SPARSE SIGNAL RECOVERY AND DETECTION UTILIZING SIDE INFORMATION

By

SHENG WANG

Bachelor of Science in Measuring and Control
Technology & Instrument
Hefei University of Technology
Hefei, Anhui, China
2008

Master of Science in Measuring Technology and
Instrument
Tianjin University
Tianjin, China
2010

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
October, 2017

SPARSE SIGNAL RECOVERY AND DETECTION UTILIZING SIDE INFORMATION

Dissertation Approved:

Professor Nazanin Rahnavard

Dissertation Advisor

Professor Martin Hagan

Professor Weihua Sheng

Professor Johnson P Thomas

Outside Committee Member

ACKNOWLEDGMENTS

There are many people who inspired and helped me along the way. First and foremost, I would like to express the deepest gratitude to my advisor, Dr. Nazanin Rahnavard. Thanks for taking me as a fresh graduate student who knew very little in signal processing. You set an example of excellence, and I truly look up to you in doing research, and being a great person. Thanks for your tremendous support, encouragement and endless hours of discussions that made this dissertation possible.

Next, I would like to thank my dissertation committee Professor Martin Hagan, Professor Weihua Sheng, and Professor Johnson P Thomas. Thanks for all your invaluable advice and relentless guidance throughout this process. In addition, I thank Professor Qi Cheng, and Professor Guoliang Fan for insightful discussions on the research reported in Chapter 6 and Chapter 7.

Further, I also would like to thank my parents Daoyin Wang, Ling Lu, and my parents-in-law, Zhenfeng Xiao, and Minghua Zhou. Thanks for your never-ending love, encouragement and support that made it possible to withstand all the challenges throughout my PhD studies.

Last but not least, to my beautiful wife and best friend, Weizhou Xiao. I can't put it in words and express how blessed and thankful I am to have you in my life. You have sacrificed so much for me. You have brighten me up every single day, cheered me up through good and through bad. You made this journey colorful, and I just want to say I love you very much, and I appreciate everything you have done for me.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A common procedure in today's data acquisition and signal processing is *capture-and-compress*, where physical information is captured and recorded by certain type of sensors, and then compressed for efficient data storage and communication. While working generally well in the past, this leads to a significant waste of data acquisition effort, as most of information captured are thrown away in the compression stage. A natural question is then raised: is it possible to combine data acquisition and compression in one stage?

The answer is positive for signals that can have a sparse representation in a proper basis, and the solution is the so-called Compressive Sensing (CS) [1–4]. In compressive sensing, a signal of length N is sampled by taking M measurements, with each being a random linear projection of the signal, and the measurement is said to be *compressed*, as $M \ll N$.

The under-sampled random measurements now become the compressed representation of the original signal. To extract, one needs to uncompress the measurement by solving an under-determined system of linear equations. Without any additional knowledge, this task is ill-posed, as there are more unknowns than the equations. What compressive sensing theory [1–4] states is that, reliable reconstruction of the signal is possible, provided that the signal is adequately sparse/compressible, and the sampling matrix satisfies the so called *Restricted Isometry Property*. Here by sparse/compressible, it is intended that the energy of the signal is primarily carried by a few coefficients (in a proper basis), referred to as significant coefficients, whereas

the energy of the rest of coefficient is insignificant.

The advantage of compressive sensing in solving under-determined systems makes it attractive in fields where increasing sampling rate is costly, and a great number of applications have been inspired. For instance, compressive sensing has found great applications in remote sensing [5], medical imaging [6], wireless communication systems [7], wireless sensor networks [8], multimedia processing [9, 10], and anomaly detection [11, 12].

Video coding employing compressive sensing is an emerging field. Since the reconstruction of under-sampled frames depends on the sparsity of the target signal, most endeavours in this area lie in exploiting the correlation, either temporally or spatially, to prompt sparsity of a video sequence. Authors in [13] studied a compressive sensing based video streaming technique for wireless multimedia sensor networks. It is shown that the difference frames of a video sequence coded by compressive sensing technique are more resilient to channel error compared to other coding techniques. In [14], a motion compensation based residual reconstruction for compressive sensing of video was proposed to explore the temporal correlation of video frames.

Aside from the temporal and spatial correlations, the structural feature of the video frames is of great importance for the compressive sensing reconstruction problem. Authors in [15] studied the sparse reconstruction task for clustered sparse signal. The clustered sparsity is modelled by the Ising model, and a novel algorithm, referred to as LaMP [15], was proposed to explore this structural feature. However, LaMP is sensitive to model parameters and the performance may degrade when a not very accurate model is selected. In order to explore the clustered sparsity, several parameters in the Ising model need to be estimated accurately, which may not be feasible for a resource limited encoder. In [16], a three-pattern model was proposed to prompt the clustered sparsity. Markov Chain Monte Carlo sampling (MCMC) is then used to infer the signal coefficients from the random samples. While no parameter needs

to be estimated before the reconstruction, CluSS [16] suffers from the huge computation time inherent in MCMC sampling. Moreover, the convergence of CluSS is not guaranteed.

Given these limitations, in Chapter 3 we study compressive sensing of difference frames in videos, and introduce a novel reconstruction method that exploits the structural characteristic, i.e., clustered sparsity in difference frames. Our method, referred to as structured reweighted ℓ_1 minimization (SRL1), estimates the signal support, and adjusts the weights associated with the signal coefficients in a weighted ℓ_1 minimization in an iterative fashion. For the signal support estimation we propose *local exploration*, and *global purification* to promote the clustered sparsity in difference frames. It is shown that by exploiting the clustered sparsity, isolated non-zero noise can be eliminated, and undiscovered signal coefficients can be retrieved. It should be noted that these steps are done based on the clustered sparsity, rather than the exact signal support distribution. This makes our method robust and distinct from many sophisticated algorithms. Experimental results show the effectiveness of our method.

Compressive sensing and reconstruction for binary sparse signals plays a key role in engineering fields including control engineering, aerospace engineering and more. One example is the fault identification problem, where the fault pattern is represented by a binary vector $\underline{x} \in \{0,1\}^n$, with "1" indicating a fault has happened. The task is to locate the set of faults pattern given a set of measurements.

What makes the binary compressive sensing unique is that, unlike conventional compressive sensing tasks, the magnitude of non-zero coefficients is fixed to "1". As discussed in [17], this congruity of the non-zero coefficients makes the reconstruction of the binary sparse signal more challenging than those signals whose non-zero coefficients have random magnitudes.

Many efforts have been taken to explore the binary feature and sparsity of the signals. In [18], the binary prior is explored by the unique sum property. In [19], a

two-modal Gaussian distribution with peaks centering at "0" and "1" is served as the prior distribution of the message passing compressive sensing framework, and the resulted algorithm is referred to as NBP, which represents the state-of-the-art and has limit approaching performance under large noise conditions. In [20], an interior algorithm is proposed to promote the binary signal reconstruction quality. In [17], to avoid the challenge in binary compressive sensing, a pre-processing stage is involved to map the binary signal to a random signal. After this pre-mapping process, the binary reconstruction problem turns to a general compressive sensing problem.

In Chapter 4, we are trying to solve the binary sparse reconstruction task from a different perspective. We handle the task by formulating it as an convex optimization task with a novel regularization term. Specifically, it is well known that among the infinite candidates, compressive sensing selects the sparest solution that agrees with the projection by the sparsity promoting property of ℓ_1-norm minimization. On the other hand, ℓ_∞-norm minimization, favors the representation whose coefficients are roughly in the same absolute magnitude [21–23]. We show that these two extremes can be combined in the binary compressive sensing problem to promote the reconstruction quality. This is done by minimizing the sum of the ℓ_1-norm and ℓ_∞-norm, up to a scaling factor and a shifting vector.

With the benefits of the two norms, our novel formulation is able to promote both sparsity and binary property effectively. The new formulation is convex and can be solved by a general convex optimization solver. We will see that although NBP [19] exhibits limit approaching property under large noise, our method gives better reconstruction under small noise. Besides, our technique turns out to be more robust when an inaccurate signal model is selected.

The sparse reconstruction techniques developed in Chapter 3 and Chapter 4 are built on the success of basis pursuit and LASSO [2, 24]. In the meantime, the sparse reconstruction task can be treated from a Bayesian aspect as well, where the distri-

bution of the signal is modelled by a mixture of density components. For example, in [25] and [26], the signal is modelled by a mixture of Laplace densities, and the coefficients are inferred by approximate message passing (AMP). In [27], two types of mixture models, *i.e.,* a Bernoulli-Gaussian mixture, and a two-state Gaussian mixture, are utilized as the prior distributions of the wavelet transform coefficients of images.

It can be seen that, the density components in these studies [25–28] are symmetrically distributed around their means. In practice, the underling density of the signal coefficients could be asymmetric. One example can be found in sensor networks, where certain type of weather data, let us say outside air temperature, when subtracted from the historical average, is asymmetrically positive or negative when the disrupting weather phenomena is heat or cool, respectively. Additionally, it is found that in microarray time course data analysis [29], the gene expressions involved in embryo are more often developed with an increasing trend.

Therefore, distributions including normal and Laplace in this case may not be a proper model to capture all the salient features, and dealing with asymmetric signals calls for more sophisticated approaches. Two related work can be found in [30] and [31]. In [30], a normal density mixture is employed as the prior distribution, and a powerful algorithm is put forward to estimate the signal while learning the mixture via Expectation Maximization [32]. In [31], an effective technique is developed to handle non-negative sparse signals by modelling the signal with a non-negative normal mixture.

While being highly effective in general, both [30] and [31] have limitations. Concretely, the mixture using normal components in [30] is known to be sensitive to outliers, and the performance degrades with smaller sample size [33]. Meanwhile, the work [31] is designed exclusively for non-negative signals, and is not capable in handling signals with mixed positive and negative significant elements.

Given these limitations, in Chapter 5, we are aiming to develop a new and more generalized framework to solve for CS-based reconstruction of asymmetrical signals. To this end, a two-state normal and skew normal mixture density is proposed. The significant coefficients of the signal are represented by a skew normal density [34], which is more general than the normal one, and comes with more flexibility in dealing with the asymmetric features. A message passing algorithm is developed to estimate the signal from the measurements. A fast gradient-based estimator is designed to infer the density of each state.

The performance of our proposed technique is examined under a variety of tests, including phase transition, noisy reconstruction, support set recovery rate, and runtime tests. Furthermore, our technique finds promising application in real world data set. We show that in weather sensor network application, the disrupting weather phenomena can be successfully learned by our proposed technique. Overall, experimental results of both simulated and real-world tests show that, our technique can effectively exploit the asymmetric feature of the signal, while being competitively efficient in solving large scale problems.

Following Chapter 5, we move one step further in Chapter 6 by approaching the compressive sensing of clustered sparse signals, where the magnitudes of each cluster are distributed asymmetrically about the corresponding cluster mean. One typical example for such signals can be found in sensor networks, where multiple events of different types and intensities are likely to occur simultaneously, and clusters of different events may in turn exhibit varying features.

To get a faithful reconstruction of the signals, we adopt a *divide-and-conquer* methodology, and a technique consisting three modules is developed.

First of all, to address the skewness feature, a finite skew normal density mixture is utilized to model the prior distribution of the signal. Skew normal density [34] generalizes normal density, and is more effective in accommodating asymmetric features.

An efficient approximate message passing algorithm, which takes the mixture density, and the hidden states of signal coefficients as inputs, is designed to iteratively derive the marginal posterior, and the Minimal Mean Squared Error (*MMSE*) estimate of the signal.

Subsequently, following the approximate message passing module, an Expectation-Maximization based algorithm is developed to estimate the mixture density from the *MMSE* estimate of the signal. The number of mixture components is estimated in an efficient and non-parametric way.

Moreover, given the *MMSE* and the mixture density estimates from previous modules, a loopy message passing based algorithm is designed, where the compatibility of neighboring coefficients is regularized by the *Potts* model [35], after which the hidden states of signal coefficients can be estimated, and the clustered property can be promoted.

Overall, our proposed technique alternates between exploiting the measurement, drawing inference of the finite mixture model, and taking advantage of the clustered property. These three modules work sequentially and iteratively, after which, a refined reconstruction of the signal can be obtained.

To the best our knowledge, our method is among the first few work taking both asymmetry and clustered sparsity into account in compressive sensing tasks. Compared to [7] which analyzed general asymmetrical sparse signals, our developed technique [36] is designed to exploit the clustered features on top of asymmetry. Moreover, compared to the two-states mixture model [7] with fixed location parameter, our technique utilizes a finite mixture model, which allows for multiple skew normal density components with arbitrary location parameters, and can therefore accommodate more general signals.

Existing studies [27, 37, 38] utilized Markov random field and *Ising* model [39] to exploit the clustered property. While being highly effective in recovering the support

sets of signals, they are incapable of discriminating diverse hidden states of significant coefficients. Taking advantages of the *Potts* model, our developed method not only can promote clustered property, but is also adequately responsive to different hidden states of signal coefficients. Therefore, compared to existing methods, clustered property is exploited in a more efficient way.

Compressive sensing combines data acquisition and compression in one stage, and permits efficient utilization of data collected by recording devices. Next, we are going to investigate wireless radio spectrum, which is another resource that calls for better utilization. Nowadays, the spectrum is pre-allocated to license holders by governmental agencies. While performed reasonably well in general, it is observed that usage of spectrum is highly imbalanced [40], where the majority of usage is carried over a certain portions of the spectrum, with the remaining being highly under-utilized.

To deal with this inefficiency, Cognitive Radio [41] was proposed to improve the usage of the valuable spectrum. This is achieved by allowing unlicensed users to operate on the licensed spectrum in an opportunistic manner.

Spectrum sensing is a critical part of cognitive radio. It involves detecting the primary/licensed user's signals which may be contaminated by noise, and enables efficient utilization of temporarily unoccupied radio frequency bands. Eigenvalue based spectrum sensing techniques [42–49], relying on the statistical characteristics of the eigenvalues of the receiving sample covariance matrix, have been recently proposed and shown to outperform classical energy detection based techniques [44, 47, 50]. This advantage comes from the inherent feature of eigenvalue based methods that no prior knowledge on primary user signal or noise power level is needed.

Spectrum sensing technique based on the statistics of eigenvalue typically consists of two steps. The first step is to describe the distributions of the extreme eigenvalues. Next, the distributions of those eigenvalues are used to calculate test statistics for

hypothesis tests. However, this is not an easy task, and the major discouragement comes from the descriptions of extreme eigenvalues. This is due to the fact that exact distributions of these eigenvalues lead to infinite series, and cannot be calculated efficiently, except under extreme small settings.

Given the difficulty in large dimensional settings, depending on the way of how the distributions of extreme eigenvalues are treated, efforts taken in this area can be divided into two directions. The first discuss the properties of extreme eigenvalue under an extreme small setting [42, 43], which cannot be extended to a more general and larger dimensional setting due to unfavorable computational cost. The second is based on asymptotic and limiting assumptions on sample size and the number of cooperated sensors [44, 46, 47, 49, 50], which is also not suitable for real application scenarios.

As we can see, neither of these two methods provide practical solutions for real world scenarios in which the dimensional setting is finite. Therefore, in Chapter 7, we investigate a more realistic region where the sample size or number of sensors is finite. We begin our efforts by analyzing the properties of the eigenvalues under a small dimensional setting where the samples and sensors' size are finite. Inspired from recent development in multivariate analysis of variance (MANOVA) [51], we derive the distribution of the largest eigenvalue of finite sample covariance matrix in the form of sum of two gamma random variables. Next, noticing the connection between moment generating function of standard condition number and confluent form of Lauricella function, we obtain compact expressions for its Probability Density Function as well as Cumulative Distribution Function. Further, these results are then applied to analyze the detection performance of Generalized Likelihood Ratio Test. Simulations results show that the proposed method outperform other eigenvalue based spectrum sensing techniques for finite number of samples and sensors.

Finally, Chapter 8 summarizes the dissertation, and discusses potential extensions

and future research directions of this work.

CHAPTER 2

BACKGROUND

In this section, we first provide a brief overview of related vector norms that are used frequently in the thesis. Next, system of linear equations is reviewed, and different settings of linear equations are discussed. Additionally, we introduce compressive sensing, and show that under certain conditions, the under-determined system of linear equations can be solved effectively by ℓ_1-norm minimization technique. Further, we show compressive sensing tasks can be treated from a Bayesian perspective, and describe *belief propagation*, which is able to take inference of the target signal by exchanging local beliefs. Moreover, *approximate message passing* is introduced as a powerful technique to solve compressive sensing tasks at a reduced complexity.

2.1 Vector Norms

Given a vector $\underline{x} = [x_1, \ldots, x_N]^\intercal \in \mathbb{R}^{N\times 1}$, vector norm is a function that assigns a non-negative magnitude to the vector. One commonly used vector norm is the ℓ_p-norm. Specifically, for a real number $p \geq 1$, the ℓ_p-norm of $\underline{x}$ is evaluated as,

$$\|\underline{x}\|_p = (|x_1|^p + |x_2|^p + \ldots + |x_N|^p)^{1/p}. \tag{2.1}$$

ℓ_p-norm is referred to as the ℓ_2-norm when setting $p = 2$ in (2.1), *i.e.,*

$$\|\underline{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_N^2}, \tag{2.2}$$

and (2.2) is also known as the *Euclidean* norm in the literature.

Similarly, ℓ_1-norm can be obtained by setting $p = 1$, and (2.1) is then reduced to

$$\|\underline{x}\|_1 = \sum_{n=1}^{N} |x_n|, \tag{2.3}$$

where $|x_n|$ represents the absolute value of x_n.

Another useful vector norm is the ℓ_∞-norm, and it is defined as the maximum of the absolute values of the entire entries, *i.e.,*

$$\|\underline{x}\|_\infty = \max_n |x_n|, \tag{2.4}$$

with $1 \leq n \leq N$.

It is also helpful to count the number of nonzero entries of a vector, as it gives a measure of complexity and sparsity of a vector. Therefore, defining $0^0 = 0$, the ℓ_0-norm is equal to,

$$\|\underline{x}\|_0 = |x_1|^0 + |x_2|^0 + \ldots + |x_N|^0. \tag{2.5}$$

It should be noted that, as (2.5) is not homogeneous [52], using the term *norm* in ℓ_0-*norm* is abuse of terminology. Nevertheless, following [2], the term *norm* is kept for its *zero-counting* property.

2.2 System of Linear Equations

System of linear equations is a collection of linear equations over a certain set of variables. Concretely, a general system of M linear equations and N variables can be

written as,

$$\begin{aligned} y_1 &= a_{11}x_1 + a_{12}x_2 + \ldots \;\; + \;\; a_{1N}x_N \\ y_2 &= a_{21}x_1 + a_{22}x_2 + \ldots \;\; + \;\; a_{2N}x_N \\ &\vdots \qquad\qquad\quad \vdots \\ y_M &= a_{M1}x_1 + a_{M2}x_2 + \ldots \;\; + \;\; a_{MN}x_N, \end{aligned} \tag{2.6}$$

with $x_1, x_2, \ldots, x_N$ denoting the unknowns variables, $y_1, y_2, \ldots, y_M$ being the measurements, and $a_{11}, a_{12}, \ldots, a_{MN}$ being the coefficients of the system.

The above linear equations can be expressed in a matrix form as well. Specifically, let $\underline{x} = [x_1, \ldots, x_N]^\intercal \in \mathbb{R}^{N\times 1}$, and $\underline{y} = [y_1, \ldots, y_M]^\intercal \in \mathbb{R}^{M\times 1}$, then (2.6) can be written as

$$\underline{y} = \mathbf{A}\underline{x}, \tag{2.7}$$

where $\mathbf{A} \in \mathbb{R}^{M\times N}$ represents the coefficients matrix, with a_{mn} being the entry at row m and column n, where $1 \leq m \leq M$, and $1 \leq n \leq N$.

Given the system of linear equations (2.7), our task is to solve for the correct solution $\underline{x}$, with $\underline{y}$ and $\mathbf{A}$ being known as a priori. When $\mathbf{A}$ is square, *i.e.,* $M = N$, and rows of $\mathbf{A}$ are independent, the system is said to be *well-determined*, and exact and unique solution of $\underline{x}$ can be found in a variety of ways. For instance, multiplying the matrix inverse $\mathbf{A}^{-1}$ on both sides of (2.7) leads to,

$$\underline{x} = \mathbf{A}^{-1}\underline{y}. \tag{2.8}$$

It should be noted that most of systems of linear equations utilized in practice are not *well-determined*. For example, predictive modelling and regression analysis tasks generally involves analyzing *over-determined* systems, where $M \gg N$. Besides, in

what will be described later, the target of interest of compressive sensing is an *under-determined* system, where $M < N$. In both cases, finding the exact and unique solution of (2.7) becomes more challenging.

2.3 Compressive Sensing

Compressive sensing [1–4] is a powerful technique for solving certain *under-determined* linear inverse problems. In compressive sensing, the signal is sampled by random linear projections as,

$$\underline{y} = \mathbf{A}\underline{x} + \underline{e}, \tag{2.9}$$

where $\underline{x} \in \mathbb{R}^{N\times 1}$ is the unknown sparse/compressible signal, $\mathbf{A} \in \mathbb{R}^{M\times N}$ is the known sampling matrix with $M \ll N$, $\underline{y} \in \mathbb{R}^{M\times 1}$ is the observed measurements, and $\underline{e} \in \mathbb{R}^{M\times 1}$ is the measurement noise. It should be noted that, by *sparse/compresisble*, it is intended that $K \ll N$ entries of the signal have significant magnitudes, with the remaining entries being insignificant, and the ratio K/N is referred to as the *sparsity rate*.

Similar to system of linear equations, in compressive sensing, the task is to solve for the unknown target signal $\underline{x}$, given the measurement $\underline{y}$ and sampling matrix $\mathbf{A}$. It is noted that $M \ll N$ in (2.9), therefore, the inverse problem has infinitely many solutions.

One plausible solution is to find an approximate $\hat{\underline{x}}$ that making $\mathbf{A}\hat{\underline{x}}$ as close to $\underline{y}$ as possible. This can be casted as the following ℓ_2-norm minimization procedure

$$\hat{\underline{x}}_{ls} = \underset{\underline{x}}{\operatorname{argmin}} \left\|\underline{y} - \mathbf{A}\underline{x}\right\|_2, \tag{2.10}$$

where $\hat{\underline{x}}_{ls}$ is referred to as the *least square* solution. However, it turns out that solving (2.10) cannot reconstruct the sparse signal. One illustrative example can be found

in Fig. 2.1, where a signal $\underline{x}$ is generated with length $N = 100$, and is made sparse by setting 80 entries to 0. The remaining $K = 20$ entries is made to be significant entries by drawing from normal distribution with mean 0, and standard deviation 20. The signal is then sampled by (2.9), where coefficients of $\mathbf{A}$ is generated from standard *Gaussian* ensemble, with each column being normalized to unit norm, *i.e.*, $\|A_{\cdot n}\|_2 = 1$, for $n = 1, \ldots, N$, and the the measurement is noiseless. As can be seen

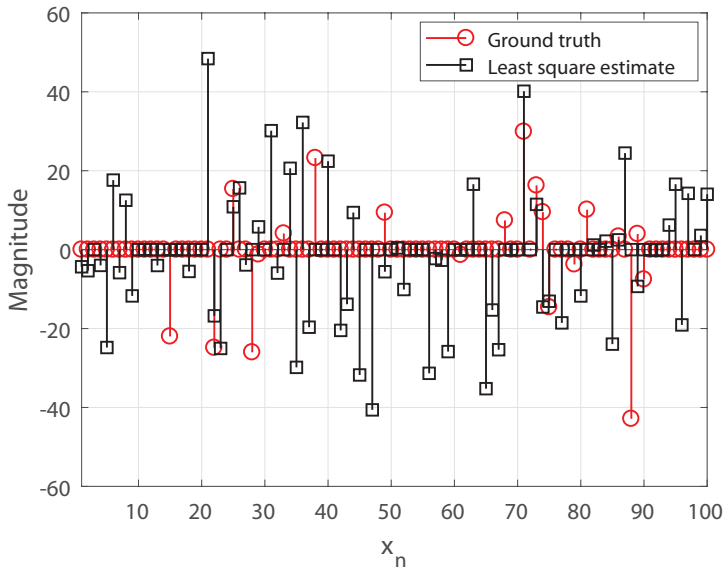


Figure 2.1: Least square estimate of a sparse signal.

in Fig. 2.1, the solution found by the least square method (2.10) deviates from the ground truth severely, and fails to recover the sparse signal.

Compressive sensing is a paradigm to solve for the under-determined system, and it permits reliable reconstruction of the signal by exploiting the sparsity. Concretely, under the conditions [53], [54]:

1. the signal $\underline{x}$ is sufficiently sparse/compressible,

2. the sampling matrix $\mathbf{A}$ obeys a *uniform uncertainty principle*,

then solving the following ℓ_1 minimization,

$$\hat{\underline{x}} = argmin\|\underline{x}\|_1, \tag{2.11}$$
$$s.t.\ \|\underline{y} - \mathbf{A} * \underline{x}\|_2 \leq \epsilon,$$

leads to a solution that is within the noise level of the unknown sparse signal, *i.e.,*

$$\|\hat{\underline{x}} - \underline{x}^*\|_2 \leq C_s * \epsilon, \tag{2.12}$$

where $\epsilon = \|\underline{e}\|_2$ is the noise level, $\underline{x}^*$ is the sparse solution and, C_s is a constant determined by the so called *S-restricted isometry constant* [3, 53] of sampling matrix $\mathbf{A}$.

As a continuation of previous example, (2.11) is utilized to reconstruct the signal, and the results are plotted in Fig. 2.2. As can be seen, ℓ_1-norm minimization faithfully reconstructs the signal, and effectively recovers the sparsity the signal.

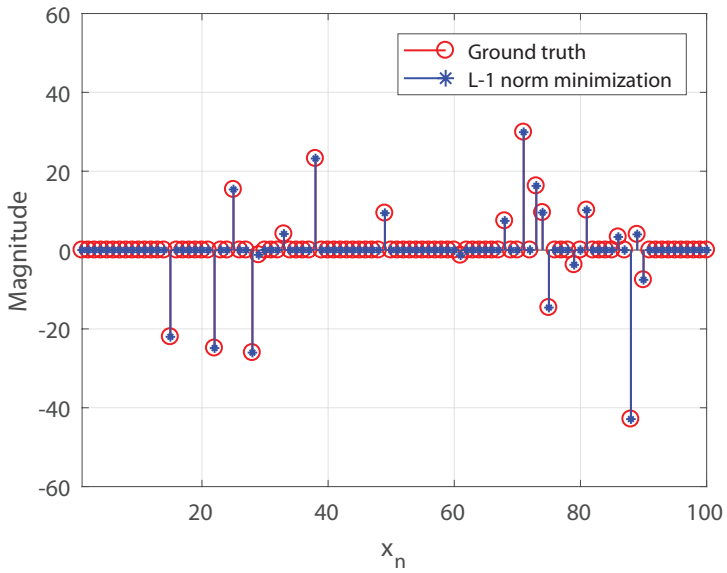


Figure 2.2: Reconstruction of a sparse signal by ℓ_1-norm minimization.

2.4 Bayesian Inference and Belief Propagation

The sparse reconstruction task of compressive sensing can be solved from a *Bayesian* perspective. Concretely, the signal $\underline{x}$ is assumed to be the outcome of random variable vector $\underline{X}$, where the distribution is determined by the prior distribution $p(\underline{X} = \underline{x})$. Besides, the measurement vector $\underline{y}$ is assumed to be realization of $\underline{Y}$. Under *Gaussian* measurement noise environment, the distribution is determined by conditional distribution as, $p(\underline{Y} = \underline{y}|\underline{X} = \underline{x}, \sigma_e)$, with σ_e being the standard deviation of *Gaussian* measurement noise. Therefore, the measurement model can be written as [55],

$$p(\underline{Y} = \underline{y}|\underline{X} = \underline{x}, \sigma_e) = (2\pi\sigma_e^2)^{-M/2} \exp\left(-\frac{1}{2\pi\sigma_e^2}\|\underline{y} - \mathbf{A}\underline{x}\|^2\right) \tag{2.13}$$

and *Bayesian* inference involves calculating the following posterior,

$$p(\underline{X} = \underline{x}|\underline{Y} = \underline{y}) = \frac{p(\underline{X} = \underline{x}, \underline{Y} = \underline{y}, \sigma_e)}{p(\underline{Y} = \underline{y})} \tag{2.14}$$

However, unless N and M are very small, $p(\underline{Y} = \underline{y})$ cannot be evaluated analytically in practice, exact derivation of (2.14) is generally intractable. Message passing, also known as belief propagation decoding [28, 56], allows efficient approximation of the marginal posterior (2.14), by exchanging messages between variable nodes $\underline{x}$ and check nodes $\underline{y}$, where the messages carry the probability distribution of the corresponding variable nodes.

Specifically, let $\nu^i_{x_n \to y_m}(x_n)$ be the message sent from variable node x_n to check node y_m at i-th iteration, and denote $\nu^i_{y_m \to x_n}(x_n)$ as the reverse, with both messages encoding the belief, namely probability density function (*pdf*) of x_n.

The message from variable node x_n to check node y_m, $\nu^i_{x_n \to y_m}(x_n)$, is calculated in a way similar to Fig. 2.3. Concretely, it is evaluated as the product of the prior distribution of x_n, *i.e.,* $f(x_n)$, and all incoming messages to x_n from check nodes $\underline{y}$,

with the exception of the one from y_m [28], *i.e.*,

$$\nu^i_{x_n \to y_m}(x_n) \cong f(x_n) \prod_{u \in \{1,\ldots,M\} \setminus m} \nu^{i-1}_{y_u \to x_n}(x_n), \tag{2.15}$$

where $\cong$ denotes identity up to a normalization constant.

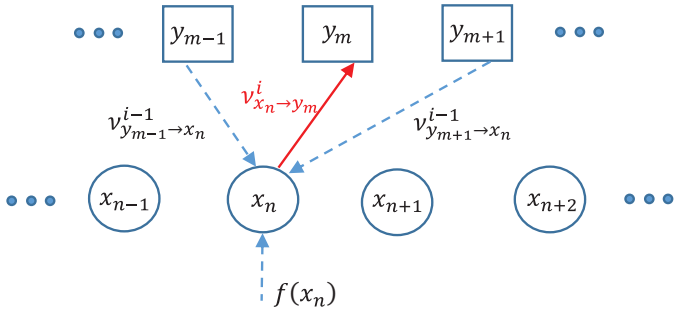


Figure 2.3: The sent message from x_n to y_m

The message from check node y_m to variable node x_n, $\nu^i_{y_m \to x_n}(x_n)$ can be evaluated in a similar way. Specifically, as shown in Fig. 2.4, $\nu^i_{y_m \to x_n}(x_n)$ is calculated as the product of the constraint on y_m, and all incoming messages of y_m from variable nodes $\underline{x}$, with the exception of the one from x_n. Under white Gaussian noise environment, the constraint on y_m is

$$con(y_m, \underline{x}) = \frac{1}{\sqrt{2\pi}\sigma_e} \exp\left(-\frac{(y_m - A^{\intercal}_{m\cdot}\underline{x})^2}{2\sigma_e^2}\right), \tag{2.16}$$

where $A_{m\cdot}$ represents the m-th row of A. In what comes next, A_{mn} is the entry at the m-th row and n-th column of A. Similarly, $A_{\cdot n}$ denote the n-th column of A.

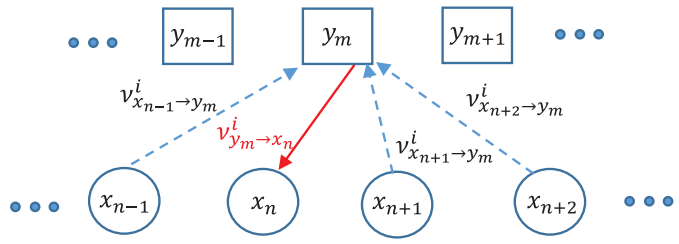


Figure 2.4: The sent message from y_m to x_n

Since $con(y_m, \underline{x})$ involves all variable nodes $\underline{x}$, the product is then marginalized

by sum over all $\underline{x}$ but x_n [28], *i.e.,*

$$\nu^i_{y_m \to x_n}(x_n) \cong$$
$$\underbrace{\int \cdots \int}_{N-1} con(y_m, \underline{x}) \prod_{\substack{t=1 \\ t \neq n}}^{N} \nu^i_{x_t \to y_m}(x_t) \underbrace{\mathrm{d}x_1 \cdots \mathrm{d}x_t \cdots \mathrm{d}x_N}_{t \in \{1,\ldots,N\} \backslash n} . \tag{2.17}$$

2.5 Approximate Message Passing

2.5.1 Minimal-Mean-Squared-Error (MMSE) Inference by Approximation

In [28] authors proposed an effective technique, *CSBP*, where the compressive sensing of sparse signal $\underline{x}$ is formulated as a graphical model, and approximate Bayesian inference of the signal is realized by exchanging messages[1] between $\underline{x}$ and $\underline{y}$. It should be noted that in [28], each message is represented by $\Delta = O(\sigma_L/\sigma_S)$ uniform samples of the corresponding *pdf*, where σ_L and σ_S denotes the standard deviation of the significant, and insignificant coefficients, respectively. Therefore, a storage of $O(\Delta N \log(N))$ is needed.

While being reasonably effective in some cases, this procedure calls for considerably large memory space, and is not satisfactorily efficient under large signal dimensionality and very small σ_S.

Approximate Message Passing (AMP), on the other hand, is more efficient. Specifically, let

$$\mathcal{N}(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.18}$$

be the Gaussian density with mean μ, and variance σ^2. Under adequately large M and N, messages in AMP are approximated by Gaussian density, which is further

[1]Each message encodes the marginal distribution of signal coefficient x_n, for $n \in [1, \ldots, N]$

parameterized by the corresponding mean and variance [25–27], *i.e.,*

$$\nu^i_{x_n \to y_m}(x_n) \approx \mathcal{N}\big(x_n; \mu^i_{x_{nm}}, \sigma^{2\,i}_{x_{nm}}\big), \tag{2.19}$$

where

$$\mu^i_{x_{nm}} = \int_{-\infty}^{\infty} x_n \nu^i_{x_n \to y_m}(x_n)\, \mathrm{d}x_n, \tag{2.20}$$

$$\sigma^{2\,i}_{x_{nm}} = \int_{-\infty}^{\infty} (x_n - \mu^i_{x_{nm}})^2 \nu^i_{x_n \to y_m}(x_n)\, \mathrm{d}x_n, \tag{2.21}$$

and

$$\nu^i_{y_m \to x_n}(x_n) \approx \mathcal{N}\big(x_n; \mu^i_{y_{mn}}, \sigma^{2\,i}_{y_{mn}}\big), \tag{2.22}$$

in which

$$\mu^i_{y_{mn}} = \frac{1}{A_{mn}} \times \left(y_m - \sum_{t \in \{1,\ldots,N\} \setminus \{n\}} A_{mt} \mu^i_{x_{tm}} \right), \tag{2.23}$$

$$\sigma^{2\,i}_{y_{mn}} = \frac{1}{A^2_{mn}} \times \left(\sigma^2_e + \sum_{t \in \{1,\ldots,N\} \setminus \{n\}} A^2_{mt} \sigma^{2\,i}_{x_{tm}} \right). \tag{2.24}$$

Following the notation in [25] and [26], define the mean operator $\mathbb{F}(\kappa, \varsigma)$, and variance operator $\mathbb{G}(\kappa, \varsigma)$ as,

$$\mathbb{F}(\kappa, \varsigma) = \mathrm{E}_{f_{v \to c}}(X), \tag{2.25}$$

$$\mathbb{G}(\kappa, \varsigma) = \mathrm{Var}_{f_{v \to c}}(X), \tag{2.26}$$

where the *pdf* of X is $f_{v \to c}(x) \cong \mathcal{N}(x; \kappa, \varsigma) f(x)$, with $f(x)$ denoting the prior distribution of X.

Therefore, with the above approximation, $\nu_{x_n \to y_m}^{i+1}$ can be written as,

$$\begin{aligned} \nu_{x_n \to y_m}^{i+1}(x_n) &\cong \mathcal{N}(x_n; \kappa_{nm}^i, \varsigma_n^i) f(x_n) \\ &\cong \mathcal{N}\big(x_n; \mu_{x_{nm}}^{i+1}, \sigma_{x_{nm}}^{2\ i+1}\big), \end{aligned} \tag{2.27}$$

where

$$\kappa_{nm}^i = \sum_{\substack{u=1 \\ u \neq m}}^{M} A_{un} \mu_{y_{un}}^i, \quad \varsigma_n^i = \frac{1}{M} \sum_{u=1}^{M} A_{un}^2 \sigma_{y_{un}}^{2\ i}, \tag{2.28}$$

$$\mu_{x_{nm}}^{i+1} = \mathbb{F}(\kappa_{nm}^i, \varsigma_n^i), \qquad \sigma_{x_{nm}}^{2\ i+1} = \mathbb{G}(\kappa_{nm}^i, \varsigma_n^i). \tag{2.29}$$

As can be seen, following the above update rule, variable node x_n sends a *unique* pair of $\big(\mu_{x_{nm}}^i, \sigma_{x_{nm}}^{2\ i}\big)$ to y_m, for $m = 1, \ldots, M$. In turn, check node y_m sends a *unique* pair of $\big(\mu_{y_{mn}}^i, \sigma_{y_{mn}}^{2\ i}\big)$ to x_n, for $n = 1, \ldots, N$. As a result, the memory requirement scales with $2MN$.

The work [25] and [26] further show that, with mild accuracy compromise, the storage requirement can be further reduced by first order approximation.

Specifically, by first order approximation, it is intended that variable node x_n sends a *uniform* message to all check nodes, *i.e.*, $\nu_{x_n \to y_m}^i = \mathcal{N}\big(\mu_{x_n}^i, \sigma_{x_n}^{2\ i}\big)$, for $m = 1, \ldots, M$. Similarly, check node y_m sends a *uniform* message to all variable nodes,

i.e., $\nu^i_{y_m \to x_n} = \mathcal{N}\big(\mu^i_{y_m}, \varsigma^i\big)$, for $n = 1, \ldots, N$, in which [25–27],

$$\mu^i_{x_n} = \mathbb{F}(\kappa^{i-1}_{x_n}, \varsigma^{i-1}), \tag{2.30}$$

$$\sigma^{2\,i}_{x_n} = \mathbb{G}(\kappa^{i-1}_{x_n}, \varsigma^{i-1}), \tag{2.31}$$

$$\mu^i_{y_m} = y_m - \sum_{n=1}^{N} A_{mn}\mu^{i-1}_{x_n} + \frac{\mu^{i-1}_{y_m}}{M}\sum_{n=1}^{N} \mathbb{F}'(\kappa^{i-1}_{x_n}, \varsigma^{i-1}), \tag{2.32}$$

$$\varsigma^i = \sigma^2_e + \frac{1}{M}\sum\nolimits_{n=1}^{N} \sigma^{2\,i}_{x_n}, \tag{2.33}$$

$$\kappa^{i-1}_{x_n} = \sum\nolimits_{m=1}^{M} A_{mn}\mu^{i-1}_{y_m} + \mu^{i-1}_{x_n}, \tag{2.34}$$

with $\mathbb{F}'(\kappa^{i-1}_{x_n}, \varsigma^{i-1})$ being the first order derivative of $\mathbb{F}(\kappa^{i-1}_{x_n}, \varsigma^{i-1})$ with respect to $\kappa^{i-1}_{x_n}$.

CHAPTER 3

SRL1: Structured Reweighted ℓ_1 Minimization for Compressive Sampling of Videos

In this chapter, we investigate the compressive sensing of difference frames in videos, and develop a novel and effective reconstruction method that is capable of boosting the reconstruction quality by exploiting the structural characteristic of video sequences.

3.1 Introduction

Video coding employing compressive sensing is an active field of research. What makes this topic unique is that, aside from the sparsity, video frames are correlated both temporally and spatially. As discussed in previous chapters, the reconstruction quality of compressive sensing task depends heavily on the sparsity of the target signal. Therefore, most endeavours in this area lie in prompting sparsity of a video sequence by exploring the corrections of video frames.

In this chapter, we aim to solve the compressive sensing task of videos by taking advantage of the clustered sparsity of the difference frames. In our technique, the difference frame is calculated as the algebraic difference of the non-reference frame *w.r.t.* the reference frame, and is then compressive sampled by projection with a random matrix.

The clustered sparsity of the difference frame in a video sequence is explored by our proposed structure-aware reconstruction technique, referred to as SRL1. The proposed method reconstructs the difference frame of a video sequence and estimates

its signal support in an iterative fashion. The clustered sparsity of current reconstruction is utilized to estimate the signal support with which the weights associated with signal coefficients can be estimated. The weights are then used to direct the reconstruction of the difference frame in the next iteration. It is shown that through what we call *local exploration* and *global purification*, unrecovered signal coefficients can be prompted, and isolated non-zero noises can be eliminated.

Our method is distinguished from other model based algorithms including LaMP [15] and CluSS [16] in two major aspects. First of all, unlike LaMP [15], our method takes advantage of the connectivity of the non-zero pixels in the difference frame, and there are few parameters which need to be tuned. In this sense, our method is more robust than LaMP [15] and will not suffer from selecting an inaccurate model. Secondly, since our algorithm is an ℓ_1 based method, the convergence of the algorithm is guaranteed. As one can see from the experiment results, our method provides more stable reconstruction results than CluSS [16], which is based on MCMC sampling.

The remainder of this chapter is organized as follows: our architecture is described in Sec. 3.2. Sec. 3.3 shows the details of the proposed algorithm. Experimental results are illustrated in Sec. 3.4, and Sec. 3.5 concludes this chapter.

3.2 System Architecture

3.2.1 Group of Pictures and System Diagram

We adopt a similar setting as [13], where the structure of a Group of Pictures (GoPs) is shown in Fig. 3.1. Let $\underline{x}_{bj}$ be the reference frame in the j^{th} GoPs in a video. $\underline{x}_{bj}$ is followed by G non-reference frames, denoted as $\underline{x}_{tj}^1, \underline{x}_{tj}^2, \ldots, \underline{x}_{tj}^G$. The difference frame between the i^{th} non-reference frame in the j^{th} group of pictures and its reference frame is calculated as a pixel-by-pixel algebraic difference:

$$\underline{d}_j^i = \underline{x}_{tj}^i - \underline{x}_{bj}, \tag{3.1}$$

for $i = 1, 2, \ldots, G$.

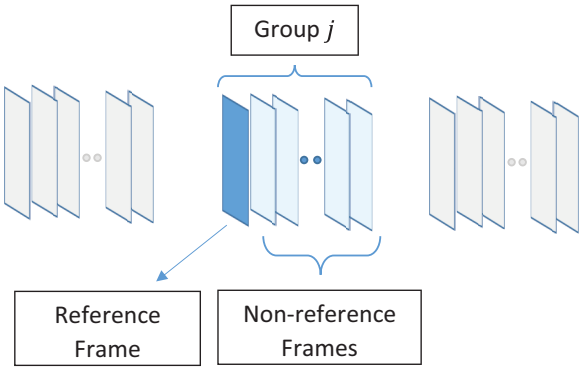


Figure 3.1: Reference frame and non-reference frame in a Group of Pictures.

Next, the difference frame is then hard thresholded and is calculated as:

$$D_j^i(n) = \begin{cases} d_j^i(n) & \text{if } d_j^i(n) \geq \tau; \\ 0 & \text{else,} \end{cases} \tag{3.2}$$

for $n = 1, 2, \ldots, N$, where τ is a threshold value. The thresholded difference frame $\underline{D}_j^i$ is then sampled using compressive sensing, *e.g.,*

$$\underline{V}_j^i = \mathbf{A} * \underline{D}_j^i. \tag{3.3}$$

In the reconstruction phase, sparsity promoting algorithm is used to recover the under-sampled difference frame. Eventually, adding non-reference frame back to the reconstruction leads to an estimate of the non-reference frame, and the above process is illustrated in Fig. 3.2.

It is known that when the number of compressed samples, M, is above the weak threshold $O\left(Klog(N/K)\right)$, the reconstruction by ℓ_1 minimization is generally very accurate. However, compressive sampling reconstruction degrades a lot when M is below the weak threshold. Now, we are interested in answering the following question: if the number of compressive sensing samples (M) is less than $O\left(Klog(N/K)\right)$, given the full knowledge of the reference frame and the compressive sampling measurements

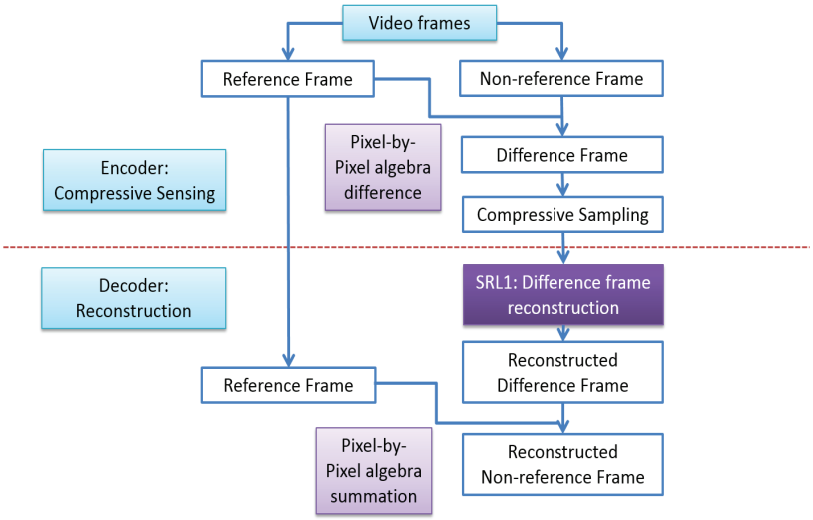


Figure 3.2: Diagram of video frame sampling and reconstruction.

of difference frames, could we reconstruct the video sequence better than the state-of-the-art methods? As we will see in the following sections, by taking advantage of the clustered sparsity, the answer is positive.

3.2.2 Iterative Reweighted L1 Minimization

Compressive sensing involves solving the under-determined system of linear equations. Without other prior knowledge, the ℓ_0 minimization (3.4),

$$\begin{aligned} \widehat{\underline{x}} &= argmin \left\| \underline{x} \right\|_0, \\ s.t.\ \ \underline{y} &= \mathbf{A} * \underline{x}, \end{aligned} \tag{3.4}$$

finds the optimal solution, as $\left\| \underline{x} \right\|_0$ represents the number of non-zero coefficients of $\underline{x}$, and (3.4) directly minimizes the number of the non-zero signal coefficients. However, the ℓ_0 minimization is NP-hard, the ℓ_1 method (2.11), which is a convex relaxation of ℓ_0, is utilized in practice in finding the sparse solution.

It should be noted that in the ℓ_0 norm minimization (3.4), the penalization is uniform regardless of the magnitude of the coefficients. On the other hand, the ℓ_1

minimization method (2.11) penalizes signal coefficients according to their magnitude [57]. This difference explains why ℓ_1 norm minimization is suboptimal in finding the sparsest solution that agrees with the measurements. To fill the gap between ℓ_0 norm method and ℓ_1 norm method, an iterative reweighted ℓ_1 norm minimization, IRWL1, is proposed in [57,58]. The basic idea of IRWL1 is to penalize large signal coefficients with weights smaller than those for small coefficients. This can be summarized as,

$$\begin{aligned} \underline{\widehat{x}} &= argmin \left\| \mathbf{W}^{iter} \underline{x} \right\|_1, \\ s.t.\ \ \underline{y} &= \mathbf{A} * \underline{x}, \end{aligned} \tag{3.5}$$

where $\mathbf{W}^{iter}$ is a diagonal reweighting matrix with entries

$$w_{n,n}^{iter} = \left(\left| x_n^{iter-1} \right| + \epsilon_0 \right)^{-1} \tag{3.6}$$

with ϵ_0 denoting the regularization constant, and $iter \geq 1$ denoting the iteration. In practice, as no prior knowledge of the signal magnitude is known, all of the entries in W^1 are set to 1 in the first iteration. As long as a current reconstruction is obtained, the reweighting matrix $\mathbf{W}^{iter}$ can be updated.

3.3 SRL1: Structured Reweighted L1 Minimization

We propose a novel structured reweighted ℓ_1 optimization technique, called SRL1, to explore the clustered sparsity of the difference frames. We will first show two key components of our technique in the first two subsections and the algorithm is then summarized in the last subsection.

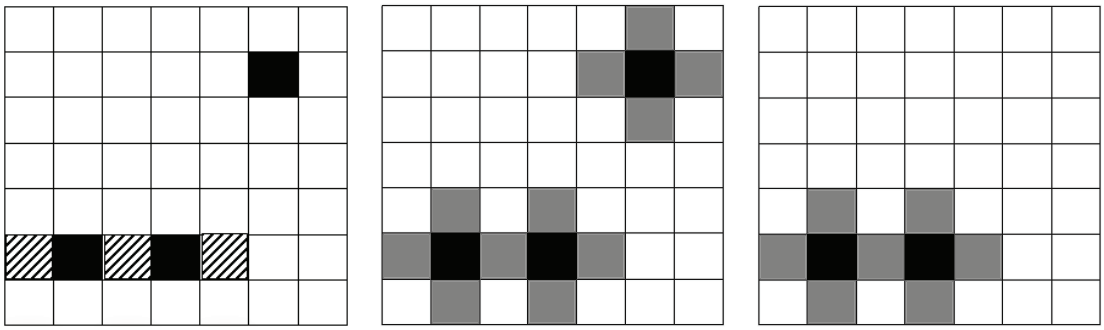
3.3.1 Clustered Sparsity

Definition 1 1 *A cluster is the set of contiguous non-zero pixels and the size of a cluster is defined as the cardinality of the set.*

Definition 2 1 *Two pixels are said to be connected in the sense of dilation by Structural Element (SE) [59] if the clusters dilated by SE from these pixels are contiguous or intersected; similarly, a non-zero pixel is said to be isolated in the sense of dilation by SE if the cluster dilated from this pixel is not contiguous or intersected with other clusters.*

In this work, the shape of the *SE* is a disk. As shown in Fig. 3.3b, the *SE* around the black pixel at the top right corner is shown by gray pixels. The size of the *SE* is 5. Besides, for brevity, *connected* and *isolated* are used as *connected in the sense of dilation by SE* and *isolated in the sense of dilation by SE*, respectively.

Fig. 3.3a is an example where each grid represents a pixel. Three out of $7 * 7$ pixels are non-zero and marked with solid black. As shown in Fig. 3.3b, the clusters (marked with solid gray) dilated from the two black pixels at the bottom left are intersected and thus these two pixels are connected; the non-zero black pixel at the top right is isolated.

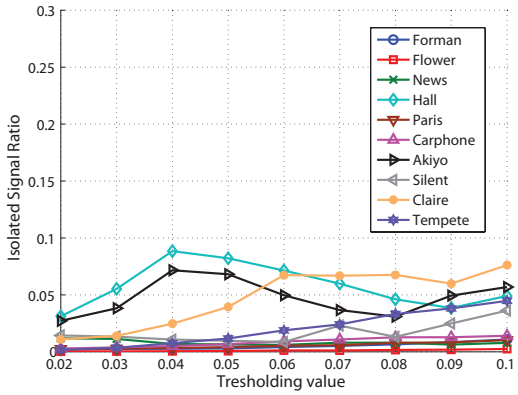


(a) (b) (c)

Figure 3.3: Local exploration and global purification steps.
(a) Initial reconstruction; (b) Local exploration by SE; (c) Global purification has removed the isolated non-zero pixel.

One of the structural characteristics of the difference frame is the clustered sparsity in which non-zero pixels tend to cluster. Here is an example. Fig. 3.4a is the

(a) (b)



(c)

Figure 3.4: Clustered Sparsity of difference frame.

thresholded difference frame between 3^{rd} frame and 1^{st} frame (reference frame) of "Foreman" video sequence while Fig. 3.4b is derived from Fig. 3.4a by removing isolated non-zero pixels. As one can see, most of the non-zero pixels in the thresholded difference frame are clustered, and just a small fraction of the non-zero pixels are isolated. Fig. 3.4c shows the ratio of the number of the isolated non-zero pixels to that of non-zero pixels. 10 video sequences each with 90 frames are tested. Clearly, in all 10 videos tested, most of the non-zero pixels are not isolated. Specifically, 7 out of the 10 videos have isolated pixel ratios smaller than 5%. Video sequences including "Hall", "Akiyo" and "Clarie" have isolated pixel ratios between 5% to 10%.

As discussed in the previous section, when the number of compressed samples (M) is less then $O\left(Klog(N/K)\right)$, the reconstruction is inaccurate. In the context

of difference frame reconstruction, that is to say, some non-zero pixels may not be recovered and some zero pixels may be reconstructed as non-zeros. The clustered sparsity of the difference frame gives rise to two heuristics that can be used to analyze and enhance the reconstructed difference frame $\underline{\widehat{D}}_j^i$ for videos:

1. If a pixel $\widehat{D}_j^i(n)$ is zero but is connected (in the sense of dilation) to other non-zero pixels, there is a high probability that this pixel is actually non-zero rather than zero;

2. If a pixel $\widehat{D}_j^i(n)$ is non-zero and is isolated (in the sense of dilation), there is a high probability that this pixel is actually zero rather than non-zero.

In the following subsections, we will see these two heuristics give rise to two key steps, Local Exploration and Global Purification, in signal support estimation. With these two steps, even when the number of compressed samples is below the weak threshold, difference frame reconstruction could be improved where unrecovered non-zero pixels can be prompted and non-zero errors can be eliminated.

3.3.2 Signal Support of Difference Frame and Weights Allocation

Signal support estimation is of great importance in the compressive sampling reconstruction step. Our signal support estimation starts from the initial reconstruction $\underline{\widehat{D}}_j^i$ and is followed by two steps, *local exploration* and *global purification*.

The local exploration stage is inspired by the first heuristic and is to find the unrecovered signal pixels. In the Local Exploration stage, we first make $\underline{\widehat{D}}_j^i$ a binary frame, and the pixel of the resulting binary frame is expressed as:

$$B_j^i(n) = \begin{cases} 1 & \text{if } \widehat{D}_j^i(n) > \tau; \\ 0 & \text{else,} \end{cases} \tag{3.7}$$

This give rise to the binary frame $\underline{B}_j^i$ and is illustrated in Fig. 3.3a. In Fig. 3.3a,

two black pixels at the bottom left represent recovered non-zero pixels. The black pixel at the top right represents a non-zero error. The three shaded pixels at the bottom left represent unrecovered non-zero pixels. Then, each non-zero pixel in $\underline{B}_j^i$ (see Fig. 3.3a) serves as an *anchor* and is morphologically dilated by the *Structuring Element* (*SE*) [59], and the dilated frame is denoted as $\underline{L}_j^i$. As a result, after the Local Exploration stage, (see Fig. 3.3b), each non-zero pixel (*anchor*) is dilated to a cluster.

The global purification stage is inspired by the second heuristic and is to eliminate non-zero errors. In the global purification stage, depending on the size, certain clusters and their corresponding *anchors* will be deleted from $\underline{L}_j^i$. There are two cases for the size of the cluster. If two *anchors* in $\underline{B}_j^i$ are connected (in the sense of dilation by *SE*), the clusters dilated from these two *anchors* will intersect and form a larger cluster. As a result, these two connected *anchors* will locate in the same larger cluster with size greater than the size of *SE*. On the other hand, if a *anchor* in $\underline{B}_j^i$ is isolated, the size of the cluster dilated from this *anchor* will be equal to the size of *SE*. To eliminate non-zero errors, the clusters with size smaller than a predefined threshold[1], N_{conn}, are deleted from $\underline{L}_j^i$ and the resulting binary object (see Fig. 3.3c), is our estimation of the signal support of $\underline{D}_j^i$, denoted as $\underline{S}_j^i$.

Then the weights vector is calculated based on the signal support $\underline{S}_j^i$ and (3.6) is restated as:

$$w^{iter}(n,n) = \begin{cases} 1/(w_1+\epsilon) & \text{if } S_j^i(n)=1; \\ 1/(w_0+\epsilon) & \text{if } S_j^i(n)=0, \end{cases} \tag{3.8}$$

where w_1, w_0 and ϵ are set to 1, 0 and 0.1 respectively in our tests, and $w^{iter}(n,n)$ is the $n \times n$ element of the re-weighting matrix W^{iter}.

[1]The threshold value is calculated based on the size of *SE* and is set to 6.

3.3.3 The Proposed Algorithm Summary

The proposed structured reweighted algorithm, SRL1, is summarized in Algorithm 1. In Step 1, an initial reconstruction of difference frame can be obtained through a variety of solvers, for example, SPGL1 [60,61]. This initial reconstruction is served as side information for further refinement and will be updated in each iteration. In Step 2, the reconstructed difference frame is then thresholded and converted to a binary frame based on (3.7). Local Exploration and Global Purification are implemented in Step 3 and Step 4. Signal support and weights vector are then updated in Step 5 and Step 6 correspondingly.

Algorithm 1: SRL1-Structured Reweighted L1 Minimization.

Input: $\underline{V}_j^i, \mathbf{A}, \underline{x}_{bj}$
Output: $\widehat{\underline{x}}_{tj}^i$
Algorithm:
Initialize: $W^1 = I, iter = 1$;
while $iter \leq ITER$ **do**
Step 1: Solve (3.5):

$$\widehat{\underline{D}}_j^i = argmin \left\| W^{iter} \underline{D}_j^i \right\|_1, s.t.\ \underline{V}_j^i = \mathbf{A} * \underline{D}_j^i$$

Step 2: Convert $\underline{D}_j^i$ to binary frame $\underline{B}_j^i$ as in (3.7);
Step 3: Perform local exploration using SE;
Step 4: Perform global purification by removing clusters with size small small than N_{conn};
Step 5: Estimate signal support: $\underline{S}_j^i$;
Step 6: Update Weight using (3.8), $iter = iter + 1$;
end while
Return: $\underline{x}_{tj}^i = \underline{x}_{bj} + \widehat{D}_j^i$

3.4 Simulation Results

Experiments are taken to show the effectiveness of the proposed algorithm. In all of the experiments, each frame is gray scale with size $128 * 128$ pixels, and the value of each pixel has been scaled to [0,1]. As in [62] and [63], to relieve the computational

burden at the video encoder which is resource-constraint, instead of using dense matrices, we use sparse random $\mathbf{A}$ with row weight 16. It should be noted that non-zero elements of $\mathbf{A}$ are drawn from normalized Gaussian distribution and are uniformly distributed across the columns.

In the first experiment, we test the number of iterations *iter* on reconstruction quality. The 1^{st} (reference frame) and 3^{rd} frames of "Foreman" are picked. The difference frame is calculated and then sampled at 40% ($M/N = 0.4$). Then, reconstructions by our proposed algorithm SRL1 and IRWL1 [58] are compared and the PSNRs of the reconstructed 3^{rd} frame are shown in Fig. 3.5. Clearly, our proposed technique outperforms IRWL1 considerably. Compared to IRWL1, SRL1 increases the PSNR of the reconstructed video frame by 3.01 dB in the second iteration and the gain is 3.85 dB after five iterations. Similar results are also observed on other pairs of frames.

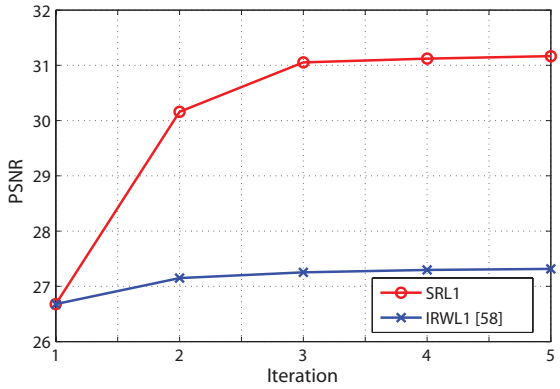
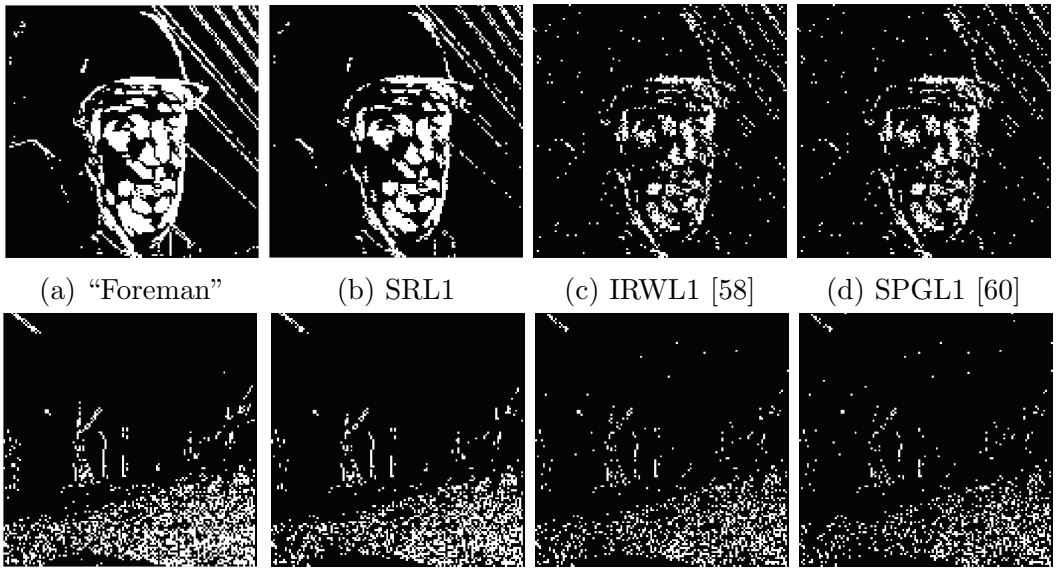


Figure 3.5: PSNR comparison between SRL1 and IRWL1 as a function of reconstruction iterations for the 3^{rd} frame of Foreman video.

In the second experiment, 1^{st} and 3^{rd} frames of "Foreman" and 1^{st} and 2^{nd} frames of "Flower" are selected. The 1^{st} frame is set as the reference frame and the difference frame is calculated as (3.1). The sampling rate (M/N) is set as 40% for both of these two tests. ITER is set 5. Threshold τ is set 0.08. The comparision of the performance of different reconstruction techniques applied for "Foreman" and

"Flower" are shown in Fig. 3.6. The reconstructed difference frames have been converted to binary frames with threshold value 0.08 for illustration purpose here. The PSNRs of the reconstructed non-reference frame for "Foreman" are 31.35 dB (SRL1), 27.31 dB (IRWL1) and 26.84 dB (SPGL1). The PSNRs of the reconstructed non-reference frame for "Flower" are 31.32 dB (SRL1), 29.22 dB (IRWL1) and 28.28 dB (SPGL1). Comparing these results, we can see that SRL1 can eliminate non-zero errors. Moreover, unrecovered non-zero pixels could be prompted.



(a) "Foreman" (b) SRL1 (c) IRWL1 [58] (d) SPGL1 [60]

(e) "Flower" (f) SRL1 (g) IRWL1 [58] (h) SPGL1 [60]

Figure 3.6: Demonstration of difference frame reconstruction using different techniques.

In the last experiment, more frames from "Flower" are tested. The size of GoP is set 5 and maximum iteration (ITER) is set 5. The sampling rate (M/N) is set based on the sparsity (K/N) of each difference frame and is set below the weak threshold. Specifically, the sampling rate (M/N) for the $1^{st}, 2^{nd}, 3^{rd}, 4^{th}$ non-reference frame in each GoP is set $40\%, 55\%, 60\%, 65\%$ for "Flower"[2]. Threshold τ is set 0.08 for each 1^{st} non-reference frame and 0.09 for $2^{nd}, 3^{rd}, 4^{th}$ non-reference frame. The PSNR of 15 reconstructed non-reference frames in the video sequence using our proposed method,

[2]The sparsity (K/N) increases as the frame distance becomes larger. As a results, we gradually increase the sampling rates (M/N) within each GoP.

SRL1, and Block-CoSaMP [64], CluSS [16], IRWL1 [58], SPGL1 [60] are shown in Fig. 3.7. It can be seen that even though the sampling rate is set below the weak threshold, taking advantage of the clustered sparsity, SRL1 still gives decent reconstruction and outperforms other schemes.

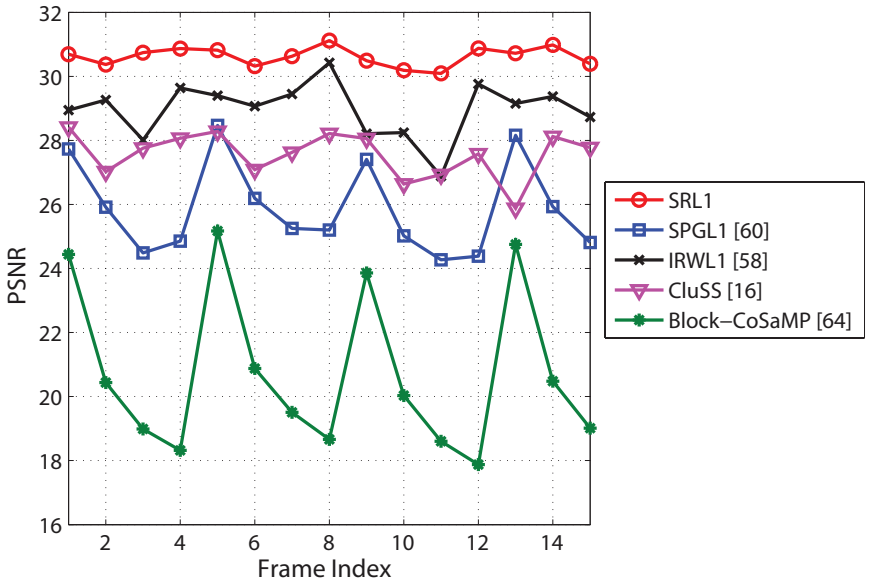


Figure 3.7: Comparison of SRL1 with other techniques.

3.5 Conclusion

In this chapter, a novel structured reweighted ℓ_1 minimization algorithm, referred to as SRL1, is proposed to reconstruct difference frames in the video sequences. It is shown that by exploiting the clustered non-zero coefficients, isolated non-zero noises could be eliminated and unrecovered signal coefficients could be prompted. We showed that SRL1 can reconstruct the difference frame much better than many other state-of-the-art algorithms.

CHAPTER 4

Binary Compressive Sensing via Sum of ℓ_1-norm and ℓ_∞ norm Regularization

In this chapter, we study the compressive sensing tasks for binary sparse signals. A novel convex optimization technique is proposed, where ℓ_∞-norm is combined with ℓ_1-norm to regularize the optimization process. Numerical results confirm the proposed technique is capable of promoting both sparsity and the binary feature of the signals.

4.1 Introduction

Conventional compressive sensing tasks involve finding correct solutions for under-determined systems of linear equations, where the target signals are sparse. In this chapter, we are interested in solving a special case of the problem, in which the target signal is binary sparse. Signal of this type is prevalent in engineering fields, including control engineering, aerospace engineering, and environment monitoring. One running example can be found in fault identification, where the fault pattern is represented by a binary vector $\underline{x} \in \{0,1\}^N$, with "1" indicating a fault has happened, and "0" indicating a normal behavior. Given a set of measurements, the task is to locate the set of faults pattern if there are any errors/events occurred in a system.

The unique binary feature of the signal makes the reconstruction more challenging than conventional compressive sensing tasks [17]. One line of research lies in adding heuristics to promote binary sparse features. For example, the binary prior is explored by the unique sum property in [18]. In [19], a density mixture model with peaks centering at "0" and "1" is served as the prior distribution of the signal, and the resulted

algorithm, NBP, achieves the state-of-the-art and limit-approaching performance under large noise conditions. In [20], a method based on the convex relaxation of the Boolean constraint is proposed to promote the binary signal reconstruction quality.

Technique proposed in [17] represents another line of research. Concretely, to avoid the challenge of binary compressive sensing, a pre-processing stage is utilized to map the non-zero entries of binary signal, *i.e.,* "1"s, to varying magnitudes. After this pre-mapping process, the binary reconstruction problem turns to a general compressive sensing problem, which after reconstruction, the signal is transformed back to binary by applying the inverse transform.

In this chapter, we are aiming to solve the binary compressive sensing task by designing a novel regularization strategy. Specifically, the convex optimization regularized by ℓ_1-norm (2.11) promotes the sparsity of the solution. On the other hand, regularization by ℓ_∞-norm favors the representation whose coefficients are roughly in the same absolute magnitude [21–23]. In this chapter, we show that these two regularization can be combined in binary compressive sensing problem to promote the reconstruction quality. This is achieved by minimizing the weighted sum of the ℓ_1-norm and ℓ_∞-norm, up to a shifting vector. Besides, the proposed new formulation is convex[1], and can therefore be solved effectively by general linear programming solvers. We will see that although NBP [19] exhibits limit approaching property in large noise, our method gives better reconstruction under small noise. Moreover, our technique turns out to be more robust when an inaccurate signal model is selected.

It should be noted that sparse regression regularized by mixed norms has been reported in several different scenarios [65–68] where the signal vector is divided into several groups according to the specific features. Then different norms are applied to two levels (individual feature level and group feature level) disjointly. The within group feature is explored by regularization such as ℓ_2-norm and ℓ_∞-norm. To promote

[1]The new formulation is convex due to the fact that both ℓ_1-norm [57] and ℓ_∞-norm [21] are convex.

the sparsity across the groups, the group norms are then summed together in the form of ℓ_1-norm. For example, [66] analyzed the multi-layered expansion problem using $\ell_{1,2}$-mixed norms. In [65], lasso (ℓ_1-norm) and group lasso ($\ell_{1,2}$-norm) are merged and the penalty function is able to induce a solution which is sparse at both individual and group feature level. In [68], within group correlation is explored by ℓ_∞-norm and sparsity is explored by ℓ_1-norm.

Our technique is distinguished from other schemes mainly in two aspects. First of all, to the best of our knowledge, we are the first to bring *sparse representation* [69–71] and *democratic representation*[2] [21–23] together to solve binary compressive sensing problem. Secondly, unlike [65–68], both ℓ_1-norm and ℓ_∞-norm are applied to the same level, *i.e.*, the whole signal vector. Thus, ℓ_1-norm minimizer and ℓ_∞-norm minimizer can work jointly to promote the binary sparsity in our problem.

The remainder of this paper is organized as follows: Our formulation of the problem is detailed in Sec. 4.2. Numerical results along with comparison with other state-of-the-art methods are illustrated in Sec. 4.3. Sec. 4.4 concludes this chapter.

4.2 Regularization by Sum-of-Norms

In this section, we will first define the binary compressive sensing problem. Next, the unit ball of ℓ_1-norm and ℓ_∞-norm is studied, and our formulation is developed.

4.2.1 Binary Compressive Sensing Problem

We follow the notations in Chapter 2, and denote the length of the signal as N. For fault identification, this indicates the total number of possible faults in the system is N. Let a binary vector $\underline{x} \in \{0,1\}^N$ be the fault pattern where $x_i = 1$ shows the fault has occurred at $i \in \{1,\ldots,N\}$. Assume faults, i.e., the "1" s, are identical and independent distributed, and let K be the total number of faults in the system. The

[2]By democratic, we mean the coefficients of the signal have roughly the same absolute magnitude.

measurement vector $\underline{y} \in \mathbb{R}^M$ is obtained by projecting $\underline{x} \in \{0,1\}^N$ with a $M \times N$ random sampling matrix, *a.k.a., fault signature matrix*, $\mathbf{A}$, and (2.9) can be restated as,

$$\underline{y} = \mathbf{A} * \underline{x} + \underline{e}, \tag{4.1}$$

where $\underline{e} \in \mathbb{R}^M$ represents a noise vector, with each component e_i identical and independent distributed following $\mathcal{N}(0, \sigma^2)$. To estimate the fault pattern, one need to solve the ill-posed problem.

4.2.2 Two extremes: ℓ_1-norm and ℓ_∞-norm minimizers

The ℓ_∞-norm solution for a linear system involves solving the following:

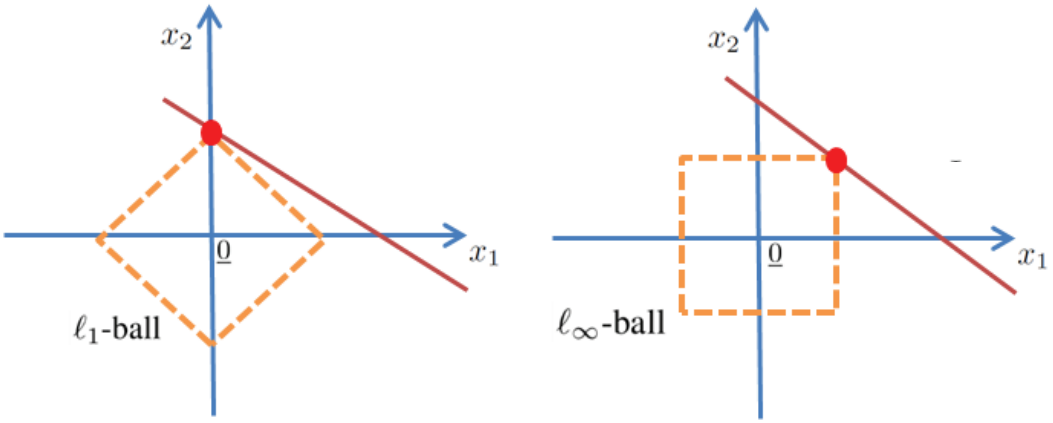
$$\hat{\underline{x}} = argmin \left\| \underline{x} \right\|_\infty, \tag{4.2}$$

$$s.t.\ \underline{y} = \mathbf{A} * \underline{x}, \tag{4.3}$$

where $\left\| \underline{x} \right\|_\infty = \max_i |x_i|$.

ℓ_1-norm and ℓ_∞-norm minimizers are two extremes in finding a solution for a linear system. This is because ℓ_1-norm seeks a solution as sparse as possible, and thus it is widely used in compressive sensing (or sparse signal representation) [69–71]. On the other hand, ℓ_∞-norm minimizer favors a dense solution and finds its application in democratic (or spread) signal representation [21–23, 72].

Fig. 4.1 is a two dimensional illustration of the comparison of ℓ_1-norm and ℓ_∞-norm minimizers in finding a solution for the linear system $\underline{y} = \mathbf{A} * \underline{x}$ with $N = 2$ and $M = 1$. As one can see in Fig. 4.1a, the ℓ_1-norm minimization finds the sparse solution, due to the shape of the ℓ_1 ball. On the contrary, we can see in Fig. 4.1b that the solution found by the ℓ_∞-norm minimizer is not sparse. Moreover, the absolute value of the magnitude of the two coefficients are the same ($|x_1| = |x_2|$).



(a) (b)

Figure 4.1: Two dimensional illustration of ℓ_1 norm and ℓ_∞ norm minimizers for the linear system. The solution in each case is marked with solid red.
(a) ℓ_1 norm minimizer finds the sparsest solution; (b) ℓ_∞ norm minimizer finds the solution with equal magnitude;

It should be noted that Fig. 4.1b is an exemplary illustration of the ℓ_∞-norm minimizer in solving a linear system. In practice, the coefficients of the optimum may not share exactly the same absolute magnitudes as stated the following proposition from [23].

Proposition 1 1 *Denote the optimal solution for a linear system yielded by ℓ_∞-norm minimizer as $\underline{x}^*$. Then $n - m + 1$ out of n signal coefficients of $\underline{x}^*$ have magnitude equal to $\pm\|\underline{x}^*\|_\infty$ and the remaining $m-1$ coefficients of $\underline{x}^*$ have magnitudes between $-\|\underline{x}^*\|_\infty$ and $\|\underline{x}^*\|_\infty$.*

Based on what so-called uncertainty principle [72], authors in [21] further analyzed the condition under which ℓ_∞-norm minimizer gives democratic representation with high probability and one may refer to [21] for more details of the property of ℓ_∞-norm minimizer.

4.2.3 Sum-of-Norms Regularization

As discussed above, ℓ_1-norm and ℓ_∞-norm minimizers are two extremes in finding a solution for a linear system. It seems that they may not have any connection in

solving the binary compressive sensing problem. However, we showed that they can be combined to promote the underlying binary sparsity. This is done by summing ℓ_1-norm and ℓ_∞-norm together, up to a scaling factor λ and a shifting factor vector $\underline{c}$. The new formulation is then expressed as follows:

$$\begin{aligned} \hat{\underline{x}} = & argmin(\|\underline{x}\|_1 + \lambda * \|\underline{x} - \underline{c}\|_\infty), \\ & s.t. \quad \|\underline{y} - \mathbf{A} * \underline{x}\|_2 \leq \|\underline{e}\|_2, \\ & and \quad 0 \leq x_n \leq 1 \quad for\ 1 \leq n \leq N \end{aligned} \tag{4.4}$$

where λ is a scalar and $\underline{c}$ represents the shifting vector which is of the same length with the signal vector $\underline{x}$. Since both ℓ_1-norm and ℓ_∞-norm are convex, the summation of these two norms is convex as well. Thus one can employ general linear programming operators, such as CVX [73] to solve (4.4).

4.2.4 Discussion on the Parameters

As one can see, unlike the classic formulation, in our novel formulation (4.4), the binary compressive sensing problem is regularized by two penalties: ℓ_1-norm and ℓ_∞-norm. Just like the classic formulation, the ℓ_1-norm term is aimed to promote the sparsity. The ℓ_∞-norm, aside with the shifting vector $\underline{c}$, on the other hand, are employed to exploit the binary property in the problem. Consequently, the balance of these two terms is controlled by the scaling factor λ. We give the following proposition regarding how to set up the shifting vector $\underline{c}$ directly without theoretical proof.

Proposition 2 1 *Provided the binary prior on the fault pattern ($\underline{x} \in \{0,1\}^N$), all the coefficients of the shifting vector $\underline{c}$ in* (4.4) *should be set to a same magnitude, which is* $1/2$ *in our case.*

To see why the proposition makes sense, one can see Fig. 4.2. It is known that the non-zero coefficients of $\underline{x}$ are located at 1. That is also to say, the coefficients of $\underline{x}$

are equally separated w. r. t. 0.5. We recall that the ℓ_∞-norm minimizer is able to spread the magnitudes evenly across all the coefficients. Therefore, by setting the shifting vector to 0.5, the two penalty terms in (4.4) can work constructively.

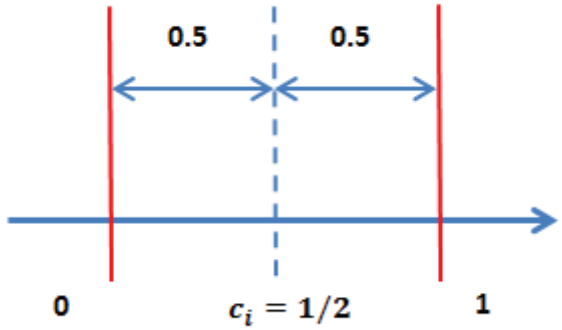


Figure 4.2: Illustration of the Formulation

To summarize, the combined effect of the two norms in (4.4) is two fold. First of all, the sparsity of the fault pattern is promoted by the ℓ_1-norm term and thus some of the coefficients are deviated from 0. The binary property is then explored by the ℓ_∞-norm term which encourages those non-zero coefficients to be centered at 1.

4.2.5 Numerical Analysis

We will evaluate the goodness of our choice of scaling factor λ and shifting vector $\underline{c}$ by feeding our method with different combinations of these two parameters. Specifically, N and M are set to 200 and 60. Besides, the sparsity K/N are set to 0.09 and the variance of the noise is set to 0.1. We use random Gaussian matrix with normalized columns as our sampling matrix, and the error metric is mean ℓ_2-norm reconstruction error (MLRE), which is calculated as $\sqrt{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}/N$. The formulation (4.4) is then solved by CVX package [73]. Fig. 4.3 shows the results averaged on 50 independent trials. It can be seen that the global minimum happens when $\underline{c} = 0.5$ and $\lambda = 100$.

Then, we further test our scheme with different combinations of scaling factor λ

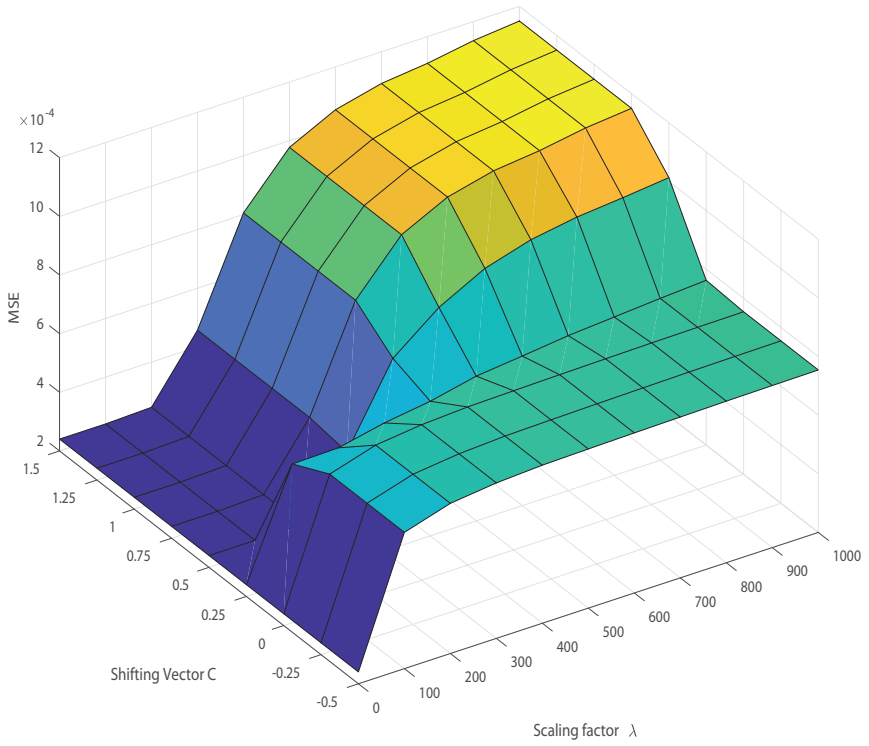


Figure 4.3: Performance on different combinations of scaling factor λ and shifting vector c.

and signal sparsity K/N while the shifting vector $\underline{c}$ is fixed at 0.5. We follow the previous setting and thus N is fixed at 200 and M is set to 60. The variance of the noise is set to 0.1. Then 50 Monte Carlo trials are executed and the results are summarized in Fig. 4.4. We can see from Fig. 4.4 that for a fixed sparsity K/N, the reconstruction error (MLRE) decreases as scaling factor λ increases from 0 ($\lambda = 0$ corresponds to solely ℓ_1 norm minimization) to 100. Besides, it can be seen that the MLRE turns out to be relatively stable when λ is set in the range $[100, 200]$ and the reconstruction deteriorates with further increment of λ ($\lambda = \infty$ corresponds to solely ℓ_∞ norm minimization).

This confirms that by proper combining ℓ_1-norm and ℓ_∞-norm, our scheme is able to do better than generic ℓ_1-norm minimization and ℓ_∞-norm minimization. Moreover, another favorable feature is that λ in the range $[100, 200]$ yields satisfactory

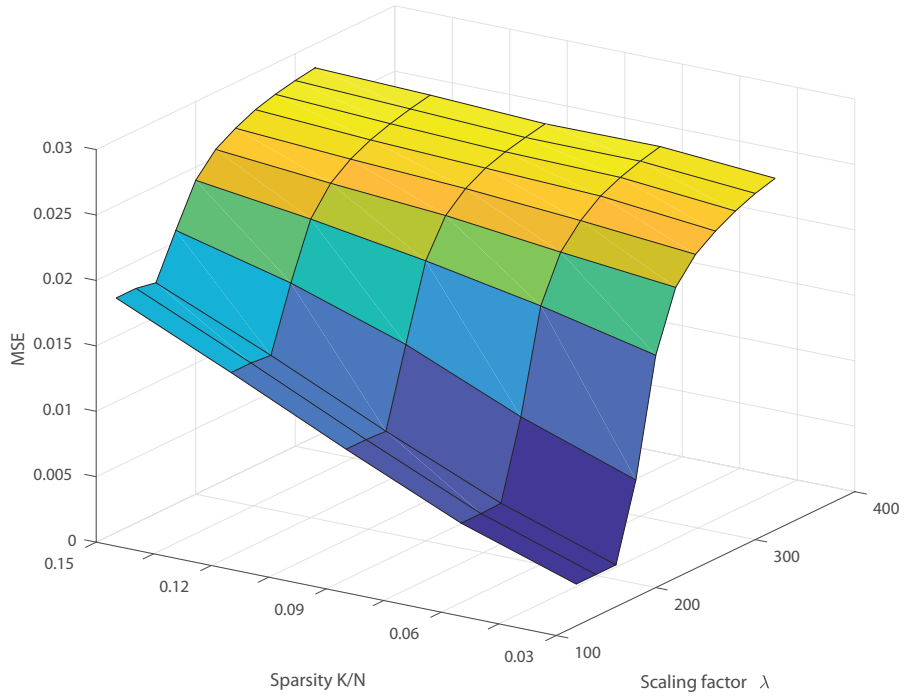


Figure 4.4: Performance on different combinations of scaling factor λ and signal sparsity K/N. c is set to 0.5.

reconstruction for all sparsity values K/N. Based on these results, we will set $\underline{c}$ to 0.5 and λ to 100 in later comparisons[3].

4.3 Simulation Results

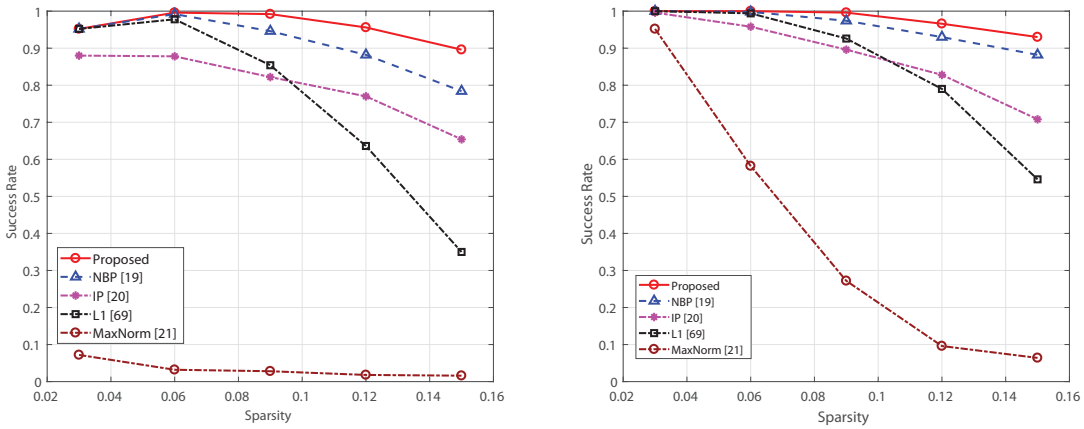
In this section, we will compare our method with several sophisticated algorithms in solving fault identification problem. Specifically, we compare our method with NBP [19], IP [20], classic ℓ_1-norm [69–71] and ℓ_∞-norm technique [21–23]. Since we are working on noisy measurements, the method in [18] will not be compared here.

To obtain a binary solution, we employ two useful packages: Variable threshold rounding and Local optimization search from [20] as the post-process stage for all the algorithms compared. We use success rate as our metric to measure the goodness of

[3]We also notice that $\lambda = 100$ turns out to be a good choice for larger signal length, such as $N = 200, 500, 1000$.

the reconstruction where a successful trial is defined as the one of which the reconstruction is exact ($\hat{\underline{x}} = \underline{x}$). Besides, as in [19], sparse sampling matrix with non-zero elements drawn independently from $\{-1, 1\}$ is employed as the sampling matrix for all the schemes. We set the percentage of the non-zero elements, i.e., the sparsity of $\mathbf{A}$, to 0.2, which is the same value as in [19]. Also, $\underline{c}$ is set to 0.5 and λ is set at 100.

The first simulation is set up to evaluate the effect of the sparsity (K/N) on reconstruction quality. In this test, N is set to 100 and M is set to 40. Noise variance σ is set to 0.1. The results averaged on 500 Monte Carlo trials are summarized in Fig. 4.5. It can be seen that, by taking advantage of the binary prior of



(a) Without Local Optimization (b) With Local Optimization

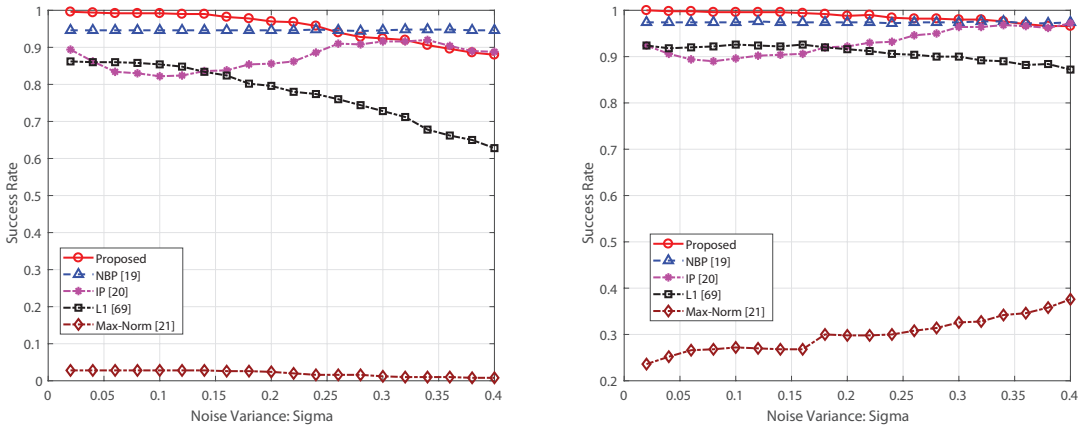
Figure 4.5: Comparing Several Schemes with varying Sparsity
(a) Without Local Optimization (b) With Local Optimization;

the fault pattern vector, our technique outperforms generic method which is based on ℓ_1-norm minimization. Also, our technique gives better fault pattern estimation quality than ℓ_∞-norm minimization technique where the sparsity prior has not been explored. Moreover, our scheme do better than the two sophisticated algorithms for fault identification problem: NBP [19] and IP [20].

It should be noted that in Fig. 4.5 the quality of Max-norm (ℓ_∞-norm) [21] is not very good. This is because Max-norm technique requires the sampling rate (M/N) to be at least 0.5 to get decent quality [22]. Also, it can be seen that the reconstruction

when sparsity K/N equals to 0.03 is worse than those when sparsity K/N equals to 0.06. This is because IP [20] requires the sparsity K/N to be known to operate properly[4], and the performance degrades when K/N is very small.

Next, we test the performance of our method with varying noise variance σ. As in the first simulation, N and M are set to 100 and 40 respectively. K/N is set to 0.09. Fig. 4.6 summarizes the results averaged on 500 trials. It can be seen that our technique outperforms NBP [19], which represents the state-of-the-art, under small noise condition. Yet we also notice that NBP [19] yields better quality when noise variance σ is larger than 0.25.



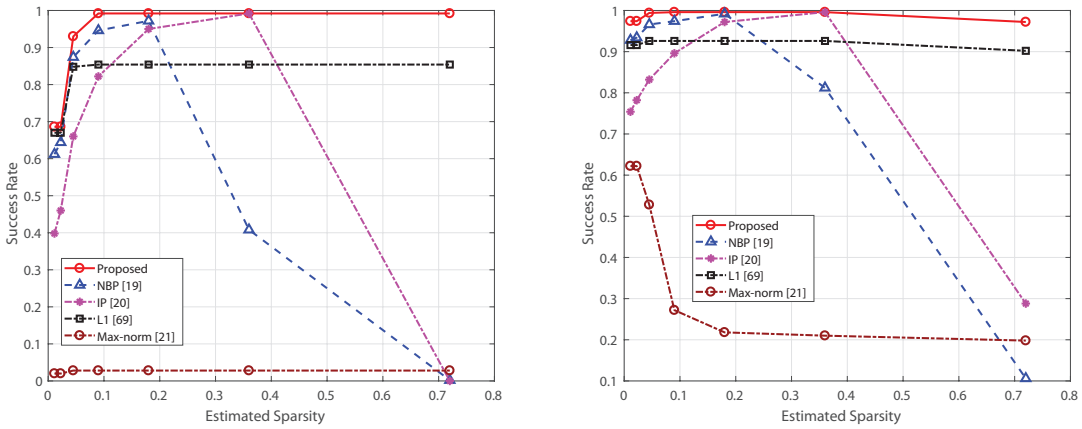
(a) Without Local Optimization (b) With Local Optimization

Figure 4.6: Comparing Several Scheme with varying Noise Variance
(a) Without Local Optimization (b) With Local Optimization;

In the last simulation, we are interested in analyzing the robustness of our method. Specifically, we test the sensitivity to the sparsity K/N on different reconstruction schemes. To do so, we fix K/N to 0.09 in generating the signal and then feed these schemes with several different sparsity values at the reconstruction stage. It is expected that our scheme could provide better results since it does not need the sparsity value to reconstruct the fault pattern. Following the setting in previous simulations, N and M are set to 100 and 40, respectively. The noise variance is set to 0.1. Fig. 4.7

[4]To get the fault pattern, sparsity K/N is required in [20] to calculate the posterior probability and loss function.

shows the results averaged on 500 Monte Carlo runs. Clearly, our technique is more robust and provides more stable reconstruction quality when an inaccurate estimation of the sparsity K/N is selected. Also, NBP [19] and IP [20] are more vulnerable to model mismatch and thus call for more efforts in tuning the algorithms.



(a) Without Local Optimization (b) With Local Optimization

Figure 4.7: Sensitivity on Sparsity
(a) Without Local Optimization (b) With Local Optimization;

4.4 Conclusion

In this chapter, we presented a novel formulation for the binary compressive sensing problem. The binary sparsity is exploited by the sum of two norms: ℓ_1-norm and ℓ_∞-norm. When applied in fault identification problem, our method is able to give decent results and outperforms other techniques especially under small noise. Besides, our scheme turns out to be more robust when an inaccurate signal model, i.e., the sparsity level, is selected.

CHAPTER 5

A Framework for Compressive Sensing of Asymmetric Signals using Normal and Skew-Normal Mixture Prior

In this chapter, we study the compressive sensing of sparse signals whose significant coefficients are distributed asymmetrically with respect to zero. To properly capture the asymmetrical features, a framework utilizing a two-state normal and skew normal mixture density as the prior distribution of the signal is developed. An efficient approximate message passing based algorithm is designed to estimate the signal. Experimental results on both synthetic data and real-world data, *i.e.,* weather sensor network, confirm the developed method is powerful in exploiting the asymmetrical feature.

5.1 Introduction

In this chapter, we are aiming to solve the compressive sensing task of asymmetrical signals. Signal of this type can be found in a number of engineering fields. One example can be found in biomedical research [29], where the gene expressions involved in embryo are more often developed with an increasing trend. Another example can be found in sensor networks, where certain type of weather data, let us say outside air temperature, when subtracted from the historical average, is asymmetrically positive or negative when the disrupting weather phenomena is heat or cool, respectively.

The sparsity promoting capability of ℓ_1-norm minimization lies in the heart of compressive sensing. While being robust and working decently in exploiting the sparse feature, optimization-based techniques in general lack the flexibility in accom-

modating other salient features of the signals. Aside from convex optimization based techniques, Bayesian inference based methods provides another effective perspective to reconstruct the signal from under sampled measurements. In a typical Bayesian inference setup, the prior knowledge of signal is modelled by a *prior distribution*, and the measurement process is represented by the *likelihood function*. The reconstruction is obtained by estimating the *posterior* distribution of the signal.

In this chapter, we adopt a Bayesian methodology. Concretely, to properly address the asymmetrical features, we develop a framework utilizing a two-state normal and skew normal mixture density as the prior distribution of the signal. The significant and insignificant coefficients of the signal are represented by skew normal and normal distributions, respectively. A novel approximate message passing based algorithm is developed to estimate the signal from its compressed measurements. A fast gradient-based estimator is designed to infer the density of each state. Experiment results on simulated data and real-world tests, *i.e.,* weather sensor network, confirm that our proposed technique is powerful in exploiting asymmetrical feature, and outperforms many sophisticated methods.

The remainder of this chapter is organized as follows. The signal model and system architecture are introduced in Section 5.2. The approximate message passing algorithm utilizing the two-state normal and skew normal mixture density are detailed in Section 5.3. Gradient-based parameter estimation is detailed in Section 5.4. The complexity of our technique is analyzed in Section 5.5. Experimental results are summarized in Section 5.6, and Section 5.7 concludes this chapter.

5.2 Approximate Message Passing based on Normal and Skew Normal Mixture Density

5.2.1 Skew Normal Density

In this work, we are aiming to estimate sparse signals whose significant coefficients are distributed asymmetrically with respect to zero.

Signals with this asymmetrical feature can be either *right-skewed*, or *left-skewed*. Specifically, for *right-skewed*, the majority of the significant coefficients are of positive sign, with the remaining few being negative. Similarly, for *left-skewed*, the majority of the significant coefficients are of negative sign, with the remaining few being negative.

As discussed, due to the symmetry, neither normal nor Laplace densities could encapsulate the asymmetric nature of signals with such prior information. In this work, we employ a normal and skew normal density mixture as the prior distribution of such signals. More specifically, the distribution of the significant coefficients is modelled by a skew normal density, the probability density function of which was formally defined in [34] as,

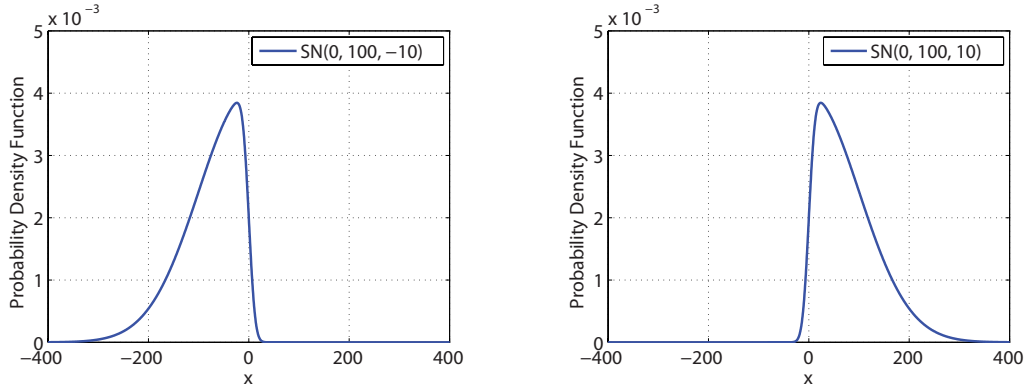
$$\mathcal{SN}(x;\xi,\omega,\alpha) = \frac{2}{\omega}\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right), \tag{5.1}$$

where $\xi, \omega,$ and α represent the location, scale, and shape parameters, $\phi(\cdot)$ and $\Phi(\cdot)$ denote the *pdf* and the cumulative density function (*cdf*) of the standard normal distributed random variable, respectively.

Compared to the normal *pdf*, a noteworthy aspect of (5.1) is the additional term $\Phi\left(\alpha\frac{x-\xi}{\omega}\right)$, which controls the skewness of the density. It is readily seen that (5.1) reduces to a normal density when α is set to 0, and approaches to positive/negative half normal density in the limits $\alpha \to \pm\infty$.

Fig. 5.1a and 5.1b show two curves of the skew normal densities with (ξ, ω, α) being set to $(0, 100, -10)$ and $(0, 100, 10)$, respectively. It can be seen that both of

these two densities are asymmetric with respect to $x = 0$, where the density with negative shape parameter α in Fig. 5.1a is left-skewed, and the density with positive α in Fig. 5.1b is right-skewed. Besides, as compared to the non-negative normal density [31], the skew normal density is more flexible in accommodating both positive and negative elements.



(a) (b)

Figure 5.1: Skew Normal Density. (a) Left-skewed with $\alpha = -10$. (b) Right-skewed with $\alpha = 10$.

Similar to [28], the distribution of the insignificant coefficients is modelled by normal density. Meanwhile, we consider the case where the location parameter $\xi = 0$. Overall, the *pdf* of the signal can be written as,

$$f(x) \cong (1 - \lambda) \times \mathcal{N}(x; 0, \sigma_S^2) + \lambda \times \mathcal{SN}(x; 0, \omega_L, \alpha_L), \tag{5.2}$$

where $\lambda = K/N$ denotes the *sparsity rate*. For convenience, let $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$ be the characterizing parameters set of the mixture.

5.2.2 System Diagram

Our proposed method consists of two functionality modules, with each module being iterative. Fig. 5.2 is the system diagram of our technique.

The first module, as shown in the left of Fig. 5.2, involves the estimation of the signal using the approximation message passing with skew normal and normal

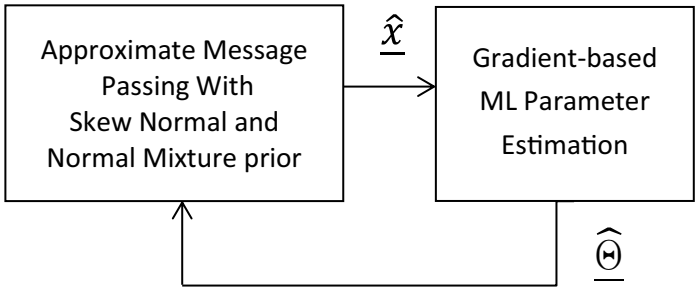


Figure 5.2: System Diagram

mixture density. After the message passing completes, the estimation $\hat{\underline{x}}$ is then fed to the second module, where the parameters of the mixture, $\hat{\underline{\Theta}} = [\hat{\sigma}_S^2, \omega_L^*, \alpha_L^*]$, are inferred. These two modules execute alternatively and repeatedly until convergence is achieved.

5.3 Bayesian Inference by Approximate Message Passing

In this section, we will detail our message passing algorithm utilizing the proposed skew normal and normal mixture density.

Specifically, given (2.19) to (2.24), and recalling the skew normal and normal mixture density (5.2), the message from x_n to y_m at $(i+1)$-th iteration can be written as,

$$\begin{aligned}\nu_{x_n \to y_m}^{i+1}(x_n) &\cong \mathcal{N}(x_n; \kappa_{nm}^i, \varsigma_n^i) f(x_n) \\ &= (1-\lambda)\,\mathcal{N}\Big(x_n; \kappa_{nm}^i, \varsigma_n^i\Big)\mathcal{N}(x_n; 0, \sigma_S^2) \\ &\quad +\lambda\,\mathcal{N}\Big(x_n; \kappa_{nm}^i, \varsigma_n^i\Big)\mathcal{SN}(x_n; 0, \omega_L, \alpha_L), \end{aligned} \tag{5.3}$$

where $\kappa_{nm}^i = \sum_{\substack{u=1 \\ u\neq m}}^{M} A_{un}\mu_{y_{un}}^i$, and $\varsigma_n^i = \frac{1}{M}\sum_{u=1}^{M} A_{un}^2 \sigma_{y_{un}}^{2\ i}$.

With the above, the next step is to approximate (5.3) by normal density (2.19). This calls for the evaluation of the mean and variance of $\nu_{x_n \to y_m}^{i+1}$. For our specific problem, in which the prior density is a normal and skew normal mixture, one needs

to analyze the product $\mathcal{N}(x|\kappa,\varsigma)\mathcal{SN}(x|0,\omega_0,\alpha_0)$ in (5.3). Therefore, ***Lemma 1*** to ***Lemma 3*** are derived below.

Lemma 1 *Let $U \sim \mathcal{N}(\mu,\sigma^2)$ be a Gaussian random variable. We have $\mathrm{E}\big(\Phi(hU+k)\big) = \Phi(\frac{k+h\mu}{\sqrt{1+h^2\sigma^2}})$ for any $h,k \in \mathbb{R}$.*

Proof. ***Lemma 1*** is a direct extension of ***Lemma 2*** in [34], which states that $\mathrm{E}\big(\Phi(hV+k)\big) = \Phi(\frac{k}{\sqrt{1+h^2}})$ for $V \sim \mathcal{N}(0,1)$. By change of variable, $V = \frac{U-\mu}{\sigma}$, ***Lemma 1*** follows. ■

Corollary 1 *Let $G(x) = \mathcal{N}(x;\kappa,\varsigma)\mathcal{SN}(x;0,\omega_0,\alpha_0)$ be the product of the pdf of normal and skew normal densities, then $C_0\int_{-\infty}^{\infty} G(x)\mathrm{d}x = 1$ for a $C_0 \in \mathbb{R}^+$.*

Proof. To prove ***Corollary 1***, it is sufficient to show that $\int_{-\infty}^{\infty} G(x)\mathrm{d}x$ has a finite value. Recalling $G(x) \geq 0$ and $\Phi(x) \leq 1$ for $x \in \mathbb{R}$, it is derived that,

$$\begin{aligned}\int_{-\infty}^{\infty} G(x)\mathrm{d}x &< 2\int_{-\infty}^{\infty} \mathcal{N}(x;\kappa,\varsigma)\mathcal{N}(x;0,\omega_0^2)\mathrm{d}x \\ &< \sqrt{\frac{2}{\pi\omega_0^2}}\int_{-\infty}^{\infty} \mathcal{N}(x;\kappa,\varsigma)\mathrm{d}x = \sqrt{\frac{2}{\pi\omega_0^2}}\ .\end{aligned} \tag{5.4}$$

■

Additionally, $\int_{-\infty}^{\infty} G(x)\mathrm{d}x$ is found to be,

$$\int G(x)\mathrm{d}x = \int \mathcal{N}(x;\kappa,\varsigma)\mathcal{SN}(x;0,\omega_0,\alpha_0)\mathrm{d}x \tag{5.5}$$

$$= \int \frac{1}{\pi\omega_0\sqrt{\varsigma}} \exp\left(-\frac{(x-\kappa)^2}{2\,\varsigma} - \frac{x^2}{2\omega_0^2}\right)\Phi\left(\frac{\alpha_0 x}{\omega_0}\right)\mathrm{d}x \tag{5.6}$$

$$= \frac{1}{\pi\omega_0\sqrt{\varsigma}} \exp\left(\frac{-\kappa^2}{2(\varsigma+\omega_0^2)}\right)\int \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\Phi\left(\frac{\alpha_0 x}{\omega_0}\right)\mathrm{d}x \tag{5.7}$$

$$= \sqrt{\frac{2}{\pi(\varsigma+\omega_0^2)}} \exp\left(\frac{-\kappa^2}{2(\varsigma+\omega_0^2)}\right) \int \mathcal{N}(\mu,\sigma^2)\Phi\left(\frac{\alpha_0 x}{\omega_0}\right) \mathrm{d}x, \tag{5.8}$$

where $\mu = \dfrac{\kappa\omega_0^2}{\varsigma+\omega_0^2}$, $\sigma^2 = \dfrac{\varsigma\omega_0^2}{\varsigma+\omega_0^2}$, and all integrals are from $-\infty$ to ∞.

Applying ***Lemma 1*** on the integral term in (5.8), it is derived that, $C_0 = \upsilon \exp(\gamma)\,\Phi(\eta)^{-1}$, in which $\upsilon = \sqrt{\dfrac{\pi(\varsigma+\omega_0^2)}{2}}$, $\gamma = \dfrac{\kappa^2}{2(\varsigma+\omega_0^2)}$, $\eta = \dfrac{h\mu}{\sqrt{1+h^2\sigma^2}}$, $h = \dfrac{\alpha_0}{\omega_0}$.

Lemma 2 *Let the pdf of the random variable X be $C_0 \times \mathcal{N}(x;\kappa,\varsigma)\mathcal{SN}(x;0,\omega_0,\alpha_0)$. The moment generating function of X is found to be,*

$$M_X(t) = \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right) \Phi^{-1}(\eta)\Phi\left(\eta + \frac{h\sigma^2 t}{\sqrt{1+h^2\sigma^2}}\right). \tag{5.9}$$

Proof.

$$M_X(t) = C_0 \int \exp(tx)\mathcal{N}(x;\kappa,\varsigma)\mathcal{SN}(x;0,\omega_0,\alpha_0)\mathrm{d}x \tag{5.10}$$

$$= \Phi(\eta)^{-1} \int \exp(tx)\mathcal{N}(\mu,\sigma^2)\Phi\left(\frac{\alpha_0 x}{\omega_0}\right) \mathrm{d}x \tag{5.11}$$

$$= \frac{\exp\left(\mu t + \dfrac{t^2\sigma^2}{2}\right)}{\Phi(\eta)} \int \mathcal{N}(\mu + t\sigma^2, \sigma^2)\Phi\left(\frac{\alpha_0 x}{\omega_0}\right) \mathrm{d}x, \tag{5.12}$$

$$= \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right) \Phi^{-1}(\eta)\Phi\left(\eta + \frac{h\sigma^2 t}{\sqrt{1+h^2\sigma^2}}\right), \tag{5.13}$$

where (5.11) holds due to ***Corollary 1***, and (5.13) holds due to ***Lemma 1***, and all integrals are from $-\infty$ to ∞. ■

With the moment generating function $M_X(t)$, the mean and variance of the density function $C_0 \times G(x)$ are derived.

Lemma 3 *Let the pdf of the random variable X be $C_0 \times \mathcal{N}(x;\kappa,\varsigma)\mathcal{SN}(x;0,\omega_0,\alpha_0)$.*

Then the mean and variance are given by

$$\mathrm{E}(X) = \mu + \frac{\theta\sigma^2}{\sqrt{2\pi}}\,\Phi^{-1}(\eta)\exp\left(-\frac{1}{2}\eta^2\right), \tag{5.14}$$

and

$$\mathrm{Var}(X) = \mu^2 + \sigma^2 + \big(\mathrm{E}(X) - \mu\big)\rho - (\mathrm{E}(X))^2, \tag{5.15}$$

respectively, where $\theta = \dfrac{h}{\sqrt{1+h^2\sigma^2}}$, *and* $\rho = \dfrac{2\mu + \mu h^2\sigma^2}{1+h^2\sigma^2}$.

Using ***Lemma 1*** to ***Lemma 3*** and omitting the iteration superscript i and subscripts n and m for coefficients, (5.3) can be approximated by normal density as,

$$\nu_{x\to y}(x) \cong \mathcal{N}(\mu_x, \sigma_x^2), \tag{5.16}$$

$$\mu_x = \mathbb{F}(\kappa, \varsigma) = p_1\mu_1 + p_2\mu_2, \tag{5.17}$$

$$\sigma_x^2 = \mathbb{G}(\kappa, \varsigma) = p_1\left(\mu_1^2 + \sigma_1^2\right) + p_2\left(\mu_2^2 + \sigma_2^2\right) - (p_1\mu_1 + p_2\mu_2)^2, \tag{5.18}$$

where $\mu_1, \sigma_1^2, p_1, \mu_2, \sigma_2^2, p_2$ are calculated in Table 5.1.

$\mu_1 = \kappa\rho_S$,	$\mu_2 = \mu_0 + \frac{\theta\sigma_0^2}{\sqrt{2\pi}}\Phi^{-1}(\eta)\exp\big(-\frac{1}{2}\eta^2\big)$,	
$\sigma_1^2 = \varsigma\rho_S$,	$\sigma_2^2 = \mu_0^2 + \sigma_0^2 - \mu_2^2 + (\mu_2 - \mu_0)\rho_0$,	
$p_1 = (1-\lambda)\frac{C}{C_1}$,	$p_2 = \lambda\frac{C}{C_2}$,	$C_1 = \upsilon_S\beta$,
$C_2 = \upsilon_L\exp(\gamma)\Phi(\eta)^{-1}$,	$\mu_0 = \kappa\rho_L$,	$\sigma_0^2 = \varsigma\rho_L$,
$\rho_S = \frac{\sigma_S^2}{\varsigma + \sigma_S^2}$,	$\rho_L = \frac{\omega_L^2}{\varsigma + \omega_L^2}$,	$\rho_0 = \frac{2\mu_0 + \mu_0 h^2\sigma_0^2}{1 + h^2\sigma_0^2}$,
$\gamma = \frac{1}{2\sigma_0^2}\big(\kappa^2\rho_L - \mu_0^2\big)$,	$\eta = \frac{h\mu_0}{\sqrt{1+h^2\sigma_0^2}}$,	$\theta = \frac{h}{\sqrt{(1+h^2\sigma_0^2)}}$,
$h = \frac{\alpha_L}{\omega_L}$,	$\beta = \exp\Big(\frac{\kappa^2}{2(\varsigma+\sigma_S^2)}\Big)$,	$\upsilon_S = \sqrt{2\pi(\varsigma+\sigma_S^2)}$,
$\upsilon_L = \sqrt{\frac{\pi(\varsigma+\omega_L^2)}{2}}$,	$C = \frac{C_1C_2}{\lambda C_1 + (1-\lambda)C_2}$.	

Table 5.1: Message Passing Parameters for $\mathbb{F}(\kappa, \varsigma)$ and $\mathbb{G}(\kappa, \varsigma)$

Omitting the iteration superscripts and coefficient subscripts, $\mathbb{F}'(\kappa_{x_n}^{i-1}, \varsigma^{i-1})$ in

(2.32) is calculated as,

$$\mathbb{F}'(\kappa,\varsigma)=(1-\lambda)\left(\mu_1\zeta_1+\frac{C}{C_1}\rho_S\right)+\lambda\left(\mu_2\zeta_2+\frac{C}{C_2}\delta\right), \tag{5.19}$$

in which ζ_1,ζ_2 and δ can be calculated as Table 5.2.

$\delta=\rho_L-\frac{\theta\sigma_0^2}{\sqrt{2\pi}}\exp\big(-\frac{1}{2}\eta^2\big)\big(\tau_0\Phi^{-2}(\eta)+\eta\rho_L\theta\Phi^{-1}(\eta)\big)$.	
$\tau=(1-\lambda)\tau_2\upsilon_L+\lambda(\tau_1\Phi(\eta)+\beta\tau_0)\upsilon_S$,	
$\zeta_1=\frac{\upsilon_L\tau_2\beta_0-\upsilon_L\exp(\gamma)\tau}{\beta_0^2}$,	$\zeta_2=\frac{\upsilon_S\beta_0(\tau_1\Phi(\eta)+\beta\tau_0)-\upsilon_S\beta\Phi(\eta)\tau}{\beta_0^2}$,
$\tau_1=\frac{\beta\kappa}{\varsigma+\sigma_S^2}$,	$\tau_2=\frac{(\kappa-\mu_0)\exp(\gamma)\rho_L}{\sigma_0^2}$,
$\tau_0=\frac{\exp\big(-0.5\eta^2\big)\rho_L\theta}{\sqrt{2\pi}}$,	$\beta_0=(1-\lambda)\upsilon_L\exp(\gamma)+\lambda\beta\upsilon_S\Phi(\eta)$.

Table 5.2: Message Passing Parameters for $\mathbb{F}'(\kappa,\varsigma)$

Therefore, similar to the approximate message passing (2.30)-(2.34) for arbitrary prior density, our approximate message passing utilizing the proposed normal and skew normal density (5.2) is concluded as (5.17), (5.18), (2.32), (2.33) and (2.34), where $\mathbb{F}'(\kappa_{x_n}^{i-1},\varsigma^{i-1})$ in (2.32) is calculated as (5.19).

5.4 Gradient Based Parameter Estimation

We now detail the parameter estimation for the density of each state. To estimate the parameters, we fit the reconstruction of AMP to the proposed normal and skew normal prior density model (5.2). It is expected that, the prior density model, and the learned parameters can regularize later AMP reconstructions.

Our strategy is *divide-and-conquer*. First of all, the reconstruction is divided into two sets, *i.e.,* large state set and small state set, according to the sparsity rate[1] $\lambda=K/N$. Specifically, Let T be the set of K largest coefficients of $\underline{\hat{x}}=[\mu_{x_1}^{I_d},\ldots,\mu_{x_N}^{I_d}]$. Meanwhile, denote T^c as the set containing the remaining $N-K$ coefficients.

[1]As [28], the sparsity rate, $\lambda=K/N$, is assumed to be known at the reconstruction stage.

For the small state, its variance can be estimated as the unbiased sample variance, *i.e.,*

$$\hat{\sigma}_S^2 = \frac{1}{N-K-1} \sum_{\hat{x}_i \in T^c} \hat{x}_i^2. \tag{5.20}$$

Given the large state set T, the parameters are estimated by maximizing the log-likelihood of the large state set T, with respect to ω_L and α_L, *i.e.,*

$$\omega_L^*, \alpha_L^* = \underset{\omega_L, \alpha_L \in \mathbb{R}}{\arg\max}\, \ell(T; \omega_L, \alpha_L), \tag{5.21}$$

where

$$\ell = K \log \frac{2}{\omega_L} - \frac{1}{2} \sum_{\hat{x}_i \in T} \left(\frac{\hat{x}_i}{\omega_L} \right)^2 + \sum_{\hat{x}_i \in T} \log \left(\Phi \left(\alpha_L \frac{\hat{x}_i}{\omega_L} \right) \right). \tag{5.22}$$

Besides, the gradients of ω_L and α_L with respect to (5.22) are found to be,

$$\frac{d\ell}{d\omega_L} = -\frac{K}{\omega_L} + \sum_{\hat{x}_i \in T} \frac{\hat{x}_i^2}{\omega_L^3} - \frac{\alpha_L}{\omega_L^2} \sum_{\hat{x}_i \in T} \frac{\phi \left(\alpha_L \dfrac{\hat{x}_i}{\omega_L} \right)}{\Phi \left(\alpha_L \dfrac{\hat{x}_i}{\omega_L} \right)} \hat{x}_i, \tag{5.23}$$

$$\frac{d\ell}{d\alpha_L} = \frac{1}{\omega_L} \sum_{\hat{x}_i \in T} \frac{\phi \left(\alpha_L \dfrac{\hat{x}_i}{\omega_L} \right)}{\Phi \left(\alpha_L \dfrac{\hat{x}_i}{\omega_L} \right)} \hat{x}_i. \tag{5.24}$$

With (5.22) and gradients (5.23) (5.24), one can choose from a variety of solvers, including *trust-region-reflective* [74,75], *interior-point* [76] algorithms to find the optimum ω_L^* and α_L^*, after which $\underline{\hat{\Theta}} = [\hat{\sigma}_S^2, \omega_L^*, \alpha_L^*]$ is fed back to the approximate message passing (6.9).

It should be noted that (5.22) is not convex in general. As a result, the proposed

gradient estimator can only find local solutions, and a good initialization strategy becomes consequential for our task.

In this work, we find that initializing ω_L and α_L such that the expected mean and variance of the skew normal density match the sample mean and variance of the large state coefficients of $\underline{\hat{x}}$ works satisfactorily. Therefore, ω_L is initialized at,

$$\omega_0 = \sqrt{\mu_T^2 + \sigma_T^2}, \tag{5.25}$$

and the initial value of α_L can be found by solving the following,

$$\frac{\alpha_0^2}{1+\alpha_0^2} = \frac{\pi}{2}\frac{\mu_T^2}{\mu_T^2 + \sigma_T^2}, \tag{5.26}$$

where μ_T and σ_T^2 are the sample mean and variance of large state set T.

Give the reconstruction, the noise variance can be estimated based on the residual, *i.e.,*

$$\hat{\sigma}_e^2 = \frac{1}{M}\sum_{m=1}^{M}(y_m - A_{m\cdot}\underline{\hat{x}})^2. \tag{5.27}$$

5.5 Complexity Analysis

Thanks to the efficient AMP framework, and together with the fast gradient-based parameter estimation, our proposed technique is highly computationally effective.

Similar to [25, 26, 77], the complexity of each message passing iteration in our AMP module is dominated by multiplying sampling matrix $A \in \mathbb{R}^{M\times N}$ with vector $\underline{\hat{x}} \in \mathbb{R}^{N\times 1}$. Besides, as can be seen in (5.23) and (5.24), the parameter estimation module involves only vector operations. This makes the complexity of our proposed technique dominated by the AMP module.

It is worth pointing out that in our derivation, evaluating functions including $\phi(\cdot)$, $\Phi(\cdot)$, as well as their division $\phi(\cdot)/\Phi(\cdot)$, will incur sizable computation overhead.

Table 5.3 summarizes the running time of several frequently evaluated functions in our scheme.

The test is implemented in Matlab [78] and is performed on a computer with dual core 2.67 GHz CPUs, and 8 GB of 1333 MHz RAM, where the input argument of each function is a scaler, and the results are the average of 10^8 random and independent trials.

add	multiply	divide	square	ϕ	Φ	ϕ/Φ
5.93	6.08	6.24	6.39	30.42	34.94	80.5

Table 5.3: Average running time (in nanoseconds, 10^{-9} seconds) of frequently evaluated functions.

It can be seen in Table 5.3, compared to scaler addition, evaluating $\phi(\cdot)$ and $\Phi(\cdot)$ are generally 5 to 6 times slower, while the division $\phi(\cdot)/\Phi(\cdot)$ is about 14 times slower.

Therefore, as a rule of thumb, a Floating Point Operations (FLOP) proportional to $10M \times (2N-1) \approx 20MN$ is expected at each iteration. As will be seen in the test, the runtime of our proposed method scales decently as the signal dimensionality N increases, making it one of most efficient techniques in the community.

5.6 Simulations

In this section, the performance of our proposed method is evaluated under phase transition, noisy reconstruction, support set recovery, and runtime tests. Besides, our technique is examined under real world application, *i.e.,* weather sensor network.

The sampling matrix $\mathbf{A}$ is generated from standard Gaussian ensemble, with each column being normalized to unit norm, *i.e.,* $\|A_{\cdot n}\|_2 = 1$, for $n = 1, \ldots, N$.

In reconstruction, our method alternates between approximate message passing and parameter estimation. Unless otherwise specified, these two modules execute up to 8 times, or stopped when consecutive normalized reconstruction difference $\|\hat{\underline{x}}^{new} - \hat{\underline{x}}^{old}\|_2^2 / \|\hat{\underline{x}}^{new}\|_2^2 \leq 10^{-6}$. In approximate message passing, $\mu_{x_n}^1$ is initialized at 0 for $n = 1, \ldots, N$, $\mu_{y_m}^1$ is set to y_m for $m = 1, \ldots, M$. Besides, ς^1 is set to 10^4

to make the inference robust. The message passing is executed up to 50 iterations, or until the convergence, which is claimed when $\|\underline{\hat{\mu}}^{i+1} - \underline{\hat{\mu}}^{i}\|_2$ is less than 10^{-7}, where $\underline{\hat{\mu}}^{i} = [\mu^{i}_{x_1}, \ldots, \mu^{i}_{x_N}]$.

In estimating the parameters, we employ the classic *trust-region-reflective* [74, 75] as the optimizer, where ω_L is bounded by $[0, \infty]$. Additionally, α_L is bounded by $[-15, 15]$ for numerical stability. The optimization is terminated after 500 iterations, or when the consecutive log-likelihood difference $\leq 10^{-6}$, whichever comes earlier.

5.6.1 Phase Transition

In the first test, the proposed method is examined under the empirical phase transition test [79]. The support set of the signal is generated uniformly at random, namely, index $n = 1, \ldots, N$ is sampled with a uniform probability $\lambda = K/N$. In generating the magnitude of the significant coefficients, two cases are considered.

In the first case, the significant coefficients are generated identically and independently from normal distribution $\mathcal{N}(0, \sigma^2)$, where the standard deviation σ follows a prior uniform distribution $\mathcal{U}[5, 25]$. Besides, the significant coefficients are made strictly non-negative by taking the absolute values.

In the second case, the significant coefficients are generated identically and independently from uniform distribution $\mathcal{U}[b_l, b_u]$, where the lower bound follows a prior uniform distribution $b_l \sim \mathcal{U}[-20, 0]$, and the upper bound follows a prior uniform distribution $b_u \sim \mathcal{U}[0, 200]$.

The insignificant coefficients are generated from normal distribution with mean 0 and variance 10^{-4}. In the first execution of approximate message passing module, $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$ is set to $[10^{-5}, 50, 0]$, which in later executions, will be updated at the solution found by the gradient based parameter estimator $\underline{\hat{\Theta}} = [\hat{\sigma}_S^2, \omega_L^*, \alpha_L^*]$. The signal length is set to $N = 1000$. Meanwhile, M/N is set from 0.05 to 0.5 at steps of 0.025. For each value of M/N, K/M is varied from 0 to 1 at steps of 0.025. 500

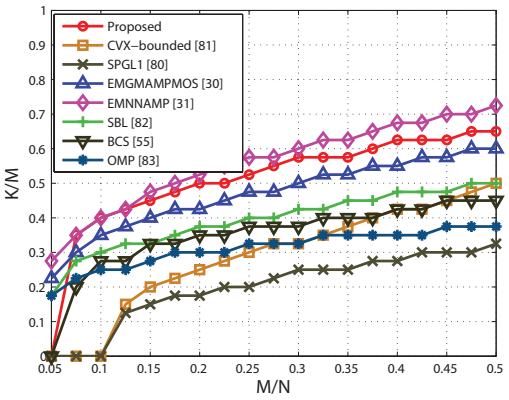
independent trials are executed for each combination of M/N and K/M, and the Normalized Square Error (NSE), evaluating as NSE $\triangleq \|\hat{\underline{x}} - \underline{x}_{true}\|_2^2 / \|\underline{x}_{true}\|_2^2$ with $\underline{x}_{true}$ denoting the ground truth, is recorded for each trial.

As in [79], the maximum value of K/M, up to which the corresponding success rate is $\geq 50\%$ is registered. Besides, a success trial is defined as the one with NSE $\leq 10^{-4}$.

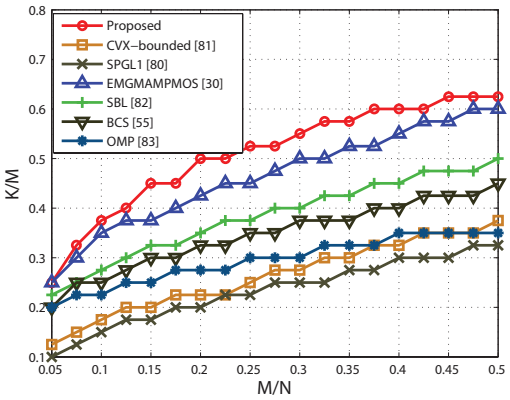
The performance is compared with two AMP based techniques, namely EMGMAMPMOS [30], and EMNNAMP [31]. Besides, SPGL1 [80], and CVX [81] are included in the comparison to solve LASSO [24]. It should be noted, CVX [81] is aided with side information, where the optimization is constrained with upper bound being the maximum of $\underline{x}_{true}$, and lower bound being the minimum of $\underline{x}_{true}$. Additionally, several powerful Bayesian and greedy algorithms, including Sparse Bayesian Learning (SBL) [82], Bayesian Compressive Sensing (BCS) [55], and Orthogonal Matching Pursuit (OMP) [83] are also included in the tests. Furthermore, since the sparsity rate λ is assumed to be known in our scheme, for fairness, the sparsity ratio in EMGMAMPMOS [30], EMNNAMP [31], and OMP [83] are fixed to $\lambda = K/N$. The simulation results are plotted in Figures 5.3a and 5.3b.

As can be seen in Fig. 5.3a where all significant coefficients are strictly non-negative, EMNNAMP [31] gives the benchmark phase transition curve by taking advantage of the non-negative normal density mixture. It is also noted that in Fig. 5.3a, although without any prior of the non-negativity, our proposed scheme is capable of exploiting the asymmetric feature of the significant coefficients, and provides very competitive performance.

Since EMNNAMP [31] is designed exclusively for non-negative signals, its plot is omitted in Fig. 5.3b, where the significant coefficients consist of both positive and negative components. As can be seen in Fig. 5.3b, comparing to many sophisticated techniques, our method provides the most competitive performance. This shows our technique can effectively exploit the underlining skewness of the signal, while being



(a)



(b)

Figure 5.3: Phase transition test. (a) Significant coefficients are strictly non-negative. (b) Significant coefficients are a mix of positive and negative elements. The signal length is set to $N = 1000$.

sufficiently flexible to accommodate both positive and negative elements.

5.6.2 Noisy Reconstruction

In the second test, the performance of our technique is examined under noisy environments. The significant coefficients are generated in ways similar to previous phase transition test. To make the reconstruction more challenging, unlike the phase transition test where the magnitudes of insignificant coefficients are negligible, in this test, the insignificant coefficients are generated from normal distribution with mean 0 and variance 0.5. It should be noted that similar setups, referred to as *heavy-tailed tests*

[2], can be found in [30] where Student's-t and log-normal prior densities are utilized to generate the signals.

The length of the signal is set to $N = 500$. The number of significant coefficients K is set to 50, and the number of samples M is set to 125. The noise vector $\underline{e}$ is sampled from Gaussian density, *i.e.,* $\underline{e} \sim \mathcal{N}(\underline{0}, \sigma_e^2 \mathrm{I}_{M\times M})$, and is added to the measurement. The variance of the noise, σ_e^2, is adjusted such that $\mathrm{SNR} = 10\log_{10}(\|A\underline{x}_{true}\|_2^2/\|\underline{e}\|_2^2)$ is varied from 10 dB to 30 dB at 2 dB increments. Meanwhile, $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$ are set to [1, 50, 0] at the first execution of approximate message passing module. The noise variance is initialized at 1, and is estimated as (5.27) in later reconstruction iterations.

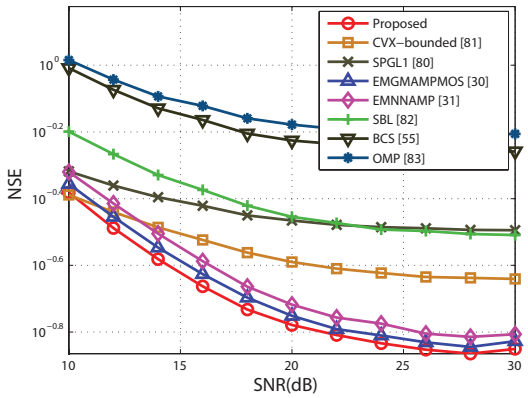
The performance of our proposed technique, EMGMAMPMOS [30], EMNNAMP [31], SPGL1 [80], and CVX (bounded) [81], SBL [82], BCS [55], and OMP [83] are compared and the results are summarized in Fig. 5.4, where each data point is the average of 500 independent trials. As can be seen, our technique yields superior results in both Fig. 5.4a and 5.4b.

5.6.3 Support Set Recovery

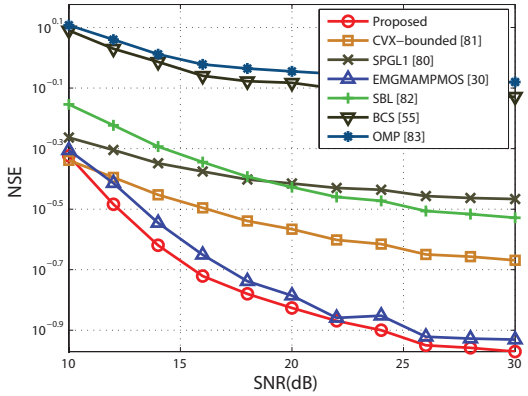
In this test, the capability of support set recovery is examined. As previous tests, two types of signals are generated, *i.e.,* strictly non-negative, and mix of positive and negative, where for each type of signals, the parameters characterizing both significant state and insignificant state, as well as the initialization of $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$, are set identical to those of phase transition test. The measurement is noiseless.

The length of signal is set to $N = 500$, and the number of random samples is fixed at $M = 125$. We gradually vary the number of significant coefficients K by adjusting K/M from 0.025 to 1, at steps of 0.025, where for each value of K, 500 independent

[2]By default, the *heavy-tailed tests* in EMGMAMPMOS [30] assumes a symmetrical signal, and the means of the density components are fixed to 0. For fairness, we turn on the update of means for EMGMAMPMOS.



(a)



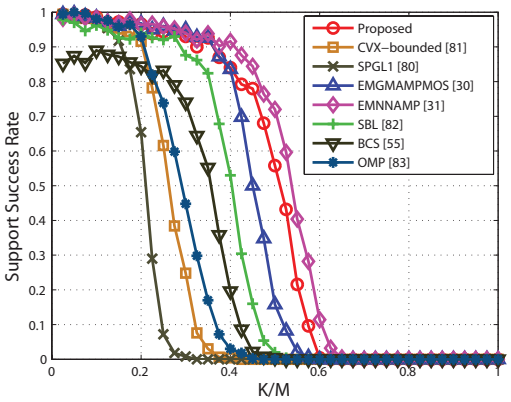
(b)

Figure 5.4: NSE vs. SNR. (a) Significant coefficients are strictly non-negative. (b) Significant coefficients are a mix of positive and negative elements.

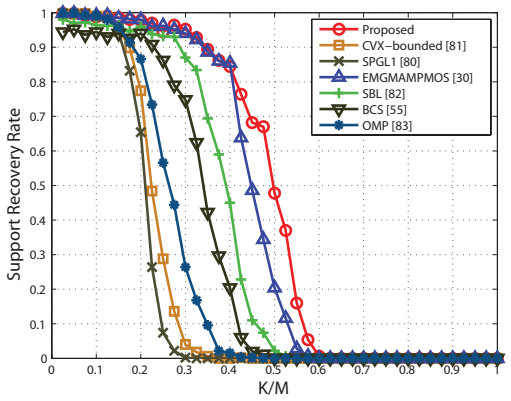
random trials are performed. Besides, the support set recovery rate is calculated by counting the trail with correct recovery of support set, *i.e.,* the trial whose estimated support set matches exactly with the ground truth. Since not all techniques are able to yield strictly sparse solutions, a threshold of 0.1 is applied to get the estimated support from the raw reconstruction.

We compare our proposed technique with EMGMAMPMOS [30], EMNNAMP [31], SPGL1 [80], and CVX (bounded) [81], SBL [82], BCS [55], and OMP [83], and the results are plotted in Figures 5.5a and 5.5b. As can be seen, for each method, the support set recovery rate decays with increasing K. Yet, thanks to ability of exploiting the asymmetrical feature of the signal, our proposed technique is capable

of providing reliable support recovery over a decently large region of K in both Figures 5.5a and 5.5b.



(a)



(b)

Figure 5.5: Support Recovery Rate vs. K/M (a) Significant coefficients are strictly non-negative. (b) Significant coefficients are a mix of positive and negative elements.

5.6.4 Runtime

We are now testing the Runtime of our proposed technique. In this test, the length of signal, N, is varied from 500 to 5000, at steps of 500. Meanwhile, without lose of generality, we fix $M/N = 0.5$, and $K/M = 0.4$, for all values of N. Signals are generated such that all significant coefficients are strictly positive, where the characterizing parameters of the densities, as well as the initialization of $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$, are set similar as phase transition test. The test is performed on a computer with

hex core 2.0 GHz CPUs, and 32 GB of 1333 MHz RAM.

We compare the runtime of our technique with EMGMAMPMOS [30], EMNNAMP [31], SPGL1 [80], and CVX (bounded) [81], SBL [82], BCS [55], and OMP [83], where all methods are implemented with Matlab [78].

The mean runtime are plotted in Fig. 5.6, where each data point is the average of 50 independent trials. Clearly, similar to the two AMP relatives, namely EMG-

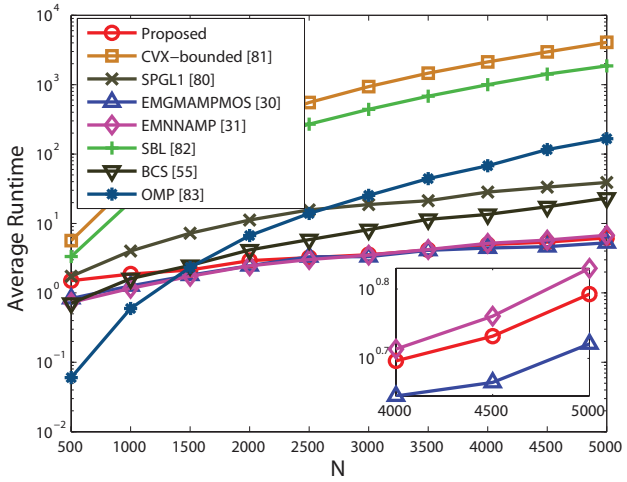


Figure 5.6: Signal Length N vs. Average Runtime (in seconds)

MAMPMOS [30] and EMNNAMP [31], our proposed technique is computationally effective. This advantage is most remarkable under relatively large signal dimensionality. For example, when $N = 5000$, our technique yields an average runtime of 6.204 Seconds (sec), which is more than 650 times faster than CVX, and 6.27 times faster than SPGL1. Besides, comparing to OMP, our technique runs 26 times faster. Moreover, our technique has advantage over Bayesian algorithms, with BCS and SBL being 3.73 and 299 times slower.

5.6.5 Weather Data Test

We evaluate our proposed technique with a dataset collected from a real weather sensor network. The data is referred to as *cooling degree day departure from normal*

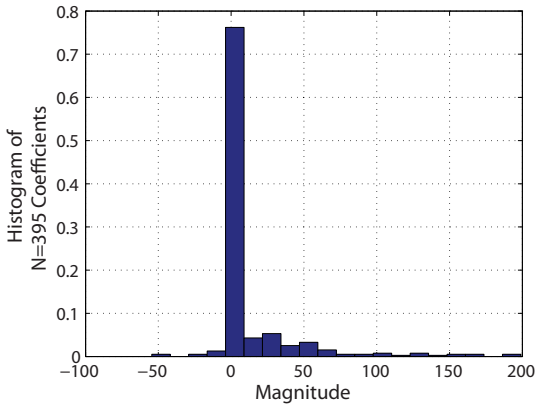
[84]. Cooling degree day is derived from outside air temperature, and is widely used in estimating the energy needed to cool a structure [84]. The phrase *departure from normal* suggests that a 30-year historical average is subtracted from the data. Our data is obtained from Automated Surface Observing System (ASOS) [85], and is accessible at National Climate Data Center [84].

The data is of length $N = 395$, and has $K = 143$ nonzero coefficients. As can be seen in the histogram plotted in Fig. 5.7a, the nonzero coefficients are asymmetrically positive. The data is down-sampled by projection with a Gaussian random sampling matrix A. The measurement is noisy, and the noise variance σ_e^2 is adjusted such that the SNR is varied from 10 dB to 30 dB at 2 dB increments. For each value of SNR, 100 realizations of random sampling matrix A are generated, and $M = 2K$. For each trial, our method performs approximate message passing decoding and parameter estimation up to 8 times, or stopped when $\|\hat{\underline{x}}^{new} - \hat{\underline{x}}^{old}\|_2^2 / \|\hat{\underline{x}}^{new}\|_2^2 \leq 10^{-2}$. Additionally, $\underline{\Theta} = [\sigma_S^2, \omega_L, \alpha_L]$ is initialized similar to phase transition test, and the noise variance is estimated as (5.27).

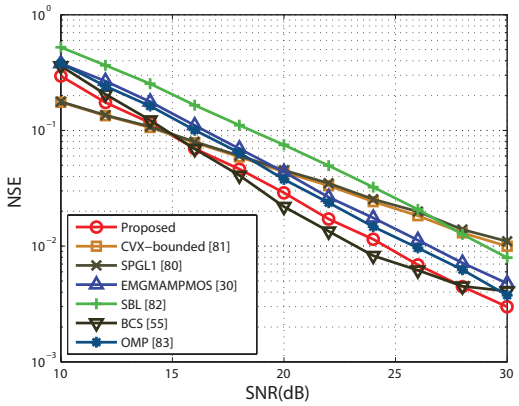
We compare our technique with EMGMAMPMOS [30], SPGL1 [80], and CVX (bounded) [81], SBL [82], BCS [55], and OMP [83]. Since the significant coefficients contain negative elements, EMNNAMP [31] is excluded from the test. Fig. 5.7b summarizes the reconstruction NSE as SNR varies. Overall, our scheme provides satisfactory results in most of the range. It is noteworthy that, although not being designed for asymmetrical signals, BCS [55] gives very competitive results by exploiting the sparsity of the signal in this test.

5.7 Conclusion

In this chapter, the compressive sensing of the sparse signals whose significant coefficients are distributed asymmetrically with respect to zero is analyzed. To properly capture the asymmetry, a two-state normal and skew normal mixture density is pro-



(a)



(b)

Figure 5.7: Temperature Data Test (a) Histogram of the temperature data. (b) NMSE vs. SNR.

posed to model the density of the signal. The significant and insignificant coefficients of such signals are represented by a skew normal distribution and a normal distribution, respectively. An approximate message passing algorithm is then designed to take inference of the signal from the compressive sensing measurement while providing fitting to the model. A gradient-based parameter estimator is put forward to infer the underlining density of each component. Experiment results on simulated data and real-world data, *i.e.,* weather sensor network, show our proposed technique can effectively exploit the asymmetrical feature, and provides competitive results compared to the state-of-the-art techniques.

CHAPTER 6

Compressive Sampling of Clustered Sparse Signals with Asymmetric features

In this chapter, we investigate the compressive sensing task of clustered sparse signals, where the magnitudes of each cluster are distributed asymmetrically *w.r.t* the cluster mean. To address the skewness feature of the signal, a finite skew-normal density mixture is utilized to model the prior distribution, where the marginal posterior of the signal is inferred by an efficient approximate message passing based algorithm. An Expectation-Maximization-based algorithm is developed to estimate the mixture density. The clustered property is then modelled by the *Potts* model, and a loopy belief propagation algorithm is designed to promote the spatial feature. Experiments results show that our technique is highly effective and efficient in exploiting both the clustered feature and asymmetrical feature of the signals, and outperforms many sophisticated techniques.

6.1 Introduction

Reconstruction of clustered sparse signal is an active line of research of compressive sensing community. In multimedia processing [10], it is found significant pixels of video difference frames tend to form clusters, due to the temporal redundancy of consecutive video frames. Another promising application can be found in sensor networks for abnormal environment event detection [85], where in the presence of abnormality, sensors close to the event give significant and correlated outputs, while those outside the scope of the event return outputs resembling the no-event average.

Many sophisticated strategies have been proposed to exploit the clustered property in compressive sensing tasks. In *Struct-OMP* [86], a pruning stage is designed to encourage clustered property based on Orthogonal Matching Pursuit (OMP). In [87], a Markov Chain Monte Carlo strategy is employed, and the proposed technique, *CluSS*, turns out to realize faithful reconstruction in dealing with block clustered sparse signals. In *SRL1* [10], a structural re-weighted ℓ_1 norm minimization technique is developed, where signal coefficients are allocated with weights determined by the magnitudes of their corresponding neighbors. In *LaMP* [37,38], the clustered sparsity of the signal is modelled by the Ising model, from which the signal support is estimated and the reconstruction is directed.

Compressive sensing of asymmetrical signals is another line of research, and signals of this type can be found in Multi-Input Multi-Output (MIMO) wireless communication systems [88], and weather sensor networks [7, 84, 85]. In [31], A *Bernoulli* non-negative *Gaussian* mixture is employed to model the distribution of sparse signals with non-negative coefficients, and an efficient approximate message passing based algorithm is proposed. An effective framework is proposed in [7] to deal with sparse signals with skewness feature, where a two-state normal and skew normal mixture density was utilized to model the prior distribution of the signals. The asymmetrical feature is captured by the skew normal density component, and the signal is estimated by a approximate message passing based algorithm.

Following Chapter 5, in this chapter, we move one step further by approaching the compressive sensing of clustered sparse signals, where the magnitudes of each cluster are distributed asymmetrically about the corresponding cluster mean. One typical example for such signals can be found in sensor networks, where multiple events of different types and intensities are likely to occur simultaneously, and clusters of different events may in turn exhibit varying features.

To get a faithful reconstruction of the signals, we adopt a *divide-and-conquer*

methodology, and decompose the task into three modules. First of all, to address the skewness feature, a finite skew-normal distribution mixture is utilized to model the prior distribution of the signal. Skew normal distribution [34] generalizes normal distribution, and is more flexible in dealing with asymmetric features. Based on the finite skew normal distribution model, an efficient approximate message passing based algorithm is developed to infer the signal by estimating the corresponding marginal posterior. Next, an Expectation-Maximization based algorithm is developed to estimate the mixture density. Additionally, the clustered property is modelled by the *Potts* model, and a loop belief propagation algorithm is designed to promote the spatial feature. A variety of experiments are conducted to test the performance of the proposed technique. Experiments results show that our developed technique is highly effective and efficient in exploiting both the clustered feature and asymmetrical feature of the signals, and outperforms many sophisticated methods.

The remainder of this paper is organized as follows. The signal model, and the framework of our proposed technique, are introduced in Sec. 6.2. Approximate message passing employing the skew normal mixture prior is detailed in Sec. 6.3. In Sec. 6.4, an Expectation-Maximization based algorithm is put forward to infer the finite skew normal density mixture. The hidden states estimate using loopy message passing and *Potts* model is derived in Sec.6.5. The complexity of our proposed technique is analyzed in Sec. 6.6, and simulation results are summarized in Sec. 6.7, and Sec. 6.8 concludes the work.

6.2 Signal Model and Problem Definition

In this section we will introduce the signal model and formally define the problem.

6.2.1 Signal Model

Signal Representations

Denote the two dimensional signal $\mathbf{x} = (x_{ij}) \in \mathbb{R}^{d\times d}$ as the outcome of random variable $\mathbf{X} = (X_{ij}) \in \mathbb{R}^{d\times d}$, where $1 \leq i, j \leq d$, and $d^2 = N$. For ease of notation, in this work, the two dimensional signal $\mathbf{x}$ is also represented as a one dimensional column vector, $\underline{x} = [x_1, \ldots, x_n, \ldots, x_N]^\intercal$, where x_{ij} is mapped to x_n in one dimensional form with $n = (i-1) \times d + j$, and $1 \leq n \leq N$. Similarly, $\underline{X} = [X_1, \ldots, X_n, \ldots, X_N]^\intercal$ is the one dimensional representation of $\mathbf{X}$.

It is also convenient to represent the signal as a concatenation of clusters. Specifically, let G be the total number of clusters, out of which, $0 \leq G_s < G$ clusters are significant, with the remaining being insignificant. Therefore, the signal can be written as, $\underline{x} = [\underline{x}_1^\intercal, \ldots, \underline{x}_g^\intercal, \ldots, \underline{x}_G^\intercal]^\intercal$, with $\underline{x}_g = [x_{g(1)}, \ldots, x_{g(d_g)}]^\intercal$ denoting the g-th cluster, where $1 \leq g \leq G$. Besides, d_g denotes the cardinality of cluster g, and $\sum_{g=1}^{G} d_g = N$.

In this work, it is assumed that signals are drawn from a probabilistic density ensemble of K density components. Let $S_n \in \{1, \ldots, K\}$ be a random variable indicating the corresponding state of signal coefficient X_n, and denote $\underline{S} = [S_1, \ldots, S_N]^\intercal \in \mathbb{R}^{N\times 1}$ as the state random vector, with the corresponding realization $\underline{s} = [s_1, \ldots, s_N]^\intercal$ being the state vector.

Without any constraint, the state vector $\underline{s}$ lies in the $\{1, \cdots, K\}^N$ subspace of $\mathbb{R}^N$. To realize clustered property, we restrict the states within a cluster to be homogenous, *i.e.,* $s(i) = s(j)$ for any $x_i, x_j \in \underline{x}_g$.

Additionally, let $\mathbf{V} = (V_{nk}) \in \mathbb{R}^{N\times K}$ be the state probability matrix, where V_{nk} denotes the probability of X_n taking state k, with the non-negative probability constraint $0 \leq V_{nk} \leq 1$, and unitary row sum constraint $\sum_{k=1}^{K} V_{nk} = 1$.

Mixture Skew Normal density Model

In this work, it is assumed that the clusters of signal coefficients are drawn independently from a mixture of K density components,

$$f(\underline{x}_g; \boldsymbol{\Theta}) \sim \sum_{k=1}^{K} \lambda_k p(\underline{x}_g; \underline{\theta}_k), \tag{6.1}$$

for $g = 1, \ldots, G$, where $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]^\intercal$ is the non-negative mixing weight vector satisfying $\sum_{k=1}^{K} \lambda_k = 1$, and $\lambda_k \geq 0$. Besides, $\underline{\theta}_k$ denotes the parameter vector specifying the k-th density component, with $\boldsymbol{\Theta} = [\underline{\theta}_1, \ldots, \underline{\theta}_K]^\intercal$ being the parameter matrix.

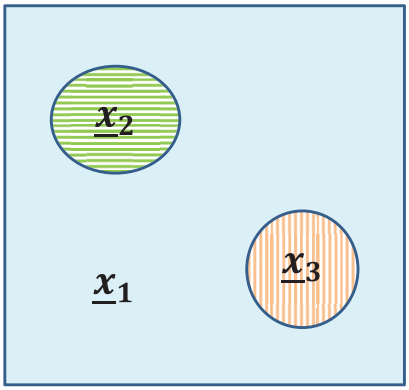
Moreover, it is further assumed that signal coefficients of any cluster are independent, conditioned on the states vector. Therefore, the joint distribution of the signal coefficients of any cluster $g \in [1, \ldots, G]$ can be factorized as,

$$\begin{aligned} p\left(\underline{X}_g = \underline{x}_g | S(\underline{x}_g) = k\right) &= \prod_{x \in \underline{x}_g} \mathcal{SN}(x | \xi_k, \omega_k, \alpha_k) \\ &= \prod_{x \in \underline{x}_g} \frac{2}{\omega_k} \phi\left(\frac{x - \xi_k}{\omega_k}\right) \Phi\left(\alpha_k \frac{x - \xi_k}{\omega_k}\right), \end{aligned} \tag{6.2}$$

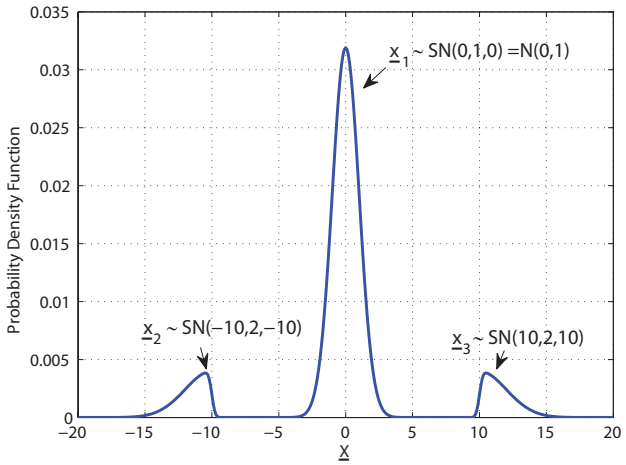
where skew normal density (5.1) is employed as the density component of the mixture (6.1).

Fig. 6.1a is a toy example of a clustered sparse signal, generated from the corresponding skew normal mixture density shown in Fig. 6.1b. Following previous notations, the signal in Fig. 6.1a can be written as a concatenation of $G = 3$ clusters, *i.e.,* $\underline{x} = [\underline{x}_1^\intercal, \underline{x}_2^\intercal, \underline{x}_3^\intercal]^\intercal$, where $\underline{x}_1^\intercal$ is insignificant cluster (a cluster with insignificant data values), $\underline{x}_2^\intercal$ and $\underline{x}_3^\intercal$ are significant clusters. Besides, $\underline{x}_1$ is drawn from $p(\underline{x}_1; \underline{\theta}_1)$, with $\underline{\theta}_1 = [\xi_1 = 0, \omega_1 = 1, \alpha_1 = 0]^\intercal$, $\underline{x}_2$ is drawn from $p(\underline{x}_2; \underline{\theta}_2)$, with $\underline{\theta}_2 = [\xi_2 = -10,\ \omega_2 = 2, \alpha_2 = -10]^\intercal$, and $\underline{x}_3$ is drawn from $p(\underline{x}_3; \underline{\theta}_3)$, with $\underline{\theta}_3 = [\xi_3 = 10, \omega_3 = 2, \alpha_3 = 10]^\intercal$. The mixing weight in Fig. 6.1b is set to $\underline{\lambda} = [\lambda_1 =$

$0.8, \lambda_2 = 0.2, \lambda_3 = 0.2]$.



(a)



(b)

Figure 6.1: Clustered Sparse Signal and Skew Normal Mixture Density (a) Signal with $G = 3$ clusters, where $G_s = 2$ clusters are significant. (b) Mixture density of $K = 3$ Skew Normal density components.

6.2.2 Problem Definition and System Architecture

We adopt a Bayesian perspective in the reconstruction phase of the compressive sensing task, with the goal being set to derive a faithful estimate of signal by maximizing the posterior distribution $p\left(\underline{x}|\underline{y}, \mathbf{V}, \mathbf{\Theta}\right)$. As neither mixture parameters $\mathbf{\Theta}$ nor the state probability $\mathbf{V}$ is known, an effective algorithm is developed to seek a reliable reconstruction of the signal by iteratively applying the sub-modules shown in Fig. 6.2.

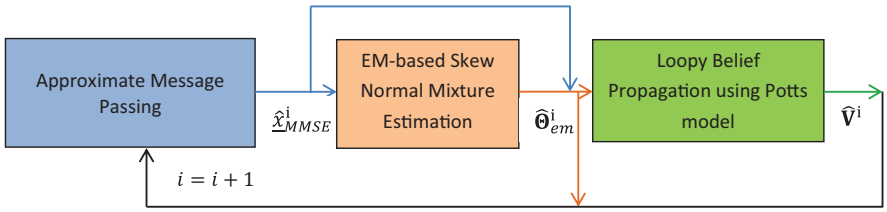


Figure 6.2: Diagram of CL-SNM-BP

As can be seen in Fig. 6.2, at iteration i, CL-SNM-BP starts with an approximate message passing module, where an *MMSE* estimate of the signal is obtained, by calculating the conditional expectation of the posterior, *i.e.,* $\hat{\underline{x}}^{i}_{\text{MMSE}} = E\left[\underline{X}|\underline{Y} = \underline{y}, \mathbf{V}^{i-1}, \boldsymbol{\Theta}^{i-1}\right]$.

Subsequently, $\hat{\underline{x}}^{i}_{\text{MMSE}}$ is fed to the second module to get an estimate of the mixture density parameters $\underline{\Theta}$. In our technique, this is realized by seeking a maximum likelihood estimate (MLE) solution, $\widehat{\boldsymbol{\Theta}}^{i}_{\text{EM}} = \text{argmax}\, p\left(\underline{X} = \hat{\underline{x}}^{i}_{\text{MMSE}}|\boldsymbol{\Theta}\right)$, using a Expectation-Maximization-based method.

The last module involves estimating the probability state $\mathbf{V}$. Specifically, taking mixture density estimate $\widehat{\boldsymbol{\Theta}}^{i}_{\text{EM}}$, and the reconstruction of signal $\widehat{\underline{x}}^{i}_{\text{MMSE}}$ as inputs, a loopy belief propagation based technique is set forth to infer the probability state, while promoting the clustered property.

The above completes the work flow of our technique. The proposed method, CL-SNM-BP, alternates between these modules, and works in an iterative fashion, where at the end of iteration i, the state probability matrix $\widehat{\mathbf{V}}^{i}$, and the parameters of the skew normal mixture $\widehat{\boldsymbol{\Theta}}^{i}_{\text{EM}}$, are fed back to the approximate message passing module, and iteration $i + 1$ starts.

6.3 Approximate Message Passing employing Skew Normal Mixture Prior

In this section, to capture the skewness feature, we employ a finite skew normal density mixture (6.1) as the prior distribution of the signals. Given $\widehat{\mathbf{V}}^{i-1}$ and $\widehat{\boldsymbol{\Theta}}^{i-1}_{\text{EM}}$,

an efficient approximate message passing algorithm is proposed to make inference of the signal by exchanging beliefs between variable nodes $\underline{x}$ and check nodes $\underline{y}$.

It is worthy noticing that, a similar technique can be found in [7], where a two-state normal and skew normal mixture was employed to model signals whose significant coefficients are skewed about the origin $x = 0$. Our work here considers a multi-state skew normal mixture with arbitrary location parameters, and is capable of accommodating varying number of mixture components. Therefore, [7] can be viewed as a special case of our work.

6.3.1 Bayesian Inference by Approximate Message Passing

Approximate message passing [25, 26] is a powerful method enabling efficient and reliable Bayesian inference of the posteriors. Following the notations of Chapter 5, let $\underline{x} = [x_1, \ldots, x_n, \ldots, x_N]^\intercal$ be the variable nodes, and denote $\underline{y} = [y_1, \ldots, y_m, \ldots, y_M]^\intercal$ as check nodes. The marginal posteriors are estimated by iteratively exchanging local beliefs between variable nodes $\underline{x}$ and check nodes $\underline{y}$. Specifically, as Chapter 5, at iteration i, let $\nu^{(i)}_{x_n \to y_m}(x_n)$ denote the message from the variable node x_n to the check node y_m, and $\nu^{(i)}_{y_m \to x_n}(x_n)$ represent the message from the check node y_m to the variable node x_n, where

$$\nu^{(i)}_{x_n \to y_m}(x_n) = \mathcal{N}(x_n; \mu^{(i)}_{x_{nm}}, \sigma^{2\,(i)}_{x_{nm}}), \tag{6.3}$$

$$\nu^{(i)}_{y_m \to x_n}(x_n) = \mathcal{N}(x_n; \mu^{(i)}_{y_{mn}}, \sigma^{2\,(i)}_{y_{mn}}), \tag{6.4}$$

with the mean and variance being evaluated as,

$$\mu^{(i)}_{x_{nm}} = \int_{-\infty}^{\infty} x_n \nu^{(i)}_{x_n \to y_m}(x_n)\, \mathrm{d}x_n, \tag{6.5}$$

$$\sigma_{x_{nm}}^{2\,(i)} = \int_{-\infty}^{\infty} (x_n - \mu_{x_{nm}}^{(i)})^2 \nu_{x_n \to y_m}^{(i)}(x_n)\, \mathrm{d}x_n, \tag{6.6}$$

$$\mu_{y_{mn}}^{(i)} = (y_m - \sum_{t \in [1,\ldots,N]\setminus\{n\}} A_{mt} \mu_{x_{tm}}^{(i)})/A_{mn}, \tag{6.7}$$

$$\sigma_{y_{mn}}^{2\,(i)} = (\sigma_e^2 + \sum_{t \in [1,\ldots,N]\setminus\{n\}} A_{mt}^2 \sigma_{x_{tm}}^{2\,(i)}\ )/A_{mn}^2. \tag{6.8}$$

Combining the skew normal mixture density prior (5.1) and (6.1), the message from x_n to y_m is updated in $(i+1)$-th iteration as,

$$\nu_{x_n \to y_m}^{(i+1)}(x_n) \cong \mathcal{N}(x_n; a_{nm}^{(i)}, b_n^{2\,(i)}) \sum_{k=1}^{K} \lambda_k \mathcal{SN}(x_n; \xi_k, \omega_k, \alpha_k). \tag{6.9}$$

where

$$a_{nm}^{(i)} \triangleq \sum_{u \in [1,\cdots,M]\setminus\{m\}} A_{un} \mu_{y_{un}}^{(i)}, \tag{6.10}$$

$$b_n^{2\,(i)} \triangleq \frac{1}{M} \sum_{u \in [1,\cdots,M]} A_{un}^2 \sigma_{y_{un}}^{2\,(i)} \quad . \tag{6.11}$$

It is noteworthy that (6.9) involves the product of normal density function and skew normal density function, *i.e.,* $\mathcal{N}(x; a_{nm}^{(i)}, b_n^{2\,(i)})\mathcal{SN}(x; \xi_q, \omega_q, \alpha_q)$. A special case of this problem, where the location parameter is fixed to $\xi = 0$, was studied in Ch. 5 for signals that are asymmetrical about the origin $x = 0$. For arbitrary value of ξ, we come up with the following **Lemma 4** and **Lemma 5** to evaluate the corresponding statistics.

Lemma 4 *Denote* $\mathcal{SN}(x; \xi, \omega, \alpha)$ *as the skew normal density with parameters being* (ξ, ω, α)*, and let* $\mathcal{N}(x; a, b^2)$ *be the normal density function with mean value* a *and*

variance b^2*, then the product* $Z(a,b,\xi,\omega,\alpha)\times \mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2)$ *is a probability density function, i.e.,* $Z(a,b,\xi,\omega,\alpha)\int_{-\infty}^{\infty}\mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2)\mathrm{d}x=1$*, with*

$$Z(a,b,\xi,\omega,\alpha)=\frac{\varsigma}{2\phi\left(\frac{a-\xi}{\varsigma}\right)\Phi(\eta)} \tag{6.12}$$

where $\varsigma=\sqrt{b^2+\omega^2}$*,* $\eta=\dfrac{\kappa+h\mu}{\sqrt{1+h^2\sigma^2}}$*,* $h=\dfrac{\alpha}{\omega}$*,* $\kappa=-h\xi$*,* $\mu=\dfrac{a\omega^2+\xi b^2}{\varsigma^2}$*, and* $\sigma^2=\dfrac{b^2\omega^2}{\varsigma^2}$*.*

Proof.

$$\begin{aligned}
&\mathcal{SN}(x;\xi,\omega,\alpha)\mathcal{N}(x;a,b^2) && (6.13)\\
&=\frac{2}{\omega b}\phi\left(\frac{x-a}{b}\right)\phi\left(\frac{x-\xi}{\omega}\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) && (6.14)\\
&=\frac{1}{\pi\omega\sigma}\exp\left(\frac{1}{2\sigma^2}\left(\mu^2-\frac{b^2\xi^2+\omega^2a^2}{\varsigma^2}-(x-\mu)^2\right)\right)\Phi\left(\alpha\frac{x-\xi}{\omega}\right) && (6.15)
\end{aligned}$$

It is noticed that (6.15) involves $\Phi(\alpha\frac{x-\xi}{\omega})$, therefore, applying ***Lemma 1*** of [7], the above ***Lemma 1*** holds. ■

As a direct extension of ***Lemma 3*** in [7], the following ***Lemma 2*** is derived.

Lemma 5 *Let a random variable* X *follows the distribution* $X\sim Z(a,b,\xi,\omega,\alpha)\times\mathcal{N}(X;a,b^2)\mathcal{SN}(X;\xi,\omega,\alpha)$*, then the mean* $\mathrm{E}(X)$ *is given by,*

$$\mathrm{E}(X)=\mu+\zeta\frac{\phi(\eta)}{\Phi(\eta)}, \tag{6.16}$$

and the variance is,

$$\mathrm{Var}(X)=\mu^2+\sigma^2+\rho\zeta\frac{\phi(\eta)}{\Phi(\eta)}-\mathrm{E}^2(X), \tag{6.17}$$

where $\zeta = \dfrac{h\sigma^2}{\sqrt{1+h^2\sigma^2}}$, *and* $\rho = \dfrac{2\mu+\mu h^2\sigma^2-\kappa h\sigma^2}{1+h^2\sigma^2}$.

As a result of **Lemma 5**, and omitting the iteration superscript, (6.9) can be approximated by the normal density as,

$$\nu_{x_n\to y_m}(x_n) \cong \mathcal{N}(\mu_{x_{nm}}, \sigma^2_{x_{nm}}), \tag{6.18}$$

in which

$$\mu_{x_{nm}} = \mathbb{F}(a_{nm}, b_n^2, \mathbf{\Theta}, \mathbf{V}) = C_n \sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}} E_{nk}, \tag{6.19}$$

$$\begin{aligned}\sigma^2_{x_{nm}} &= \mathbb{G}(a_{nm}, b_n^2, \mathbf{\Theta}, \mathbf{V}) \\ &= \sum_{k=1}^{K} p_{nk}(E_{nk}^2 + \ Var_{nk}) - (\sum_{k=1}^{K} p_{nk} E_{nk})^2,\end{aligned} \tag{6.20}$$

where E_{nk} and Var_{nk}^2 can be calculated as (5.14) and (5.15) with corresponding parameters $\kappa_{nm}^{(i)}, \varsigma_n^{(i)}, \xi_k,\ \omega_k$ and α_k of (6.9). It should be noted that, in evaluating the mean and variance of (6.9), instead of using a uniform mixing weight $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]$ for all coefficients, the state probability matrix $\mathbf{V}$ is utilized, where signal coefficients are assigned with non-uniform weights. More specifically, in (6.9), $\underline{\lambda} = [\lambda_1, \ldots, \lambda_K]$ is replaced with $[V_{n1}, \ldots, V_{nK}]$ for signal coefficient x_n, where $n \in [1, \ldots, N]$. Therefore, $p_{nk} = C_n \dfrac{V_{nk}}{Z_{nk}}$, $C_n = (\sum_k \dfrac{V_{nk}}{Z_{nk}})^{-1}$, and Z_{nk} can be calculated in (6.12).

6.3.2 First Order Approximation by Chain Rule and Matrix Operations

The above message updating strategies (6.3), (6.4) and (6.18) enable an approximate *MMSE* solution by tracking $\mathcal{O}(MN)$ messages. To further simplify the belief propagation, we adopt a first order approximation strategy [26], where a variable node x_n sends a uniform message to all check nodes $\underline{y} = [y_1, \ldots, y_M]$. Similarly, a check node y_m sends a uniform message back to all variable nodes $\underline{x} = [x_1, \ldots, x_N]$, after which

only $\mathcal{O}(N)$ messages are needed to be updated in each belief propagation iteration.

It should be noted that the first order approximate strategy involves taking the derivatives of (6.19) with respect to κ_{nm}. As a_{nm} is involved in equations, taking the derivative directly on (6.19) as [7] is complicated, and intractable for varying number of mixture density components K. Therefore, we apply the *Chain Rule*, where the derivative is obtained by decomposing (6.19) into simpler constituent functions, the derivatives of which are then evaluated, and eventually chained together to form the target derivative.

To this end, the following update rules (6.21) to (6.25) are derived,

$$a_{x_n}^{(i)} = \sum\nolimits_{m=1}^{M} A_{mn}\mu_{y_m}^{(i)} + \mu_{x_n}^{(i)}, \tag{6.21}$$

$$\mu_{x_n}^{(i+1)} = \mathbb{F}_n(a_{x_n}^{(i)}, b^{2\,(i)}) = \sum\nolimits_{k=1}^{K} p_{nk}E_{nk}^{(i)}, \tag{6.22}$$

$$\begin{aligned}\sigma_{x_n}^{2\ (i+1)} \ &= \mathbb{G}_n(a_{x_n}^{(i)}, b^{2\,(i)}) \\ &= \sum\nolimits_{k=1}^{K} p_{nk}[(E_{nk}^{(i)})^2 \ + Var_{nk}^{(i)}] - (\sum\nolimits_{k=1}^{K} p_{nk}E_{nk}^{(i)})^2,\end{aligned} \tag{6.23}$$

$$\mu_{y_m}^{(i+1)} = y_m - \sum\nolimits_{n=1}^{N} A_{mn}\mu_{x_n}^{(i)} + \frac{\mu_{y_m}^{(i)}}{M}\sum\nolimits_{n=1}^{N} \mathbb{F}_n^{'}(a_{x_n}^{(i)}, b^{2\,(i)}), \tag{6.24}$$

$$b^{2\,(i+1)} = \hat{\sigma}_e^2 + \frac{1}{M}\sum\nolimits_{n=1}^{N} \sigma_{x_n}^{2\,(i+1)} \ , \tag{6.25}$$

where $\mathbb{F}_n^{'} \triangleq \dfrac{\mathrm{d}\mathbb{F}_n}{\mathrm{d}a_{x_n}}$ and related parameters are calculated as Table 6.1, with iteration i being omitted for simplicity.

At implementation, $\mu_{y_m}^{(1)}$ in (6.21) is initialized at y_m for $m \in [1, \ldots, M]$, and $\mu_{x_n}^{(1)}$ is set to 0 for $n \in [1, \ldots, N]$. Besides, b^2 in (6.22) to (6.24) is initialized at 10^4 for robustness. Additionally, a maximum iteration of 100 is set for the approximate message passing module, and the convergence criteria is set to $\|\underline{\hat{\mu}}^{i+1} - \underline{\hat{\mu}}^{i}\|_2 \leq 10^{-8}$, where $\underline{\hat{\mu}}^{(i)} = [\mu_{x_1}^{(i)}, \ldots, \mu_{x_N}^{(i)}]$.

$$\frac{\mathrm{d}\mathbb{F}_n}{\mathrm{d}a_{x_n}} = [\sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}} E_{nk}] \frac{\mathrm{d}C_n}{\mathrm{d}a_{x_n}} + C_n \sum_{k=1}^{K} V_{nk} \frac{\mathrm{d}\left(E_{nk}/Z_{nk}\right)}{\mathrm{d}a_{x_n}}, \tag{I.1}$$

$$\frac{\mathrm{d}C_n}{\mathrm{d}a_{x_n}} = C_n^2 \sum_{k=1}^{K} \frac{V_{nk}}{Z_{nk}^2} \frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}}, \tag{I.2}$$

$$\frac{\mathrm{d}\left(E_{nk}/Z_{nk}\right)}{\mathrm{d}a_{x_n}} = \frac{1}{Z_{nk}^2} \left(Z_{nk} \frac{\mathrm{d}E_{nk}}{\mathrm{d}a_{x_n}} - E_{nk} \frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}} \right), \tag{I.3}$$

$$\frac{\mathrm{d}E_{nk}}{\mathrm{d}a_{x_n}} = \frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}} + \zeta_{nk} \frac{\mathrm{d}\left(\phi(\eta_{nk})/\Phi(\eta_{nk})\right)}{\mathrm{d}a_{x_n}}, \tag{I.4}$$

$$\frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}} = \frac{\omega_k^2}{b^2 + \omega_k^2}, \tag{I.5}$$

$$\frac{\mathrm{d}\left(\phi(\eta_{nk})/\Phi(\eta_{nk})\right)}{\mathrm{d}a_{x_n}} = \frac{\mathrm{d}\phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \Phi^{-1}(\eta_{nk}) - \frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \frac{\phi(\eta_{nk})}{\Phi^2(\eta_{nk})}, \tag{I.6}$$

$$\frac{\mathrm{d}\phi(\eta_{nk})}{\mathrm{d}a_{x_n}} = -\eta_{nk}\phi(\eta_{nk}) \frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}}, \tag{I.7}$$

$$\frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} = \phi(\eta_{nk}) \frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}}, \tag{I.8}$$

$$\frac{\mathrm{d}\eta_{nk}}{\mathrm{d}a_{x_n}} = \frac{h_k}{\sqrt{1 + h_k^2 \sigma_{nk}^2}} \frac{\mathrm{d}\mu_{nk}}{\mathrm{d}a_{x_n}}, \tag{I.9}$$

$$\delta_{nk} = \frac{a_{x_n} - \xi_k}{\sqrt{b^2 + \omega_k^2}}, \tag{I.10}$$

$$\tau_{nk} = -\frac{1}{2} \sqrt{b^2 + \omega_k^2} \left(\phi(\delta_{nk})\Phi(\eta_{nk})\right)^{-2}, \tag{I.11}$$

$$\frac{\mathrm{d}Z_{nk}}{\mathrm{d}a_{x_n}} = \tau_{nk} \left(\frac{\mathrm{d}\phi(\delta_{nk})}{\mathrm{d}a_{x_n}} \Phi(\eta_{nk}) + \frac{\mathrm{d}\Phi(\eta_{nk})}{\mathrm{d}a_{x_n}} \phi(\delta_{nk}) \right), \tag{I.12}$$

$$\frac{\mathrm{d}\phi(\delta_{nk})}{\mathrm{d}a_{x_n}} = -\frac{a_{x_n} - \xi_k}{b^2 + \omega_k^2} \phi(\delta_{nk}) \tag{I.13}$$

Table 6.1: Message Passing Parameters

6.4 Parameter Estimation: an Expectation-Maximization approach

In this section, given the current reconstruction of the signal $\hat{\underline{x}}^i_{\mathrm{MMSE}}$ from the approximate message passing module, a novel Expectation-Maximization based algorithm is designed to learn the underlying parameters $\mathbf{\Theta}$ that specifying the mixture.

6.4.1 Learning the Parameters

In our technique, the mixture density $\mathbf{\Theta}$ is obtained by seeking a *MLE* solution, $\widehat{\mathbf{\Theta}}^i_{\mathrm{EM}} = \operatorname{argmax} p\left(\underline{X} = \hat{\underline{x}}^i_{\mathrm{MMSE}} | \mathbf{\Theta}\right)$, using a Expectation-Maximization based method.

For ease of derivation, in estimating the density parameters, it is assumed that signal coefficients are jointly independent. Therefore, the log-likelihood function can

be written as,

$$\ln p(\underline{\hat{x}}|\underline{\lambda}, \boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k) \right\} + \pi \left(\sum_{k=1}^{K} \lambda_k - 1 \right), \tag{6.26}$$

where the last term comes from the constraint $\sum_{k=1}^{K} \lambda_k = 1$, and π is a *Lagrange* multiplier.

Taking the derivative of (6.26) with respect to the mixing weight λ_k, and set it to 0, the following is derived,

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \boldsymbol{\Theta})}{\mathrm{d}\lambda_k} = \sum_{n=1}^{N} \frac{SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)}{\sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)} + \pi = 0. \tag{6.27}$$

Meanwhile, let

$$\gamma_{nk} = \frac{\lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)}{\sum_{k=1}^{K} \lambda_k SN(\hat{x}_n|\xi_k, \omega_k, \alpha_k)} \tag{6.28}$$

be the probability[1] of density component k on signal coefficient x_n. Given the above, and multiplying λ_k with (6.27), it is derived that,

$$\pi = -N, \tag{6.29}$$

$$\hat{\lambda}_k = \frac{\sum_{n=1}^{N} \gamma_{nk}}{N}, \tag{6.30}$$

where (6.29) holds due to fact $\sum_{k=1}^{K} \sum_{n=1}^{N} \gamma_{nk} = N$, and $\sum_{k=1}^{K} \lambda_k = 1$.

Besides, denote $\psi_{nk} = \phi\left(\alpha_k \frac{\hat{x}_n - \xi_k}{\omega_k}\right) \Big/ \Phi\left(\alpha_k \frac{\hat{x}_n - \xi_k}{\omega_k}\right)$, and ξ_k can then be updated by taking the derivative of (6.26) with respect to ξ_k, and setting it to 0,

$$\frac{\mathrm{d} \ln p(\underline{\hat{x}}|\underline{\lambda}, \Theta)}{\mathrm{d}\xi_k} = \sum_{n=1}^{N} \gamma_{nk} \left[\frac{\hat{x}_n - \xi_k}{\omega_k^2} - \frac{\alpha_k}{\omega_k} \psi_{nk} \right] = 0. \tag{6.31}$$

[1] Also known as soft responsibility in [89].

Similarly, taking the derivative of (6.26) with respect to ω_k gives

$$\frac{\mathrm{d}\ln p(\underline{\hat{x}}|\underline{\lambda},\Theta)}{\mathrm{d}\omega_k}=\sum_{n=1}^{N}\frac{\gamma_{nk}}{\omega_k^3}[(\hat{x}_n-\xi_k)^2-\omega_k^2-\omega_k\alpha_k(\hat{x}_n-\xi_k)\psi_{nk}], \tag{6.32}$$

and ω_k is updated as

$$\omega_k^2\sum_{n=1}^{N}\gamma_{nk}+\omega_k\alpha_k\sum_{n=1}^{N}\gamma_{nk}\psi_{nk}(\hat{x}_n-\xi_k)-\sum_{n=1}^{N}\gamma_{nk}(\hat{x}_n-\xi_k)^2=0. \tag{6.33}$$

Additionally, α_k can be updated by solving

$$\frac{\mathrm{d}\ln p(\underline{\hat{x}}|\underline{\lambda},\Theta)}{\mathrm{d}\alpha_k}=\sum_{n=1}^{N}\gamma_{nk}\psi_{nk}\frac{(\hat{x}_n-\xi_k)}{\omega_k}=0. \tag{6.34}$$

Therefore, (6.28), (6.31), (6.33) and (6.34) complete one iteration of the Expectation-Maximization update for γ_{nk}, ξ_k, ω_k, and α_k, where $k\in[1,\ldots,K]$, and $n\in[1,\cdots,N]$.

To summarize, our proposed Expectation-Maximization module starts with an initialization $\Theta^{(0)}$ and $\underline{\lambda}^{(0)}=[\lambda_1,\ldots,\lambda_K]$, and alternates between the following Expectation and Maximization steps,

1. *Expectation* step: Given the current mixture parameters $\mathbf{\Theta}^{(i)}$, evaluate the soft responsibility γ_{nk} for $k\in[1,\ldots,K]$, and $n\in[1,\cdots,N]$.

2. *Maximization* step: With updated soft responsibility, for $k\in[1,\ldots,K]$, re-estimate ξ_k, ω_k, and α_k using (6.31), (6.33), and (6.34), respectively.

where as in [90, 91], parameters are updated sequentially in our proposed method.

It should be pointed out that the learning rules (6.31), (6.33), and (6.34) for ξ_k, ω_k, and α_k are not in closed forms, and thus the solutions cannot be calculated explicitly. In this case, one can take advantage of *root-finding* routines, including Golden Section, Newton's method, or Secant's Method [92], to solve for the solution.

6.4.2 Approximate ψ_{nk} using piecewise functions

It is worth noticing that the learning rules of ξ_k (6.31), ω_k (6.33), and α_k (6.34) involve evaluating the *inverse mills ratio* [93], $\psi(t) = \dfrac{\phi(t)}{\Phi(t)}$, where $t = \alpha_k \dfrac{x_n - \xi_k}{\omega_k}$, for $k \in [1, \ldots, K]$, and $n \in [1, \cdots, N]$.

Since $\Phi(t) \to 0$ as $t \to -\infty$, the *inverse mills ratio* $\psi(t)$ is evaluated as *Not a Number* (NaN) when the operand goes to extremes, which prevents the *Expectation-Maximization* and *root finding* procedure from updating properly. As a motivating example, $\psi(t)$ is evaluated as *NaN* at $t = -40$, which will cause the root finding procedure terminate before convergence, and thus the correct solution cannot be found.

Given the fact $\psi(t)$ is not an elementary function[2], our strategy is to substitute it with an approximate that allows for reliable and efficient evaluation for all real numbers $t \in \mathbb{R}$.

Inspecting the limit of $\phi(t)/\Phi(t)$ as $t \to -\infty$, and recall the *L'Hospital's* rule [94], the following is derived,

$$\lim_{t \to -\infty} \frac{\phi(t)/\Phi(t)}{t} = \lim_{t \to -\infty} \frac{(\phi(t))'}{(t\Phi(t))'} \tag{6.35}$$

$$= \lim_{t \to -\infty} \frac{(\phi(t))''}{(t\Phi(t))''} \tag{6.36}$$

$$= \lim_{t \to -\infty} \frac{(t^2 - 1)\exp(-t^2/2)}{(2 - t^2)\exp(-t^2/2)} = -1, \tag{6.37}$$

where (6.35) holds due to

$$\lim_{t \to -\infty} t\,\Phi(t) = \lim_{t \to -\infty} \frac{\Phi(t)}{1/t} = \lim_{t \to -\infty} \frac{-t^2}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) = 0, \tag{6.38}$$

[2] $\psi(t) = \phi(t)/\Phi(t)$ is not elementary because the denominator $\Phi(t)$ is not elementary. As [7], evaluating $\psi(t) = \phi(t)/\Phi(t)$ is more than 10 times slower than scaler operations.

and (6.36) holds due to

$$\begin{aligned}\lim_{t\to-\infty}(t\Phi(t))' &= \lim_{t\to-\infty}(\Phi(t)+t\phi(t)) \\ &= \lim_{t\to-\infty}\frac{t}{\sqrt{2\pi}}\exp\left(\frac{-t^2}{2}\right)=0. \end{aligned} \tag{6.39}$$

Meanwhile, taking the limit of $\psi(t)$ as $t\to+\infty$ gives,

$$\lim_{t\to+\infty}\ \frac{\phi(t)}{\Phi(t)}=\frac{\phi(t)}{1}=\phi(t). \tag{6.40}$$

The above limits suggest that $\psi(t)$ is asymptotically equivalent to $-t$, and $\phi(t)$, in the limit of $t\to-\infty$, and $t\to+\infty$, respectively. Therefore, a plausible approximate of $\psi(t)$ can be formed by joining an affine function, and a normal *pdf* function. To be more specific, it is intended to approximate $\psi(t)$ by $\widehat{\psi}(t)$ as,

$$\widehat{\psi}(t)=\begin{cases} a_1t+a_2, & \text{if } t\le\Delta \\ c_0\phi(\dfrac{t-\mu_0}{\sigma_0}), & \text{if } t>\Delta \end{cases} \tag{6.41}$$

where Δ is the boundary dividing the domain, a_1 and a_2 are the parameters defining the affine function, and c_0,μ_0,σ_0 are the corresponding parameters specifying the scaled normal *pdf* function.

We adopt a numerical approach, where the goal is set to solve for the approximate $\widehat{\psi}(t)$ by fitting (6.41) to the samples of $\psi(t)=\phi(t)/\Phi(t)$. Since the approximate (6.41) is not piecewise linear, finding the optimal parameters $(\Delta,a_1,a_2,c_0,\mu_0,\sigma_0)$ is intractable [95]. To this end, an effective *k-means* [89] based greedy algorithm is designed in ***Algorithm 2*** to find the parameters of (6.41).

Algorithm 2 starts with a *pre-partition* step, and is followed by a loop that alternates between *piecewise fitting* and *re-partition* steps. In *pre-partition*, a set of evenly spaced sampling points $\underline{\delta}=[\delta_1,\ldots,\delta_q]$ are drawn from the interval $[\delta_-,\delta_+]$,

Algorithm 2: Approximating $\psi(t) = \phi(t)/\Phi(t)$ by a piecewise function

Initialize: $\Delta^{(0)} = 10^3$, $\Delta^{(1)} = -2$, $\epsilon = 10^{-4}$, $tol = 10^{-8}$, $\delta_- = -30$, $\delta_+ = 30$, $I_{max} = 100$, and $i = 1$
Algorithm:
Pre-partition:

1. Build the sampling vector $\underline{\delta} = [\delta_1, \ldots, \delta_q]$ by drawing samples evenly from the interval $[\delta_-, \delta_+]$, with a step ϵ
2. Split $\underline{\delta}$ as $\underline{\delta}_l$ and $\underline{\delta}_u$ at the boundary Δ^1, such that $[\underline{\delta}_l, \underline{\delta}_u] = \underline{\delta}$, and $v \leq \Delta^1 < w$ holds for $v \in \underline{\delta}_l$, $w \in \underline{\delta}_u$.
3. Build the regressands vectors $\underline{\psi}_l^{(1)}$ and $\underline{\psi}_u^{(1)}$ by applying $\psi(t)$ to $t \in \underline{\delta}_l$, and $t \in \underline{\delta}_u$, respectively.

while $i \leq I_{max}$ and $|\Delta^{(i)} - \Delta^{(i-1)}| \leq tol$, **do**
4) Fit affine function $a_1 t + a_2$ to $\underline{\psi}_l^{(i)}$,

$$[\hat{a}_1^{(i)}, \hat{a}_2^{(i)}] = \text{fit}(\underline{\psi}_l^{(i)})$$

5) Fit scaled normal *pdf* function $c_0\phi(\frac{t-\mu_0}{\sigma_0})$ to $\underline{\psi}_u{}^{(i)}$,

$$[\hat{c}_0{}^{(i)}, \hat{\mu}_0{}^{(i)}, \hat{\sigma}_0{}^{(i)}] = \text{fit}(\underline{\psi}_u{}^{(i)})$$

6) Find the intersection t^* of two fitted functions by solving,

$$\hat{a}_1^{(i)} t^* + \hat{a}_2^{(i)} = \hat{c}_0{}^{(i)} \phi\left(\frac{t^* - \hat{\mu}_0{}^{(i)}}{\hat{\sigma}_0{}^{(i)}}\right),$$

and update the boundary $\Delta^{(i+1)} = t^*$
7) Update $\underline{\psi}_l^{(i+1)}$ and $\underline{\psi}_u^{(i+1)}$ as of the steps in Pre-partition using the boundary $\Delta^{(i+1)}$
8) $i = i + 1$
end while
Return: $\Delta = \Delta^{(i)}$, $a_1 = \hat{a}_1^{(i)}$, $a_2 = \hat{a}_2^{(i)}$, $c_0 = \hat{c}_0^{(i)}$, $\mu_0 = \hat{\mu}_0^{(i)}$, and $\sigma_0 = \hat{\sigma}_0^{(i)}$.

with a step size ϵ. Subsequently, $\underline{\delta}$ is split at the boundary Δ into two vectors as $\underline{\delta}_l$ and $\underline{\delta}_u$, where $[\underline{\delta}_l, \underline{\delta}_u] = \underline{\delta}$, and $v \leq \Delta < w$ holds for $v \in \underline{\delta}_l$, $w \in \underline{\delta}_u$. Additionally, applying $\psi(t)$ to elements of $\underline{\delta}_l$, and $\underline{\delta}_l$, leads to the regressands $\underline{\psi}_l^{(1)}$, and $\underline{\psi}_u^{(1)}$, respectively.

To find the parameters of the approximate, at iteration i, $\underline{\psi}_l^{(i)}$ and $\underline{\psi}_u^{(i)}$ are fitted by the affine function, and normal function (6.41), respectively. In Matlab [78], the *least square error* fits of (6.41) can be obtained by calling *polyfit* and *fit* functions, leading to $\hat{a}_1^{(i)} t + \hat{a}_2^{(i)}$, and $\hat{c}_0{}^{(i)} \phi\left(\frac{t - \hat{\mu}_0{}^{(i)}}{\hat{\sigma}_0{}^{(i)}}\right)$, correspondingly.

Moreover, the intersection of two fitted functions can be found by solving for t^*

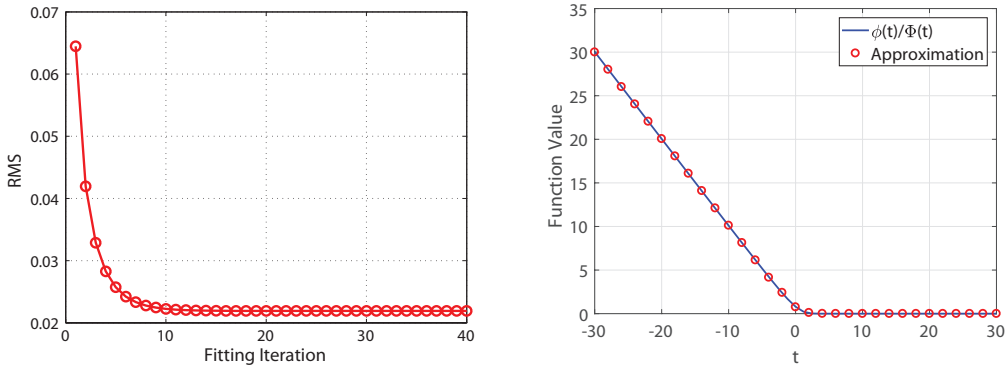
of the following,

$$\hat{a}_1^{(i)} t^* + \hat{a}_2^{(i)} = \hat{c}_0^{(i)} \phi\left(\frac{t^* - \hat{\mu}_0^{(i)}}{\hat{\sigma}_0^{(i)}}\right). \tag{6.42}$$

The above completes one iteration of the *piecewise fitting* step. At iteration $i+1$, the data is re-partitioned by setting the boundary to the intersection of two fitted functions, *i.e.*, $\Delta^{(i+1)} = t^*$, and the loop continues until the convergence of the boundary.

The fitted results utilizing ***Algorithm 2*** are shown in Fig. 6.3, where for numerical stability and efficiency, the interval $[\delta_l, \delta_u]$ is fixed to a limited range with $\delta_l = -30$, $\delta_u = 30$, and the sampling step is set to $\epsilon = 10^{-4}$.

Let Root Mean Square (RMS) of a vector $\underline{\epsilon} \in \mathbb{R}^n$ be $\underline{\epsilon}_{rms} = \sqrt{\frac{1}{N}(\epsilon_1^2 + \epsilon_2^2 + \cdots \epsilon_n^2)}$. As can been seen in Fig. 6.3a, the RMS of the fit error gradually decreases as the iteration increases, and eventually converges to RMS $= 0.022$, where the parameters are found to be $\Delta = -3.1727, a_1 = -0.994, a_2 = 0.1795, c_0 = 8.944, \mu_0 = -4.0153$, and $\sigma_0 = 2.2836$. Moreover, as can be seen in Fig. 6.3b, the approximate (6.41) resembles the $\psi(t) = \phi(t)/\Phi(t)$ quite decently.



(a) (b)

Figure 6.3: Fit piecewise function to $\phi(t)/\Phi(t)$. (a) Root Mean Square (RMS) Errors of Fit. (b) Comparison of $\psi(t) = \phi(t)/\Phi(t)$ and its piecewise approximate $a_1 t + a_2$ for $t \leq \Delta$, and normal function $c_0\phi(\frac{t-\mu_0}{\sigma_0})$ for $t > \Delta$, where $\Delta = -3.1727, a_1 = -0.994, a_2 = 0.1795, c_0 = 8.944, \mu_0 = -4.0153$, and $\sigma_0 = 2.2836$.

6.4.3 Initialization Strategy

It is worth noticing that, as *Expectation-Maximization* only finds local optimums, a good initialization strategy is critical in building an effective parameter estimation procedure. In our work, given the number of mixture components K, the parameters are initialized by matching the moments of mixture component.

Specifically, the coefficients of current estimate $\underline{\hat{x}}$ is divided into K groups, $\underline{\hat{x}} = [\underline{\hat{x}}_1, \ldots, \underline{\hat{x}}_K]$, by utilizing *k-means* algorithm [89].

Additionally, given the K clusters, the parameters for each density component is initialized in a way where sample mean, variance, and skewness match the population mean, variance, and skewness, respectively. Concretely, denote m_k as sample mean, ${v_k}^2$ as sample variance, and g_k as sample skewness, respectively. Then the location, scale, and shape parameters of skew normal density component k is initialized at ξ_k, ω_k, and α_k by solving,

$$m_k = \xi_k + \omega_k \frac{\alpha_k}{\sqrt{\pi(1+\alpha_k^2)/2}}, \tag{6.43}$$

$${v_k}^2 = \omega_k^2 \left(1 - \frac{2\alpha_k^2}{\pi(1+\alpha_k^2)}\right), \tag{6.44}$$

$$\left|\frac{\alpha_k}{\sqrt{1+\alpha_k^2}}\right| = \left(\frac{\pi}{2}\frac{|g_k|^{\frac{2}{3}}}{|g_k|^{\frac{2}{3}} + ((4-\pi)/2)^{\frac{2}{3}}}\right)^{\frac{1}{2}}, \tag{6.45}$$

where the sample skewness g_k is capped to a maximum absolute value of 0.95 for numerical stability, and the sign of α_k is same as g_k.

6.4.4 Estimate the number of density components K

Selection of the number of components K is fundamental for techniques utilizing mixture model, and a variety of methods have been proposed to develop effective way for estimating K. In our work where the mixture component is skew normal, a non-parametric method is developed, where the number of components is estimated

based on the modality of the kernel density estimate.

Specifically, given the signal coefficients, $\underline{\hat{x}} = [\hat{x}_1, \ldots, \hat{x}_N]$, a kernel $U : \mathbb{R} \rightarrow \mathbb{R}_+$, is placed at sample point $t \in \mathbb{R}$, and each signal coefficient $\hat{x}_n \in \underline{\hat{x}}$ contributes a non-negative density mass $U(t - \hat{x}_n)$. Utilizing the Gaussian kernel $U(t) = \phi(t)$, the density at sample point $t \in \underline{t}$, can be estimated by summing up the normalized contributions from all coefficients as,

$$\hat{f}(t) = \frac{1}{NW} \sum_{n=1}^{N} \phi\left(\frac{t - \hat{x}_n}{W}\right), \tag{6.46}$$

where $\underline{t} = [t_1, \ldots, t_L]$ is a vector of $L = 200$ evenly spaced sampling points drawn in the range of $\underline{\hat{x}}$, and W is the bandwidth that controls the spread of the density mass, and ultimately the smoothness of the density estimate.

It should be noted that the kernel density estimate found by (6.46) is highly sensitive to the choice of bandwidth W, where a large value leads to an *over-smoothed* estimate that under-fits the real density, and a small value makes the estimate *under-smoothed*, and over-fits the real one. Therefore, a proper value of W is a good balance of *under-smoothing* and *over-smoothing*, where a well-behaved W is generally set manually by cross validation procedures.

In our work, the problem is tackled by a robust *two-stage* procedure. In the first place, the kernel density is estimated as (6.46), where the bandwidth is set to $W = 0.05$ to pick up the local variability of the density. Subsequently, a Gaussian weighted moving average filter [78] is followed as the second stage to capture the overall modality of the underline density, *i.e.,*

$$\hat{f}_g(t) = \sum_{j=1}^{W_f} \hat{f}(t - j + 1)V(j), \tag{6.47}$$

where $V(i) = exp(\frac{-i^2}{2\sigma_f^2})$ is the Gaussian smoothing kernel, with window size $W_f = 10$,

and standard deviation $\sigma_f = 0.2 \times W_f = 2$. It is found out that although a good choice of W, W_f and σ_f are problem dependent, the above settings work decently in practice.

Given the above, the number of components K is estimated by counting the number of modes, *i.e.,* $\hat{f}_g(i-1) < \hat{f}_g(i) < \hat{f}_g(i+1)$ for $i \in [1, \ldots, L]$. In Matlab [78], this can be obtained by calling the function *findpeaks*.

6.4.5 Evaluations of Parameter Estimation

Fig. 6.4 is a demonstration of the proposed *Expectation-Maximization* based mixture density estimation. To test the effectiveness of the module, a signal $\underline{x}$ is generated by drawing $N = 2000$ random samples from a mixture of $K = 4$ skew normal density components, with the parameters being shown in Table 6.2. Specifically, the insignificant coefficients of $\underline{x}$ are generated from skew normal density with parameter $\underline{\theta}_1$. Besides, the significant coefficients are generated from $\underline{\theta}_2$, $\underline{\theta}_3$, and $\underline{\theta}_4$. The mixing weights are set to $\lambda_1 = 0.7$, $\lambda_2 = 0.1$, $\lambda_3 = 0.1$, and $\lambda_4 = 0.1$, respectively. The density of significant coefficients is plotted in Fig. 6.4a as solid line.

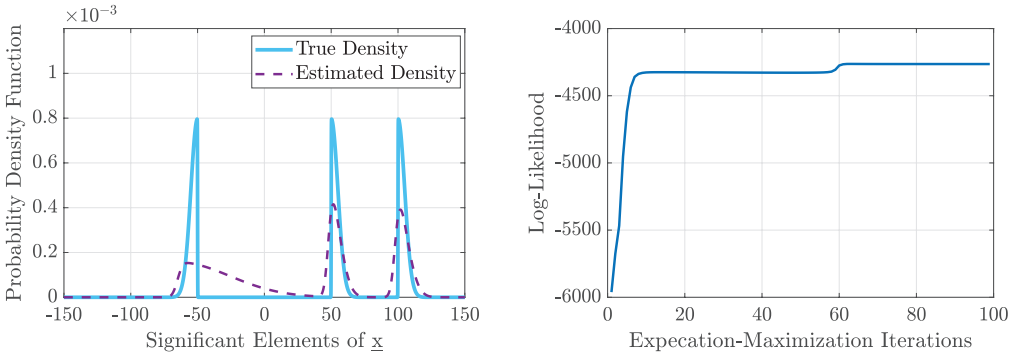
The signal $\underline{x}$ is then sampled by (2.9), *i.e.,* $\underline{y} = \mathbf{A}\underline{x} + \underline{e}$, with length of $\underline{y}$ being set to $M = 1650$. Meanwhile, the measurement white Gaussian noise $\underline{e}$ is added such that $\text{SNR} = 10\log_{10}(\frac{\|\mathbf{A}\underline{x}\|}{\|\underline{e}\|}) = 30$ dB, where $\|\underline{e}\| = \sum_{m=1}^{M} |e_m|^2$. Additionally, the signal reconstruction $\hat{\underline{x}}$ is obtained by employing the proposed signal inference module with an uninformative prior.

The proposed *Expectation-Maximization* module is applied to $\hat{\underline{x}}$ to estimate the mixture density, with the maximum iteration being set to 100. The log-likelihood of each iteration is tracked by evaluating (6.26), where convergence is reached when the consecutive difference of log-likelihood $\leq 10^{-6}$. Besides, the parameters found at each iteration are tracked, and the proposed module returns the one that leads to maximum log-likelihood as the solution.

As can be seen in Fig. 6.4b, the log-likelihood of the density estimate improves gradually as the iteration increases, and eventually converges with a gain of 1697.3. The estimated significant densities are plotted in Fig. 6.4a as dashed line. The true and estimated density parameters are compared at Table 6.2. As can be seen, the proposed module recovered the number of mixture components as $\hat{K} = 4$ precisely. Besides, although deviated mildly in $\hat{\underline{\theta}}_2$, our technique faithfully recovered the overall modality, and skewness of the signal.

Table 6.2: True and Estimated Parameters

	Density Parameters			Weight
$\underline{\theta}_1$	$[\ \xi_1 = 0,$	$\omega_1 = 1,$	$\alpha_1 = 0\ ]$	$\lambda_1 = 0.7$
$\hat{\underline{\theta}}_1$	$[\ \hat{\xi}_1 = 0,$	$\hat{\omega}_1 = 0.19,$	$\hat{\alpha}_1 = 0\ ]$	$\hat{\lambda}_1 = 0.65$
$\underline{\theta}_2$	$[\ \xi_2 = -50,$	$\omega_2 = 5,$	$\alpha_2 = -50\ ]$	$\lambda_2 = 0.1$
$\hat{\underline{\theta}}_2$	$[\ \hat{\xi}_2 = -65,$	$\hat{\omega}_2 = 39.05,$	$\hat{\alpha}_2 = 12.73\ ]$	$\hat{\lambda}_2 = 0.15$
$\underline{\theta}_3$	$[\ \xi_3 = 100,$	$\omega_3 = 5,$	$\alpha_3 = 50\ ]$	$\lambda_3 = 0.1$
$\hat{\underline{\theta}}_3$	$[\ \hat{\xi}_3 = 97.85,$	$\hat{\omega}_3 = 8.38,$	$\hat{\alpha}_3 = 2.96\ ]$	$\hat{\lambda}_3 = 0.1$
$\underline{\theta}_4$	$[\ \xi_4 = 50,$	$\omega_4 = 5,$	$\alpha_4 = 50\ ]$	$\lambda_4 = 0.1$
$\hat{\underline{\theta}}_4$	$[\ \hat{\xi}_4 = 47.90,$	$\hat{\omega}_4 = 7.92,$	$\hat{\alpha}_4 = 3.02\ ]$	$\hat{\lambda}_4 = 0.1$



(a) (b)

Figure 6.4: Expectation-Maximization Mixture Density Estimate. (a) True and estimated mixture density of the significant coefficients. (b) Log-likelihood evaluated at Expectation Maximization iterations.

6.5 States Estimation using Belief Propagation and Potts Model

Given the reconstruction of the signal $\hat{\underline{x}}^i_{MMSE}$, and the estimated mixture density parameters $\hat{\mathbf{\Theta}}^i_{EM}$, in this section, we are aiming to promote the clustered property, and take inference of the underlining hidden states $\mathbf{S}$, by estimating the state probability matrix $\mathbf{V}$.

We approach the task by modelling the clustered property using the *Potts* model [35], where neighboring hidden state pairs are encouraged to be consistent, through the regularization of the compatibility function. A belief propagation based technique is then employed to infer the hidden states, and exploit clustered property by exchanging local beliefs.

6.5.1 Potts Model

In this work, a K-state *Potts* model is considered. Specifically, let $S_{i,j} \in [1, \ldots, K]$ be the hidden state variable of signal coefficient $X_{i,j}$, and $1 \leq i, j \leq d$. Besides, S_n and X_n correspond to $S_{i,j}$ and $X_{i,j}$ respectively, with the transform $n = (i-1) \times d + j$, and $1 \leq n \leq N$.

Borrowing the terminology from *Statistical Mechanics*, the energy of a hidden state configuration $\mathbf{S} = \mathbf{s} \in [1, \ldots, K]^N$ is defined as [56],

$$E(\mathbf{s}) = -\sum_{\langle u,v \rangle} J_0(s_u, s_v) - \sum_{n=1}^{N} H_0(s_n, \hat{x}_n), \tag{6.48}$$

where $J_0(s_u, s_v)$ is the *interaction* function that measures the consistency of neighboring hidden state pairs, $H_0(s_n, \hat{x}_n)$ is the *field* function that quantifies the coherence between estimated signal coefficients, and the corresponding hidden states, and $\langle u, v \rangle$ denotes neighboring pairs.

Subsequently, denote $J(s_u, s_v) = \exp(J_0(s_u, s_v))$ as the *compatible* function, and let $H(s_n, \hat{x}_n) = \exp(H_0(s_n, \hat{x}_n))$ be the *evidence* function, the joint probability func-

tion of a hidden states $\mathbf{s}$ can be evaluated by *Boltzmann's law* as [56],

$$\begin{aligned} P(\mathbf{s}) =& \frac{1}{Z_p} \exp(-E(\mathbf{s})) \\ =& \frac{1}{Z_p} \prod_{\langle u,v \rangle} J(s_u, s_v) \prod_n H(s_n, \hat{x}_n), \end{aligned} \tag{6.49}$$

where Z_p is a normalization constant.

As can be seen from (6.48) and (6.49), *Potts* model can be configured by proper choice of *compatibility* and *evidence* functions[3], such that *compatible* and *evident* hidden state configurations are preferred probabilistically over the *chaotic* counterparts.

6.5.2 Hidden State Inference by Belief Propagation

Given the *Potts* model, our goal is set to build appropriate *compatibility* and *evidence* functions, and then estimate the hidden state s_n for $n \in [1, \ldots, N]$, by computing the corresponding marginal probability from the joint probability (6.49). It should be noted that calculating the marginal probability involves summing over all other hidden state nodes, and unless N is very small, exact derivation is intractable in practice.

To this end, belief propagation is utilized to get an approximate estimate of the marginal probability by exchanging local beliefs.[4] Specifically, in the work, each density component of $\widehat{\mathbf{\Theta}}^i_{EM}$ is associated to a value of $s_n \in [1, \ldots, K]$, where the *evidence* $H(s_n, \hat{x}_n)$ is utilized to measure the responsibilities of mixture components on the specific signal coefficient. Therefore, the *evidence* function can be written as a *K-by-1* column vector,

$$\underline{H}_n = \underline{H}(s_n, \hat{x}_n) \cong [H_{n1}, \ldots, H_{nK}]^\intercal, \tag{6.50}$$

[3]Or equivalently, *interaction* function and *field* function.

[4]Similar to the *message* in Sec. slowromancapiii@, belief in this context encodes the marginal probability.

with $H_{nk} = \mathrm{SN}(\hat{x}_n|\hat{\xi}_k, \hat{\omega}_k, \hat{\alpha}_k)$.

Additionally, to promote clustered property, the compatibility function is defined in a way where neighboring pairs are encouraged to take identical hidden state. Therefore, following the vector representation of *evidence* function, the compatibility function is defined accordingly as a *K-by-K* state transition matrix [96],

$$\mathbf{J}^{(t)}(s_u, s_v) = \mathbf{J}^{(t)} = \tau^{(t)}\mathbf{I}_{K\times K} + \upsilon^{(t)}(\mathbf{1}_{K\times K} - \mathbf{I}_{K\times K}), \tag{6.51}$$

where t represents iteration, $\mathbf{I}_{K\times K}$ denotes identity matrix of size *K-by-K*, and $\mathbf{1}_{K\times K}$ represents matrix consisting of all ones. Besides, to promote compatible pairs, the compatibility function is made to be diagonally dominant by setting $\tau^{(t)} \gg \upsilon^{(t)}$, with the constraints $\tau^{(t)} + (K-1)\upsilon^{(t)} = 1$, and $0 \leq \tau^{(t)}, \upsilon^{(t)} \leq 1$.

Given the above, the state probability vector hidden state s_n can be calculated as the of product of corresponding evidence, and all incoming messages as [56, 96],

$$\underline{\hat{b}}_n^{(t)} \cong \underline{H}_n \bullet \prod_{j\in Neighbor(n)}^{\bullet} \underline{m}_{jn}^{(t)}, \tag{6.52}$$

where $\underline{m}_{jn}^{(t)} \in \mathbb{R}^{K\times 1}$ denotes the message sending from s_j to its neighbor s_n, and can be evaluated as,

$$\underline{m}_{jn}^{(t)} \cong \mathbf{J}^{(t)}\left(\underline{H}_n \bullet \prod_{k\in Neighbor(j)\backslash n}^{\bullet} \underline{m}_{kj}^{(t-1)}\right), \tag{6.53}$$

with $\bullet$ representing the *Hadamard* product [97] of vectors[5], and $_{Neighbor(j)\backslash n}$ denoting the set of neighboring nodes s_j except s_n.

At implementation, the messages are initialized non-informatively at

$$\underline{m}_{ij}^{(0)} = [\frac{1}{K}, \ldots, \frac{1}{K}]^{\intercal},$$

[5] *Hadamard* product of two vectors $\underline{a} = [a_1, a_2]^{\intercal}$ and $\underline{b} = [b_1, b_2]^{\intercal}$ gives another vector $\underline{a} \bullet \underline{b} = [a_1 b_1, a_2 b_2]^{\intercal}$.

for all neighboring pairs $\langle i, j \rangle$. The messages are then propagated, and updated asynchronously [96, 98] by iteratively calling the message update rule (6.53) for 3 iterations. Besides, a first order neighborhood system is employed, where the hidden state $S_{u,v}$ statistically interacts with the four adjacent neighbors, *i.e.,* $S_{u,v+1}$, $S_{u,v-1}$, $S_{u+1,v}$, and $S_{u-1,v}$, for $1 \leq u, v \leq d$.

Additionally, the hyper-parameters $\tau^{(t)}$ and $\upsilon^{(t)}$ are set based on the compatibility as,

$$\tau^{(t)} = \frac{r_s^{(t)}}{r_s^{(t)} + r_d^{(t)}}, \tag{6.54}$$

$$\upsilon^{(t)} = \frac{1}{K-1}\big(1 - \tau^{(t)}\big), \tag{6.55}$$

where $r_s^{(t)}$ and $r_d^{(t)}$ are updated with the corresponding momentum, and compatibility measure as,

$$r_s^{(t)} = r_s^{(t-1)} + \frac{\kappa^{(t)}}{\vartheta^{(t)} + \kappa^{(t)}}, \tag{6.56}$$

and

$$r_d^{(t)} = r_d^{(t-1)} + \frac{\vartheta^{(t)}}{\vartheta^{(t)} + \kappa^{(t)}}. \tag{6.57}$$

It should be noted that in the above, the compatibility measures $\kappa^{(t)}$ and $\vartheta^{(t)}$ are evaluated as the number of compatible pairs, and incompatible pairs, respectively, where at iteration t, a pair $\langle u, v \rangle$ are said to be compatible if they have identical dominant state, *i.e., argmax*$\big(\hat{\underline{b}}_u^{(t)}\big)$ = *argmax*$\big(\hat{\underline{b}}_v^{(t)}\big)$, and incompatible otherwised.

6.6 Complexity Analysis

Similar to other approximate message passing based techniques [7, 25, 26], our signal inference module is highly efficient. Concretely, the complexity of the module is dominated by two major operations. The first comes from evaluating (6.21), which when implemented by matrix, leads to the multiplication of a matrix of size $\mathbb{R}^{M\times N}$, with a vector of size $\mathbb{R}^{N\times 1}$. Therefore, a Floating Point Operations ($FLOP$) proportional to $\mathcal{O}(M(2N-1))$ is expected. The second rises from (6.22) and (6.23), which calls for the element-wise product of size $\mathbb{R}^{N\times K}$, leading to a $FLOP$ of $\mathcal{O}(NK)$. As $K \ll M$ holds in practice, the overall FLOP of the approximate message passing module is $\mathcal{O}(M(2N-1))$.

The parameter estimation module involves finding the root of the function consisting of N terms, for each of K density components. Considering the overhead [99] of root finding procedure[6], and the fact that each density component has 3 parameters, a $FLOP$ of $\mathcal{O}(15KN)$ is needed for each Expectation-Maximization iteration.

The state estimation module enjoys great computation efficiency as well. Specifically, as (6.53) involves only element-wise product, a $FLOP$ of $\mathcal{O}(4KN)$ is expected for each iteration of belief propagation, where the leading constant comes from the size of neighborhood.

Therefore, although involving multiple modules, our proposed technique is highly efficient in exploring the salient features of the signals. As a rule of thumb, the time complexity of our proposed technique is estimated to be $\mathcal{O}(M(2N-1))$ $FLOP$.

6.7 Experiments

In this section, the performance of our proposed method is evaluated under a variety of numerical simulations. For each test, the signal $\underline{x}$ is sampled by (2.9), where the

[6] A factor of $\log_2(32) = 5$ is anticipated for root finding procedure using *Newton's* method with a 32 digits precision representation.

coefficients of the sampling matrix $\mathbf{A}$ are drawn from *i.i.d.* Gaussian ensemble, with the columns of $\mathbf{A}$ being normalized to unit ℓ_2 norm, *i.e.,* $A = [\underline{A}_1{}^\intercal, \ldots, \underline{A}_N{}^\intercal]^\intercal$, and $\|\underline{A}_n\|_2 = (\sum_{m=1}^{M} A_{mn}^2)^{\frac{1}{2}}$=1, for $1 \leq n \leq N$.

At the reconstruction phase, the signal is estimated by the proposed technique that alternates between signal inference, mixture density estimate, and hidden state inference modules. The process is executed for a maximum of $i = 4$ iterations, or till the convergence of reconstruction, *i.e.,* $\|\hat{\underline{x}}^i - \hat{\underline{x}}^{i-1}\|_2^2 / \|\hat{\underline{x}}^i\|_2^2 \leq 10^{-4}$.

At iteration $i = 1$, an un-informative setting is adopted, where the mixture is assumed to consist $K = 2$ normal density components, and the parameters are set to $\hat{\mathbf{\Theta}}^0 = [\underline{\theta}_1, \underline{\theta}_2]$, where $\underline{\theta}_1 = [\xi_1 = 0, \omega_1 = 0.5, \alpha_1 = 0]$, and $\underline{\theta}_2 = [\xi_2 = 0, \omega_2 = 50, \alpha_2 = 0]$. Besides, the corresponding mixing weights are assumed to be $\lambda_1 = 0.8$, and $\lambda_2 = 0.2$. At iteration $i = 1$, the state probability matrix is set to $\widehat{\mathbf{b}}^0 = [\underline{b}_1, \ldots, \underline{b}_N]^\intercal$, with $\underline{b}_n = [\lambda_1, \lambda_2]^\intercal$, for $n \in [1, \ldots, N]$. The variance of measurement noise in (6.25) is initialized at $\hat{\sigma}_e^2 = 1$, and can be estimated based on residual as $\hat{\sigma}_e^2 = \frac{1}{M}\|\underline{y} - \mathbf{A} * \hat{\underline{x}}\|_2^2$.

6.7.1 Pictorial Demonstration

As a demonstration, in this test, our proposed technique is examined by reconstructing an artificial signal $\mathbf{x} \in \mathbb{R}^{63\times 63}$ shown in Fig. 6.5a, with the length of the signal being $N = 3969$. The coefficients are drawn from a mixture consisting of $K = 6$ density components shown in Table 6.3, where without loss of generality, $\underline{\theta}_1$ denotes insignificant density component, and $\underline{\theta}_2$ to $\underline{\theta}_6$ represent significant density components.

As can be seen in Fig. 6.5a, the signal $\underline{x}$ consists of $G_s = 13$, *disk*-like significant clusters, with each cluster composing of 69 coefficients. In Table 6.3, the *Weight* of each density component is adjusted by the number of clusters, which are set to 3, 2, 3, 2, and 3, for $\underline{\theta}_2$, $\underline{\theta}_3$, $\underline{\theta}_4$, $\underline{\theta}_5$, and $\underline{\theta}_6$, respectively. The signal is sampled by (2.9), where the number of samples is set to $M = 1794$, and the measurement is noisy with

SNR $= 35\ dB$.

Table 6.3: Mixture Density Parameters

Density Parameters				
$\underline{\theta}_1$	[$\xi_1 = 0$,	$\omega_1 = 0.5$,	$\alpha_1 = 0$	]
$\underline{\theta}_2$	[$\xi_2 = 50$,	$\omega_2 = 20$,	$\alpha_2 = 5$	]
$\underline{\theta}_3$	[$\xi_3 = -50$,	$\omega_3 = 20$,	$\alpha_3 = -5$	]
$\underline{\theta}_4$	[$\xi_4 = 200$,	$\omega_4 = 20$,	$\alpha_4 = -10$	]
$\underline{\theta}_5$	[$\xi_5 = -200$,	$\omega_5 = 20$,	$\alpha_5 = -10$	]
$\underline{\theta}_6$	[$\xi_6 = 300$,	$\omega_6 = 120$,	$\alpha_6 = -10$	]

The signal is then reconstructed by our proposed technique, and Fig. 6.5b, and 6.5c show the reconstruction obtained at 1^{st}, and 4^{th} iteration, respectively. As can be seen in Fig. 6.5b, the reconstruction of 1^{st} iteration missed 5 clusters, and the signal estimate is corrupted by a large number of *salt-and-pepper* noises. After a few iterations, our proposed technique manages to recover all clusters, and as can be seen in Fig. 6.5c, the reconstruction of the last iteration reliably resembles the ground truth of the signal.

The reconstruction error is tracked by evaluating NMSE $\triangleq \frac{1}{N}\|\hat{\underline{x}} - \underline{x}\|_2^2 / \|\underline{x}\|_2^2$ at each iteration, and is plotted in Fig. 6.5d. As can be seen in Fig. 6.5d, our proposed technique faithfully reduces the reconstruction error, which eventually delivers $NSE = 0.0104$ at the last iteration.

6.7.2 Phase Transition

In the second test, the performance of our proposed algorithm is evaluated under the phase transition test. Concretely, the size of the signal is fixed to 54-by-54, with the length $N = 2916$. Besides, M/N is varied from 0.1 to 0.5, at 0.05 intervals. Additionally, for each value of M, the number of significant clusters G_s, is varied from 1 to $\lfloor \frac{M}{d} \rfloor$, at steps of 1, where similar to previous tests, the shape of cluster is *disk*, and each cluster consists 69 coefficients[7]. The signal coefficients are drawn from the

[7] $\lfloor \frac{M}{d} \rfloor$ represents the largest integer $\leq \frac{M}{d}$.

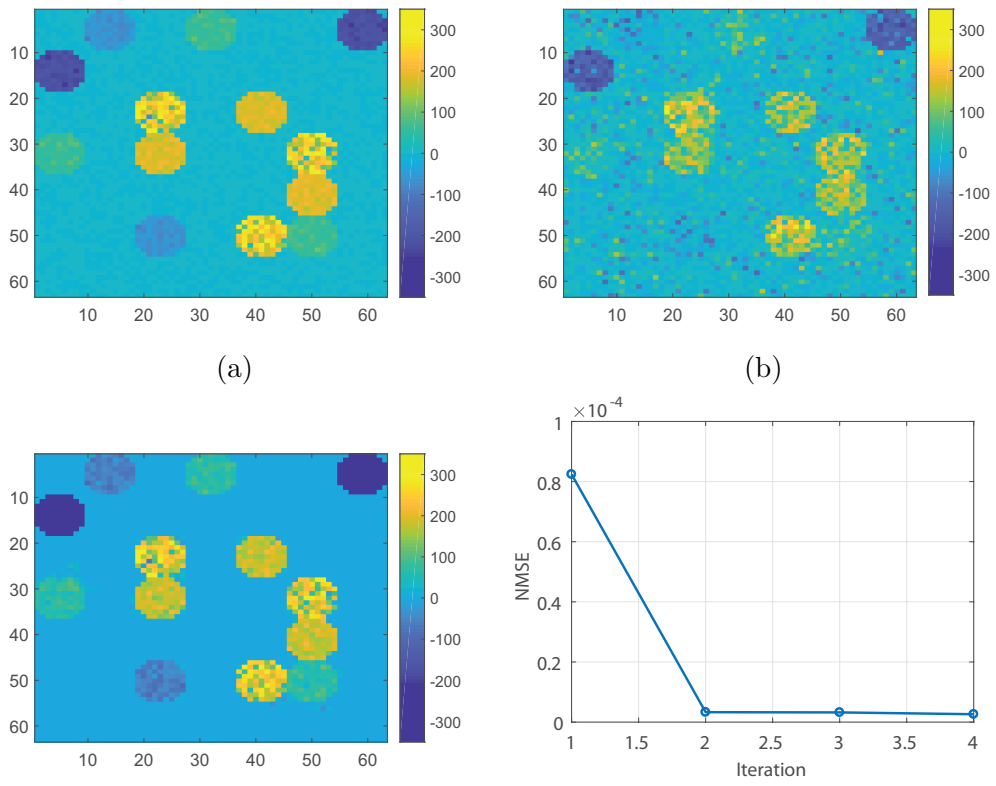


Figure 6.5: Pictorial Demonstration. (a) Ground truth of the signal of size 63-by-63, that consists of $G_S = 13$ significant clusters. (b) Reconstruction at iteration $i = 1$, with $NMSE = 8.24 \times 10^{-5}$. (c) Reconstruction at iteration $i = 4$, with $NMSE = 2.62 \times 10^{-6}$. (d) *NMSE* vs. iterations.

density mixture shown in Table 6.3, where a maximum of 5 significant densities, *i.e.,* $\underline{\theta}_2$ to $\underline{\theta}_6$, are considered. 20 independent trials are performed for each combination of M and G_s, and for each trial, the number of clusters corresponding to each significant density, are generated uniform randomly.

Our proposed method is compared to several sophisticated *structure-aware* methods, including *Struct-OMP* [86], *Turbo-AMP* [27], and *SRL1* [10]. Besides, for completeness, we also include a number of general purpose sparse reconstruction techniques, including *EM-GM-AMP* [30], *SPGL1* [80], *BCS* [55], and *MSBL* [82]. Additionally, our proposed algorithm also compared to *SNAMP* [7] which is designed for asymmetrical sparse signals. It should be noted that, *Struct-OMP* requires the prior

knowledge of the number of significant coefficients. Therefore, for fairness, similar to the setting of our proposed technique, the sparsity in *Struct-OMP* is set to 0.2.

Similar to [79] and [7], success rate is employed to measure the goodness of the methods, and a successful trial is defined as the one with $NMSE \leq 10^{-4}$. The results are summarized in Fig. 6.6, where Q/M vs M/N is depicted, and $Q = 69 \times G_s$ represents the number of significant coefficients. Similar to [79], the area under each curve represents the range at which the corresponding success rate $\geq 50\%$.

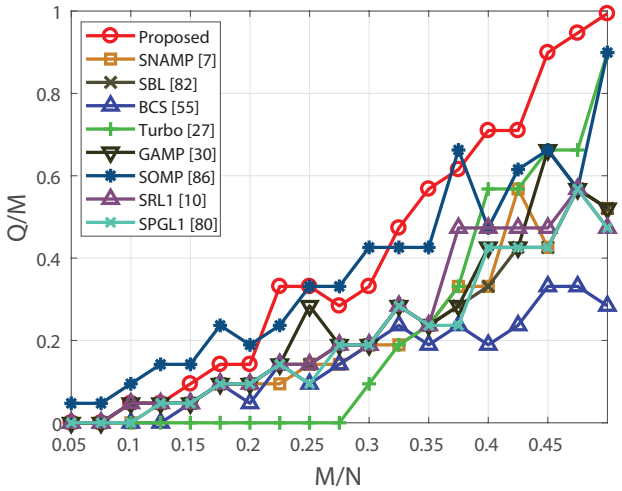


Figure 6.6: Phase Transition tests. The size of significant cluster is set to $d = 69$, and the number of significant coefficients is $Q = 69 \times G_s$. M/N is varying from 0.1 to 0.5 at 0.05 intervals, and Q/M is varying by increasing G_s from 1 to $\lfloor \frac{M}{d} \rfloor$ at steps of 1.

It can be seen in Fig. 6.6 that our proposed method gives competitive results in the phase transition tests. Specifically, our technique is most effective when $M/N > 0.3$. We believe this advantage comes from the fact that mixture estimation requires sizable significant coefficients to be efficient. It is also worthy pointing out that, our proposed technique managed to outperform the *Approximate Message Passing* relatives , *i.e.*, *Turbo-AMP*, *EM-GM-AMP*, and *SN-AMP*, confirming that the proposed technique is highly effective in taking advantage of both clustered property and the skewness features.

6.7.3 Noisy Reconstruction

In this test, our scheme is tested under noisy environments. Specifically, Gaussian random noise $\underline{e}$ is added to the measurements as in (2.9). Similar to *Phase Transition tests*, the size of signal is set to 54-by-54. The signal coefficients are drawn from the density mixture defined in Table 6.3. A total of $G_s = 15$ significant clusters are generated, with each significant density, *i.e.*, $\underline{\theta}_2$ to $\underline{\theta}_6$, contributing 3 clusters.

Fig. 6.7 shows the reconstruction *NMSE* under noisy environments, where *SNR* is varied from 12.5 dB to 30 dB, at 2.5 dB intervals, and each data point is averaged over 200 independent trials. It can be seen from Fig. 6.7 that, our proposed technique CL-SNM-BP gives superior results under varying *SNRs*.

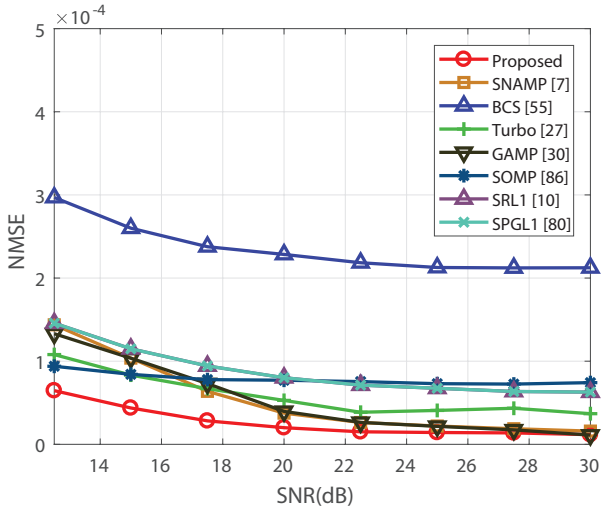


Figure 6.7: NMSE vs. SNR

use same markers as Runtime.

6.7.4 Runtime tests

The time complexity of our proposed algorithm is evaluated by the *Runtime tests*. The size of signal is set to d-by-d, where d varies from 18 to 72, at steps of 9. The shape of significant clusters is *disk*, with each containing 69 coefficients. Besides, the number of significant clusters are fixed to $G_s = 2$, with one cluster drawing

from $\underline{\theta}_3$, and the other sampling from $\underline{\theta}_4$ of Table 6.3. Additionally, the number of measurements is set to $M = 276$.

The experiments are performed on a desktop with hex core 3.2 GHz CPUs, and 16 GB of 1333 MHz memory. 20 independent trials are performed for each value of d, and Fig. 6.8 shows the average runtime of each method as the size of the signal N increases.

It can be seen that, as multiple modules are involved in our proposed technique, the runtime of our scheme is slightly longer than the other approximate message passing relatives, *i.e., Turbo-AMP*, *EM-GM-AMP*, and *SN-AMP*. Yet it should be pointed out that, our proposed algorithm scales decently with the increment of N. Specifically, reconstruction of the signal with $N = 324$ leads to an average runtime of 1.94 seconds, which is then scaled to 12.08 seconds when $N = 5184$.

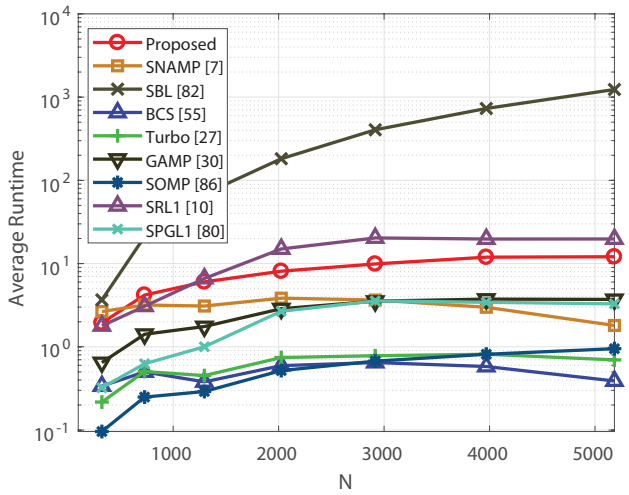


Figure 6.8: Signal Length N vs. Average Runtime (in seconds)

6.7.5 Robustness Test

In this experiment, we are interested in analyzing the robustness of our scheme by feeding signals with density components of different levels of skewness. This is done by generating the signal coefficients from Table 6.4, and varying shape parameters

from -40 to 40, at steps of 10.

The size of the signals is 54-by-54, with $N = 2916$, and $M = 1449$. Besides, $G_s = 12$ significant clusters are generated, with each significant density contributing 6 *disk* clusters of size 69. Our proposed scheme is tested under noisy environments, where *SNRs* is varied from 10 *dB* to 25 *dB*.

The results are summarized in Fig. 6.9, where each data point is averaged over 200 independent trials. It should be noted that, in Fig. 6.9, $\alpha_r = +40$ ($\alpha_r = -40$) represents approximately the positive (negative) half-normal density. On the other hand, $\alpha_r = 0$ resembles the normal density. As can be seen, in general, our proposed technique can adapt to different skewness, and provides robust and consistent reconstruction when the signal is generated from varying shape parameters α_r.

Table 6.4: Robustness Test Mixture Density Parameters

Density Parameters				
$\underline{\theta}_1$	[$\xi_1 = 0,$	$\omega_1 = 0.5,$	$\alpha_1 = 0$	]
$\underline{\theta}_2$	[$\xi_2 = 200,$	$\omega_2 = 20,$	$\alpha_2 = \alpha_r$	]
$\underline{\theta}_3$	[$\xi_3 = -200,$	$\omega_3 = 20,$	$\alpha_3 = \alpha_r$	]

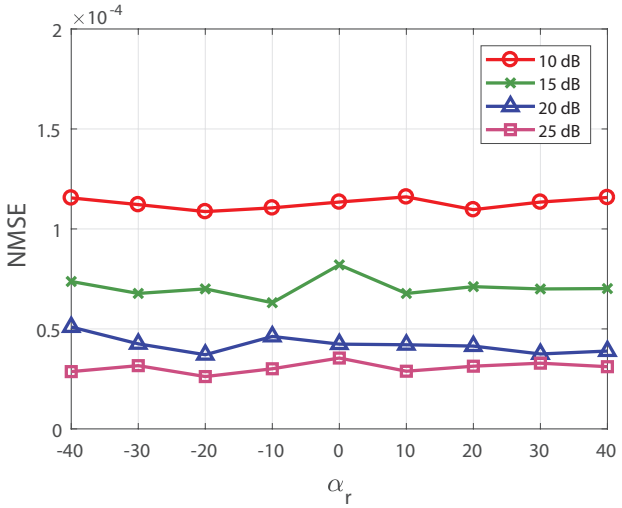


Figure 6.9: NMSE vs. shape parameter α_r

6.8 Conclusion

In this chapter, we investigate the compressive sensing task of clustered sparse signals, where the magnitudes of each significant cluster are distributed asymmetrically $w.r.t$ the cluster mean. To capture the skewness feature, a finite skew normal density mixture is utilized to model the prior distribution of the signal. The clustered property is modelled by the *Potts* model. An effective algorithm is developed to estimate the signal by alternating between exploiting the measurement, drawing inference of the finite skew normal mixture, and taking advantage of the clustered property. Experiments under a variety of settings show that our technique is effective in exploring both the skewness, and the clustered features of the signals.

CHAPTER 7

Eigenvalue-based Cooperative Spectrum Sensing with Finite Samples/Sensors

In this chapter, we study the spectrum sensing problem for cognitive radios. Based on the statistics of the eigenvalues of sample covariance matrix, an effective algorithm is developed to detect the presence of primary user.

7.1 INTRODUCTION

Cognitive Radio [41] is a technique that has the potential to improve the usage of the valuable wireless spectrum. This is achieved by allowing unlicensed users to operate on the licensed spectrums, and therefore vacant spectrum bands can be used more efficiently. One prerequisite of cognitive radio is that the licensed bands are used by unlicensed users, *a.k.a* secondary users, in an opportunistic manner, where unlicensed users shall stop the occupancy and vacate whenever primary users are present. Due to the variance of wireless channels and noise levels, sophisticated spectrum sensing techniques is needed, such that the spectrum can be monitored reliably.

Eigenvalue-based spectrum sensing techniques have drawn lots of attention recently. The major challenge in this field lies in the fact that, exact descriptions of extreme eigenvalues of sample covariance matrix lead to infinite series, and cannot be evaluated explicitly and efficiently. Therefore, existed research [44, 46, 47, 49, 50] mainly focused on the asymptotic or limiting distributions of extreme eigenvalues, which require extremely large numbers of samples and sensors.

In this chapter, we investigate a more realistic region where the sample size or

number of sensors is finite. Exploiting a recent result on multivariate analysis of variance, we derive a new expression for the distribution of the largest eigenvalue of the sample covariance matrix, which is more accurate than existing methods based on asymptotic or limiting distributions. Next, noticing the connection between the Moment Generating Function (MGF) of the distribution of the largest eigenvalue and Lauricella function, compact expressions for the probability density function (pdf), and cumulative distribution function (cdf) of largest eigenvalue of non-central Wishart matrix are derived. These results are further applied to analyse the detection performance of the presence of primary user. Experiments results show the proposed method outperform other eigenvalue based spectrum sensing techniques for finite number of samples and sensors.

The remainder of the paper is organized as follows. The system model and previous results are presented in Sec. 7.2. The distribution of the largest eigenvalue of non-central Wishart matrix is derived in Sec. 7.3. Analysis of the detection performance of Generalized Likelihood Ratio Test using proposed distribution is presented in Sec. 7.4. Simulation results are summarized in Sec. 7.5, and Sec. 7.6 concludes this chapter.

7.2 System Model and Existing Results

7.2.1 System Model

Here we assume a system model similar to [46]. We consider a cooperative detection setting in which K sensors collaborate to detect the presence of a signal, with each sensor taking N samples during the sensing period. Let $y_k(n)$ be the discrete baseband complex sample at receiver k at time n and $\underline{y}(n) = [y_1(n), \ldots, y_K(n)]^\intercal$ denotes the $K \times 1$ received vector at time n. Given this, the task is to differentiate between two hypotheses: under null hypothesis $\mathcal{H}_0$(no primary signal), the received vector

contains only noise

$$\underline{y}(n)|_{\mathcal{H}_0} = \underline{v}(n), \tag{7.1}$$

where $\underline{v}(n) \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_{K\times 1}, \sigma_v^2 I_{K\times K})$ satisfy multivariate circularly symmetric complex Gaussian distribution.

Under alternative hypothesis $\mathcal{H}_1$(presence of primary signal), the received vector consists of both signal and noise

$$\underline{y}(n)|_{\mathcal{H}_1} = \underline{x}(n) + \underline{v}(n) = \underline{h}s(n) + \underline{v}(n), \tag{7.2}$$

where $s(n) \sim \mathcal{N}(0, \sigma_s^2)$ is the transmitted sample of primary user signal, and $\underline{h} = [h_1, \ldots, h_K]^\intercal$ is a $K \times 1$ unknown complex vector, with element h_k representing the channel coefficient associated with sensor k, for $1 \leq k \leq K$.

For simplicity, it is assumed that the channel is constant and memoryless during sensing period. Under $\mathcal{H}_1$, we define the signal to noise ration (SNR) at the receiver as

$$\rho \triangleq \frac{\mathrm{E}\|\underline{x}\|^2}{\mathrm{E}\|\underline{v}\|^2} = \frac{\sigma_s^2 \|\underline{h}\|^2}{K\sigma_v^2}. \tag{7.3}$$

The received sample matrix $\mathbf{Y}$ is a $K \times N$ matrix, with n-th column being $\underline{y}(n)$, *i.e.,*

$$\boldsymbol{Y} \triangleq [\underline{y}(1), \ldots, \underline{y}(N)] = \underline{h} * \underline{s}^\intercal + \boldsymbol{v}, \tag{7.4}$$

where $\boldsymbol{v} \triangleq [\underline{v}(1), \ldots, \underline{v}(N)]$ is a $K \times N$ matrix noise matrix, and $\underline{s} \triangleq [s(1) \ldots s(N)]^\intercal$ is the received signal vector.

Therefore, the sample covariance matrix $\boldsymbol{R}$ can be written as,

$$\boldsymbol{R} \triangleq \frac{1}{N} \boldsymbol{Y}\, \boldsymbol{Y}^H, \tag{7.5}$$

where superscript H represents conjugate transpose, with the non-ascending sequence $\lambda_1 \geq \ldots \geq \lambda_K$ being the eigenvalues of $\boldsymbol{R}$.

In hypothesis testing, due to the fluctuations caused by inherent probabilistic behavior of test statistics, two error events are considered, namely the *probability of false alarm* defined as

$$P_{fa} = Pr(T > t|\mathcal{H}_0) \tag{7.6}$$

and the *probability of detection* defined as

$$P_d = Pr(T > t|\mathcal{H}_1) \tag{7.7}$$

where T is the test statistic used by detector and t is the threshold usually set by probability of false alarm.

We focus on blind detection methods with no prior knowledge regarding the targeted primary signal. Besides, noise level is assumed to be unknown. In this case, several tests can be employed, such as Standard Condition Number detector [44, 47, 49], defined as

$$T_{SCN} = \frac{\lambda_1}{\lambda_K}, \tag{7.8}$$

and Generalized Likelihood Ratio Test (GLRT) [46, 100]:

$$T_{GLRT} = \frac{\lambda_1}{\frac{1}{K}tr(\boldsymbol{R})}. \tag{7.9}$$

Compared to T_{GLRT}, as noticed in [46], T_{SCN} is suboptimal unless $K = 2$. We choose GLRT as our method due to its optimality of statistical power over SCN.

7.2.2 Previous Results

Eigenvalues of Central Wishart Matrix

Under $\mathcal{H}_0$, the receiving sample covariance matrix follows a Central Wishart distribution. By proper centering and scaling, λ_1 and λ_K follow a second-order Tracy-Widom distribution [101] asymptotically as $K, N \to \infty$.

Eigenvalues of non-Central Wishart Matrix

Under $\mathcal{H}_1$, the receiving sample covariance matrix follows a non-central Wishart distribution. Paul [102] and Nadler [45] studied the distribution of the eigenvalues of a non-Central Wishart Matrix, asymptotic in K and N. λ_1 follows Gaussian distribution with,

$$\mathrm{E}[\frac{\lambda_1}{\sigma_v^2}] = (1 + K\rho)(1 + \frac{K-1}{NK\rho}), \tag{7.10}$$

and,

$$\mathrm{Var}[\frac{\lambda_1}{\sigma_v^2}] = \frac{1}{N}(1 + K\rho)^2. \tag{7.11}$$

7.3 LARGEST EIGENVALUE DISTRIBUTION

Due to the difficulty in describing the exact distribution of the largest eigenvalue of a Wishart matrix, computable expressions of the approximated distribution is given by [47, 49, 50, 100] which are asymptotic in K and N. Question raised here whether there is any method to better describe the distribution of λ_1. In this section, exploiting the latest development in MANOVA, we express the approximated distribution of λ_1 in small dimensional setting.

Lemma 6 *In the limit $\sigma_v^2 \to 0$, the distribution of the ratio of largest eigenvalue λ_1 and noise variance σ_v^2 satisfies*

$$\frac{\lambda_1}{\sigma_v^2} \sim \Gamma(\alpha_1, \beta_1) + \Gamma(\alpha_2, \beta_2), \tag{7.12}$$

with expected value and variance:

$$\mathrm{E}[\frac{\lambda_1}{\sigma_v^2}] = \alpha_1\beta_1 + \alpha_2\beta_2, \tag{7.13}$$

$$\mathrm{Var}[\frac{\lambda_1}{\sigma_v^2}] = \alpha_1\beta_1^2 + \alpha_2\beta_2^2, \tag{7.14}$$

where α_1=$N/2$, β_1=$2(1+K\rho)/N$, α_2=$(K-1)/2$ and β_2=$2/N$.

Proof. As discussed in previous section, in the presence of primary user signal the matrix $\boldsymbol{R}$ follows a non-central Wishart distribution. Instead of asymptotic in sample size or the number of sensors, using matrix perturbation and asymptotic in the *SNR*, recent work [51] gives almost accurate approximation under small dimensional setting. It is stated that in the limit $\sigma_v^2 \to 0$, the distribution of the largest eigenvalue λ_1 can be expressed as the weighted sum of two independent Chi-squared random variables:

$$\lambda_1 \sim \frac{1}{N} \cdot ((\sigma_s^2 + \sigma_v^2)\chi_N^2 + \sigma_v^2\chi_{K-1}^2 + O(\sigma_v^2)). \tag{7.15}$$

Based on this, plugging (7.3) into (7.15), we get:

$$\frac{\lambda_1}{\sigma_v^2} \sim \frac{1}{N} \cdot ((1+K\rho)\chi_N^2 + \chi_{K-1}^2). \tag{7.16}$$

Considering the relation between χ^2 and Gamma random variables, **Lemma 6** holds.

■

It is noted that in **Lemma 6**, the distribution of λ_1/σ_v^2 is given as the sum of two independent gamma random variables. Inconsistent with the concise form of (7.12), the exact expression of this distribution is an open problem and always leads to infinite series [103, 104]. Recent work [105] investigates the distribution of the sum of correlated gamma variables. Taking advantages of the connection between the

MGF of targeted function and Confluent form of Lauricella function, the author gives the exact expression of the distribution for the sum of correlated gamma variables. It should be noted that in [105], however, only the case that all gamma random variables having the same shape parameter α has been treated. In the following part, we extend the results to a more general case in which shape parameters α_k are not necessarily the same. By employing the relation between Lauricella function and a certain type of hypergeometric function, we first give the exact distribution of considered statistic that can effective be numerically evaluated.

Lemma 7 *In the limit* $\sigma_v^2 \to 0$*, the* pdf *of the ratio of largest eigenvalue* λ_1 *to noise variance* σ_v^2 *is*

$$f_Q(q) = \frac{q^{\alpha_1+\alpha_2-1}}{\beta_1\beta_2\Gamma(\alpha_1+\alpha_2)} \tag{7.17}$$

$$\times e^{-\frac{q}{\beta_2}} \times {}_1F_1(\alpha_1; \alpha_1+\alpha_2; \frac{q}{\beta_2} - \frac{q}{\beta_1}), \tag{7.18}$$

where ${}_1F_1(;;)$ stands for confluent hypergeometric functions of the first kind.

Proof. Let $Q = \sum_{m=1}^{M} P_m$ be sum of M independent distributed gamma random variables with parameters α_m and β_m. Denote the Laplace transform of an arbitrary function $\Psi(x)$ as $\mathcal{L}[\Psi(x); s]$. The *MGF* of Q is given in [103] as:

$$\mathcal{M}_Q(s) = \mathcal{L}[f_Q(q); -s] = \prod_{m=1}^{M} (1-\beta_m s)^{-\alpha_m}. \tag{7.19}$$

Thus the Laplace transform of the *pdf* of Q can be written as:

$$\mathcal{L}[f_Q(q);s] = \prod_{m=1}^{M}(1+\beta_m s)^{-\alpha_m}\left\{\frac{\prod_{m=1}^{M}\beta_m^{-\alpha_m}}{\Gamma(\sum_{m=1}^{M}\alpha_m)}\right\} \tag{7.20}$$

$$\times\left\{\frac{\Gamma(\sum_{m=1}^{M}\alpha_m)}{s^{\sum_{m=1}^{M}\alpha_m}}\right\}\left\{\prod_{m=1}^{M}(1+\frac{1}{\beta_m s})^{-\alpha_m}\right\}. \tag{7.21}$$

Employing the inverse Laplace transform function for Hypergeometric function, the *pdf* of Q can be expressed as:

$$f_Q(q) = \frac{q^{\sum_{m=1}^{M}\alpha_m-1}}{\prod_{m=1}^{M}(\beta_m)\Gamma(\sum_{m=1}^{M}\alpha_m)} \tag{7.22}$$

$$\times\,\Phi_2^M(\alpha_1,\ldots,\alpha_M;\sum_{m=1}^{M}\alpha_m;-\frac{q}{\beta_1},\ldots,-\frac{q}{\beta_M}), \tag{7.23}$$

which in our case $M=2$, and (7.22) can be written as:

$$f_Q(q) = \frac{q^{\alpha_1+\alpha_2-1}}{\beta_1\beta_2\Gamma(\alpha_1+\alpha_2)} \tag{7.24}$$

$$\times\,\Phi_2^2(\alpha_1,\alpha_2;\alpha_1+\alpha_2;-\frac{q}{\beta_1},-\frac{q}{\beta_2}), \tag{7.25}$$

where in (7.24), $\Phi_2^M(;;)$ stands for the confluent form of Lauricella function and the exact calculation of general confluent form of Lauricella function [106,107] again leads to infinite series. However, in our case, using the result [108]:

$$\Phi_2^2(a,c-a;c;x,-y) = e_1^{-y}F_1(a;c;x+y), \tag{7.26}$$

(7.24) can be further reduced to (7.17). ■

The *cdf* of λ_1/σ_v^2 is given as:

Lemma 8 *In the limit $\sigma_v^2 \to 0$, the* cdf *of the ratio of largest eigenvalue λ_1 and noise variance σ_v^2 is,*

$$F_Q(q) = \frac{q^{\alpha_1+\alpha_2}}{\beta_1\beta_2\Gamma(1+\alpha_1+\alpha_2)} \tag{7.27}$$

$$\times \Phi_2^2(\alpha_1, \alpha_2; 1+\alpha_1+\alpha_2; -\frac{q}{\beta_1}, -\frac{q}{\beta_2}). \tag{7.28}$$

To the best of our knowledge, there is no simple reduction formula for (7.27). Nevertheless, as we will show in Sec. 7.5, approximated expression for the *cdf* works reasonably well under a relaxed condition.

7.4 DETECTION PERFORMANCE EVALUATION

In this section, by using *Theorem 1*, *probability of detection* can be better described under a small dimensional setting.

7.4.1 Threshold Setting

Proper centering and scaling, the distribution of T_{GLRT} under $\mathcal{H}_0$ is similar to the well studied distribution of λ_1 of a central Wishart matrix [46, 47, 49, 50, 100] where T_{GLRT} follows a second-order Tracy-Widom distribution[1] in the joint limit $K, N \to \infty$:

$$Pr[\frac{T_{GLRT} - \mu}{\xi} < s] \to F_{TW2}(s), \tag{7.29}$$

where

$$\mu = [(\frac{K}{N})^{1/2} + 1]^2, \tag{7.30}$$

$$\xi = N^{-2/3}[(\frac{K}{N})^{1/2} + 1][(\frac{K}{N})^{-1/2} + 1]^{1/3}. \tag{7.31}$$

[1]Readers are referred to [101] for definition and calculation of Tracy-Widom distribution $F_{TW2}(s)$.

It should be noted that even though (7.29) is asymptotic in K and N, it has been tested to approximate the real distribution well even with small K and N [101].

Given this and (7.6), an approximate expression for the threshold t is given as

$$t(\alpha) \approx \mu + F_{TW2}(1-\alpha)\xi. \tag{7.32}$$

7.4.2 Probability of detection

As noted in [46], (7.9) can be written as:

$$\frac{\lambda_1}{U} > \tilde{t}(\alpha), \tag{7.33}$$

where

$$U \triangleq \frac{1}{K-1}\sum_{i=2}^{K}\lambda_i, \tag{7.34}$$

$$\tilde{t}(\alpha) = \frac{K-1}{K-t(\alpha)}t(\alpha), \tag{7.35}$$

and U is Gaussian distributed:

$$\mathrm{E}[\frac{U}{\sigma_v^2}] = 1 - \frac{1}{N}\frac{K\rho+1}{K\rho}, \tag{7.36}$$

$$\mathrm{Var}[\frac{U}{\sigma_v^2}] = O(\frac{1}{N^2}). \tag{7.37}$$

Comparing (7.36) and (7.37) with (7.13) and (7.14), we can see that fluctuations of λ_1 is much bigger than those of U. Thus an approximation of (7.33) is:

$$\frac{\lambda_1}{\sigma_v^2} > \tilde{t}(\alpha)\mathrm{E}[\frac{U}{\sigma_v^2}], \tag{7.38}$$

where asymptotically in *SNR*, λ_1/σ_v^2 satisfies the distribution of the sum of two gamma random variables given in **Lemma 6**. Its *pdf* and *cdf* are given in **Lemma 7** and

Lemma 8, respectively. Thus the P_d can be calculated by plugging the right hand side of (7.38) into (7.27). However, as been discussed in previous section, the *cdf* given in (7.27) involves infinite series and is hard to numerically evaluate. It is noted in [51] that, in (7.14), the fluctuation of $\Gamma(\alpha_1, \beta_1)$ is much larger compared to that of $\Gamma(\alpha_2, \beta_2)$ under large *SNR*. Thus the variation of λ_1/σ_v^2 mainly comes from $\Gamma(\alpha_1, \beta_1)$ and it is plausible to approximate $\Gamma(\alpha_2, \beta_2)$ as its mean value. Following this, the probability of detection is:

$$P_d = 1 - F_G(\tilde{t}(\alpha)\mathrm{E}[\frac{U}{\sigma_v^2}] - \alpha_2\beta_2; \alpha_1; \beta_1), \tag{7.39}$$

where

$$F_G(x; a; b) = \frac{\gamma(a, x/b)}{\Gamma(a)} \tag{7.40}$$

is *cdf* of Gamma distributed random variable X with scale parameter b and shape parameter a.

7.5 SIMULATION RESULTS

We first compare the proposed approximated density of λ_1/σ_v^2 given in (7.17) to its empirical density and the classical Gaussian approximation described in (7.10) and (7.11). Simulation are taken with $K = 8$, $N = 10$ and $SNR = 0.75$. As shown in Figure 7.1, the proposed approximation is very close to actual value and can better describe the distribution than classical Gaussian approximation.

Next, in Figure 7.2, we compare the probability of detection of GLRT using the proposed theoretical approximation given in (7.39) to the empirical probability and approximated expression using classical Gaussian distribution. A small size setting where $K = 8$ and $N = 10$ is tested. False alarm rate α is set to 0.005. As can be seen, the proposed theoretical approximation is more accurate than the results given in [46] where the P_d is calculated based on classical Gaussian approximation.

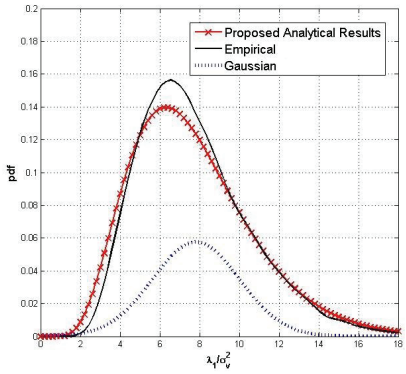


Figure 7.1: Comparison of the density of the ratio of the largest eigenvalue and noise variance. K=8, N=10, and SNR=0.75.

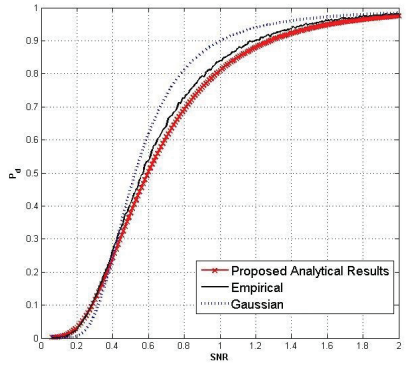


Figure 7.2: Comparison of the detection performance curves as a function of Signal to Noise Ratio (K=8, N=10, $\alpha = 0.005$).

7.6 CONCLUSION

The performance of eigenvalue-based spectrum sensing under a small dimensional setting has been studied in this chapter. Asymptotically in signal to noise ratio, the distribution of the largest eigenvalue of receiving sample covariance matrix has been given as sum of two Gamma random variables. Utilizing the relation of Moment Generating Function and confluent form of Lauricella function, a closed-form expression of the *pdf* has been given. Simulation results show the proposed expression of *pdf* is almost accurate and better than classical Gaussian approximation under a small dimensional setting. Besides, the detection performance using proposed approximation is analyzed. Simulation shows that the proposed method can describe real detection

performance faithfully.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

In this dissertation, we investigated signal recovery and detection tasks utilizing compressive sensing. We designed an effective structural aware reconstruction technique for the compressive sensing task of videos [10]. Next, we investigated the reconstruction task for binary sparse signals, and a novel optimization based algorithm is proposed to exploit both the binary and sparse features [11]. Additionally, compressive sensing for asymmetrical signals are studied [7], and we developed an efficient algorithm that is capable of learning the skewness of the signals, while promoting the sparsity features. Further, sparse reconstruction of clustered sparse signals with asymmetrical features are investigated [36], and a powerful technique is developed to take inference of the signal, estimate the mixture density, and exploit the clustered features. Moreover, eigenvalue based wireless spectrum sensing for cognitive radio is studied [109]. We summarize our contributions and our suggested future research in what follows.

8.1 SRL1: Structured Reweighted ℓ_1 Minimization for Compressive Sampling of Videos

Although ℓ_1 minimization is able to promote the sparsity of signals in compressive sensing tasks, it is incapable of recovering other salient features. To solve the compressive sensing task of videos, we began our efforts by analyzing the difference frames of video sequence. It is found that due to the temporal redundancy of consecutive frames, the difference frames of videos are dominated by clusters formed by non-zero

pixels, with only a few of nonzero pixels being isolated. Therefore, if a pixel in the reconstructed difference frame is zero but is connected to other non-zero pixels, there is a high probability that this pixel is actually non-zero rather than zero. Similarly, if a pixel is non-zero and is isolated, there is a high probability that this pixel is actually zero rather than non-zero.

Noticing this characteristic, a two-step strategy is developed to exploit the clustered features of by refining the signal support of difference frame. The first step involves identifying unrecovered non-zero pixels, and is achieved by exploring the local neighborhoods of recovered non-zero pixels. The second step is designed to eliminate isolated non-zero noises, by analyzing the connectivity of clusters. The refined support estimate is then transformed and served as the weights of the iterative reweighted ℓ_1 minimization scheme.

Our proposed technique reconstructs the difference frame and estimates the signal supports in an iterative fashion, and experimental results show that by exploiting clustered property, isolated non-zero noise can be eliminated, and undiscovered signal coefficients can be recovered.

8.2 Binary Compressive Sensing via Sum of ℓ_1-norm and ℓ_∞-norm Regularization

We considered the task of reconstructing a sparse binary signal vector from a limited number of noisy measurements employing compressive sensing technique. Compared to general sparse signals, a unique feature of this type of signals lies in the fact that signal entries are equally separated with respect to 0.5.

We approached the task based on convex optimization, and a novel regularization term is developed. Concretely, it is known that among the infinite candidates, ℓ_1-norm minimization selects the sparest solution that agrees with the projection. On the other hand, ℓ_∞-norm minimization, favors the representation whose coefficients

are roughly in the same absolute magnitude. We showed that these two extremes can be combined in binary compressive sensing problem to promote the reconstruction quality. This is done by minimizing the sum of the ℓ_1-norm and ℓ_∞-norm, up to a scaling factor and a shifting factor.

The new formulation is convex, and can be solved effectively by many convex optimization operators. Experimental results confirmed that our developed technique is able to promote both the sparsity and binary features of the signals, and outperformed many sophisticated techniques.

8.3 A Framework for Compressive Sensing of Asymmetric Signals using Normal and Skew-Normal Mixture Prior

We investigated the compressive sensing task of sparse signals whose significant coefficients are distributed asymmetrically with respect to zero. We developed a framework utilizing a two-state normal and skew normal mixture density as the prior distribution of the signal, where the significant and insignificant coefficients of the signal are represented by skew normal and normal distributions, respectively. Next, an efficient approximate message passing based algorithm is developed to estimate the signal from its compressed measurements. Further, a fast gradient-based estimator is designed to infer the density of each state.

The performance of our proposed technique is examined under a variety of tests, including phase transition, noisy reconstruction, support set recovery rate, and runtime tests. Our developed technique finds promising applications in real world data set. We show that in weather sensor network application, the disrupting weather phenomena can be successfully learned by our proposed technique. Overall, experimental results show that our technique can effectively exploit the asymmetric feature of the signal, while being competitively efficient in solving large scale problems.

8.4 Compressive Sampling of Clustered Sparse Signals with Asymmetric features

We investigated the compressive sampling task of clustered sparse signals, where the magnitudes of each cluster are distributed asymmetrically *w.r.t* the cluster mean. To address the skewness feature, a finite skew-normal density mixture is utilized to model the prior distribution of the signal. An efficient approximate message passing algorithm, which takes the mixture density, and the hidden states of signal coefficients as inputs, is designed to iteratively derive the estimate of the signal, by propagating local beliefs between the measurements and the signal estimates.

Next, following the approximate message passing module, an Expectation Maximization based algorithm is developed to estimate the mixture density from the estimate of the signal. The number of mixture components is estimated in an efficient and non-parametric way.

Moreover, given the estimate of the signal, and the mixture density estimates, a loopy message passing based algorithm is designed, where the compatibility of neighboring coefficients is regularized by the *Potts* model, after which the hidden states of signal coefficients is estimated, and the clustered property is promoted.

Overall, the proposed technique alternates between exploiting the measurement, drawing inference of the finite mixture model, and taking advantage of the clustered property. These three modules work sequentially and iteratively, after which, a refined reconstruction of the signal can be obtained. Experiments results showed that our technique is highly effective and efficient in exploiting both the clustered feature and asymmetrical feature of the signals, and outperformed many sophisticated techniques.

8.5 Eigenvalue-based Cooperative Spectrum Sensing with Finite Samples/Sensors

In this chapter, we studied the spectrum sensing problem for cognitive radios. Based on the statistics of the eigenvalues of sample covariance matrix, an effective algorithm is developed to detect the presence of primary user.

We derived a new expression for the distribution of the largest eigenvalue of the sample covariance matrix, which is more accurate than existing methods based on asymptotic or limiting distributions. Next, noticing the connection between the Moment Generating Function of the distribution of the largest eigenvalue and Lauricella function, compact expressions for the *pdf*, and *cdf* of largest eigenvalue of non-central Wishart matrix are derived. These results are further applied to analyse the detection performance of the presence of primary user. Experiments results confirmed the proposed method outperform other eigenvalue based spectrum sensing techniques for finite number of samples and sensors.

8.6 Suggestion for Future Research

In this dissertation, we investigated several new research areas in compressive sensing and spectrum sensing. In the following, we summarize potential future research directions.

- In the study of compressive sensing for videos, a fixed threshold is utilized to obtain the signal support from the difference frame. Although this static strategy works reasonably well in a variety of tests, a fixed setting is not necessarily optimal in general. Therefore, optimal threshold value can be potentially investigated for better reconstruction quality.

- Although convex optimization based technique with mixed norm regularization solved the binary compressive sensing tasks decently, it is an interesting exten-

sion to develop a greedy, and faster algorithm to recover the binary and sparse features of the signals.

- In the study of sparse reconstruction of asymmetrical signals, aside from the numerical study, theoretical analysis of the developed technique is an interesting topic for future research.

BIBLIOGRAPHY

[1] E. J. Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] E. J. Candes, "Compressive sampling," *Int. Congress of Mathematics*, 2006.

[4] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag*, vol. 24, no. 4, pp. 118–121, 2007.

[5] Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma, and W. Sun, "Super-resolution based on compressive sensing and structural self-similarity for remote sensing images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 51, no. 9, pp. 4864–4876, 2013.

[6] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang, "Low-dose x-ray ct reconstruction via dictionary learning," *IEEE Trans. Medical Imaging*, vol. 31, no. 9, pp. 1682–1697, 2012.

[7] S. Wang and N. Rahnavard, "A framework for compressive sensing of asymmetric signals using normal and skew-normal mixture prior," *IEEE Trans. Commun*, vol. 63, no. 12, pp. 5062–5072, 2015.

[8] A. Talari and N. Rahnavard, "Cstorage: Distributed data storage in wireless sensor networks employing compressive sensing," in *Proceedings of IEEE Global Telecommun. Conf*, pp. 1–5, 2011.

[9] S. Pudlewski, A. Prasanna, and T. Melodia, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *IEEE Trans. Mobile Comput*, vol. 11, no. 6, pp. 1060–1072, 2012.

[10] S. Wang, B. Shahrasbi, and N. Rahnavard, "SRL1: Structured reweighted l1 minimization for compressive sampling of videos," *in Proceedings of Int. Symp. Inf. Theory*, pp. 301–305, 2013.

[11] S. Wang and N. Rahnavard, "Binary compressive sensing via sum of L-1 norm and L-infinity norm regularization," *in Proceedings of IEEE Military Commun. Conf*, pp. 1616–1621, 2013.

[12] U. Nakarmi and N. Rahnavard, "BCS: Compressive sensing for binary sparse signals," in *Proceedings of IEEE Military Commun. Conf*, pp. 1–5, 2012.

[13] S. Pudlewski, T. Melodia, and A. Prasanna, "Compressed-sensing-enabled video streaming for wireless multimedia sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 11, pp. 1060–1072, 2008.

[14] S. Mun and J. E. Fowler, "Residual reconstruction for block-based compressed sensing of video," *Data Compression Conference (DCC)*, vol. 15, 2011.

[15] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk, "Sparse signal recovery using markov random fields," in *Proceedings of the Workshop on Neural Information Processing Systems (NIPS)*, 2008.

[16] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for clustered sparse signals," *Acoustics, Speech and Signal Processing (ICASSP)*, vol. 92, no. 1, pp. 259–269, 2011.

[17] X. Zhang, Z. Chen, J. Wen, J. Ma, Y. Han, and J. Villasenor, "Relaxed maximum a posteriori fault identification," in *Data Compression Conference (DCC)*, 2011.

[18] N. Ukash and N. Rahnavard, "Bcs: Compressive sensing for binary sparse signals," in *Military Communication Conference (MILCOM)*, 2012.

[19] D. Bickson, D. Baron, A. Ihler, H. Avissar, and D. Dolev, "Fault identification via nonparametric belief propagation," *Signal Processing, IEEE Transactions on*, vol. 59, no. 6, pp. 2602–2613, 2011.

[20] A. Zymnis, S. Boyd, and D. Gorinevsky, "Relaxed maximum a posteriori fault identification," *Signal Processing, IEEE Transactions on*, vol. 89, no. 6, pp. 989–999, 2009.

[21] C. Studer, W. Yin, and R. G. Baraniuk, "Signal representations with minimum ℓ_∞-norm," in *Allerton Conf. on Comm. Control, and Computing (Allerton)*, 2012.

[22] O. L. Mangasarian and B. Recht, "Probability of unique integer solution to a system of linear equations," *European Journal of Operational Research*, vol. 214, no. 1, pp. 27–30, 2011.

[23] J. J. Fuchs, "Spread representations," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pp. 814–817, 2011.

[24] R. Tibshirani, "Regression shrinkage and selection with the lasso," *J. Royal. Statist. Soc*, vol. B(Methodological), pp. 267–288, 1996.

[25] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *Proceedings of IEEE Inf. Theory Workshop*, pp. 1–5, 2010.

[26] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: II. analysis and validation," in *Proceedings of IEEE Inf. Theory Workshop*, 2010.

[27] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process*, vol. 60, no. 7, pp. 3439–3448, 2012.

[28] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process*, vol. 58, no. 1, pp. 269–280, 2010.

[29] A. Farcomeni and S. Arima, "A Bayesian autoregressive three-state hidden Markov model for identifying switching monotonic regimes in microarray time course data," *Stat. Appl. Genet. Molec. Biol*, vol. 11, p. 4, 2012.

[30] J. Vila and P. Schniter, "Expectation-Maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process*, vol. 61, no. 19, pp. 4658–4672, 2013.

[31] J. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," *IEEE Trans. Signal Process*, vol. 62, no. 18, pp. 4689–4703, 2014.

[32] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Royal. Statist. Soc*, vol. 39, pp. 1–17, 1977.

[33] M. Svensen and C. M. Bishop, "Robust Bayesian mixture modelling," *Neurocomputing*, vol. 64, pp. 235–252, 2005.

[34] A. Azzalini, "A class of distributions which includes the normal ones," *Scandinavian J. of Stat*, vol. 12, pp. 171–178, 1985.

[35] R. B. Potts, "Some generalized order-disorder transformations," *Mathematical proceedings of the cambridge philosophical society*, vol. 48, no. 1, pp. 106–109, 1952.

[36] S. Wang and N. Rahnavard, "Compressive sampling of clustered sparse signals with asymmetric features," 2017.

[37] V. Cevher, M. F. Duarte, C. Hegde, and R. Baraniuk, "Sparse signal recovery using markov random fields," *Advances in Neural Information Processing Systems*, pp. 257–264, 2009.

[38] V. Cevher, P. Indyk, L. Carin, and R. Baraniuk, "A tutorial on sparse signal acquisition and recovery with graphical models.".

[39] B. M. McCoy and T. T. Wu, *The two-dimensional Ising model*. Courier Corporation, 2014.

[40] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," vol. 50, no. 13, pp. 2127–2159, 2006.

[41] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE personal communications*, vol. 6, no. 4, pp. 13–18, 1999.

[42] G. T. F. de Abreu, W. Zhang, and Y. Sanada, "Spectrum sensing algorithms via finite random matrix theory," in *IEEE Int Communications (ICC) Conf*, pp. 1–5, 2011.

[43] G. T. F. De Abreu, W. Zhang, and Y. Sanada, "Finite random matrices for blind spectrum sensing," in *Conference of Asilomar Conference on Signals, Systems and Computers*, pp. 116–120, 2010.

[44] L. S. Cardoso, M. Debbah, P. Bianchi, and J. Najim, "Cooperative spectrum sensing using random matrix theory," in *Proceedings of 3rd Int. Symp. Wireless Pervasive Computing ISWPC*, 2008.

[45] B. Nadler, "Finite sample approximation results for principal component analysis: A matrix perturbation approach," *Annals of Statistics*, vol. 36, pp. 2791–2817, 2008.

[46] B. Nadler, F. Penna, and R. Garello, "Performance of eigenvalue-based signal detectors with known and unknown noise level," in *Proceedings of IEEE Int Communications (ICC) Conf*, 2011.

[47] F. Penna, R. Garello, and M. A. Spirito, "Probability of missed detection in eigenvalue ratio spectrum sensing," in *Proceedings of IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications*, 2009.

[48] T. Ratnarajah, C. Zhong, A. Kortun, M. Sellathurai, and C. B. Papadias, "Complex random matrices and multiple-antenna spectrum sensing," in *Proceedings of IEEE Int Acoustics Speech and Signal Processing (ICASSP) Conf*, 2011.

[49] Y. Zeng and Y. chang Liang, "Eigenvalue-based spectrum sensing algorithms for cognitive radio," *IEEE Trans. on COMM.*, vol. 57, no. 6, pp. 1784–1793, 2009.

[50] S. Kritchman and B. Nadler, "Non-parametric detection of the number of signals: Hypothesis testing and random matrix theory," *IEEE J. on Signal Proc.*, vol. 57, pp. 3930–3941, 2009.

[51] B. Nadler and I. M. Johnstone, "Detection performance of roy's largest root test when the noise covariance matrix is arbitrary," in *in Proceedings of IEEE Statistical Signal Processing Conf*, 2011.

[52] "Lp-space." https://en.wikipedia.org/wiki/Lp_space.

[53] E. J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[54] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[55] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process*, vol. 56, pp. 2346–2356, June 2008.

[56] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[57] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[58] M. A. Khajehnejad, W. Xu, A. S. Avestimehr, and B. Hassibi, "Improved sparse recovery thresholds with two-step reweighted l1 minimization," *International Symposium on Information Theory (ISIT)*, pp. 1603–1607, 2010.

[59] R. Boomgard and R. Balen, "Methods for fast morphological image transforms using bitmapped images," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, vol. 54, no. 3, pp. 254–258, 1992.

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

[61] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[62] B. Dror, S. Sarvotham, and R. G. Barniuk, "Bayesian compressive sensing via belief propagation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 1, pp. 269–280, 2010.

[63] A. Wani and N. Rahnavard, "Compressive sampling for energy efficient and loss resilient camera sensor networks," in *Proceedings of IEEE Military Communication Conference (MILCOM)*, 2011.

[64] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *Information Theory, IEEE Transactions on*, vol. 56, pp. 1982–2001, 2010.

[65] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso." arXiv preprint, 2010.

[66] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[67] H. Ohlsson, L. Ljung, and S. Boyd, "Segmentation of arx-models using sum-of-norms regularization," *Automatica*, vol. 46, no. 6, pp. 1107–1111, 2010.

[68] Y. Chen and A. O. Hero, "Recursive $\ell_{1,\infty}$ group lasso," *Signal Processing, IEEE Transactions on*, vol. 60, no. 8, pp. 3978–3987, 2012.

[69] D. L. Donoho, "Comprssive sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–13061, 2006.

[70] R. Baraniuk, "Comprssive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[71] E. Candes, "Comprssive sensing," in *Int. Congress of Mathematics*, Congress of Mathematics, 2006.

[72] Y. Lyubarskii and R. Vershynin, "Uncertainty principles and vector quantization," *Information Theory, IEEE Transactions on*, vol. 56, no. 7, pp. 3491–3501, 2010.

[73] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." http://cvxr.com/cvx, 2014.

[74] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM J. on Optimization*, vol. 6, pp. 418–445, 1996.

[75] T. F. Coleman and Y. Li, "On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, no. 2, pp. 189–224, 1994.

[76] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *SIAM J. on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[77] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proceedings of National Academy of Sciences*, vol. 106, pp. 18914–18919, November 2009.

[78] "Matlab 2013a." The MathWorks Inc.

[79] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Phil. Trans. R. Soc. A: Math., Phys. and Eng. Sci*, vol. 367, no. 1906, pp. 4273–4293, 2009.

[80] E. V. D. Berg and M. P. Friedlander., "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. on Sci. Comp*, vol. 31, no. 2, pp. 890–912, 2008.

[81] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in Recent advances in learning and control, pp. 95–110, 2008.

[82] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process*, vol. 52, pp. 2153–2164, August 2004.

[83] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *in Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 40–44, 1993.

[84] "National Centers for Environmental Information." http://www1.ncdc.noaa.gov/pub/download/asos/.

[85] "National Weather Service Automated Surface Observing System," http://www.nws.noaa.gov/asos/.

[86] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," *Journal of Machine Learning Research*, vol. 12, pp. 3371–3412, Nov 2011.

[87] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal Processing*, vol. 92, no. 1, pp. 259–269, 2012.

[88] M. E. Eltayeb, T. Y. Al-Naffouri, and H. R. Bahrami, "Compressive sensing for feedback reduction in MIMO broadcast channels," *IEEE Trans. Commun*, vol. 62, pp. 3209–3222, September 2014.

[89] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[90] T. I. Lin, J. C. Lee, and S. Y. Yen, "Finite mixture modelling using the skew normal distribution," *Statistica Sinica*, pp. 909–927, 2007.

[91] X. L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ecm algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[92] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[93] J. P. Mills, "Table of the ratio: area to bounding ordinate, for any portion of normal curve," *Biometrika*, pp. 395–400, 1926.

[94] M. Abramowitz and I. A. Stegun, "Handbook of mathematical functions: with formulas, graphs, and mathematical tables," *Courier Corporation*, vol. 55, 1964.

[95] A. Magnani and S. P. Boyd, "Convex piecewise-linear fitting," *Optimization and Engineering*, vol. 10, no. 1, pp. 1–17, 2009.

[96] Z. Yin and R. Collins, "Belief propagation in a 3d spatio-temporal mrf for moving object detection," in *Proceedings of IEEE. Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

[97] "https://en.wikipedia.org/wiki/hadamard_product_(matrices)."

[98] M. F. Tappen and W. T. Freeman, "Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters," in *in Proceedings of IEEE Int. Conf. on Computer Vision*, 2003.

[99] "Newton's method." http://en.citizendium.org/wiki/Newton%27s_method.

[100] P. Bianchi, M. Debbah, M. Maida, and J. Najim, "Performance of statistical tests for single-source detection using random matrix theory," *IEEE Trans. on Information Theory*, vol. 57, no. 4, pp. 2400–2419, 2011.

[101] I. M. Johnstone, "On the distribution of the largest eigenvalue in principal components analysis," *Annals of statistics*, pp. 295–327, 2001.

[102] D. Paul, "Asymptotics of sample eigenstructure for a large dimensional spiked covariance model," *STATISTICA SINICA*, vol. 17, pp. 1617–1642, 2007.

[103] P. Moschopoulos, "The distribution of the sum of independent gamma random variables," *Annals of the Institute of Statistical Mathematics*, vol. 37, pp. 541–544, 1985.

[104] D. G. Kabe, "On the exact distribution of a class of multivariate test criteria," *The Annals of Mathematical Statistics*, vol. 33, pp. 1197–1200, 1962.

[105] J. F. Paris, "A note on the sum of correlated gamma random variables," *arXiv preprint arXiv:1103.0505*, 2011.

[106] L. J. Slater, *Generalized hypergeometric functions*. Cambridge, England: Cambridge University Press, 1966.

[107] H. Exton, *Multiple hypergeometric functions and applications*. New York: E. Horwood, 1976.

[108] J. L. Burchnall and T. W. Chaundy, "Expansions of appell's double hypergeometric functions (ii)," *The Quarterly Journal of Mathematics*, vol. 12, pp. 112–128, 1941.

[109] S. Wang and N. Rahnavard, "Eigenvalue-based cooperative spectrum sensing with finite samples/sensors," *in Proceedings IEEE Information Sciences and Systems*, pp. 1–5, 2012.

VITA

Sheng Wang

Candidate for the Degree of

Doctor of Philosophy

Dissertation: SPARSE SIGNAL RECOVERY AND DETECTION UTILIZING SIDE INFORMATION

Major Field: Electrical and Computer Engineering

Biographical:

Personal Data: Born in Hefei, Anhui, China on October 24, 1986.

Education:

Received the B.S. degree from Hefei University of Technology, Hefei, Anhui, China, 2008, Measuring and Control Technology & Instrument,
Received the M.S. degree from Tianjin University, Tianjin, China, 2010, Measuring Technology and Instrument,
Completed the requirements for the degree of Doctor of Philosophy with a major in Electrical and Computer Engineering, Oklahoma State University in October, 2017.

Name: Sheng Wang Date of Degree: October, 2017

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: SPARSE SIGNAL RECOVERY AND DETECTION UTILIZING SIDE INFORMATION

Pages in Study: 136 Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

In this dissertation, we investigate the signal recovery and detection task for compressive sensing and wireless spectrum sensing. First, we investigate the compressive sensing task for the difference frames of videos. Exploiting the clustered property, we design an effective structural aware reconstruction technique that is capable of eliminating isolated nonzero noisy pixels, and promoting undiscovered signal coefficients.

Further, we develop a novel optimization based method for the compressive sensing of binary sparse signals. We formulate the reconstruction task as a least square minimization procedure, and propose a novel regularization term based on the weighted sum of ℓ_1 norm and ℓ_∞ norm.

Moreover, we study the compressive sensing for asymmetrical signals. We devise an efficient algorithm that greatly improves the reconstruction quality of asymmetrical sparse signals. Further, we investigate sparse reconstruction of clustered sparse signals with asymmetrical features. We develop a powerful technique that is capable of taking inference of the signal, estimating the mixture density, and exploiting the clustered features.

Finally, we investigate the spectrum sensing task for cognitive radio. We develop an eigenvalue based technique that notably improve the primary user detection performance under finite number of sensors and samples.

ADVISOR'S APPROVAL: _____