding (S-IHT), to address the problem of reconstruction of jointly sparse signals with different sparsity basis and sensing matrices. We show that our algorithm is much faster than the state of the art algorithms, such as DCS-SOMP, while showing similar reconstruction rate.

In Chapter 4, we study the problem of reconstructing a set of non-uniformly sparse signals that are sensed by a set of spatially distributed sensors in a sensor network. We first propose an approach called Modified Iterative Hard Thresholding (MIHT) to reconstruct non-uniformly sparse signals in the single sensor case and then extend the MIHT algorithm to jointly sparse signals and exploit the information of spatial correlation as well as temporal correlation in the sensor networks.

In Chapter 5, we propose a Generalized Joint Sparsity Model (G-JSM) that is able to model more practical cases. The proposed model decomposes each sparse signal into two sparse components. The first component has a common support across all sensed signals. The second component is an innovation part that is specific to each sensor and might have a support that is different from the support of the other innovation signals. In this model, we use Principal Component Analysis followed by Minimum Description Length to reconstruct the signals.

Finally, in Chapter 6, we end this work with our concluding remarks and provide directions for future work.

1.3 Contribution

The contributions of this thesis include:

1. Proposing simultaneous iterative hard thresholding algorithm as a very fast algorithm for reconstructing jointly sparse signals in Chapter 3 [3].

2. Proposing a modified iterative hard thresholding algorithm (MIHT) to incorporate additional information about temporal correlation in the sensor nodes in Chapter 4.

3. Extension of MIHT algorithm to the distributed scenarios and proposing the DCS-MIHT algorithm that can exploit both the spatial and temporal correlation in Chapter 4.

4. Proposing a Generalized Joint Sparsity Model (G-JSM) to model sparse correlated signals as well as a reconstruction algorithm for this class of signals using principal component analysis in Chapter 5 [4].

Chapter 2

Background

2.1 Compressive Sensing Theory

2.1.1 Sparse Representation

A length N signal $\mathbf{x}$ is called a $K-$sparse signal if it could be sparsely represented as a linear combination of K vectors chosen from a predefined sparsity basis $\Psi = [\psi_1, ..., \psi_N]$. In matrix notation we have

$$\mathbf{x} = \Psi\theta$$

where Ψ is an $N \times N$ sparsity basis such as wavelet, Gabor bases, curvelets, etc., that are widely used for representation of natural signals and θ is a vector that has K non-zero coefficients.

2.1.2 Incoherent Projections

The measurement scheme in Compressed Sensing is $M < N$ incoherent projection onto a second basis function as follows.

$$\mathbf{y} = \Phi\mathbf{x}$$

where $\mathbf{y}$ is an $M \times 1$ column vector and Φ is an $M \times N$ measurement matrix. Since $M < N$, for a given $\mathbf{y}$, there are infinitely many solutions for $\mathbf{x}$, however, the CS theory suggests that if certain conditions, in addition to sparsity hold, there will be a unique sparse solution. These conditions are called "incoherence" and "restricted isometry property".

According to the incoherence condition, the basis vectors ψ_i cannot be sparsely represented in the overcomplete dictionary of Φ. The coherence of a matrix can be defined as follows.

$$\mu = \max_{i \neq j} \langle \phi_i, \phi_j \rangle$$

This incoherence property holds for many pairs of bases such as delta spikes and the sine waves of a Fourier basis, or the Fourier basis and wavelets. Interestingly, this incoherence condition also holds with high probability between an arbitrary fixed basis and a randomly generated one. It can be proved that if $K < (1 + \mu^{-1})$, the sparsest solution of $\mathbf{y} = \Phi\mathbf{x}$ will be unique [5].

Another condition is called Restricted Isometry Property (RIP). For a measurement matrix Φ, we say that Φ satisfies the RIP with parameters (s, ϵ) if

$$(1-\epsilon)\|v\|_2 < \|\Phi v\|_2 < (1+\epsilon)\|v\|_2$$

holds for all $s-$sparse vectors. Often the notation ϵ_s is used to denote the smallest ϵ for which the above holds for all $s-$sparse signals. The intuition behind the RIP constant is that we want the measurement matrix to preserve the distance of all $s-$sparse signals after projecting them to a lower dimensional space.

One of the important properties of compressed sensing is universality which states that we can use the same measurement matrix to sense the signal no matter what basis the signal is sparse in. Thus, without loss of generality, we can assume that the sparsity basis matrix, Ψ, is the identity matrix, and hence $\mathbf{x}$ is the signal in the sparse domain.

2.1.3 Signal Recovery via ℓ_1 optimization

In order to find the sparsest solution of $\mathbf{y} = \Phi\mathbf{x}$, we have to solve the following optimization problem.

$$\hat{\mathbf{x}} = \arg\min \|\mathbf{x}\|_0 \text{ s.t. } \mathbf{y} = \Phi\mathbf{x}$$

Unfortunately solving the ℓ_0 optimization is known to be NP-complete. However, it has been recently shown that instead of solving the ℓ_0 optimization, we can convexify the problem and solve an ℓ_1 minimization and obtain exactly the same result as long as $M > cK\log N$ where c is a constant.

$$\hat{\mathbf{x}} = \arg\min \|\mathbf{x}\|_1 \text{ s.t. } \mathbf{y} = \Phi\mathbf{x}$$

This is a more approachable optimization problem which is called Basis Pursuit and can be solved using linear programming algorithm whose computational complexities are polynomial in N. However, while in the ℓ_0 minimization we need $K+1$ measurements for perfect reconstruction, we would need more measurements in order to recover sparse signals via Basis Pursuit. We require $M = cK$ measurements, where $c > 1$ is an oversampling factor. If we define $S = \frac{K}{N}$ as the sparsity rate, we will have the following result.

Theorem 2.1: [6] Set $K = SN$ with $0 < S \ll 1$. Then there exists an oversampling factor $c(S)$ such that, for a $K-$sparse signal $\mathbf{x}$ in basis Ψ, the following statements hold.

1. The probability of recovering x via Basis Pursuit from $(c(S)+\epsilon)K$ random projections, $\epsilon > 0$ converges to one as $N \to \infty$

2. The probability of recovering x via Basis Pursuit from $(c(S)-\epsilon)K$ random projections, $\epsilon > 0$ converges to zero as $N \to \infty$

As a rule of thumb, the oversampling factor $c(S)$ in Theorem 2.1 satisfies $c(S) \simeq log_2(1+S^{-1})$.

2.1.4 Signal Recovery via Matching Pursuit Algorithms

In this section, we describe *Orthogonal Matching Pursuit* algorithm (OMP) proposed in [7] for reconstruction of sparse signals. Let $\mathbf{x}$ be a K-sparse signal in R^N and our measurement vector is $\mathbf{y} = \Phi\mathbf{x}$. Since $\mathbf{y}$ is a linear combination of K columns of Φ, it is natural to think of signal recovery as a problem dual to sparse approximation. Therefore, sparse approximation algorithms can be used to recover sparse signals. We use an iterative algorithm to recover the support of the sparse signal. The idea is that at each iteration, we pick a column of Φ that has the maximum correlation with the remaining part of the measurement vector and then subtract its contribution from $\mathbf{y}$ and iterate on the residual. More precisely, the algorithm is as follows.

OMP for Signal Recovery:
Reconstruction: 1) Initialize the residual $\mathbf{r}_0 = \mathbf{y}$ and the index set $\Lambda_0 = \varnothing$ and the iteration counter $t = 1$. 2) Find λ_t such that: $\lambda_t = \arg\max_{j=1,\ldots,N} \lvert\langle \mathbf{r}_{t-1}, \varphi_j \rangle\rvert$ 3) Augment the index set by the column that is selected in the previous stage: $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$ and $\Phi_t = [\Phi_{t-1}, \varphi_{\lambda_t}]$ 4) Solve the following least square problem to obtain the new estimate: $\mathbf{x}_t = \arg\min_x \lVert \mathbf{y} - \Phi_t \mathbf{x} \rVert_2$ 5)$\mathbf{r}_t = \mathbf{y} - \Phi_t \mathbf{x}_t$ 6)if $t < K$ go to step 2, otherwise proceed to the next step. 7)Estimate the non-zero coefficients of the sparse signal by multiplying the pseudoinverse of the matrix of chosen atoms by the measurement vector: $\mathbf{x}_{\Lambda_t} = {\Phi_t}^{\dagger} \mathbf{y}$

In Compressed Sensing applications, OMP requires $M > 2K \ln(N)$ measurements to succeed with high probability. It can be proved that if $K < \frac{1}{2}(1 + \mu^{-1})$, both $\ell_1$minimization and OMP will recover the unique sparsest solution of $\mathbf{y} = \Phi\mathbf{x}$. Gilbert and Tropp in [7] proved the following theorem that predicts the reconstruction rate of OMP algoritm.

Theorem 2.2: [7] Fix $\delta \in (0, 0.36)$ and let Φ be an $M \times N$ Gaussian matrix with $M \geq CMlog(N/K)$. Let $\mathbf{x}$ be a $K-$sparse signal with the length N. Then with probability exceeding $1 - 2\delta$, OMP correctly recovers the signal $\mathbf{x}$ from the measurements $\mathbf{y} = \Phi\mathbf{x}$.

It can be proved that a similar theorem can be stated for the subgaussian matrices.

2.1.5 Signal Recovery via Thresholding Algorithms

Iterative Hard Thresholding (IHT) [8] is a class of low computational algorithms, which has recently been proposed for reconstruction of sparse signals. Starting from $\mathbf{x}^0 = 0$, IHT iteratively finds the sparsest solution of $\mathbf{y} = \Phi\mathbf{x}$ using the following iteration.

$$\mathbf{x}^{n+1} = H_K(\mathbf{x}^n + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^n)) \tag{2.1}$$

where H_K is a nonlinear operator that sets all but the K largest (in magnitude) coefficients of its input vector to zero. It can be shown that if $\lVert\Phi\rVert_2 < 1$ the algorithm is guaranteed to converge to the local minimum of the cost function $\lVert \mathbf{y} - \Phi\mathbf{x} \rVert_2$ under the constraint that $\mathbf{x}$ is $K-$sparse.

Starting from zero vector for all signals, at each iteration, the algorithm moves in the direction $\Phi^T(\mathbf{y} - \Phi\mathbf{x}^n)$ which is perpendicular to the plane $\mathbf{y} = \Phi\mathbf{x}$. Thus, this direction is the best direction in terms of minimizing $\lVert \mathbf{y} - \Phi\mathbf{x} \rVert_2$. But the question is how far we should go along this direction? It is easy to see that we have: $\mathbf{x}^n + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^n) = \mathbf{x} + (\Phi^T\Phi - I)(\mathbf{x} - \mathbf{x}^n)$. So we can regard $(\mathbf{x} - \mathbf{x}^n)$ as the "error term" and $(\Phi^T\Phi - I)(\mathbf{x} - \mathbf{x}^n)$ as the "noise term" that is added to our actual signal and its contribution should be reduced at each iteration. But for a random Gaussian matrix, due to the pseudo-orthogonality of the columns, we have $E[\Phi^T\Phi] = MI_{N\times N}$. Thus, if we scale the sensing matrices by a fixed scaling factor of $\frac{1}{\sqrt{M}}$, the matrix $(\Phi^T\Phi - I)$ would become a Gaussian matrix that doesn't correlate with the error term. Therefore, by multiplying the error term by this random matrix, the variance of error would get reduced. We can show that we can further improve the performance of reconstruction by using an adaptive scaling factor to minimize the cost function $\|\mathbf{y}-\Phi\mathbf{x}\|_2$ at each iteration. The essential properties of the noise term ($\mathbf{n}$) can be found in the following lemma [9].

Lemma 1. Let Φ be a random matrix whose entries have i.i.d. distribution with $E(\Phi_{ij}) = 0$, $E(\Phi_{ij}^2) = \mathcal{O}(\frac{1}{M})$ and $E(\Phi_{ij}^4) = \mathcal{O}(\frac{1}{M^2})$. If $\mathbf{n} = (\Phi^T\Phi - I)\mathbf{e}$, then $E(n_i) = 0$, $E(n_i n_j) = \mathcal{O}(\frac{1}{M})$ for $i \neq j$ and $E(n_i^2) = \mathcal{O}(\frac{1}{M}) + \frac{1}{M}\|\mathbf{e}\|_2^2$.

According to Lemma 1 and the Central Limit Theorem, we can conclude that if $M \to \infty$, the noise term has an i.i.d. Gaussian distribution with $E(n_i) = 0$, $E(n_i^2) = \frac{1}{M}\|\mathbf{e}\|_2^2$ and $E(n_i n_j) = 0$ for $i \neq j$.

Thus, at each iteration, we are essentially dealing with a detection problem where we wish to detect whether a coefficient of $\mathbf{x}$ is zero or non-zero based on its noisy observation $\mathbf{z}^{n+1} = \mathbf{x} + \mathbf{n}$. This detection should be such that the probability of correct decision is maximized. Without any prior knowledge (uniform probability) the maximum-likelihood (ML) function is given by $f(z_i^{n+1}|x_i = 0) = \frac{1}{\sqrt{2\pi\sigma_n^2}}e^{-(z_i^{n+1})^2/2\sigma_n^2}$. So if we evaluate the ML-function at each coefficient of $\mathbf{z}^{n+1}$, we will see that the k largest coefficients are the coefficients that minimize the ML-function the most (have higher chance of being non-zero). Thus, the maximum-likelihood decision on the support set leads to selecting the K largest coefficients in absolute value for the case of uniform prior probability of being zero. After finding the K largest coefficients at each iteration, we keep them and set the rest of the coefficients to zero and the whole process will be repeated again until the algorithm converges. A sufficient condition for the convergence of IHT algorithm has been derived in [10] as follows.

Theorem 2.3: Suppose that $K < \frac{1}{3.1}\mu^{-1}$ and $\frac{|x(i)|}{|x(i+1)|} < 3^{l_i-4}, \forall i, 1 \leq i < K$. Then IHT finds the true support set in at most $\sum_{i=1}^{K} l_i + k$ steps and after this step the support set would remain the same and the reconstruction error will go towards zero exponentially.

2.1.6 Probabilistic Approaches of Reconstruction

Recently, the probabilistic and Bayesian approaches of compressed sensing have received a lot of attentions. In the Bayesian viewpoint of compressed sensing, it is assumed that the coefficients of the signal are drawn from a "sparsity-promoting" or "compressible" prior [11]. Sparsity-promoting priors, in simple words, are the distributions whose independent and identically distributed realization result in compressible signals. In these priors, the small values are usually heavily weighted while the large values have a small probability of occurrence. Several sparsity-promoting priors have been proposed in the literature. Laplacian prior is used in [12] for compressed sensing reconstruction. It is easy to show that if we ask for the solution that maximizes the Laplacian likelihood, we will obtain the famous ℓ_1 minimization algorithm. In [13], the authors have placed a Student's t prior on the coefficients of the signal. Student's t distribution has received a lot of attention in the literature of Bayesian inference and relevance vector machines (RVMs) [14]. It has been shown that by exploiting a probabilistic sparse Bayesian learning framework we can reconstruct the sparse signals. A mixture of Gaussian model has also been proposed in [15] and the signal is reconstructed using factor graphs. Approximate message passing (AMP) [9, 16, 17] is a recent breakthrough in sparse reconstruction algorithms and could be considered as a fast approximation of a sum-product belief propagation algorithm in a factor graph.

2.2 Joint Sparsity Models

In this section, we generalize the concept of a signal being sparse in some basis to the notion of an ensemble of signals being jointly sparse. We consider a joint sparsity model proposed in [18, 19] which could be applied in different situations.

A practical situation well-modeled by this joint sparsity model is where multiple sensors acquire replicas of the same Fourier-sparse signal but with phase shifts and attenuations caused by signal propagation. In many cases it is critical to recover each one of the sensed signals, such as in many acoustic localization and array processing algorithms. Consider for example a sensor network, in which we have J signals of length N, where $\mathbf{x}_j, j \in \{1, 2, ..., J\}$ represents the signals of each sensor node. For each sensor node, we have an $M \times N$ measurement matrix called Φ_j. Thus, for each sensor node, we can find a measurement vector as $\mathbf{y}_j = \Phi_j \mathbf{x}_j$. We will use a *Joint Sparsity Model* (JSM) that considers both intra-sensor correlation (sparsity) and inter-sensor correlation among sensors. In this model, all signals are constructed from the same sparse set of basis vectors, but with different coefficients; that is

$$\mathbf{x}_j = \Psi \theta_j \ , \ j \in \{1, 2, ..., J\}$$

In this model, all signals share a *common support set* meaning that the location of non-zero elements of θ_j is the same for different sensors.

Another application is modeling the "temporal correlation" of video frames, or in general, time-sequences of sparse signals whose sparsity patterns change slowly over time [20]. In many applications due to the strong temporal dependencies, the support set of the signals change slowly over time and by exploiting this temporal correlation, we can do better than performing CS at each time separately. So, by using this model, we can capture the temporal correlations in the time-varying sparse signals.

2.3 Reconstruction of Jointly Sparse Signals

In this section, we review the conventional algorithms for reconstructing jointly sparse signals. The literature has mostly focused on problems where the sensing matrix is the same among different sensors. In this case, we can form the matrix $Y = [\mathbf{y}_1, ..., \mathbf{y}_J]$ and $X = [\mathbf{x}_1, ..., \mathbf{x}_J]$ and thus we will have

$$Y = \Phi X$$

Now we review some of the reconstruction algorithms of jointly sparse signals in the literature.

2.3.1 Simultaneous Orthogonal Matching Pursuit (SOMP)

Tropp and Gilbert in [21] have proposed an algorithm called *Simultaneous Orthogonal Matching Pursuit* (SOMP), which is an extension of OMP for multiple signals with the same support. The key idea in SOMP is that at each iteration, we select the column index that accounts for the greatest amount of residual energy *across all signals*. It can be shown that if the number of signals tends to infinity the number of measurements per signal required for perfect reconstruction will tend to the sparsity level. The SOMP algorithm is a greedy algorithm that performs the following procedure:

1. At the first iteration, set $Y_0 = Y$, $S_0 = \varnothing$ and $t = 1$.

2. Select ℓ_t such that $\|\phi_{\ell_t}^* Y_{t-1}\|_2 = \max_{1\le\ell\le N} \|\phi_\ell^* Y_{t-1}\|_2$ and set $S_t = S_{t-1} \bigcup \{\ell_t\}$.

3. After the t iterations, $S_t = \{\ell_j\}_{j=1}^t$ and $Y_t = (I - P_{S_t})Y$, where P_{S_t} is the orthogonal projection onto $\text{span}\{\phi_{\ell_j}\}_{j=1}^t$.

4. $t \leftarrow t + 1$ and goto step 2.

The worst case analysis of SOMP shows that a sufficient condition for SOMP to succeed is

$$\max_{j \in \text{supp} X} \|\Phi_S^\dagger \phi_j\| < 1$$

where $S = supp\{X\}$ and $\Phi_S^\dagger$ is the pseudoinverse of the columns of Φ restricted to the set S. A sufficient condition for the convergence of the algorithm is given by

$$\|X\|_0 < \frac{1}{2}(\frac{1}{\mu} + 1)$$

where μ is the mutual coherence of the matrix Φ defined by $\mu = \max_{i\neq j} \langle \phi_i, \phi_j \rangle$. Note that the above sufficient condition is the same as the sufficient condition of the convergence of the orthogonal matching pursuit algorithm stated in section 2.1.4 implying that the condition does not improve by increasing the number of signals. This motivates the need for the average case analysis of SOMP algorithm discussed in [22].

2.3.2 2-Thresholding

At each iteration of SOMP algorithm, we subtract the contribution of the estimated support set from the measurement matrix. A faster approach is to estimate the support set S of the signal in just one iteration. This is the key idea of 2-thresholding algorithm [22]. In this algorithm, we select a set S with $|S| = K$ such that

$$\|\phi_l^* Y\|_2 > \|\phi_j^* Y\|_2 \text{ for all } l \in S \text{ and } j \notin S$$

If we can estimate the support set of X, then we can recover the nonzero coefficients of X by the equation $X_S = \Phi_S^\dagger Y$. This algorithm is much faster than SOMP algorithm at the expense of worse reconstruction rate.

2.3.3 ReMBO algorithm

Reduce Multiple Measurement Vector and Boost (ReMBo) algorithm has been proposed in [23] by Mishali and Eldar. This algorithm addresses the reconstruction of jointly sparse signals by reducing it to a series of single sensor compressed sensing problems. The key idea of ReMBo algorithm is that if we have the following equation

$$Y = \Phi X$$

Then for any random vector $\mathbf{v}$ we have

$$\mathbf{y} = Y\mathbf{v} = \Phi X \mathbf{v} = \Phi \overline{\mathbf{x}} \tag{2.2}$$

Using the above equation, Mishali and Eldar [23] proposed the ReMBo algorithm as follows.

1. Set the maximum number of iterations as MaxIters, set $i = 1$ and $Flag = F$.

2. While $i \le MaxIters$ and $Flag = F$, generate a random vector $\mathbf{v}$ and then using equation (2.2) to generate a random single sensor compressed sensing problem.

   - If the SMV problem has a $K-$sparse solution, then we let S be the support of the solution vector, and set $Flag = T$.
   - Otherwise, $i \leftarrow i + 1$

3. If $Flag = T$, find the nonzero components of X by the equation $X_S = \Phi_S^{\dagger} Y$

2.3.4 Mixed Norm Approach

The mixed norm approach [24] addresses reconstruction of the jointly sparse signals based on a convex relaxation method. Instead of solving the original problem, the mixed norm approach solves the following convex optimization problem.

$$\text{minimize} \|X\|_{p,q} \qquad 1 < p, q < 2$$

$$\text{subject to } Y = \Phi X$$

where

$$\|X\|_{p,q} = (\sum_{j=1}^{J} \|\mathbf{x}_j\|_p^q)^{\frac{1}{q}}.$$

When $p = 2$ and $q = 1$, it can be showed [24] that a sufficient condition for perfect reconstruction is as follows.

$$\max_{j \notin S} \|\Phi_S^{\dagger} \phi_j\|_2 < 1$$

where S is the support set of the signal.

2.3.5 Distributed Compressed Sensing SOMP

The DCS-SOMP algorithm proposed in [19] is a modified version of SOMP algorithm for the case that the measurement matrices are different for each sensor node. So we are assuming that $\mathbf{y}_j = \Phi_j \mathbf{x}_j$ and $\mathbf{x}_j$s are sharing a same support set.

In each step of DCS-SOMP, we pick the column that accounts for the greatest amount of residual energy across all signals and then orthogonalize the remaining columns. More precisely, the algorithm is as follows.

1. Initialize $\ell = 1$, $\hat{\beta}_j = 0$, $r_{j,0} = y_j$ and $\hat{S} = \varnothing$

2. Select the dictionary vector that maximizes the value of the sum of the magnitudes of the projections of the residual, and add its index to the set of selected indices.

$$n_\ell = \text{argmax}_{n=1,\ldots,N} \sum_{j=1}^{J} \frac{\langle r_{j,\ell-1}, \phi_{j,n} \rangle}{\|\phi_{j,n}\|_2}$$

$$\hat{S} = [\hat{S}; n_\ell]$$

3. Orthogonalize the selected basis vector against the orthogonalized set of previously selected dictionary vectors
$$\gamma_{j,\ell} = \phi_{j,n_\ell} - \sum_{t=0}^{\ell-1} \frac{\langle \gamma_{j,t}, \phi_{j,n_\ell} \rangle}{\|\gamma_{j,t}\|_2^2} \gamma_{j,t}$$

4. Update the estimate of the coefficients for the selected vector and residuals
$$\hat{\beta}_j(\ell) = \frac{\langle r_{j,\ell-1}, \gamma_{j,\ell} \rangle}{\|\gamma_{j,\ell}\|_2^2}$$
$$r_{j,\ell} = r_{j,\ell-1} - \frac{\langle r_{j,\ell-1}, \gamma_{j,\ell} \rangle}{\|\gamma_{j,\ell}\|_2^2} \gamma_{j,\ell}$$

5. Check for convergence. If the algorithm has not converged, increment ℓ and go to step 2.

It can be proved that for the perfect reconstruction of all signals, we would need $M = \hat{c}K$ measurements per signal where $\hat{c} \to 1$ as $J \to \infty$.

Chapter 3

Reconstruction of Jointly Sparse Signals via Iterative Hard Thresholding

3.1 Introduction

In Section 2.2, we introduced the joint sparsity models and its reconstruction algorithms such as SOMP [21], Mixed norm approach [24], M-FOCUSS [25], ReMBo [23], and M-SBL [26]. All of these algorithms are fundamentally based on the assumption that the sensing matrix and sparsity basis are fixed among all sensors. However, there might be cases that our signals are sparse in different basis. Also, one can intuitively expect that using different sensing matrices in the sensors could introduce more diversity into the measurements and improve the performance of the algorithm. In [18], an extension of SOMP called DCS-SOMP is proposed to address the problem of distributed compressed sensing with different sensing matrices. The DCS-SOMP algorithm has been discussed in Section 2.3.5. This algorithm needs the solution of a least square problem at each of its iterations, which can result in high computational cost of DCS-SOMP and hence limits its application in practice.

In many of the applications of the joint sparsity model, we encounter extremely large problem sizes for which algorithms with low computational complexity are required. We discussed the *iterative hard thresholding* in Section 2.1.5 which has been recently proposed and is faster than the ℓ_1−minimization and greedy algorithms for compressed sensing. In this chapter, we extend the iterative hard thresholding algorithm to jointly sparse signals and propose a very low computational complexity reconstruction algorithm, which we call it *Simultaneous Iterative Hard Thresholding* (S-IHT), to address the problem of reconstruction of jointly sparse signals with different sparsity basis and sensing matrices. We show that our algorithm is much faster than the state of the art algorithms, such as DCS-SOMP, while showing similar reconstruction rate. We will also investigate the performance of our proposed algorithm analytically by giving conditions under which the exact reconstruction could happen.

3.2 Simultaneous Iterative Hard Thresholding Algorithm

In this section, we first formally state our algorithm [3] and investigate its performance analytically. We then use a simple modification of the algorithm to increase its rate of convergence.

Iterative Hard Thresholding (IHT) has been discussed in Section 2.1.5. In this work, we extend the iterative hard thresholding to jointly sparse signals and propose a new reconstruction algorithm, which we call it the *Simultaneous Iterative Hard Thresholding* (S-IHT). In this algorithm, it is assumed that the $K-$sparse signals are sharing a common support set. We will show that by utilizing this additional information and simultaneously reconstructing the signals, we can obtain substantial improvement in reconstruction rate over separate compressed sensing. We further assume that the measurement scheme is a random projection by a matrix whose entries are i.i.d. Gaussian random variables with the variance 1. Starting from zero vector for all signals, at each iteration, the algorithm moves in the direction $\Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^n)$ which is perpendicular to the plane $\mathbf{y}_j = \Phi_j\mathbf{x}_j$. Thus, this direction is the best direction in terms of minimizing $\|\mathbf{y}_j - \Phi_j\mathbf{x}_j\|_2$. But the question is how far we should go along this direction? It is easy to see that we have: $\mathbf{x}_j^n + \Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^n) = \mathbf{x}_j + (\Phi_j^T\Phi_j - I)(\mathbf{x}_j - \mathbf{x}_j^n)$. So we can regard $(\mathbf{x}_j - \mathbf{x}_j^n)$ as the "error term" and $(\Phi_j^T\Phi_j - I)(\mathbf{x}_j - \mathbf{x}_j^n)$ as the "noise term" that is added to our actual signal and its contribution should be reduced at each iteration. But for a random Gaussian matrix, due to the pseudo-orthogonality of the columns, we have $E[\Phi_j^T\Phi_j] = MI_{N\times N}$. Thus, if we scale the sensing matrices by a fixed scaling factor of $\frac{1}{\sqrt{M}}$, the matrix $(\Phi_j^T\Phi_j - I)$ would become a Gaussian matrix that doesn't correlate with the error term. Therefore, by multiplying the error term by this random matrix, the variance of error would get reduced. Although we are using a fixed scaling factor to satisfy the convergence conditions, in Section 3.2.2, we will show that we can further improve the performance of reconstruction by using an adaptive scaling factor to minimize the cost function $\|\mathbf{y}_j - \Phi_j\mathbf{x}_j\|_2$ at each iteration. At the end of each iteration, we have a rough estimate of our signals $\mathbf{x}_j$. Ideally, all of the estimates should be $K-$sparse with a same support set, but this is not the case due to the existence of the noise term. So we can simply force all of the signals to have a same support set as follows. We first take the average of absolute values of the estimates and choose the indices of the $K-$largest coefficients of the average as the estimated common support set. By averaging over the absolute values of estimates, we are essentially reducing the contribution of the Gaussian noise term of each signal. Now we make each signal $K-$sparse according to the estimated common support set of previous stage and direct the algorithm to the $K-$sparse solutions of $\mathbf{y}_j = \Phi_j\mathbf{x}_j$, $j \in \{1, 2, ..., J\}$ with a common support set constraint. Thus, the proposed algorithm is summarized in Figure 3.1.

where H_{Γ^n} is a nonlinear operator that sets all but the indices of Γ^n to zero.

3.2.1 Theoretic Properties of S-IHT

Theorem 3.1. *Assume that the non-zero values of $\mathbf{x}_j$ are all equal to 1. If $J \to \infty$ and $M > 0.63K$, then S-IHT can recover the true support set of the signals at the first iteration with a probability approaching 1.*

Proof: Without loss of generality we assume that the non-zero elements of $\mathbf{x}_j$ are the first K elements. The idea of the proof is as follows. We will show that if $J \to \infty$ and $M > 0.63K$, the variance of the coefficients of $\mathbf{w}^1$ tends to zero while the expectation of the first K coefficients of $\mathbf{w}^1$ is greater than the expectation of the rest of the coefficients. So as $J \to \infty$ with a probability approaching 1, we

Simultaneous Iterative Hard Thresholding:
Sensing: 1) $\Phi'_j = randn(M,N)$, $\Phi_j = \frac{1}{\sqrt{M}}\Phi'_j$, $\mathbf{y}_j = \Phi_j \mathbf{x}_j$
Reconstruction: 1) Initialize: $\mathbf{x}_j^0 = 0, n = 0$ 2) **while** $\sum_{j=1}^{J} \lVert \mathbf{y}_j - \Phi_j \mathbf{x}_j \rVert_2 > \epsilon$ **do** 3) $\mathbf{z}_j^{n+1} = \left| \mathbf{x}_j^n + \Phi_j^T(\mathbf{y}_j - \Phi_j \mathbf{x}_j^n) \right|$, $j \in \{1,2,...,J\}$ 4) $\mathbf{w}^{n+1} = \frac{1}{J}\sum_{j=1}^{J} \mathbf{z}_j^{n+1}$ 5) $\Gamma^n = \{$indices of K largest coefficients of $\mathbf{w}^{n+1}\}$ 6) $\mathbf{x}_j^{n+1} = H_{\Gamma^n}(\mathbf{x}_j^n + \Phi_j^T(\mathbf{y}_j - \Phi_j \mathbf{x}_j^n))$, $j \in \{1,2,...,J\}$ 7) $n \leftarrow n+1$ 8) **end while**

Figure 3.1: Simultaneous Iterative Hard Thresholding

can distinguish the true support set. We know that

$$\mathbf{z}_j^1 = \left| \mathbf{x}_j^0 + \Phi_j^T(\mathbf{y}_j - \Phi_j \mathbf{x}_j^0) \right| = \left| \Phi_j^T \mathbf{y}_j \right| = \left| \Phi_j^T \Phi_j \mathbf{x}_j \right| \tag{3.1}$$

If $1 \le i \le K \Rightarrow E[\mathbf{z}_j^1(i)] = E[| \, \lVert \Phi_{j,i} \rVert^2 + \sum_{\substack{p=1 \\ p \neq i}}^{K} \langle \Phi_{j,p}, \Phi_{j,i} \rangle |]$

By using the Central Limit Theorem, we have $\langle \Phi_{j,p}, \Phi_{j,i} \rangle \sim \mathcal{N}(0, \frac{1}{M})$ and $\lVert \Phi_{j,i} \rVert^2 \sim \mathcal{N}(1, \frac{2}{M})$. Thus $\mathbf{z}_j^1(i) = | \, \lVert \Phi_{j,i} \rVert^2 + \sum_{\substack{p=1 \\ p \neq i}}^{K} \langle \Phi_{j,p}, \Phi_{j,i} \rangle |$ has a folded normal distribution and its expectation is given as $E[|\mathbf{z}_j^1(i)|] = \sigma\sqrt{\frac{2}{\pi}} e^{\frac{-\mu^2}{2\sigma^2}} + \mu[1 - 2\mathcal{Q}(\frac{\mu}{\sigma})]$ where (μ, σ) are the mean and standard deviation of the normal random variable, respectively. Now using $\mathcal{Q}(x) < \frac{1}{x}\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ we can easily conclude $E[\mathbf{w}^1(i)] = \frac{1}{J}\sum_{j=1}^{J} E[\mathbf{z}_j^1(i)] > 1$.

The $VAR[\mathbf{z}_j^1(i)]$ can be found as follows.

$$VAR[\mathbf{z}_j^1(i)] = E[\mathbf{z}_j^1(i)^2] - E[\mathbf{z}_j^1(i)]^2 \tag{3.2}$$

$$E[\mathbf{z}_j^1(i)^2] = E[\{\lVert \Phi_{j,i} \rVert^2 + \sum_{\substack{p=1 \\ p \neq i}}^{K} \langle \Phi_{j,p}, \Phi_{j,i} \rangle\}^2] = 1 + \frac{K+1}{M} \tag{3.3}$$

$$\Rightarrow VAR[\mathbf{z}_j^1(i)] < \frac{K+1}{M} \tag{3.4}$$

$$VAR[\mathbf{w}^1(i)] = \frac{1}{J^2}\sum_{j=1}^{J} VAR[\mathbf{z}_j^1(i)] < \frac{K+1}{MJ} \tag{3.5}$$

$$\boxed{\lim_{J \to \infty} VAR[\mathbf{w}^1(i)] = 0} \tag{3.6}$$

If $K+1 \le i \le N \Rightarrow E[\mathbf{z}_j^1(i)] = E[| \sum_{p=1}^{K} \langle \Phi_{j,p}, \Phi_{j,i} \rangle |]$

By using the Central Limit Theorem, we can conclude that $|\sum_{p=1}^{K}\langle\Phi_{j,p},\Phi_{j,i}\rangle|$ has a folded normal distribution. Thus, we would have $E[\mathbf{w}^1(i)] = \frac{1}{J}\sum_{j=1}^{J}E[\mathbf{z}_j^1(i)] = \sqrt{\frac{2K}{\pi M}}$.

The $VAR[\mathbf{z}_j^1(i)]$ can be found as follows.

$$VAR[\mathbf{z}_j^1(i)] = E[\mathbf{z}_j^1(i)^2] - E[\mathbf{z}_j^1(i)]^2 \tag{3.7}$$

$$E[\mathbf{z}_j^1(i)^2] = E[\{\sum_{p=1}^{K}\langle\Phi_{j,p},\Phi_{j,i}\rangle\}^2] = \frac{K}{M} \tag{3.8}$$

$$\Rightarrow VAR[\mathbf{z}_j^1(i)] = \frac{(\pi-2)K}{\pi M} \tag{3.9}$$

$$VAR[\mathbf{w}^1(i)] = \frac{1}{J^2}\sum_{j=1}^{J}VAR[\mathbf{z}_j^1(i)] = \frac{(\pi-2)K}{\pi MJ} \tag{3.10}$$

$$\boxed{\lim_{J\to\infty}VAR[\mathbf{w}^1(i)] = 0} \tag{3.11}$$

So we just need to prove that the expectation of the first K coefficients of $\mathbf{w}^1$ is greater than the expectation of the other coefficients and this happens if we have: $1 > \sqrt{\frac{2K}{\pi M}} \Rightarrow \boxed{M > 0.63K}$.

In Theorem 3.1, we proved the performance of the algorithm just for the case that the non-zero coefficients are all equal to 1. However, our simulation results have shown that the results also hold for general non-zero coefficients. In this theorem, we also proved that we are able to recover the location of non-zero coefficients of the signals with less than K measurements per sensor. However, it is clear that we would need more than K measurements to find the values of the non-zero coefficients. In the following, we will show that if the algorithm converges to the true support set at some iteration and $M \geq K+1$, then the algorithm will converge to the non-zero elements up to any precision.

Let us assume we have found the true support set at iteration n. If we define the error vectors as $\mathbf{e}_j^{new} = \mathbf{x}_j - \mathbf{x}_j^{n+1}$ and $\mathbf{e}_j^{old} = \mathbf{x}_j - \mathbf{x}_j^{n}$, we will show that $E[\mathbf{e}_j^{new}] = 0$ and if $E[\mathbf{e}_j^{old}(i)^2] < \sigma^2, \forall i$ then $E[\mathbf{e}_j^{new}(i)^2] < \frac{(K+1)}{M}\sigma^2, \forall i$.

For the first K coefficients of $\mathbf{x}_j^{n+1}$ we have:

$$\begin{aligned}\mathbf{x}_j^{n+1} &= \mathbf{x}_j^n + \Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^n)\\ &= \mathbf{x}_j + (\Phi_j^T\Phi_j - I)(\mathbf{x}_j - \mathbf{x}_j^n)\\ &= \mathbf{x}_j + (\Phi_j^T\Phi_j - I)\mathbf{e}_j^{old}\end{aligned} \tag{3.12}$$

Here we prove that $E[\mathbf{e}_j^{new}] = 0$.

$$E[-\mathbf{e}_j^{new}(i)] = E[\mathbf{x}_j(i)^{n+1} - \mathbf{x}_j(i)] = E[\mathbf{e}_j^{old}(i)]E[(\|\Phi_{j,i}\|^2 - 1)] + \sum_{\substack{p=1\\p\neq i}}^{K}E[\mathbf{e}_p^{old}(i)]E[\langle\Phi_{j,p},\Phi_{j,i}\rangle] \tag{3.13}$$

$\Rightarrow E[\mathbf{e}_j^{new}(i)] = 0$

So we just prove that the upper bound of the variance of the error is decreasing after each iteration.

In other words, the error at iteration $n+1$ is a random variable whose variance is less than the variance of the error at iteration n. But we know that the support set of $\mathbf{e}_j^{old}$ and $\mathbf{e}_j^{new}$ are the same. Thus, we have

$$E[\mathbf{e}_j^{new}(i)^2] = E[(\mathbf{x}_j(i) - \mathbf{x}_j(i)^{n+1})^2] \tag{3.14}$$

$$= E[\mathbf{e}_j^{old}(i)^2(\|\Phi_{j,i}\|^2 - 1)^2] + E[\sum_{\substack{p=1\\p\neq i}}^{K} \mathbf{e}_j^{old}(p)^2\langle\Phi_{j,p},\Phi_{j,i}\rangle^2]$$

We know that $\mathbf{e}_j^{old}$ and $\langle\Phi_{j,p},\Phi_{j,i}\rangle$ are almost independent, thus if we assume $E[\mathbf{e}_j^{old}(i)^2] < \sigma^2, \forall i$ we would have

$$E[\mathbf{e}_j^{new}(i)^2] < \frac{(K+1)}{M}\sigma^2 \tag{3.15}$$

Therefore, we can conclude that if $M \geq K+1$, the upper bound of the variance of the error will be reduced at each iteration.

In Theorem 3.1, we showed the performance of the algorithm for the first iteration, however, in Section 4.5, we will show by simulation results that as we increase the number of iterations, the performance of the algorithm gets better. Here, we just provide an intuitive explanation of this behavior. When we run the algorithm, at the end of the first iteration, some of the indices of the true support set will get into the estimated support set. It can be shown that with high probability, these indices will remain in the estimated support set of the next iterations. But after a few iterations the variance of the error of the coefficients of those indices gets smaller and the algorithm essentially removes their contributions from the measurement vectors. Thus, other indices of the true common support set find the chance to get into the estimated support set of the subsequent iterations and the same process will happen again until the true common support set is identified. Once the true support set is found, according to the previous argument, the algorithm converges to the non-zero elements up to any precision.

3.2.2 S-IHT Algorithm with Adaptive Scaling

In order to increase the rate of convergence of the algorithm, we use an adaptive scaling factor at each iteration. Instead of using $\frac{1}{\sqrt{M}}$ as the normalization factor, we use a μ factor for scaling Φ matrix that is determined adaptively at each iteration.

Let us assume at some iteration, we have found the true common support set, which we denote by Γ. The idea behind this adaptive factor is that we would like to minimize $f(\mathbf{x}_{j\Gamma}^{n+1}) \triangleq \|\mathbf{y}_j - \Phi_{j\Gamma}\mathbf{x}_{j\Gamma}^{n+1}\|_2^2$ and this would be done using gradient descent algorithm by moving along the direction of $g(\mathbf{x}) = -\frac{1}{2}\nabla f(\mathbf{x}) = \Phi^T(\mathbf{y} - \Phi\mathbf{x})$ at each iteration as follows.

$$\mathbf{x}_{j\Gamma}^{n+1} = \mathbf{x}_{j\Gamma}^{n} + \mu_j^n \Phi_{j\Gamma}^T(\mathbf{y}_j - \Phi_{j\Gamma}\mathbf{x}_{j\Gamma}^{n}) \tag{3.16}$$

Thus, the optimum step to maximally reduce $f(\mathbf{x}_{j\Gamma}^{n+1})$ is :

$$\begin{aligned}\mu_j^n &= \arg\min_{\mu_j^n} f(\mathbf{x}_{j\Gamma}^{n} + \mu_j^n \Phi_{j\Gamma}^T(\mathbf{y}_j - \Phi_{j\Gamma}\mathbf{x}_{j\Gamma}^{n}))\\ &= \frac{\mathbf{g}_{j\Gamma}^T\mathbf{g}_{j\Gamma}}{\mathbf{g}_{j\Gamma}^T\Phi_{j\Gamma}^T\Phi_{j\Gamma}\mathbf{g}_{j\Gamma}}\end{aligned} \tag{3.17}$$

where

$$\mathbf{g}_{j\Gamma} = \Phi_{j\Gamma}^T(\mathbf{y}_j - \Phi_{j\Gamma}\mathbf{x}_{j\Gamma}^n))$$

The first iteration of S-IHT with adaptive scaling is the same as the non-adaptive case and starts with $\mu_j^0 = \frac{1}{M}$. But for the subsequent iterations, μ_j^n is computed adaptively for the j–th signal using (5.1). A similar idea for increasing the convergence rate of IHT has been proposed in [27].

The modified algorithm will be as follows.

Adaptive Simultaneous Iterative Hard Thresholding:
Sensing:
1) $\Phi_j = randn(M,N)$, $\mathbf{y}_j = \Phi_j\mathbf{x}_j$
Reconstruction:
1) Initialize: $\mathbf{x}_j^0 = 0$, $n = 0$, $\mu_j^0 = \frac{1}{M}$
2) **while** $\sum_{j=1}^{J} \|\mathbf{y}_j - \Phi_j\mathbf{x}_j\|_2 > \epsilon$ **do**
3) $\mathbf{z}_j^{n+1} = \left|\mathbf{x}_j^n + \mu_j^n\Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^n)\right|$, $j \in \{1,2,...,J\}$
4) $\mathbf{w}^{n+1} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{z}_j^{n+1}$
5) $\Gamma^n = \{$indices of K largest coefficients of $\mathbf{w}^{n+1}\}$
6) $\mathbf{x}_j^{n+1} = H_{\Gamma^n}(\mathbf{x}_j^n + \mu_j^n\Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^n))$, $j \in \{1,2,...,J\}$
7) $\mathbf{g}_j = \Phi_j^T(\mathbf{y}_j - \Phi_j\mathbf{x}_j^{n+1})$, $j \in \{1,2,...,J\}$
8) $\mu_j^{n+1} = \frac{\mathbf{g}_{j\Gamma^n}^T\mathbf{g}_{j\Gamma^n}}{\mathbf{g}_{j\Gamma^n}^T\Phi_{j\Gamma^n}^T\Phi_{j\Gamma^n}\mathbf{g}_{j\Gamma^n}}$, $j \in \{1,2,...,J\}$
9) $n \leftarrow n+1$
10) **end while**

3.2.3 Complexity of S-IHT

Simultaneous iterative hard thresholding algorithm just requires simple matrix manipulations at each iteration and thus its computational bottleneck is due to operators Φ and Φ^T [28]. So for J sensors the computational complexity is in general $O(IJNM)$ where I is the number of iterations. However, if these operators are Fourier or Wavelet transforms, the computational complexity could be as low as $O(IJN\log M)$.

3.3 Simulations

We run our proposed S-IHT algorithm with adaptive scaling factor, on the signals of length $N = 100$ with sparsity level of $K = 5$ and average the results over 5000 simulation runs. In our simulations, we set the non-zero values of the sparse signals to 1. It can be shown that other values of non-zero coefficients result in the same performance.

In this work, we just compare our algorithm with DCS-SOMP [18], since other MMV algorithms such as [23–25] are fundamentally based on the assumption of fixed sensing matrices and thus are not comparable with our algorithm.

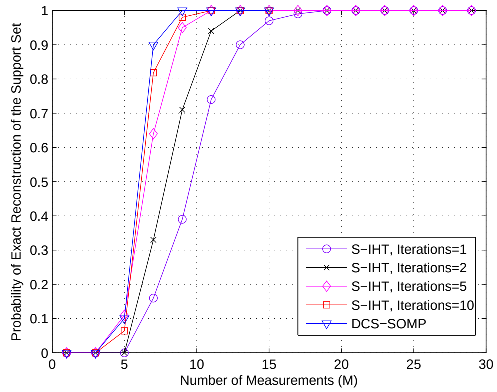

Figure 3.2: Reconstruction of the common support set in Joint Sparsity Model using S-IHT and SOMP for signals with $J = 16$, $N = 100$ and $K = 5$.

3.3.1 Comparison of Reconstruction Rate of S-IHT and DCS-SOMP

We now present a simulation result comparing the proposed S-IHT algorithm versus the conventional DCS-SOMP [18] algorithm for the case of $J = 16$ sensors. Figure 3.2 plots the probability of exact reconstruction of the *support set* versus the number of measurements per signal (M) for different number of iterations. We can see that the performance of our algorithm gets better as we increase the number of iterations. While Figure 3.2 indicates that the reconstruction performance of S-IHT with 10 iterations is very close to DCS-SOMP, we will show in the next part that the computation time of S-IHT is much less than that of DCS-SOMP.

3.3.2 Comparison of Computation Time of S-IHT and DCS-SOMP

In this part, we compare the computation time of our algorithm with that of DCS-SOMP for a signal of length $N = 100$ and sparsity level $K = 5$ with $M = 20$ measurements. It should be noted that DCS-SOMP requires the solution of a least square problem at each iteration that might be done by QR factorization which makes the algorithm costly. However, in contrast, our algorithm just requires simple matrix multiplications at each iteration and thus has a better computational complexity. Figure 3.3 compares the computation time of DCS-SOMP and that of S-IHT algorithm with different number of iterations. For instance, in the case of $J = 20$, the computation time for 10 iterations is roughly four times lower than the computation time needed for DCS-SOMP.

3.3.3 Effect of Number of Sensors

In this part, we investigate the effect of the number of sensors on the performance of the algorithm. We run the algorithm with 5 iterations for $J \in \{1, 2, 4, 8, 16, 32, 64\}$ signals of length $N = 100$ and sparsity level $K = 5$. Figure 3.4 indicates that by increasing the number of sensors, the performance of the algorithm gets closer to an oracle decoder that has prior knowledge about the location of the non-zero coefficients. We can see from this Figure that we are able to recover the support set of the signals with less than $M = 5$ measurements. However, we would need more than K measurements to recover the values of non-zero coefficients.

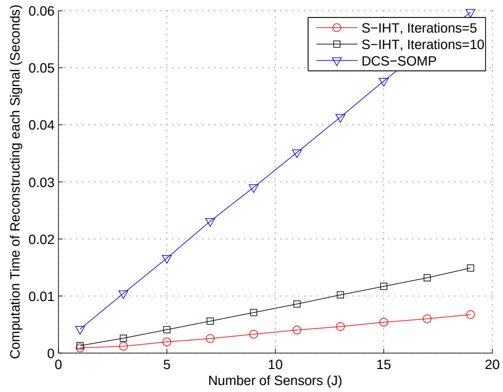

Figure 3.3: Comparison between the computation time of DCS-SOMP with S-IHT with various number of iterations for a signal with $N = 100$, $K = 5$ and $M = 20$.

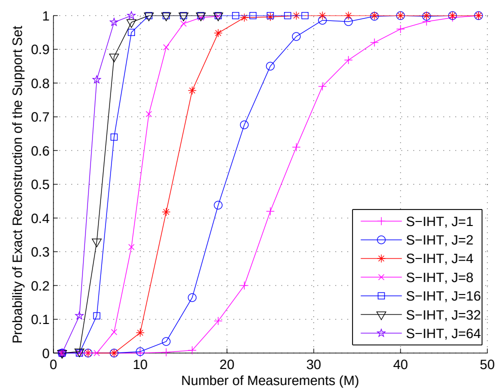

Figure 3.4: Reconstruction of the common support set versus M in Joint Sparsity Model using S-IHT for different values of J for signals with $N = 100$, $K = 5$.

3.3.4 Effect of Noise

In real situations, the signals are not exactly sparse and they are often approximated with $K-$sparse signals. Moreover, the measurements are always affected by noise. In this section, we model these two factors by a white Gaussian noise added to our measurements and investigate the effect of noise on the performance of our algorithm. We use the following model:

$$\mathbf{y}_j = \Phi_j \mathbf{x}_j + \mathbf{n}_j \tag{3.18}$$

The performance of our algorithm with 10 iterations for $N = 100$, $K = 5$, $J = 16$ and SNR values of $0dB$ and $10dB$ is plotted in Figure 3.5. We have also compared the performance of our algorithm with that of DCS-SOMP. In the DCS-SOMP algorithm, we find the non-zero values by using the pseudo-inverse of the columns of sensing matrix corresponding to the estimated support set. The result suggests that for large number of measurements, the performance of both algorithms are similar, which is due to the fact that both achieve perfect reconstruction and their estimates of the noise are the same.

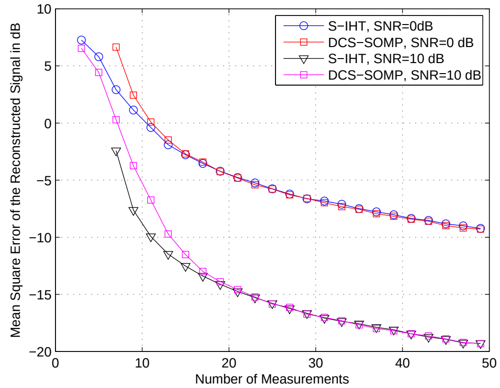

Figure 3.5: Mean Square Error of reconstructed signals with $N = 100$, $K = 5$, $J = 16$ in the presence of noise using S-IHT and DCS-SOMP.

3.4 Conclusion

In this Chapter, we proposed a very fast iterative algorithm with low computational complexity to reconstruct jointly sparse signals, which could arise in a large number of applications such as sensor networks. Our method iteratively refines its estimate of the common support set and converges to non-zero coefficients of the signals once it finds the true support set. In contrast to many conventional Multi Measurement Vector algorithms, in our proposed algorithm, the signals do not need to be sparse in the same basis, nor they need to be sensed using the same sensing matrix. Moreover, we showed that the computation time of our algorithm is much less than that of the DCS-SOMP algorithm while both of the algorithms show a similar performance in terms of reconstruction rate and robustness to noise. In addition, the popular recovery techniques of compressed sensing have a fixed performance in terms of the computation time and the reconstruction performance, however, in our S-IHT algorithm, by tuning the number of iterations, we are able to trade the reconstruction performance with the computational time, which makes this algorithm suitable for a wide variety of applications.

Chapter 4

Reconstruction of Jointly Sparse Signals with Non-Uniform Support Set

In this chapter, we study the problem of reconstructing a set of non-uniformly sparse signals that are sensed by a set of spatially distributed sensors in a sensor network. We use the *iterative hard thresholding* algorithm discussed in 2.1.5 to reconstruct the signals. Here, we first focus on reconstructing non-uniformly sparse signals in the single sensor case and modify the IHT algorithm to account for the additional information about the non-uniform support set of the signal. We then extend our algorithm to the multiple sensor case and exploit the spatial correlation that exists between different sensor nodes as well. We will show that the suggested scheme substantially improves the reconstruction rate over the standard IHT algorithm in compressed sensing by exploiting the prior information in the sensor networks and is much faster than the state-of-the-art algorithms.

4.1 Introduction

In this chapter, we focus on the reconstruction of non-uniformly sparse signals that are being sensed by a set of spatially distributed sensors. In the first part of this chapter, we focus on the single sensor case and propose an algorithm to reconstruct non-uniformly sparse signals. In the second part of the chapter, we extend our algorithm to the multiple sensor case and exploit the additional information that comes from the spatial correlation between different sensor nodes in addition to the prior information about the non-uniform support set of the signals. More precisely, this chapter is structured as follows.

First part of this work is focused on the single sensor case and we investigate the reconstruction problem from measurements of a sensor signal when additional information about the support set of the sensed signal is available. This occurs in various sensors. For instance, if the sensor is sensing smooth signals, we know that the low frequencies have higher probability of being non-zero. Similarly, for the wide-band sensors exposed to bandpass signals, we can assume that there is a higher probability for having non-zero components in the neighborhood of the carrier frequencies. Another application of the non-uniform support set is modeling the "temporal correlation" of video frames, or in general, time-sequences of sparse signals whose sparsity patterns change slowly over time [20]. In many applications due to the strong temporal dependencies, the support set of the signals change slowly over time and by exploiting this temporal correlation, we can do better than performing CS at each time separately. In these scenarios, we can use sufficient number of measurements to find the support set of the initial video frame and then at future times, we can use the fact that the corresponding coefficients in the next frames are more likely to be non-zero. So we can use prior probabilities to capture the temporal correlations in the time-varying sparse signals.

In recent years, many algorithms have been proposed to incorporate the additional information about the support set in compressed sensing [20, 29–32]. A weighted ℓ_1−minimization algorithm has been proposed and analyzed in [29–31]. In this approach, a properly selected weight is assigned to each coefficient of the signal and the reconstruction is done by minimizing the weighted ℓ_1 norm of the signal. The weighting scheme is such that the algorithm favors the coefficients that are more likely to be non-zero and thus the weighted norm of the signals, which are consistent with the prior information, is lower. The ℓ_1−minimization is solved by linear programming with the complexity of $\mathcal{O}(N^3)$. In the first part of this Chapter, we propose a new algorithm based on iterative hard thresholding [8] to incorporate this information into CS algorithm. We will show that our modified IHT algorithm is an order of magnitude faster than the weighted ℓ_1 minimization in reconstructing non-uniformly sparse signals.

In the second part of this chapter, we extend our proposed algorithm of the first part to the distributed sensor networks and modify it such that it can exploit the spatial correlation as well. In the present chapter, we consider the joint sparsity model discussed in Section 2.2 where the support set of all signals is the same. We call this new algorithm DCS-MIHT and will show that it can successfully exploit both spatial and temporal correlation in the distributed scenarios.

This chapter is structured as follows. The problem is stated in Section 4.2. We propose a new algorithm which we call it *modified iterative hard thresholding* (MIHT) to incorporate the additional information about the non-uniform support set into CS algorithm in Section 4.3. We will show that our algorithm is an order of magnitude faster than the weighted ℓ_1−minimization and achieves a substantial improvement in reconstruction rate over the standard iterative hard thresholding (IHT) algorithm. We extend the modified MIHT algorithm to the multiple sensor case and propose DCS-MIHT algorithm in section 4.4. The simulation results of our algorithm are given in Section 4.5. Section 4.6 concludes this chapter.

4.2 Problem Formulation and Notations

Assume J sensors measuring signals that might be correlated both in space and time. Each sensor node transmits its data to a fusion center. At the fusion center, the data of all sensor nodes are decoded jointly and the spatial correlation is exploited using the Joint Sparsity Models (JSM). In this scheme, measurements are taken independently and there is no need for collaboration among sensors.

Let $\mathbf{x}_j, j = 1, \ldots, J$ represent the $N \times 1$ signal of the j−th sensor node and $x_{i,j}^n$ denotes the i−th coefficient of the estimation of vector $\mathbf{x}_j$ at the n−th iteration. We model the spatial correlation by assuming that the signals have exactly the same support set across all sensors but maybe with different coefficients. Due to the universality of compressive sensing, without loss of generality, we can assume that the sparsity basis matrix is the identity matrix, and hence $\mathbf{x}_j$ is the signal in the sparse domain. We further assume that $\mathbf{x}_j$ is a "non-uniformly sparse" vector and each coefficient of the signals is coming from the prior distribution of $x_{i,j} \sim p_i\delta(x_{i,j}) + (1-p_i)G_{i,j}(x_{i,j})$, where p_i is close to one and is known in advance for each coefficient and $G_{i,j}(x_{i,j})$ is the "unknown" conditional prior of $x_{i,j}$ given that it is non-zero.

At each sensor node, we use a different $M \times N$ random Gaussian measurement matrix Φ_j, whose columns are normalized to have a unit ℓ_2 norm, to get the measurement vector $\mathbf{y}_j = \Phi_j \mathbf{x}_j$ and transmit the sampled vector to the fusion center. At the fusion center, we jointly reconstruct the signals using the received measurements with the assumption that the vector $\mathbf{p} = \{p_1, ..., p_N\}$ is available as an additional information that comes from the non-uniform common support set.

In the rest of this chapter, Φ_Γ represents the submatrix collecting the columns of Φ indexed by Γ and $\mathbf{x}_\Gamma$ denotes coefficients of $\mathbf{x}$ in the set Γ. Γ^c represents the complement of Γ and then mutual coherence is $\mu = \max\limits_{i \neq j} \langle \phi_i, \phi_j \rangle$.

In Section 4.3, we focus on the single sensor case and thus the index j is dropped.

4.3 Modified Hard Thresholding Algorithm for The Single Sensor Case

In this section, we first formally state our algorithm and then investigate a scenario that can be considered as an especial case of our algorithm, namely compressed sensing with partially known support set.

Iterative Hard Thresholding (IHT) [8] is a class of low computational algorithms, which has recently been proposed for the reconstruction of sparse signals. In this chapter, we use "iterative thresholding with inversion" (ITI) algorithm that is proposed in [10]. Starting from $\mathbf{x}^0 = 0$, ITI iteratively finds the sparsest solution of $\mathbf{y} = \Phi\mathbf{x}$ using the following iteration:

$$\begin{aligned} \Gamma^{n+1} &= \text{supp}\{|\mathbf{x}^n + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^n)|\} \\ \mathbf{x}^{n+1}_{\Gamma^{n+1}} &= (\Phi^T_{\Gamma^{n+1}}\Phi_{\Gamma^{n+1}})^{-1}\Phi^T_{\Gamma^{n+1}}\mathbf{y} \\ \mathbf{x}^{n+1}_{(\Gamma^{n+1})^c} &= 0 \end{aligned} \tag{4.1}$$

where "supp" is an operator that returns the indices of k largest coefficients of its input vector.

In this work, we modify the iterative thresholding with inversion algorithm to account for the prior information about the support set. Let us define the vector $\mathbf{z}^{n+1}$ as

$$\mathbf{z}^{n+1} = \mathbf{x}^n + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^n) = \mathbf{x} + (\Phi^T\Phi - I)(\mathbf{x} - \mathbf{x}^n)$$

Let us regard $\mathbf{e}^n = (\mathbf{x} - \mathbf{x}^n)$ as the "error term" and $\mathbf{n} = (\Phi^T\Phi - I)(\mathbf{x} - \mathbf{x}^n) = (\Phi^T\Phi - I)\mathbf{e}^n$ as the "noise term" that is added to our actual signal. It is needless to say that the contribution of the noise term should be reduced at each iteration. For a random Gaussian matrix with normalized columns, due to the pseudo-orthogonality of the columns, we have $E[\Phi^T\Phi] = I_{N\times N}$. Thus, by the Central Limit Theorem, the matrix $(\Phi^T\Phi - I)$ is a Gaussian matrix that does not correlate with the error term. Therefore, by multiplying the error term by this random matrix, the variance of error will be reduced.

According to the Central Limit Theorem, we can conclude that if $M \to \infty$, the noise term has an i.i.d. Gaussian distribution with $E(n_i) = 0$, $E(n_i^2) = \frac{1}{M}\|\mathbf{e}^n\|_2^2$ and $E(n_i n_j) = 0$ for $i \neq j$. Thus, at each iteration, we are essentially dealing with a detection problem where we wish to detect whether a coefficient of $\mathbf{x}$ is zero or non-zero based on its noisy observation $\mathbf{z}^{n+1} = \mathbf{x} + \mathbf{n}$. This detection should be such that the probability of correct decision is maximized. Without any prior knowledge (uniform probability) the maximum-likelihood (ML) function is given by $f(z_i^{n+1}|x_i = 0) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-(z_i^{n+1})^2/2\sigma_n^2}$. So if we evaluate the ML-function at each coefficient of $\mathbf{z}^{n+1}$, we will see that the k largest coefficients are the coefficients that minimize the ML-function the most (have higher chance of being non-zero). Thus, the maximum-likelihood decision on the support set leads to selecting the k largest coefficients in absolute value for the case of uniform probability. However, if the a priori probabilities of coefficients being zero are not equal, instead of selecting the k largest coefficients at each iteration, we select the coefficients that maximize the posterior probability of being non-zero (MAP detection). The posterior probabilities of coefficients being zero is

$$\Pr(x_i = 0|z_i^{n+1}) = \int_{x_i=0^-}^{x_i=0^+} \frac{f(z_i^{n+1}|x_i)}{f(z_i^{n+1})} f(x_i)dx_i$$

$$= \int_{x_i=0^-}^{x_i=0^+} \frac{f(z_i^{n+1}|x_i)}{f(z_i^{n+1})} (p_i\delta(x_i) + (1-p_i)G_i(x_i))dx_i$$

The second term of prior probability vanishes in the integration and thus we have

$$\Pr(x_i = 0|z_i^{n+1}) \propto f(z_i^{n+1}|x_i = 0)p_i$$

$$= \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{\frac{-(z_i^{n+1})^2}{2\sigma_n^2}} p_i = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(z_i^{n+1})^2 - 2\sigma_n^2 ln p_i}{2\sigma_n^2}}$$

For non-uniform prior probability, it is necessary to know the variance of the noise term (σ_n^2) to compute the posterior probability and make the decision. We have $E(n_i^2) = \frac{1}{M}\|\mathbf{x} - \mathbf{x}^n\|_2^2$. Also we know that $\|\mathbf{x} - \mathbf{x}^n\|_0 < 2k$ and the matrix Φ preserves the norm of $2k$ sparse signals since it obeys the restricted isometry property of order $2k$. Thus, we have $\sigma_n^2 = E[n_i^2] = \frac{1}{M}\|\mathbf{x} - \mathbf{x}^n\|_2^2 \approx \frac{1}{M}\|\Phi(\mathbf{x} - \mathbf{x}^n)\|_2^2 = \frac{1}{M}\|(\mathbf{y} - \Phi\mathbf{x}^n)\|_2^2$. Therefore, the proposed algorithm is as follows.

Modified Iterative Hard Thresholding:
Reconstruction:
1) Initialize: $\mathbf{x}^0 = 0$, $n = 0$
2) **while** $\|\mathbf{y} - \Phi\mathbf{x}\|_2 > \epsilon$ **do**
3) $\mathbf{z}^{n+1} = \mathbf{x}^n + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^n)$
4) $\sigma_n^2 = \frac{1}{M}\|(\mathbf{y} - \Phi\mathbf{x}^n)\|_2^2$
5) $\Gamma^{n+1} = \text{supp}\{(\mathbf{z}^{n+1})^2 - 2\sigma_n^2 ln(\mathbf{p})\}$
6) $\mathbf{x}_{\Gamma^{n+1}}^{n+1} = (\Phi_{\Gamma^{n+1}}^T\Phi_{\Gamma^{n+1}})^{-1}\Phi_{\Gamma^{n+1}}^T\mathbf{y}$ and $\mathbf{x}_{(\Gamma^{n+1})^c}^{n+1} = 0$
7) $n \leftarrow n + 1$
8) **end while**

4.3.1 Compressed Sensing with Partially Known Support Set

In some compressed sensing applications such as MR image reconstruction using wavelet transform, a part of support set of sparse signal may be available from prior knowledge [4, 20, 32]. The question is how we can incorporate this prior information to improve the performance of compressed sensing. In this part, we show how the MIHT algorithm can address this problem and investigate the performance of the algorithm analytically by giving sufficient conditions under which exact reconstruction could happen.

Let us assume that the size of the known support set is k_c, the size of the unknown support set is k_i and the sparsity level is $k = k_c + k_i$. We assign the probability of zero to k_c indices of the known support set and a uniform probability to the other indices. Thus, for the known support set, we have $(z_i^{n+1})^2 - 2\sigma_n^2 ln(p_i) \to \infty$ which means that the algorithm forces the indices of the known support set to be in the estimated support set at every single iteration and the other k_i indices of the estimated support set are the k_i largest coefficients of $\mathbf{z}^{n+1}$ in absolute value excluding the k_c coefficients of the known support set.

We next derive a sufficient condition for the performance of the algorithm. In [10], it has been proved that a sufficient condition for the convergence of hard thresholding algorithm is $k < \frac{1}{3}\mu^{-1}$ where μ is the mutual coherence of the matrix Φ and k is the sparsity level. Here, we state a more general theorem for the case where a part of the support set (k_i elements) is known in advance.

Theorem 4.1. Suppose we have $\frac{\mu k_i}{1-\mu k} < \frac{1}{2}$. Then, the modified iterative hard thresholding algorithm achieves perfect reconstruction of the sparse signal.

Proof: Without loss of generality we can assume that the known part of the support set is the first k_c elements of the signal, and the next k_i elements are the unknown part of the support set such that $x(k_c+1) > x(k_c+2) > ... > x(k_c+k_i)$ and the other elements are 0.

The algorithm forces the first k_c indices of $x_1, x_2, ..., x_{k_c}$ to be in the estimated support set of all iterations. The idea of the proof is as follows. If we have $\frac{\mu k_i}{1-\mu k} < \frac{1}{2}$, after the first the iteration the largest element of the unknown support set x_{k_c+1} will get into the estimated support set. In the second iteration, x_{k_c+1} will remain in the estimated support set and also x_{k_c+2} will get into the estimated support set and this process happens again for k_i iterations after which all elements of the unknown support set will get into the estimated support set.

Suppose $x_1, ..., x_{i-1}$ are in the support set at iteration n. we will show that

$$\max_{j \in \Gamma^n} |x_j^n - x_j| < \frac{k_i \mu |x_i|}{1 - k\mu} \tag{4.2}$$

We have:

$$\|x_{\Gamma^n}^n - x_{\Gamma^n}\|_\infty = \|(\Phi_{\Gamma^n}^T \Phi_{\Gamma^n})^{-1} \Phi_{\Gamma^n}^T \Phi_{(\Gamma^n)^c} x_{(\Gamma^n)^c}\|_\infty < \|(\Phi_{\Gamma^n}^T \Phi_{\Gamma^n})^{-1}\|_{\infty,\infty} \|\Phi_{\Gamma^n}^T \Phi_{(\Gamma^n)^c} x_{(\Gamma^n)^c}\|_\infty \tag{4.3}$$

where in (4.3), $\|A\|_{a,b}$ is the operator norm on matrix A defined as follows.

$$\|A\|_{a,b} = \max_x \frac{\|Ax\|_b}{\|x\|_a}$$

But we know that:

$$\|(\Phi_{\Gamma^n}^T \Phi_{\Gamma^n})^{-1}\|_{\infty,\infty} < \|I\|_{\infty,\infty} + \|I - \Phi_{\Gamma^n}^T \Phi_{\Gamma^n}\|_{\infty,\infty} + ... < \frac{1}{1 - \|I - \Phi_{\Gamma^n}^T \Phi_{\Gamma^n}\|_{\infty,\infty}} < \frac{1}{1 - k\mu} \tag{4.4}$$

$$\|\Phi_{\Gamma^n}^T \Phi_{(\Gamma^n)^c} x_{(\Gamma^n)^c}\|_\infty < k_i \mu |x_i| \tag{4.5}$$

From (4.3), (4.4) and (4.5) we can conclude (4.2). Now assume that $k_c + 1 \le i \le k$ and $x_1, ..., x_{i-1}$ are already in the support set. For $k_c + 1 \le m \le i$ we have

$$|z_m^{n+1}| = |x_m + \sum_{j \neq m} \langle \phi_m, \phi_j \rangle (x_j - x_j^n)|$$

$$= |x_m + \sum_{\substack{j\neq m \\ j\in\Gamma^n}} \langle \phi_m, \phi_j \rangle (x_j - x_j^n) + \sum_{\substack{j\neq m \\ j\notin\Gamma^n}} \langle \phi_m, \phi_j \rangle x_j|$$

$$> |x_m| - |\sum_{\substack{j\neq m \\ j\in\Gamma^n}} \langle \phi_m, \phi_j \rangle (x_j - x_j^n))| - |\sum_{\substack{j\neq m \\ j\notin\Gamma^n}} \langle \phi_m, \phi_j \rangle x_j|$$

$$> |x_i| - \mu k \ \max_{j\in\Gamma^n} |(x_j - x_j^n)| - \mu k \ \max_{j\notin\Gamma^n} |x_j|$$

$$> |x_i| - \mu k \ \max_{j\in\Gamma^n} |(x_j - x_j^n)| - \mu k |x_i|$$

$$> |x_i| - \mu k \frac{k_i \mu |x_i)|}{1 - k\mu} - \mu k_i |x_i| = |x_i| - \frac{\mu k_i}{1 - \mu k} |x_i|$$

The above inequality is not generally true for $1 \leq m \leq k_c$ since there is no guarantee that $|x_m| > |x_i|$ is satisfied when $1 \leq m \leq k_c$. However, as we know a priori that these indices are in the actual support set, they are forced to remain in the estimated support set and this is indeed where the gain in reconstruction rate comes from.

If $k + 1 \leq i \leq N \Rightarrow |z_i^{n+1}| = |\sum_{j\neq i} \langle \phi_i, \phi_j \rangle (x_j - x_j^n)|$

$$= |\sum_{\substack{j\neq i \\ j\in\Gamma^n}} \langle \phi_i, \phi_j \rangle (x_j - x_j^n) + \sum_{\substack{j\neq i \\ j\notin\Gamma^n}} \langle \phi_i, \phi_j \rangle x_j|$$

$$< |\sum_{\substack{j\neq i \\ j\in\Gamma^n}} \langle \phi_i, \phi_j \rangle (x_j - x_j^n)| + |\sum_{\substack{j\neq i \\ j\notin\Gamma^n}} \langle \phi_i, \phi_j \rangle x_j|$$

$$< \mu k \ \max_{j\in\Gamma^n} |(x_j - x_j^n)| - \mu k \ \max_{j\notin\Gamma^n} |x_j|$$

$$< \mu k \ \max_{j\in\Gamma^n} |(x_j - x_j^n)| - \mu k |x_i|$$

$$< \mu k \frac{k_i \mu |x_i|}{1 - k\mu} + \mu k_i |x_i| = \frac{\mu k_i}{1 - \mu k} |x_i|$$

So the sufficient condition for detecting x_i in the estimated support set is

$$|x_i| - \frac{\mu k_i}{1 - \mu k} |x_i| > \frac{\mu k_i}{1 - \mu k} |x_i| \Rightarrow \frac{\mu k_i}{1 - \mu k} < \frac{1}{2}$$

If we assume $x_{k_c+1}, ...x_{k_c+r}$ are already in the support set, it is not hard to see that a sufficient condition for x_{k_c+r+1} to get into the support set is that

$$\max |(\Phi^T \Phi - I)(x - x_i)| < \frac{|x_{k_c+r+1}|}{2}$$

since in that case $z_{k_c+i} > \frac{|x_{k_c+r+1}|}{2}$, $\forall 0 < i < r$ and $z_{k_c+i} < \frac{|x_{k_c+r+1}|}{2}$, $\forall r < i < N$. Also the conditions of the theorem guarantee that $x_{k_c+1}, ...x_{k_c+r}$ will remain in the support set. But we know

$$\max |(\Phi^T \Phi - I)(x - x_i)| < \sum_{\substack{j\neq i \\ j\in\Gamma^n}} \langle \phi_i, \phi_j \rangle (x_j - x_j^m) + \sum_{\substack{j\neq i \\ j\notin\Gamma^n}} \langle \phi_i, \phi_j \rangle x_j| < k_i \mu \frac{k \mu x_{k_c+1}}{1 - k\mu} + \mu k_i x(k_c+1) = x(k_c+r+1) \frac{\mu k_i}{1 - \mu k}$$

Thus, by having

$$\frac{\mu k_i}{1 - \mu k} < \frac{1}{2}$$

we achieve perfect reconstruction.

If $k_c = 0$ our algorithm performs similar to iterative thresholding with inversion and the sufficient condition for perfect reconstruction would become $k < \frac{1}{3}\mu^{-1}$. As we increase k_c, the algorithm is provided with more prior information and the sufficient condition of perfect reconstruction gets weaker and thus the performance rate will be improved. If we have $k_c = k$ and $k_i = 0$, the sufficient condition is always satisfied for any mutual coherence, in other words, the algorithm forces the estimated support set of the first iteration to be the same as the known support set and then in just one step the algorithm recovers the sparse signal using the pseudoinverse of the columns of sensing matrix corresponding to the known support set.

4.4 DCS-MIHT algorithm for Jointly Sparse Signals

Using the same technique used in Chapter 3, we can extend the MIHT algorithm to the multiple sensor case. At the end of each iteration, we have a rough estimate of our signals $\mathbf{x}_j$. Ideally, all of the estimates should be $k-$sparse with a same support set, but this is not the case due to the existence of the noise term. So we can simply force all of the signals to have a same support set as follows. If the a priori probabilities of each coefficient being zero are not equal, instead of selecting the k largest coefficients at each iteration, we select the coefficients that maximize the posterior probability of being non-zero (MAP detection). The probability of each coefficient of the support set being zero could be found as follows.

$$\Rightarrow \Pr(x_i = 0 | z_{i,1}^{n+1}, z_{i,2}^{n+1}, ..., z_{i,J}^{n+1}) \propto p_i \prod_{j=1}^{J} f(z_{i,j}^{n+1} | x_i = 0)$$

$$= p_i \prod_{j=1}^{J} \frac{1}{\sqrt{2\pi\sigma_{n,j}^2}} e^{\frac{-(z_{i,j}^{n+1})^2}{2\sigma_{n,j}^2}} \propto e^{-\sum_{j=1}^{J} \frac{(z_{i,j}^{n+1})^2}{2\sigma_{n,j}^2} + ln(p_i)}$$

So the idea is that the indices of the $k-$ coefficients corresponding to the largest probability of being non-zero will be the estimated common support set. Now that we have estimated the support set, we force all signals to have the same support set and find their coefficients by using the pseudo-inverse of the measurement matrix of that sensor. So the DCS-MIHT algorithm will be as follows.

Modified Iterative Hard Thresholding:
Reconstruction:
1) Initialize: $\mathbf{x}_j^0 = 0$, $n = 0$, $j \in \{1, 2, ..., J\}$
2) **while** $\sum_{j=1}^{J} \|\mathbf{y}_j - \Phi_j \mathbf{x}_j\|_2 > \epsilon$ **do**
3) $\mathbf{z}_j^{n+1} = \mathbf{x}_j^n + \Phi_j^T(\mathbf{y}_j - \Phi_j \mathbf{x}_j^n)$, $j \in \{1, 2, ..., J\}$
4) $\sigma_{n,j}^2 = \frac{1}{M}\|(\mathbf{y}_j - \Phi_j \mathbf{x}_j^n)\|_2^2$, $j \in \{1, 2, ..., J\}$
5) $\Gamma^{n+1} = \text{supp}\{\sum_{j=1}^{J} \frac{(\mathbf{z}_j^{n+1})^2}{2\sigma_{n,j}^2} - ln(\mathbf{p})\}$
6) $\mathbf{x}_{j_{\Gamma^{n+1}}^{n+1}} = (\Phi_{j_{\Gamma^{n+1}}}^T \Phi_{j_{\Gamma^{n+1}}})^{-1} \Phi_{j_{\Gamma^{n+1}}}^T \mathbf{y}_j$, $j \in \{1, 2, ..., J\}$
7) $\mathbf{x}_{j_{(\Gamma^{n+1})^c}}^{n+1} = 0$, $j \in \{1, 2, ..., J\}$
8) $n \leftarrow n + 1$
9) **end while**

4.5 Simulations

4.5.1 Performance of MIHT algorithm

In this part, we investigate the performance of the MIHT algorithm and compare it with the weighted ℓ_1- minimization [30] for signals of length $N = 200$. For each signal, we consider three sparsity regions R_1, R_2, R_3. R_1 includes the first 20 coefficients, R_2 includes the next 40 coefficients and R_3 includes the last 140 coefficients. The probability of being non-zero for the coefficients of R_1, R_2, R_3 are $0.9, 0.5, 0.1$ respectively. Fig. 4.1 shows the reconstruction rate of standard IHT, modified IHT (MIHT), standard ℓ_1-minimization and weighted ℓ_1-minimization. The number of iterations in MIHT algorithm is 10 and the weights in weighted ℓ_1-minimization are proportional to the probability of being zero ($w_i = p_i$) [30]. In Fig. 4.1, We can observe that MIHT algorithm can successfully exploit the additional information and substantially reduce the number of measurements compared to standard IHT. However, the performance of weighted ℓ_1-minimization is better than that of our algorithm. This behavior is expected since it is well-known that the methods based on the convex relaxation like ℓ_1-minimization have better reconstruction rate than the greedy algorithms like IHT at the expense of computation time. We can also see in this figure that the performance gap between the standard ℓ_1-minimization and the standard IHT is the same as the gap between MIHT and the weighted ℓ_1-minimization. In the next subsection, we show that the computation time of our algorithm is an order of magnitude better than that of the weighted ℓ_1-minimization.

Robustness: In this part, we investigate the robustness of our algorithm in the presence of non-accurate or biased prior probabilities. Here, instead of providing the algorithm with the actual prior probabilities, we provide it with a random probability vector whose elements are drawn uniformly at random from the interval $(0, 1)$. As we can see from Fig. 4.1, even in the case of non-accurate prior information, MIHT algorithm eventually performs almost as good as the standard IHT algorithm but needs more iterations to converge. Here, we just provide an intuitive explanation of this behavior. If the prior information is accurate, in the first few iterations, the prior information helps most of the indices of the true support set get into the estimated support set. In the next iterations, the variance of the error of the coefficients of those indices gets smaller and the algorithm essentially removes their contributions from the measurement vector. Thus, other indices of the true common support set find the chance to get into the estimated support set and this substantially improves the reconstruction rate. However, if the prior information is non-accurate, the algorithm will not converge in the first few iterations. But as the number of iterations increases, σ_n gets smaller and thus the contribution of prior information on the estimation of support set vanishes. Therefore, if we let the algorithm to have a sufficient number iterations, the reconstruction rate will eventually remain almost the same as the standard IHT.

4.5.2 Computation Time of MIHT algorithm

In this subsection, we investigate the computation time of both MIHT and weighted ℓ_1-minimization on the signals of the previous part. We keep the sparsity ratio of R_1, R_2, R_3 the same and change N from 200 to 500. We can see from Fig. 4.2, that the computation time of MIHT is substantially (an order of magnitude) better than the weighted ℓ_1-minimization. This reduction in the complexity comes at the expense of only about 10% increase in the number of measurements required for perfect reconstruction as illustrated in Fig. 4.1.

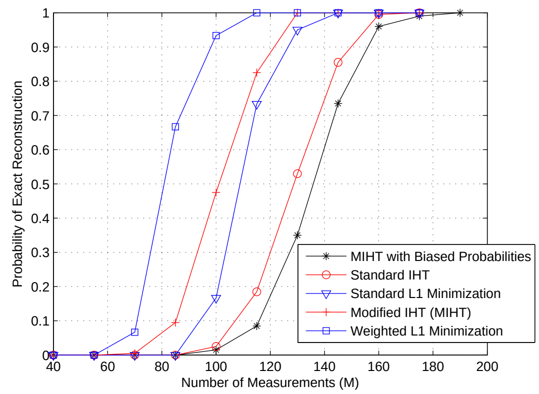

Figure 4.1: Reconstruction rate of signals with $N = 200$ using MIHT and weighted ℓ_1−minimization.

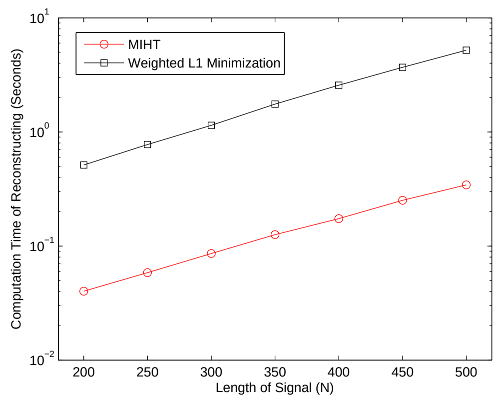

Figure 4.2: Computation time of MIHT and weighted ℓ_1.

4.5.3 Compressed Sensing with Partially Known Support Set

We now investigate the performance of the MIHT algorithm in the scenarios where we partially know the support set of the sparse signal. Let us assume k_c elements of the support set are known a priori and k_i elements are unknown. Fig. 4.3 shows the performance of the algorithm on the signals of length 200 and sparsity level $k = k_c + k_i = 50$ for different values of $k_c \in \{0, 10, 20, 30, 40\}$. The case where $k_c = 0$ corresponds to the standard IHT. We can see from this figure that as we increase the size of the known support set (k_c), the algorithm is provided with more additional information about the support set and the reconstruction rate gets better.

4.5.4 DCS-MIHT algorithm for Jointly Sparse Signals

In this part, we investigate the effect of the number of sensors on the performance of the algorithm. We run the DCS-MIHT algorithm with 30 iterations on the signals of the previous subsection, for $J \in \{1, 2, 4, 16\}$ without taking into account the non-uniform support set of the signals. We then incorporate the additional information of the non-uniform support and run the algorithm for the case $J = 4$. Fig. 4.4 indicates that by increasing the number of sensors, the performance of the algorithm gets better. We

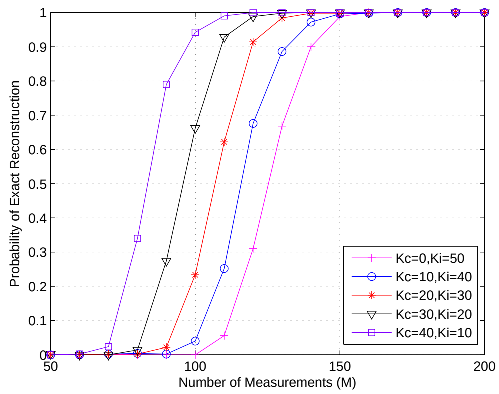

Figure 4.3: Reconstruction rate of signals with $N = 200$, $k = 50$ for different sizes of known sparsity support (k_c).

can also see from this figure that for the case of "$J = 4$ with non-uniform support set", the algorithm is able to exploit the information of both spatial correlation and non-uniform support and shows a better performance than the case of "$J = 4$ with uniform support set".

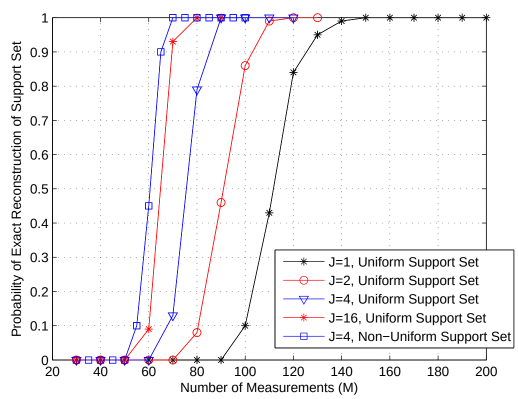

Figure 4.4: Reconstruction of the common support set versus M in Joint Sparsity Model using DCS-MIHT.

4.6 Conclusion

In this work, we proposed a very fast algorithm to reconstruct non-uniformly sparse signals and compared it with the state-of-the-art algorithms such as weighted ℓ_1-minimization for the single sensor case. Our algorithm (MIHT) is essentially a modified version of iterative hard thresholding. We showed how we can use the MIHT algorithm to address the problem of compressed sensing with partially known support set. In this case, we also derived a sufficient condition under which the exact reconstruction could happen. We showed that our algorithm is much faster than the weighted ℓ_1−minimization and achieves substantially better reconstruction rate compared to the standard iterative hard thresholding.

Afterwards, we extended our algorithm to the multiple sensor case and proposed the DCS-MIHT algorithm for jointly sparse signals with different sensing matrix. We showed that the DCS-MIHT algorithm is able to capture the spatial correlation of the signals as well as the additional information about the non-uniform support set in a sensor network.

Chapter 5

Generalized Joint Sparsity Models

In this chapter, we define a new Joint Sparsity Model (JSM) and use Principal Component Analysis followed by Minimum Description Length and Compressive Sensing to reconstruct spatially and temporally correlated signals in a sensor network. The proposed model decomposes each sparse signal into two sparse components. The first component has a common support set across all sensed signals. The second component is an innovation part that is specific to each sensor and might have a support set that is different from the support set of the other innovation signals. We use the fact that the common component generates a common subspace that can be found using the principal component analysis and the minimum description length. We show that with this general model, we can reconstruct the signal with smaller samples that are needed by the direct application of the compressive sensing on each sensor.

In [18, 33], three models for joint sparsity have been proposed, which are represented by JSM-1, JSM-2 and JSM-3. The three Joint Sparsity Models (JSM) impose certain structure for the sparsity that might be too restrictive in many cases. In JSM-1 and JSM-3, it is assumed that the sensed signals have a common component, which is identical for all sensors. This common component is assumed to be sparse in JSM-1 and non-sparse in JSM-3. However, the assumption of having an exact common component across all signals might not be satisfied in practical applications. JSM-2 assumes that the support set of all signals are exactly the same. Algorithms such as SOMP [21], Mixed norm approach [24], M-FOCUSS [25], M-SBL [26], ReMBo [23] and model-based compressed sensing [34] have been proposed for reconstructing JSM-2 signals under the concept of Multiple Measurement Vectors (MMV). However, this model does not allow any variations among the support sets and the assumption of fixed support set is not valid if we are interested in the fine features of the signals. Therefore, despite much improvement obtained in the JSM models, their restrictive assumptions limit their application in practice.

In this chapter, motivated by the signal models proposed in [18], we propose a more general model called *Generalized Joint Sparsity Model* (G-JSM) that can model more practical cases. We show that JSM-1 and JSM-2 are special cases of this model and then propose a sensing and reconstruction algorithm for G-JSM. We will see that even in this general model we can still have a better performance compared to separate compressed sensing. Our reconstruction algorithm is based on Principal Component Analysis (PCA) and Minimum Description Length (MDL) and we will show that this algorithm outperforms SOMP [21] in the G-JSM model. We show that with this general model, we can reconstruct the signal with smaller samples that are needed by the direct application of the compressive sensing on each sensor.

This chapter is structured as follows. Our Generalized Joint Sparsity Model is introduced in Section

5.1. In Section 5.2, we will propose our reconstruction algorithm. We will compare the performance of our algorithm with that of SOMP and separate compressed sensing using Orthogonal Matching Pursuit [7] in Section 5.3. Section 5.4 concludes the chapter.

5.1 Generalized Joint Sparsity Model

Assume J sensors measuring signals that might be correlated both in space and time. Each sensor node transmits its data to a fusion center. At the fusion center, the data of all sensor nodes are decoded jointly and the spatial correlation is exploited using the Joint Sparsity Models (JSM). In this scheme, measurements are taken independently and there is no need for collaboration among sensors.

Let $\mathbf{x}_j, j = 1, \ldots, J$ represents the $N \times 1$ signal of the jth sensor node. We assume that each of the signals consists of two components. The first component ($\mathbf{w}_j$) has exactly the same support set across all sensors, with the sparsity level denoted by K_C, but maybe with different coefficients and the second component is called the *innovation component* ($\mathbf{z}_j$) whose sparsity level is K_j. Therefore, for $j = 1, \ldots, J$,

$$\mathbf{x}_j = \mathbf{w}_j + \mathbf{z}_j$$

$$\|\mathbf{w}_j\|_0 = K_C \ , \ \|\mathbf{z}_j\|_0 = K_j$$

Due to the universality of compressive sensing, without loss of generality, we can assume that the sparsity basis matrix is the identity matrix, and hence $\mathbf{x}_j$, $\mathbf{w}_j$ and $\mathbf{z}_j$ are the signals in the sparse domain.

We call the above model the *Generalized Joint Sparsity Model* (G-JSM). The G-JSM model is less restrictive than JSM-1 [18] since it does not assume that the first components are exactly the same among all signals. It only assumes that the support set of the first components are the same. Furthermore, it is more general than JSM-2 since it assumes that each signal has an innovation component and thus the support set of the signals could be different. We will show that even in this more general model we can achieve perfect reconstruction with fewer measurements than separate compressed sensing.

A practical situation that could be well modeled by G-JSM is where several acoustic sensors are listening to a speech signal. Each signal experiences different attenuation and multi-path effect which cause different amplitudes and phases. However, we expect that the location of the excited coefficients be roughly the same. So we can model the intersection of the support sets by $\mathbf{w}_j$ and the variations by $\mathbf{z}_j$.

At each sensor node, we use the same $M \times N$ random Gaussian measurement matrix $\mathbf{\Phi}$ to get the measurement vector $\mathbf{y}_j = \mathbf{\Phi}\mathbf{x}_j$ and transmit the sampled vector to the fusion center. At the fusion center, we form an $N \times J$ matrix $\mathbf{X}$ whose columns are the J signals and an $M \times J$ matrix $\mathbf{Y}$ whose columns are the measurements. Thus,

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{X}$$

Let $C \subset \{1, ..., N\}$ denotes the common sparse support set of the signals (the support of $\mathbf{w}_j$) and $\mathbf{\Phi}_C$ represents the matrix that is formed by the columns of $\mathbf{\Phi}$ indexed by C. Each measurement vector $\mathbf{y}_j$ is a linear combination of the columns of $\mathbf{\Phi}_C$ plus a linear combination of the columns of $\mathbf{\Phi}$ that correspond to the sparse support set of $\mathbf{z}_j$. If we ignore the innovation components $\mathbf{z}_j$, all the measurements will lie in a K_C−dimensional subspace which is the span of the columns of $\mathbf{\Phi}_C$ and is equal to the span of the columns of $\mathbf{Y}$ if and only if $J > K_C$. But if we have innovation components, the dimension of the span of the columns of $\mathbf{Y}$ could be as large as M. However we can interpret the departure of measurements from the span of the columns of $\mathbf{\Phi}_C$ as noise and conclude that in the case of G-JSM, the intrinsic dimensionality of the span of the columns of $\mathbf{Y}$ is still K_C.

To estimate the shared support set of the signals, we first estimate the span of the columns of $\mathbf{\Phi}_C$ as the principal subspace of the span of the columns of $\mathbf{Y}$ using the dominant eigenvalues of the covariance matrix of $\mathbf{Y}$. The principal subspace would be the span of the eigenvectors corresponding to the dominant eigenvalues. A model order selection method such as the *Minimum Description Length* (MDL) can be used to find the size of the shared support set (K_C). Afterward, we project all the columns of $\mathbf{\Phi}$ onto the principal subspace and choose the K_C columns that have the minimum projection error and introduce them as the indices of the shared support set.

The above discussion motivates the application of *principal component analysis* (PCA) [35], which is a method to project signals onto a lower dimensional subspace given that the signals lie close to a manifold of lower dimensionality than that of the original data space.

The sample mean $\bar{\mathbf{y}}$ and the sample covariance matrix $\bar{\mathbf{S}}$ of the vectors $\mathbf{y}_j, j = 1, \ldots, J$ are given as:

$$\bar{\mathbf{y}} = \frac{1}{J}\sum_{j=1}^{J}\mathbf{y}_j \ , \ \bar{\mathbf{S}} = \frac{1}{J}\sum_{j=1}^{J}(\mathbf{y}_j - \bar{\mathbf{y}})(\mathbf{y}_j - \bar{\mathbf{y}})^T$$

Now consider the projection of signals onto a K_C–dimensional space whose direction is defined by orthonormal vectors $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{K_C}$. Maximizing the variance of the projected data is equivalent to maximizing $\sum_{j=1}^{K_C} \mathbf{b}_j{}^T \bar{\mathbf{S}} \mathbf{b}_j$ and according to the theory of PCA the maximum is $\sum_{j=1}^{K_C} \lambda_j$ which is achieved by the eigenvectors $\mathbf{u}_1, ..., \mathbf{u}_{K_C}$ of the sample covariance matrix $\bar{\mathbf{S}}$ corresponding to the K_C largest eigenvalues $\lambda_1, \lambda_2, ..., \lambda_{K_C}$.

5.2 Reconstruction of Correlated Signals

Our reconstruction algorithm has two stages. In the first stage, we estimate the common sparse support set of the first component $\mathbf{w}_j$ using PCA followed by MDL. In the second stage, we subtract the contribution of the estimated common sparse support set and use Orthogonal Matching Pursuit [7] to reconstruct the innovation components of the signals.

5.2.1 Recovery of the Shared Support Set

As discussed earlier, the common components of the measurement vectors span a subspace with dimension K_C. We can estimate this subspace by computing the largest eigenvalues of the sample correlation matrix $\bar{\mathbf{S}}$. A simple thresholding can detect the largest eigenvalues. However, thresholding is only useful if the K_Cth largest eigenvalue is much larger than the $(K_C + 1)$st eigenvalue, which is the case if K_C is very small compared to M. In this chapter we use MDL, which is a method used to find the similarity of the smallest eigenvalue. Although the equal eigenvalue assumption may not hold in our example, it is nevertheless a proper approximation. This is in particular a good approximation when the innovative components of the signal have support sets that are random across all available dimensions. We are motivated to use PCA followed by MDL since even for a perfect model with spherical white noise, the so-called *signal subspace* remains unaffected by noise. Hence, MDL can successfully identify the largest eigenvalues of the sample covariance matrix.

The proposed algorithm is as follows.

1. Apply PCA on the subspace created by the span of the columns of Y and find the eigenvalues and corresponding eigenvectors:
$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$$

2. Use MDL to find K_C as the point that minimizes the MDL cost function:
$$MDL(i) = -\log \left[\frac{\prod_{k=i+1}^{M} \lambda_k}{\left(\frac{1}{M-i} \sum_{k=i+1}^{M} \lambda_k \right)^{M-i}} \right]^N + \frac{1}{2} i(2M - i) \log N$$

3. Pick the first K_C eigenvalues and form a subspace by the span of their corresponding eigenvectors.

4. Find the error of the projection of all the columns of $\mathbf{\Phi}$ onto the subspace found in the previous step.

5. The shared support set is the indices of the columns that have the minimum projection error.

5.2.2 Recovery of the Innovations

After finding the indices of the shared support set, we are essentially dealing with a compressed sensing problem with partially known support set in which the size of the known support set is K_C and the size of the unknown support set is K_j. In [20] , a modified-CS algorithm is proposed which uses convex relaxation to find the sparsest signal outside of the known part of the support set. However, in many of applications, ℓ_1 minimization approaches are not fast enough to meet the needs. Here, we use the approach proposed in [36]. In this algorithm, we can treat $\mathbf{w}_j$ as the sparse noise and subtract its contribution from the measurements $\mathbf{y}_j$.

Thus we just need to find a matrix $\mathbf{P}$ such that when we multiply both sides of equation (5.1) by $\mathbf{P}$, the contribution of $\mathbf{w}_j$ is removed.

$$\begin{aligned} \mathbf{x}_j &= \mathbf{w}_j + \mathbf{z}_j \\ \mathbf{y}_j &= \mathbf{\Phi}(\mathbf{w}_j + \mathbf{z}_j) \\ \mathbf{P}\mathbf{y}_j &= \mathbf{P}\mathbf{\Phi}\mathbf{w}_j + \mathbf{P}\mathbf{\Phi}\mathbf{z}_j \end{aligned} \tag{5.1}$$

We know that $\mathbf{\Phi}\mathbf{w}_j$ lies in the subspace created by the span of the columns of $\mathbf{\Phi}_C$. So by choosing matrix $\mathbf{P}$ as the projection matrix that projects the vectors of $\mathbb{R}^M$ onto the orthogonal complement subspace of the span of the columns of $\mathbf{\Phi}_C$, we can null out all the possible vectors of the form $\mathbf{\Phi}\mathbf{w}_j$. More formally if we choose $\mathbf{P}$ as
$$\mathbf{P} = \mathbf{I} - \mathbf{\Phi}_C(\mathbf{\Phi}_C^T \mathbf{\Phi}_C)^{-1} \mathbf{\Phi}_C^T$$

we will have

$$\begin{aligned} \forall \mathbf{w}_j,\ \mathbf{P}\mathbf{\Phi}\mathbf{w}_j &= (\mathbf{I} - \mathbf{\Phi}_C(\mathbf{\Phi}_C^T \mathbf{\Phi}_C)^{-1} \mathbf{\Phi}_C^T)\mathbf{\Phi}\mathbf{w}_j = 0 \\ \Rightarrow \mathbf{P}\mathbf{y}_j &= \mathbf{P}\mathbf{\Phi}(\mathbf{w}_j + \mathbf{z}_j) = \mathbf{P}\mathbf{\Phi}\mathbf{z}_j \end{aligned} \tag{5.2}$$

It can be shown that $\mathbf{P\Phi}$ satisfies the RIP condition and thus we can use standard compressed sensing algorithms such as the Orthogonal Matching Pursuit [7] to reconstruct each $\mathbf{z}_j$ separately.

The intuition behind our algorithm is that we are essentially reducing the sparsity level of the signals by subtracting the contribution of the common support set, thus we can achieve better performance compared to separate compressed sensing.

5.3 Simulation Results

In this section, we investigate the performance of the proposed algorithm through simulations. We run the algorithm on $J \in \{75, 100\}$ signals of length $N = 50$ and different sparsity levels and number of measurements. We then find the probability of exact reconstruction and average the results over 1000 simulation runs.

5.3.1 Comparison of PCA and SOMP in Reconstructing the Shared Support Set

In this part, we compare the performance of PCA with $J \in \{75, 100\}$ and that of SOMP with $J = 100$, which is proposed in [21] in terms of finding the shared support set of signals with sparsity level $K_j = 5$, $K_C = 5$. One obvious advantage of PCA is the ability to find K_C using statistical techniques such as MDL, while in SOMP we would need to know K_C in advance to determine the number of iterations. Moreover, as Figure 5.1 indicates, PCA outperforms SOMP in terms of the probability of exact reconstruction of the shared support set in G-JSM. This is mainly due to the fact that SOMP is tailor-made for the case that support set of the signals are exactly the same, while PCA considers the existence of noise and tries to minimize it.

5.3.2 Comparison of Joint and Separate Reconstruction of the Signals

In this part, we use $J \in \{75, 100\}$ signals with sparsity level $K_j = 3$, $K_C = 3$ to compare the performance of the proposed algorithm with separate compressed sensing in terms of probability of exact reconstruction of the signals. This comparison is shown in Figure 5.2. We first use PCA to find the common sparse support, subtract its contribution from the measurements and then use OMP to reconstruct the innovation components. The second plot corresponds to applying OMP directly on each signal and computing the probability of exact reconstruction. As the Figure illustrates, we can exploit the inter-sensor correlation and substantially decrease the number of measurements.

5.3.3 Performance of PCA Followed by MDL in an Imperfect G-JSM

Up to now, we have assumed that all signals have the shared support set ($\mathbf{w}_j$), However in practical situations, it is likely that a few number of sensors do not share the common support set. Here, we investigate the performance of applying PCA followed by MDL on the measurements in the scenario where some of the signals are not obeying the G-JSM model. Assume that we have $J = 100$ signals with $N = 50$, $M = 25$ and α percent of them share a common support set with $K_C = 3$, $K_j = 3$ and the other signals do not share the common support set and the sparsity level of them is $K_C + K_j = 6$. Figure 5.3 shows the probability of reconstruction of the common support set versus α using MDL and

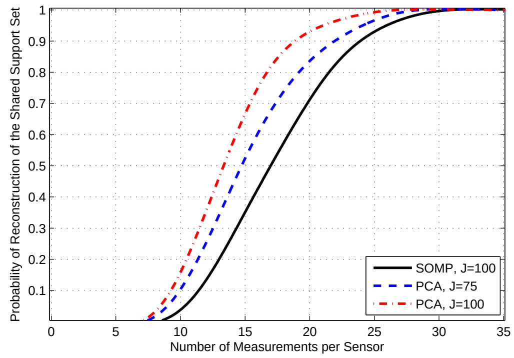

Figure 5.1: Reconstruction of the common support set in G-JSM using PCA and SOMP for $J \in \{75, 100\}$, $N = 50$, $K_C = 5$ and $K_j = 5$.

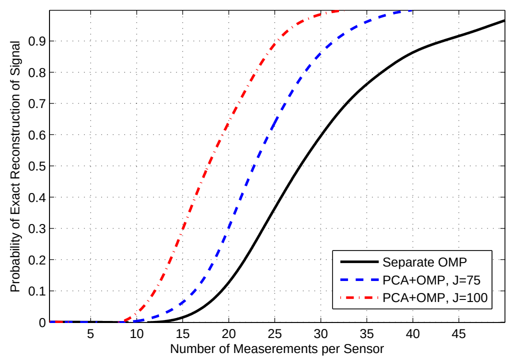

Figure 5.2: Reconstruction of signals in G-JSM using PCA-OMP and Separate OMP for $J \in \{75, 100\}$, $N = 50$, $K_C = 3$ and $K_j = 3$.

PCA. As this Figure indicates, even in the case where 10% of the signals do not obey the G-JSM model, our reconstruction algorithm achieves perfect reconstruction of the shared support set. In this case, subtracting the contribution of the common support set does not affect 10% of the signals but improves the probability of reconstruction of 90% of the signals as it does in a perfect G-JSM model. Thus we can conclude that the performance of the whole algorithm in this scenario is the same as G-JSM model for 90% percent of the signals and the same as separate CS for 10% of them.

5.4 Conclusion

In this chapter, we proposed the G-JSM model which is more general than the joint sparsity models in the literature and is suitable for practical applications. We applied PCA on the measurements to determine the common component of the signals and then used MDL to find the sparsity level of the common component. After finding the common component, we subtracted its contribution from the measurements and used standard compressive sensing on each individual signal to find the innovation components. We further showed that our proposed sampling and reconstruction algorithm outperforms

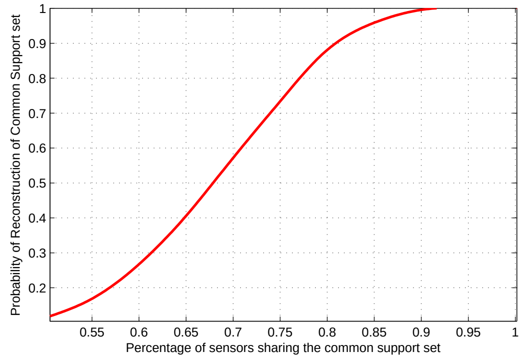

Figure 5.3: Reconstruction of the common support set in an imperfect G-JSM model for $J = 100$, $N = 50$, $M = 25$, $K_C = 3$ and $K_j = 3$.

the conventional SOMP algorithm for the reconstruction of the common support set of the signals in this particular joint sparsity model (G-JSM). Our simulation results show that even in this general model for sensor networks, we can achieve a much better performance than separate compressed sensing reconstruction.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this thesis, we have proposed and studied reconstruction algorithms for jointly sparse signals. Our focus was on applications that are well suited to the properties of CS, such as sensor/camera networks and video acquisition. The fact that in these applications there exists certain correlations among signals provides us with an additional information that if successfully exploited, can help us to get a better reconstruction rate in compressed sensing.

In Chapter 3, we proposed a very fast algorithm called simultaneous iterative hard thresholding (S-IHT) with a very low computational complexity to reconstruct jointly sparse signals. This algorithm could be used where we are dealing with the extremely large problem sizes. In this scenario, the signals do not need to be sparse in the same basis, nor they need to be sensed using the same sensing matrix. Moreover, we showed that this algorithm is faster than the DCS-SOMP algorithm while both of the algorithms show a similar performance in terms of reconstruction rate and robustness to noise.

In Chapter 4, we first proposed an approach (MIHT) based on iterative hard thresholding to reconstruct non-uniformly sparse signals. We then extended this approach to the joint sparsity models and multiple sensor case and proposed the DCS-MIHT algorithm to reconstruct signals with both spatial and temporal correlation. These types of correlations were both converted to a non-uniform prior information on the support set and the true support set is estimated through an iterative procedure.

In Chapter 5, we focused on the joint sparsity models and proposed the G-JSM, model which is a more general model than the other joint sparsity models proposed in the literature. We showed that the G-JSM model can be used in more practical applications. We then proposed a reconstruction algorithm for the signals of this model. As for the reconstruction, we applied PCA followed by MDL to find the common component of the signals and then used standard compressed sensing techniques to reconstruct the innovation components. We further showed that our proposed sampling and reconstruction algorithm outperforms the conventional SOMP algorithm for the reconstruction of the common support set of the signals in this particular joint sparsity model (G-JSM). Our simulation results show that even in this general model for sensor networks, we can achieve a much better performance than separate compressed sensing reconstruction.

6.2 Future Works

As a possible extension of this work, we propose to use a graphical model approach to address the problem of compressed sensing for jointly sparse signals. The idea is that we can incorporate the additional information of spatial correlation in the sparsity-promoting prior of a factor graph. One can show that for any additional information, we can design an "intelligent" prior whose i.i.d. statistical realization results in a structured sparsity model that is consistent with our additional information. We can then use an MMSE decoder on each coefficient of the signal to estimate the sparse signal. This MMSE estimation is done by a message passing algorithm over a factor graph with the sparsity-promoting priors. It can also be shown that after the simplifications of the message passing algorithm, we will reach to a modified version of the iterative thresholding algorithm. Moreover, it can be proved that the incorporated additional information in the factor graph will just be reflected in the de-noising operator of the iterative thresholding algorithm. Thus, with this framework, we can bridge the gap between probabilistic and deterministic approaches of compressed sensing with any kind of additional information. Making this framework mathematically rigorous is still an ongoing research in the compressed sensing community.

Bibliography

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006.

[2] E. J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling." November 2006.

[3] A. Makhzani and S. Valaee, "Reconstruction of jointly sparse signals using iterative hard thresholding," in *Communications (ICC), 2012 IEEE International Conference on*, IEEE, 2012.

[4] A. Makhzani and S. Valaee, "Reconstruction of a generalized joint sparsity model using principal component analysis," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on*, pp. 269–272, IEEE, 2011.

[5] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *Information Theory, IEEE Transactions on*, vol. 56, no. 4, pp. 1982–2001, 2010.

[6] D. Donoho and J. Tanner, "Neighborliness of randomly projected simplices in high dimensions," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, p. 9452, 2005.

[7] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655 –4666, 2007.

[8] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[9] A. Maleki and A. Montanari, "Analysis of approximate message passing algorithm," in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pp. 1–7, IEEE, 2010.

[10] A. Maleki, "Coherence analysis of iterative thresholding algorithms," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pp. 236–243, IEEE, 2009.

[11] V. Cevher, "Learning with compressible priors," *NIPS, Vancouver, BC, Canada*, pp. 7–12, 2008.

[12] S. Babacan, R. Molina, and A. Katsaggelos, "Bayesian compressive sensing using laplace priors," *Image Processing, IEEE Transactions on*, vol. 19, no. 1, pp. 53–63, 2010.

[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2346–2356, 2008.

[14] M. Tipping, "Sparse bayesian learning and the relevance vector machine," *The Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[15] S. Sarvotham, D. Baron, and R. Baraniuk, "Compressed sensing reconstruction via belief propagation," *preprint*, 2006.

[16] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *Information Theory, IEEE Transactions on*, vol. 57, no. 2, pp. 764–785, 2011.

[17] D. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Information Theory Workshop (ITW), 2010 IEEE*, pp. 1–5, IEEE, 2010.

[18] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *In Asilomar Conf. Signals, Sys., Comput*, pp. 1537–1541, 2005.

[19] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, "Distributed compressed sensing," tech. rep., 2005.

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

[21] J. Tropp, A. Gilbert, and M. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 5, pp. v/721 – v/724 Vol. 5, 2005.

[22] R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst, "Atoms of all channels, unite! average case analysis of multi-channel sparse recovery using greedy algorithms," *Journal of Fourier analysis and Applications*, vol. 14, no. 5, pp. 655–687, 2008.

[23] M. Mishali and Y. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4692–4702, 2008.

[24] J. Tropp, "Algorithms for simultaneous sparse approximation. part ii: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[25] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *Signal Processing, IEEE Transactions on*, vol. 53, no. 7, pp. 2477–2488, 2005.

[26] D. Wipf, *Bayesian methods for finding sparse representations*. PhD thesis, University of California, San Diego, 2006.

[27] T. Blumensath and M. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 298–309, 2010.

[28] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[29] M. Khajehnejad, W. Xu, S. Avestimehr, and B. Hassibi, "Weighted ℓ_1 minimization for sparse recovery with prior information," *Arxiv preprint arXiv:0901.2912*, 2009.

[30] S. Oymak, M. Khajehnejad, and B. Hassibi, "Weighted compressed sensing and rank minimization," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 3736–3739, IEEE, 2011.

[31] T. Tanaka and J. Raymond, "Optimal incorporation of sparsity information by weighted l1 optimization," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pp. 1598–1602, IEEE, 2010.

[32] L. Jacques, "A short note on compressed sensing with partially known signal support," *Signal Processing*, vol. 90, no. 12, pp. 3308–3312, 2010.

[33] D. Baron, M. Wakin, M. Duarte, S. Sarvotham, and R. Baraniuk, "Distributed compressed sensing," 2005.

[34] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *Information Theory, IEEE Transactions on*, vol. 56, no. 4, pp. 1982–2001, 2010.

[35] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[36] M. Davenport, P. Boufounos, and R. Baraniuk, "Compressive domain interference cancellation," *Structure et parcimonie pour la représentation adaptative de signaux (SPARS), Saint-Malo, France*, 2009.