

Syracuse University

SURFACE at Syracuse University

Dissertations - ALL

SURFACE at Syracuse University

Summer 7-16-2021

One-bit Compressed Sensing in the Presence of Noise

Swatantra Kafle
Syracuse University

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Kafle, Swatantra, "One-bit Compressed Sensing in the Presence of Noise" (2021). *Dissertations - ALL*. 1415.

<https://surface.syr.edu/etd/1415>

This Dissertation is brought to you for free and open access by the SURFACE at Syracuse University at SURFACE at Syracuse University. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE at Syracuse University. For more information, please contact surface@syr.edu.

ABSTRACT

Many modern real-world systems generate large amounts of high-dimensional data stressing the available computing and signal processing systems. In resource-constrained settings, it is desirable to process, store and transmit as little amount of data as possible. It has been shown that one can obtain acceptable performance for tasks such as inference and reconstruction using fewer bits of data by exploiting low-dimensional structures on data such as sparsity. This dissertation investigates the signal acquisition paradigm known as one-bit compressed sensing (one-bit CS) for signal reconstruction and parameter estimation.

We first consider the problem of joint sparse support estimation with one-bit measurements in a distributed setting. Each node observes sparse signals with the same but unknown support. The goal is to minimize the probability of error of support estimation. First, we study the performance of maximum likelihood (ML) estimation of the support set from one-bit compressed measurements when all these measurements are available at the fusion center. We provide a lower bound on the number of one-bit measurements required per node for vanishing probability of error. Though the ML estimator is optimal, its computational complexity increases exponentially with the signal dimension. So, we propose computationally tractable algorithms in a centralized setting. Further, we extend these algorithms to a decentralized setting where each node can communicate only with its one-hop neighbors. The proposed method shows excellent estimation performance even in the presence of noise.

In the second part of the dissertation, we investigate the problem of sparse signal reconstruction from noisy one-bit compressed measurements using a signal that is statistically dependent on the compressed signal as an aid. We refer to this signal as side-information. We consider a generalized measurement model of one-bit CS where noise is assumed to be added

at two stages of the measurement process- a) before quantization and b) after quantization. We model the noise before quantization as additive white Gaussian noise and the noise after quantization as a sign-flip noise generated from a Bernoulli distribution. We assume that the SI at the receiver is noisy. The noise in the SI can be either in the support or in the amplitude, or both. This nature of the noise in SI suggests that the noise has a sparse structure. We use additive independent and identically distributed Laplacian noise to model such sparse nature of the noise. In this setup, we develop tractable algorithms that approximate the minimum mean square error (MMSE) estimator of the signal. We consider the following three different SI-based scenarios:

- The side-information is assumed to be a noisy version of the signal. The noise is independent of the signal and follows the Laplacian distribution. We do not assume any temporal dependence in the signal.
- The signal exhibits temporal dependencies between signals at the current time instant and the previous time instant. The temporal dependence is modeled using the birth-death-drift (BDD) model. The side-information is a noisy version of the previous time instant signal, which is statistically dependent on the signal as defined by the BDD model.
- The SI available at the receiver is heterogeneous. The signal and side-information are from different modalities and may not share joint sparse representation. We assume that the SI and the sparse signal are dependent and use the Copula function to model the dependence.

In each of these scenarios, we develop generalized approximate message passing-based algorithms to approximate the minimum mean square error estimate. Numerical results show the effectiveness of the proposed algorithm.

In the final part of the dissertation, we propose two one-bit compressed sensing reconstruction algorithms that use a deep neural network as a prior on the signal. In the first algorithm,

we use a trained Generative model such as Generative Adversarial Networks and Variational Autoencoders as a prior. This trained network is used to reconstruct the compressed signal from one-bit measurements by searching over its range. We provide theoretical guarantees on the reconstruction accuracy and sample complexity of the presented algorithm. In the second algorithm, we investigate an untrained neural network architecture so that it acts as a good prior on natural signals such as images and audio. We formulate an optimization problem to reconstruct the signal from one-bit measurements using this untrained network. We demonstrate the superior performance of the proposed algorithms through numerical results. Further, in contrast to competing model-based algorithms, we demonstrate that the proposed algorithms estimate both direction and magnitude of the compressed signal from one-bit measurements.

ONE-BIT COMPRESSED SENSING IN THE PRESENCE
OF NOISE

By

Swatantra Kafle

B.E., Tribhuvan University, 2011
M. Phil., Syracuse University, 2019

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical and Computer Engineering

Syracuse University
July 2021

Copyright © 2021 Swatantra Kafle

All rights reserved

ACKNOWLEDGMENTS

First, I would like to thank my advisor, Prof. Pramod K. Varshney, for his continued guidance, support, and encouragement throughout my doctoral studies. His work ethic, curiosity, persistence, high standards, engagement, motivation, and patience have heavily influenced me as a researcher and will stay with me forever. I am tremendously grateful for all kinds of support he provided me while going through uncertain and disturbing times. I want to express my gratitude to Thakshila Wimalajeewa for mentoring me during my studies and dissertation. I would also like to thank my defense committee members Prof. Lixin Shen, Prof. Mustafa Gursoy, Prof. Makan Fardad, Thakshila Wimalajeewa, and Prof. Venkanta Gandikota for their insightful comments and valuable suggestions.

I have been fortunate to be a part of the Sensor Fusion Lab and enjoy the companionship of many intelligent, supporting, and friendly labmates Prashant, Pranay, Saikiran, Baocheng, Shan, Quan, Nandan, Aditya, Bhava, Hao, Nianxia, Qunwei, Raghed, Sid, and Arun. All the intellectual and philosophical discussions we had helped me grow as a researcher and as a person. Time spent with Amar, Pratik, Prabesh, Basanta, Prashant, Pranay, and Sai is engraved in my memory and will be cherished forever. I would also like to thank Aunty for her kindness, support, and motherly love. She made me feel at home during my stay at Syracuse.

Finally, and most importantly, this thesis would not have been possible without the constant encouragement, faith, patience, and support that my father, mother, sisters, brothers-in-law, and wife have bestowed on me over the years of my graduate studies.

I am eternally grateful for their unwavering love and support. I would like to thank my wife, Renisha, who has been smiling, encouraging, and cheering me on this roller coaster journey. I greatly appreciate her unconditional love, trust, and understanding.

TABLE OF CONTENTS

Acknowledgments	vi
List of Tables	xii
List of Figures	xiii
1 Introduction	1
1.1 Compressed Sensing	2
1.2 One-Bit Compressed Sensing	6
1.3 Literature Review	11
1.4 Major Contributions	14
1.5 Organization of the Dissertation	15
1.6 Bibliographic Note	17
2 Joint Sparsity Pattern Recovery in a Distributed Network	19
2.1 Introduction	19
2.2 Observation Model	20
2.3 Joint Sparse Support Recovery with One-Bit Compressed Sensing via the Maximum Likelihood Decoder	21
2.3.1 Probability of Error and Maximum Likelihood Decoder	21
2.3.2 Sufficient Conditions for Support Recovery	22
2.4 Centralized Algorithms with One-Bit Compressed Sensing	26

2.4.1	Centralized $\ell_{1,\infty}$ Regularized Maximum Likelihood Based Algorithm	27
2.4.2	Centralized-BIHT	33
2.5	Decentralized Algorithms with One-bit Compressed Sensing	35
2.5.1	Decentralized-BIHT	36
2.6	Simulation Results	38
2.7	Summary	49
3	One-bit Compressed Sensing with Homogeneous Side-Information	50
3.1	Introduction	50
3.2	Signal and Measurement Models	52
3.2.1	Signal Model	52
3.2.2	Measurement Model	52
3.2.3	Noisy One-Bit Compressed Sensing	54
3.3	Noisy One-Bit Compressed Sensing Algorithm	54
3.3.1	Noisy One-Bit Compressed Sensing (Noisy1bG)	55
3.4	Noisy One-bit Compressed Sensing with Side-Information	60
3.4.1	Noisy one-bit Compressed Sensing with Laplacian Noise (laplacianSI)	61
3.4.2	Estimation of the v_s	65
3.4.3	Noisy one-bit Compressed Sensing with Gaussian Noise (GaussianSI)	67
3.5	Noisy One-bit Compressed Sensing with support as side-information	69
3.6	Simulation Results	72
3.7	Conclusion	77
4	One-bit Compressed Sensing with Side-Information in Time-Varying Environ-	
	ments	78
4.1	Introduction	78
4.2	Signal and Measurement Models	79

4.2.1	Signal Model	79
4.2.2	Measurement Model	80
4.2.3	Noisy One-Bit Compressed Sensing	82
4.3	Noisy One-Bit Compressed Sensing Algorithm with SI in Time-varying Environments	82
4.3.1	Noisy One-Bit Compressed Sensing in a Time-Varying Environment	82
4.3.2	Variations of Algorithm 4.1	90
4.4	Simulation Results	92
4.5	Conclusion	97
5	One-Bit CS with Heterogeneous SI	98
5.1	Introduction	98
5.1.1	Signal and Measurement Model	99
5.1.2	Signal Model	99
5.1.3	Copula Functions	99
5.1.4	Measurement Model	100
5.1.5	Bayesian Formulation	102
5.2	GAMP Algorithm Update Equations	102
5.2.1	Heterogeneous Side-Information (HSI)	102
5.2.2	Copula Functions	103
5.2.3	Update Equations	104
5.3	Simulation Results	106
5.4	Conclusion	108
6	One-bit Compressed Sensing using Network prior	109
6.1	Introduction	109
6.2	System Model and Algorithm	110

6.3	Compressed Sensing using Trained Network as a Prior	111
6.3.1	Solution Space	111
6.3.2	Theoretical Analysis	113
6.3.3	Simulation Results	115
6.4	Compressed Sensing using Untrained Network as a Prior	125
6.4.1	Solution Space	126
6.4.2	Design of the Loss Function \mathcal{L}	126
6.5	Solution Methodology	128
6.5.1	Simulation Results	129
6.6	Summary	134
7	Summary and Future Directions	135
7.1	Summary	135
7.2	Future Directions	136
	Appendix	139
A	Proof of Theorem 2.2	139
B	Derivation of (3.16)	144
C	Derivation of Result 3.1:	146
D	Derivation of Result 3.3	149
E	Evaluation of Result 4.1	152
F	Results for 1bitdynG	159
G	Proof of Theorem 6.1	161
	References	165

LIST OF TABLES

2.1	Comparison of run times of C-BIHT and MLA in seconds to obtain the sparsity pattern when $N=100$ and $N=500$	
	46	
3.1	GAMP Equations for Side-Information	63
4.1	GAMP Equations for SI in a dynamic setting	87

LIST OF FIGURES

1.1	A pictorial representation of the noiseless compressed sensing measurement model	3
2.1	PSPR for MLA when $\eta = -3.01$ dB, $N = 100$ and $K = 5$ in a network of 10 nodes.	38
2.2	PSPR for C-BIHT when $\eta = -3.01$ dB, $N = 100$ and $K = 5$ in a network of 10 nodes.	38
2.3	PSPR when $\eta = -3.01$ dB for $N = 100$ and $K = 5$	40
2.4	PESPR when $\eta = -3.01$ dB for $N = 100$ and $K = 5$	40
2.5	PSPR when $\eta = 6.99$ dB for $N = 100$ and $K = 5$	40
2.6	PESPR when $\eta = 6.99$ dB for $N = 100$ and $K = 5$	40
2.7	PSPR for D-BIHT, D-BIHTm, DMLA, C-BIHT, MLA, BIHT (SMV) as a function of η when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.	42
2.8	PESPR for D-BIHT, D-BIHTm, DMLA, C-BIHT, MLA, BIHT (SMV) as a function of η when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.	42
2.9	Comparison of SIHT with C-BIHT, D-BIHT, and D-BIHTm when $N = 100$, $P = 10$, $K = 5$, and $\eta = 16.99$ dB	43

2.10	Comparison of SIHT with C-BIHT, D-BIHT, and D-BIHTm when $N = 100$, $P = 10$, $K = 5$, and $M = 60$	43
2.11	Minimum number of measurements for support recovery vs. Signal SNR, γ when $N = 1000$, $K = 20$, $\sigma_v^2 = 0.01$, $\mu_j = 1$, for $j \in \mathcal{U}$, and $P = 5$	44
2.12	PSPR values of C-BIHT and MLA algorithms as a function of η when different random Gaussian \mathbf{A}_p are used when $N = 100$, $M = 60$, $K = 5$, and $P = 10$	45
2.13	PSPR values of C-BIHT and MLA algorithms as a function of η when random partial DCT matrix is used when $N = 100$, $M = 60$, $K = 5$, and $P = 10$	45
2.14	Percentage of Sparsity Pattern Recovery for D-BIHT, D-BIHTm, C-BIHT, BIHT, and IHT as a function of M/N when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.	48
3.1	One-bit CS with pre-quantization and post-quantization noise.	53
3.2	Comparison of reconstruction performance of the proposed method as a function of sign-flip, $1 - \gamma$ and number of measurements, M when $N = 50$, $\lambda = 0.15$	72
3.3	Comparison of reconstruction performance of the proposed methods as a function of sign-flip probability and measurements in presence of SI when $N = 200$, $\lambda = 0.15$	73
3.4	Comparison of the effect of noise in SI on the reconstruction performance of proposed algorithms when $\lambda = 0.1$, $1 - \gamma = 0.15$, and $v = 0.15$,	75
3.5	reconstruction performance of the proposed algorithms when $\lambda = 0.1$, $1 - \gamma = 0.15$, and $v = 0.15$, $N = 200$, and $M = 600$	76
4.1	1-bit CS with pre-quantization and post-quantization noise.	81

4.2	Comparison of reconstruction performance of the proposed method with 1bitdynG, 1bdyn and 1bLaplacian as a function of sign-flip probability and number of measurements, respectively when $N = 200$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.1$. In first plot $M = 200$, and in second plot $\gamma = 0.8$	92
4.3	Comparison of the reconstruction performance of the proposed method with 1bitdynG, 1bdyn and 1bLaplacian as a function of noise-variance in SI when $N = 200$, $M = 200$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$	94
4.4	NMSE performance as a function of ρ when $N = 200$, $M = 200$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$	95
4.5	NMSE performance as a function of time instants when $N = 200$, $M = 400$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$ of a sequence of sparse signal with slow time varying dynamics	96
5.1	one-bit CS with pre-quantization and post-quantization noise.	101
5.2	Reconstruction performance of the proposed method when $\rho = 0.7$, $\gamma = 0.10$, $v_n = 0.1$, $\lambda = 0.1$	106
5.3	Reconstruction performance of the proposed method when $M/N = 2$, $v_n = 0.1$, $\lambda = .1$, $\rho = 0.7$	106
5.4	Reconstruction performance of the proposed method. $M/N = 2$, $v_n = 0.1$, $\lambda = .1$, $\gamma = 0.1$	107
6.1	MSE of the proposed algorithm compared with BIHT and YP as a function of number of measurements M in a noiseless setting.	117
6.2	NMSE of the proposed algorithm compared with BIHT and YP as a function of number of measurements M in a noiseless setting.	117
6.3	The first rows show the original images, the second, third and fourth rows are the reconstruction images using BIHT, YP and proposed algorithms, respectively when $m = 784$ in a noiseless setting for MNIST dataset.	118

6.4	The first rows show the original images, the second, third and fourth rows are the reconstruction images using BIHT, YP and proposed algorithms, respectively when $m = 784$ in a noiseless setting for FashionMNIST dataset.	118
6.5	MSE of the proposed algorithm compared with BIHT, and YP as a function of number of measurements m in a noisy setup ($v_n = 0.1$ and $\alpha = 0.85$).	120
6.6	NMSE of the proposed algorithm compared with BIHT, and YP as a function of number of measurements m in a noisy setup ($v_n = 0.1$ and $\alpha = 0.85$).	120
6.7	MSE of the proposed algorithm compared with BIHT, and YP as a function of number of sign-flip probability when $M = 784$ and $v_n = 0.1$	121
6.8	NMSE of the proposed algorithm compared with BIHT, and YP as a function of sign-flip probability when $M = 784$ and $v_n = 0.1$	121
6.9	Reconstruction performance of the proposed algorithm, BIHT and YP as a function of measurement matrix uncertainty, v_Δ , when $M = 1500$ in a noiseless setting ($\alpha = 1$ and, $v_n = 0$).	122
6.10	NMSE values of proposed algorithm and BIHT for MNIST and Omniglot dataset as a function of number of measurements	123
6.11	MSE of the proposed algorithm compared with GenModel_pgd as a function of number of Measurements when $v_n = 0.1$	124
6.12	NMSE of the proposed algorithm compared with GenModel_pgd as a function of number of Measurements when $v_n = 0.1$	124
6.13	Comparison of CSI estimation of the proposed algorithm with the state-of-the-art algorithm from one-bit compressed measurements when $\alpha = 0.9$ and $v_n = 0.05$	125
6.14	Reconstruction performance of our algorithm compared with the state-of-the-art algorithm.	130
6.15	Reconstruction when $M = 1500$	132

6.16 Reconstruction when $M = 1500$	133
---	-----

CHAPTER 1

INTRODUCTION

We generate an enormous amount of digital data daily, and the pace of generation is only accelerating with the adoption of new technologies such as the Internet of Things. The tremendous increase in the volume of data has posed a number of important challenges. A few of the critical challenges include acquisition, storage, transmission, and inference from these data. For example, consider modern camera systems that are capable of generating several gigabytes of raw data in a short period of time. Traditional acquisition systems usually adopt a sample-then-compress acquisition model that at first samples data and then discards redundant information. The task at hand could be an inference task such as classification, which usually requires much less data to yield acceptable performance. So, it is desired to devise an acquisition system that compresses data at the time of sampling. In many other applications such as CCTV, the storage requirement is very high as the system is required to store video information from all the cameras for a significant number of days. One would like to store as little amount of data as possible while being able to reconstruct the entire signal whenever desired. Further, in resource-constrained networks and/or power-constrained devices, one would like to measure, store and transmit as little amount of data as possible. For example, in a cognitive radio network where mobile devices cooperate to estimate the occupied and unoccupied frequency

slots, these devices want to communicate as few bits of information as possible to make a final decision regarding occupancy.

In all of these examples, we can observe some common traits. First, all of these examples involve signals that are high-dimensional. Second, in all these applications, it is strongly desired to decrease the amount of data we obtain, store, or transmit without compromising on the ability to reconstruct the original data or make inferences from the data. In this dissertation, we focus on one such acquisition technology known as compressed sensing. To be more precise, we are interested in an extreme case of compressed sensing when the observations from the compressed sensing acquisition system are quantized element-wise to one-bit. This is popularly known as one-bit compressed sensing.

Next, we introduce the problem of compressed sensing and then describe the special case namely the one-bit compressed sensing considered in this dissertation.

1.1 Compressed Sensing

Compressed Sensing (CS) [15, 26] is a relatively new framework for signal acquisition that provides a huge saving in the sampling and computation costs for high dimensional signals having finite sparse or compressible representation. The traditional Nyquist-Shanon sampling theorem suggests that signals can be accurately reconstructed from the uniformly sampled measurements if they are acquired at a rate at least twice the signal bandwidth. CS, however, employing a complex sampling processing, reconstructs the sparse signal from a small set of linear, nonadaptive measurements. The CS recovery algorithms are highly non-linear and computationally intensive compared to the sinc-interpolation used for the Nyquist-Shannon-based algorithm.

Specifically, let $\mathbf{x} \in \mathbb{R}^N$ be the signal which has a sparse representation and $\mathbf{y} \in \mathbb{R}^M$ be

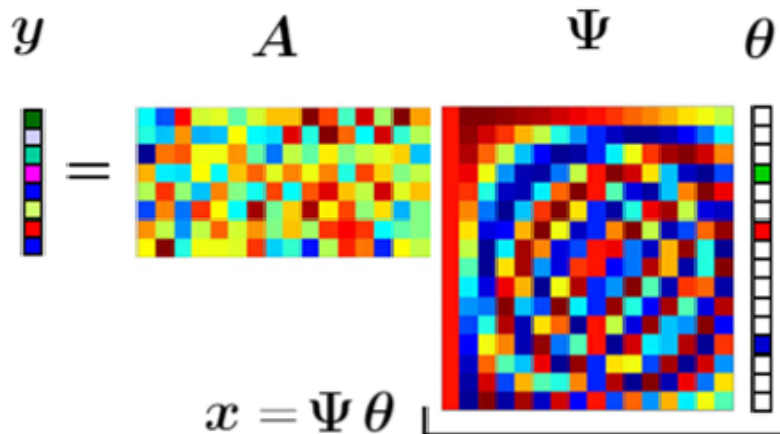


Fig. 1.1: A pictorial representation of the noiseless compressed sensing measurement model

the compressed measurements. The measurement model can be represented as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad (1.1)$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a measurement matrix and \mathbf{n} is an additive white Gaussian noise (AWGN). Let $\mathbf{x} = \Psi\boldsymbol{\theta}$ where $\boldsymbol{\theta}$ is a K -sparse signal, and Ψ is the basis where \mathbf{x} has the sparse representation. Any signal is called K -sparse, if the signal has at most K non-zero elements. Without loss of generality, we assume that the signal is sparse in canonical basis, i.e., $\Psi = \mathbb{I}_N$, where \mathbb{I}_N is the $N \times N$ identity matrix. The goal is to recover the sparse signal \mathbf{x} from \mathbf{y} when $M < N$. When the problem is under-determined, for any \mathbf{x} , it is impossible to recover the signal from \mathbf{y} . But to reconstruct all K -sparse signals, it is necessary that linear measurements for two distinct signals are different, i.e., for all K -sparse signals $\mathbf{x}_1 \neq \mathbf{x}_2$,

$$\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2) \neq \mathbf{0}. \quad (1.2)$$

In a noiseless setup, if (1.2) is satisfied, we can recover \mathbf{x} by solving the following optimization

problem

$$\hat{\mathbf{x}} = \arg \min_{\theta} \|\theta\|_0 \quad s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (1.3)$$

where $\|\mathbf{x}\|_0$ is the ℓ_0 norm of \mathbf{x} . The optimization problem (1.3) is NP-hard, and is computationally intractable for high-dimensional signals. The problem in (1.3) can be relaxed to a convex problems as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad s.t. \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (1.4)$$

This problem can be solved in a polynomial time. While (1.2) is necessary and sufficient for reconstruction guarantees in a noiseless setting, the measurement matrix should satisfy a stronger condition, known as restricted isometry property (RIP), when measurements are noisy. A measurement matrix \mathbf{A} satisfies RIP of order K if there exists some $\delta_K \in (0, 1)$ such that

$$(1 - \delta_K)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_K)\|\mathbf{x}\|_2^2 \quad (1.5)$$

holds for all K -sparse signals \mathbf{x} . In the noisy case, the equivalent optimization problem for sparse signal reconstruction is

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\| \leq \epsilon, \quad (1.6)$$

where the ϵ measures the fidelity of the signal reconstruction to the compressed measurements. The value of ϵ can be determined by the noise or the quantization error. When matrix \mathbf{A} satisfies RIP of order $2K$, the solution of the optimization problem (1.6) can recover all K -sparse signals [14]. The unconstrained version of (1.6) is quite popular and can be expressed as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \quad (1.7)$$

where λ is a tuning parameter which is used as a trade-off between the fidelity of the reconstruction and the sparsity. For some value of λ , problems (1.6) and (1.7) are equivalent. How-

ever, the value of λ that makes these problems equivalent is not known *a priori* and several approaches have been proposed to choose the value of λ [29, 33, 36].

In addition to these convex optimization-based approaches to reconstruct sparse signals, several greedy/iterative methods have been proposed for solving such reconstruction problems [7, 9, 10, 22, 28, 88]. These methods either iteratively estimate supports in each iteration until a certain convergence criterion is met or iteratively improve the signal estimate. Some of the well-known algorithms are Orthogonal Matching Pursuit [88], and Iterative hard thresholding [10]. These algorithms have similar performance guarantees with respect to the convex optimization-based approaches. Beyond signal reconstruction, several works have addressed the problem of inference such as detection, estimation, and classification from compressed measurements with or without full signal reconstruction [23, 42, 51, 95].

Recently, compressed sensing based approaches have been extended to incorporate different classes of low-dimensional signals other than sparsity. In [11], the authors have explored the idea of compressed sensing when the signal lies in or near the range space of a trained neural network $G(\cdot)$. We call these methods deep-learning based methods as they use a feed forward neural network as a prior on the sparse signal. When a trained neural network is used as the prior, the compressed signal can be recovered by solving the following optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{g}: \mathbf{x}=G(\mathbf{g})} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2, \quad (1.8)$$

where $\mathbf{g} \in \mathbb{R}^k$ is an input to the neural network. When the signal lies in a set $\mathcal{S} \in \mathbb{R}^N$, the set restricted eigenvalue condition on \mathbf{A} is a sufficient condition for robust signal recovery, which is defined as

$$\mathbf{A}(\mathbf{x}_1 - \mathbf{x}_2) \geq \gamma \|\mathbf{x}_1 - \mathbf{x}_2\| - \delta, \quad (1.9)$$

for some $\gamma > 0$, $\delta \geq 0$ and $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}$. Note that, in the problem of compressed sensing using generative model, the unknown signal lies in or near the set $\mathcal{S} = \{\mathbf{x} | \mathbf{x} = G(\mathbf{g}) \in \mathbb{R}^N\}$.

Generally, the $G(\mathbf{g})$ is a feed-forward neural network and the optimization problem (1.8) is non-convex with respect to a latent variable \mathbf{g} . Any solution to (1.8) does not guarantee the optimal solution. However, the empirical performance has been found to be promising.

We note that the standard problem of CS assumes that the measurements are real-valued. However, practical implementations require quantization of these measurements. When high precision quantization is employed, compressed sensing can perform similar to the real-valued compressed sensing as the error or noise due to quantization could be modeled as additive noise. Coarse quantization, however, is more attractive in practice, as it significantly reduces bandwidth usage and power consumption leading to its use in low-power devices and resource-constrained networks. In this dissertation, we focus on the most extreme form of quantization, i.e., one-bit quantization. We study the performance of sparse signal reconstruction and parameter estimation from noisy one-bit measurements with the goal of improving the performance of existing algorithms in the literature. Next, we introduce the problem of one-bit compressed sensing.

1.2 One-Bit Compressed Sensing

The problem of one-bit compressed sensing deals with the reconstruction of sparse signals from one-bit measurements. The linear measurements of low-dimensional signal \mathbf{x} are quantized to one-bit element-wise. The measurement model of one-bit compressed sensing is

$$\mathbf{y} = Q(\mathbf{A}\mathbf{x}), \tag{1.10}$$

where the quantizer $Q : \mathbb{R}^M \rightarrow \{q_1, q_2\}^M$ is the element-wise quantizer. The m -th element at the output of the quantizer, $[Q(\zeta)]_m$ is

$$[Q(\zeta)]_m = \begin{cases} q_1, & \text{if } \zeta_m \leq \text{Th}, \\ q_2, & \text{if } \zeta_m > \text{Th}. \end{cases} \quad (1.11)$$

In one-bit quantization, the quantizer is most often implemented as a sign quantizer, i.e., $q_1 = -1, q_2 = +1$, and $\text{Th} = 0$. The goal of one-bit CS is to reconstruct signal \mathbf{x} from one-bit quantized vector \mathbf{y} . Note that, with one-bit quantization amplitude information is lost and hence, the signal are reconstructed on a unit sphere. One candidate optimization problem to reconstruct a sparse signal from one-bit measurements is

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{S}^{N-1}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \text{sign}(\mathbf{A}\mathbf{x}), \quad (1.12)$$

where \mathcal{S}^{N-1} represents the unit-sphere. The problem in (1.12) has exponential complexity. Alternative algorithms have been proposed that are computationally feasible. Some of the one-bit compressed sensing algorithms are Binary Iterative Hard Thresholding algorithm [47], Matching Sign Pursuit [4], and Adaptive Outlier Pursuit [99]. We introduce one of these algorithms that is used for signal recovery from one-bit measurements in detail as as we employ it in the dissertation.

Binary Iterative Hard Thresholding

Binary Iterative Hard Thresholding (BIHT) [47] is one of the most popular one-bit CS algorithms that iteratively estimates the K -sparse signal \mathbf{x} . The signal estimate at the t -th iteration with quantized measurements \mathbf{y} is given as

$$\mathbf{x}^k = \Theta_K \left(\mathbf{x}^{t-1} - \tau \mathbf{A}^T (\text{sign}(\mathbf{A}\mathbf{x}^{k-1}) - \mathbf{y}) \right), \quad (1.13)$$

where $\Theta_K(\mathbf{v})$ computes the best K -term approximation of \mathbf{v} by thresholding. BIHT can be thought of as trying to solve the following optimization problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|[\mathbf{y} \odot (\mathbf{A}\mathbf{x})]_-\|_1 \quad \text{s.t.} \quad \|\mathbf{x}\|_0 = K, \|\mathbf{x}\|_2 = 1, \quad (1.14)$$

where \odot is a Hadamard product and $([\mathbf{x}]_-)_i = [x_i]_-$ with $[x_i] = x_i$ if $x_i < 0$ and 0 otherwise. Notice that $\mathbf{y} \odot (\mathbf{A}\mathbf{x})$ is positive if the sign of $(\mathbf{A}\mathbf{x})$ is the same as that of \mathbf{y} . Minimizing the one-sided ℓ_1 objective forces $\mathbf{A}\mathbf{x}$ to have the same sign as \mathbf{y} , i.e.,

$$\mathbf{y} \odot (\mathbf{A}\mathbf{x}) \geq 0 \quad (1.15)$$

The condition in (1.15) enforces consistency between \mathbf{y} and the linear measurements of the reconstructed signal $\mathbf{A}\mathbf{x}$.

Note that the measurement model in (1.10) is noiseless and hence, is of limited use. Next, we introduce a general noisy one-bit compressed sensing measurement model as:

$$\mathbf{y} = f_{\zeta \sim \mathcal{D}}(\mathbf{A}\mathbf{x}, \zeta), \quad (1.16)$$

where ζ models the randomness due to noise and is assumed to follow distribution \mathcal{D} . Any noisy one-bit compressed sensing model can be expressed as an instance of (1.16). Consider the measurement model

$$\mathbf{y} = \boldsymbol{\eta} \odot Q(\mathbf{A}\mathbf{x} + \mathbf{n}), \quad (1.17)$$

where $\boldsymbol{\eta}$ and \mathbf{n} are noise vectors. We can consider ζ as the vector formed by combining noise vectors $\boldsymbol{\eta}$ and \mathbf{n} . In this setup, the distribution \mathcal{D} is the joint distribution of $\boldsymbol{\eta}$ and \mathbf{n} .

In this dissertation, we are interested in devising reliable algorithms for reconstruction and parameter estimation of low-dimensional signal $\mathbf{x} \in \mathbb{R}^N$ from noisy one-bit measurements $\mathbf{y} \in \mathbb{R}^M$. For all the problems of interest, we focus on the following two types of low-dimensional

signals:

- *Sparse Signals:* Sparse signals have only a few elements that are essentially non-zero. If a signal has at most K non-zero elements, then we say the signal is K -sparse. In general, for K -sparse signals $K \ll N$.
- *Signals on the Range Space of a Neural Network:* Let $\mathbf{g} \in \mathbb{R}^k$ and \mathbf{w} be the latent variable (or input to a neural network) and the weight parameter of a neural network, respectively. We represent a neural network by a mapping $G(\mathbf{w}; \mathbf{g})$ that maps any input vector $\mathbf{g} \in \mathbb{R}^k$ to some signal $\mathbf{x} \in \mathbb{R}^N$. The low-dimensional signal is assumed to lie in the range space of this network, i.e., $\mathbf{x} = G(\mathbf{w}; \mathbf{g}) \in \mathbb{R}^N$. The mapping function $G(\mathbf{w}; \mathbf{g})$ represents a trained neural network or untrained neural network. Note that, we assume both the input \mathbf{g} and the signal \mathbf{x} to be vectors. We can represent even two-dimensional signals such as images with this problem setup. In this case, the signal is the vectorized form of the actual image. If we like to represent both the input and the output as matrices, the network architecture of $G(\mathbf{w}; \mathbf{g})$ should be adjusted accordingly.

We approach the problem of inference and signal reconstruction from two different perspectives.

- *Bayesian Perspective:* In this setup, we look at the problem of joint sparse support estimation in a distributed setting and the problem of sparse signal reconstruction from noisy one-bit measurements. For the task of support estimation, we use the maximum likelihood estimation of support using noisy one-bit measurements from distributed nodes and provide a lower bound on the number of one-bit measurements required per node for vanishing probability of error. Specifically, we consider the following problem.

$$\hat{\mathcal{U}} = \underset{\mathcal{U}_k, n_0=1, \dots, N_0}{\operatorname{argmax}} p(\mathbf{Y} | \mathcal{U}_{n_0}), \quad (1.18)$$

where \mathbf{Y} is the measurement matrix at the Fusion Center, \mathcal{U}_{n_0} is the n_0 -th support set

where $n_0 \in \{1, \dots, N_0\}$, and N_0 is the total number of sparse support sets. Next, we address the problem of signal reconstruction by evaluating minimum mean square error (MMSE) estimate of the signal. We use a prior on the signal to impose a sparse structure. Let $p(\mathbf{x})$ be the pdf of the prior on \mathbf{x} and $p(\mathbf{x}|\mathbf{y})$ be the posterior density function of the signal \mathbf{x} . The MMSE estimate of the signal is evaluated as

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}], \quad (1.19)$$

where $\mathbb{E}[\mathbf{x}|\mathbf{y}]$ represents the expectation of the signal with respect to $p(\mathbf{x}|\mathbf{y})$. We then investigate several scenarios when the receiver has access to a signal that is statistically dependent on the compressed signal, which we call side-information (SI). In several applications such as dynamic MRI reconstruction [68], video signal reconstruction [53], sequential estimation [17], and wireless channel estimation [69] SI is available and can be used as an aid to improve reconstruction performance. In this setup, we evaluate the MMSE estimate of the signal as

$$\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}], \quad (1.20)$$

where the expectation is with respect to the posterior pdf of the signal when SI is available at the receiver. As the evaluation of the MMSE estimator is intractable in high-dimensions, we propose computationally tractable algorithms to approximate the estimators. We use the Bayesian framework in these problems because it allows us to model the sparse structure of the signal, higher-order dependence, and randomness in support and measurements with ease.

- *Machine-learning Perspective:* In this approach, we assume that the signal lies in the range space of a feed-forward neural network. In this dissertation, we look into two different cases when a feed-forward neural network is used as a prior. In the first case, we

assume that the neural network used as a prior is trained to the desired dataset. For example, the trained neural network could be the decoder of the Variational Autoencoders (VAEs) [55] or the generator network of Generative Adversarial Networks (GANs) [37]. Let $G(\mathbf{g}, \mathbf{w})$ be the trained network. We compute $\hat{\mathbf{x}}$ from one-bit measurements \mathbf{y} by optimizing a loss function $\mathcal{L}(\mathbf{A}\mathbf{x}; \mathbf{y})$ as

$$\min_{\mathbf{g}} \mathcal{L}(\mathbf{A}G(\mathbf{w}; \mathbf{g}), \mathbf{y}). \quad (1.21)$$

The loss function is optimized over the input of the trained neural network. In the second case, we use an untrained feed-forward neural network as a prior and consider the following optimization problem to reconstruct a signal from one-bit measurements.

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{A}G(\mathbf{w}; \mathbf{g}), \mathbf{y}). \quad (1.22)$$

In this case, the loss function is optimized over the weights of the untrained neural network. In both settings, we formulate optimization problems and provide algorithms to reconstruct low-dimensional signals from one-bit measurements.

1.3 Literature Review

In this section, we review some recent progress in the area of one-bit compressed sensing. We discuss recent works focused on the task of signal reconstruction and inference. At the end, we introduce a few other works which deal with problems beyond signal reconstruction and inference.

The idea of one-bit compressed sensing was first introduced in [13], and since then, it has been an active field of research. The popularity of one-bit CS is because the quantizer is easier to build and is cost-effective. Further, it provides savings in the number of bits of information

for storage and transmission. These are the reasons why one-bit CS has found applications in several areas such as wireless sensor networks [18, 85, 97], radar [25, 63, 100], cognitive radios [32, 60], bio-signal processing [2, 41], and wireless communications [67, 71, 79, 84, 98].

Most of the works in one-bit compressed sensing are focused on the sparse signal reconstruction problem [1, 4, 5, 13, 21, 31, 46, 48, 50, 52, 57, 62, 66, 74, 77, 78, 80, 105]. The reconstruction algorithms can be categorized into optimization-based methods and greedy/iterative algorithms. Optimization-based methods [1, 57, 77, 78] usually minimize the ℓ_1 norm under some constraints. These constraints force consistency between sign measurements and their linear measurements. Greedy/Iterative methods in [4, 12, 48] reconstruct the sparse signal by iteratively improving the signal estimate until the specified convergence criterion is satisfied. While many of these approaches estimate the signal as a point in a high-dimensional space, some works have addressed one-bit compressed sensing from the Bayesian perspective [52, 62, 74, 82, 105]. These methods use probability density function (pdf) as a prior on the signal to impose sparse structure and approximate the posterior pdf of the sparse signal rather than providing a point estimate. These posterior pdfs can be used to perform minimum mean square error and maximum *a posteriori* estimation of the sparse signal. All these methods, both Bayesian and non-Bayesian, require sparse structure on the signal. However, in recent works [46, 50, 66, 80] authors have extended the framework of one-bit CS to other low-dimensional structures. Specifically, in [50, 66, 80], the low-dimensional signals are assumed to lie on the range space of trained neural networks, whereas in [46], the signal is assumed to lie on a manifold. In most of these one-bit CS signal reconstruction algorithms, the authors have assumed noiseless one-bit measurements or noisy measurements where the noise is introduced either before or after quantization. In a general noisy one-bit compressed sensing setup, noise is introduced both before quantization and after quantization. The signal reconstruction performance decreases with the presence of noise and can be improved by using side-information (SI). Authors in [76, 79] have exploited the side-information available at the

receiver to improve the sparse signal reconstruction performance. In [76] and [79], the authors assume partial knowledge and complete knowledge of the support set of the sparse signals as the side-information, respectively. These works assume that the SI at the receiver does not have erroneous information. The performance of these algorithms suffers when SI is imperfect. So, we look at the problem of general noisy one-bit compressed sensing where the noise is introduced at two stages, i.e., before quantization and after quantization when a) the signal is sparse and b) the signal lies on the range space of a neural network. Further, we aim to improve the reconstruction performance of noisy one-bit CS by incorporating SI. In contrast to the existing works, we consider noisy one-bit CS measurements with erroneous SI at the receiver and develop reconstruction algorithms.

Beyond signal reconstruction, we are also interested in inference tasks from one-bit compressed measurements. We study the sparse support estimation problem in a distributed setup and provide the lower bound on the number of one-bit measurements required for support set estimation with vanishing probability of error. Several works have tackled different inference problems such as detection, estimation, and classification using one-bit compressed measurements. These inference works may not require entire signal reconstruction. Authors in [61, 92, 101, 102] have studied the problem of sparse signal detection, whereas the authors in [34, 35, 45] have explored the problem of sparse signal parameter estimation performance from one-bit compressed measurements. The works in [40, 65, 106] have studied the performance of compressive classification from one-bit measurements.

One-bit compressed sensing has been a very fertile research field, and several problems have been studied beyond signal reconstruction and inference tasks. For example, authors have studied the performance of one-bit CS algorithms when different sensing matrices are used other than the Gaussian matrices. The works in [1] and [24] have studied the performance of one-bit compressed sensing using subgaussian and partial Gaussian circulant sensing matrices, respectively. Further, dictionary learning [103] and sensing matrix design [39] are a few other

directions that have been explored in the domain of one-bit compressed sensing.

1.4 Major Contributions

The goal of this dissertation is to develop algorithms for parameter estimation and/or signal reconstruction from one-bit compressed sensing measurements. In the following, we list the major contributions of the dissertation:

- We investigate the problem of joint sparse support recovery from one-bit compressed measurements. In this work, we analyze the performance of the maximum likelihood (ML) based decoder to estimate the joint support set. We provide a lower bound on the number of one-bit measurements required per node to estimate the joint support set with a vanishing probability of error when multiple measurement vectors are available at a centralized fusion center (FC). We then develop tractable algorithms in centralized and decentralized settings to recover the joint-support set of the sparse signal. We show the superior performance of the proposed algorithms through numerical simulations.
- We consider the problem of one-bit compressed sensing in the presence of noise from the Bayesian framework. The one-bit measurements we consider include both pre-quantization and post-quantization noise. We then extend the problem to a setup where the receiver has access to a side-information (SI). We investigate two different scenarios of side-information: a. Homogeneous SI and b. Heterogeneous SI.

Homogeneous SI: In this setup, we study two different scenarios. In the first scenario, we assume that the SI is a noisy version of the actual signal. We investigate the cases when the SI has support information only or both amplitude and support information. In the second scenario, we assume that the sparse signal and the SI are dependent and use the Birth-Death-Drift (BDD) model to account for the dependence.

Heterogeneous SI: In this case, the signal and the heterogeneous SI are signals from

different modalities and are modeled using different priors. In this case, we use copula functions to model the dependence between the SI and the signal.

In all these scenarios, we develop reconstruction algorithms using the generalized approximate message passing (GAMP) framework. We provide closed-form expressions for the evaluation of all the non-linear update equations in the GAMP algorithm. This makes the proposed algorithms more computationally efficient. We provide numerical results to demonstrate the superior performance of the proposed algorithms.

- We develop two one-bit CS algorithms which use feed-forward neural networks as priors on the compressed signal. In the first algorithm, we use a trained neural network as a prior. The trained neural networks are generative models such as VAEs and GANs. We formulate an optimization problem and use a gradient descent-based algorithm to optimize the representation learned by the model that matches the given measurements. We establish that, as long as gradient descent finds a good approximate solution to our optimization problem, the algorithm output will be close to the true vector in the range of the generator. The second algorithm uses an untrained neural network as a prior on a signal. We formulate an optimization problem which when solved, optimizes untrained neural networks to reconstruct the signal from its one-bit measurements. A projected gradient descent-based algorithm is used to update the weights of the neural network to reconstruct the signal from one-bit measurements. We provide detailed numerical results to demonstrate the superior performance of the proposed algorithms.

1.5 Organization of the Dissertation

The dissertation is organized into seven chapters. Chapter 2 introduces the problem of sparse support estimation using the multiple one-bit noisy measurement vectors (MMVs). We provide a lower bound on the number of measurements required per node to minimize the estimation

error using a maximum likelihood estimator. As the evaluation of the maximum likelihood estimator is computationally intractable for signals with large dimensions, we propose tractable algorithms for sparse support estimation in a centralized setting. We then extend these algorithms to a decentralized environment. We provide numerical experiments to demonstrate the lower bound of the ML estimator and the performance of the proposed algorithm.

In Chapter 3, we study the problem of sparse signal reconstruction from one-bit measurements in the presence of SI. We first pose the signal reconstruction problem in a Bayesian framework and reconstruct the signal by computing the MMSE estimator. We present the system model considered in which one-bit measurements obtained are corrupted by both additive noise and sign-flip noise. We first develop a reconstruction algorithm using noisy one-bit measurements. We then extend the results to the scenario where the receiver has access to SI. We use Laplacian distribution to model noise between the side-information and the signal. As the MMSE estimator is computationally intractable, we develop a tractable algorithm using a GAMP-based algorithm. We provide closed-form expressions for all of the non-linear equations required to approximate the MMSE estimator. We provide numerical experiments to demonstrate the superior performance of the proposed algorithm.

In Chapter 4, we extend the problem considered in Chapter 3 to a setup where sparse signals have temporal dependence. We model the stochastic dependence of these signals by using the Birth-Death-Drift (BDD) [69, 83] model. We assume that the amplitudes of the sparse signal that continues to be non-zero between the consecutive time instant signals are positively correlated. In this setup, we develop a computationally tractable algorithm to approximate the MMSE estimator of the signal. We show the effectiveness of the proposed algorithm through numerical simulations.

In Chapter 5, we consider the problem of SI-aided signal reconstruction when the SI available at the receiver is heterogeneous. We assume that the SI available at the receiver is of a different modality than the compressed signal. The distribution of these signals could be differ-

ent, and these signals may not have a joint sparse representation. We model dependence among these signals of different modalities using Copula functions. We develop a tractable algorithm in this setup. Numerical results are provided to demonstrate the superior performance of the proposed algorithm.

In Chapter 6, we consider the problem of signal reconstruction from one-bit measurements when deep learning-based networks are used as the prior on the signal. First, we investigate the problem that uses trained generative models such as generative adversarial networks (GANs) and Variational Autoencoders (VAEs) as a prior on the signal during the signal reconstruction. In this setup, we provide sample complexity and an upper bound on the reconstruction error. We also provide numerical results that demonstrate the superiority of the proposed method. Second, we investigate the signal reconstruction performance from one-bit measurements numerically when an untrained neural network is used as a prior on the signal. We formulate an optimization problem for this task and estimate the compressed signal by minimizing the loss function using the projected gradient descent-based method. Numerical results are provided to demonstrate the superior reconstruction performance.

Finally, in Chapter 7, we summarize the findings and results of this dissertation. Several directions and ideas for future research are also presented.

1.6 Bibliographic Note

Most of the work presented in this dissertation has appeared in the following publications

Journal Papers

1. Swatantra Kafle, Vipul Gupta, Bhavya Khailkhura, Thakshila Wimalajeewa, and Pramod K. Varshney, "Joint sparsity pattern recovery with 1-b compressive sensing in distributed sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp.15-30, March 2019.

2. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy one-bit compressed sensing with side-information," *IEEE Transactions on Signal Processing*, submitted, 2020.
3. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy one-bit compressed sensing with side-information in Time-varying Environments," *IEEE Transactions on Signal Processing*, submitted, 2021.
4. Swatantra Kafle, Geethu Joseph, and Pramod K. Varshney, "One-bit compressed sensing with Generative models", *IEEE Transactions on Signal Processing*, In preparation, 2021

Conference Papers

1. Swatantra Kafle, Bhavya Khaikhura, Thakshila Wimalajeewa, and Pramod K. Varshney, "Decentralized joint sparsity pattern recovery using 1-bit compressed sensing," in *Proc. IEEE Global Conference on Signal and Information Processing(GlobalSIP)*, 2016.
2. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Generalized approximate message passing for noisy 1-bit compressed sensing with side-information," in *Proc. 52nd IEEE Asilomar Conference on Signals Systems and Computer*, 2019.
3. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy 1-bit compressed sensing with heterogeneous side-information," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
4. Swatantra Kafle, Geethu Joseph, and Pramod K. Varshney, "One-bit Compressed Sensing with Untrained Prior", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021

CHAPTER 2

JOINT SPARSITY PATTERN RECOVERY IN A DISTRIBUTED NETWORK

2.1 Introduction

In this chapter, we investigate the problem of joint support recovery from one-bit compressed measurements. We assume that signals across the distributed networks have joint sparse support. First, we consider a centralized setting where the fusion center (FC) has access to one-bit compressed measurement vectors from all the nodes and develop the maximum likelihood (ML) decoder for sparsity pattern recovery. We determine the sufficient condition to have vanishing probability via the ML approach. This gives us a lower bound on the number of one-bit measurements per node for vanishing probability of error for joint support recovery. Though the ML-based estimation provides optimal results on joint sparse support estimation, the approach becomes intractable as the signal dimension and the number of sensors increase. Hence, we propose two tractable algorithms for joint sparse support estimation in a centralized setting. We extend these centralized algorithms to a decentralized setting. We illustrate the performance of the proposed algorithms with numerical experiments.

2.2 Observation Model

We consider a distributed network with multiple nodes that observe sparse signals. Let the number of sensors be P . At a given node, consider the following $M \times 1$ real valued observation vector collected via random projections:

$$\mathbf{y}_p^r = \mathbf{A}_p \mathbf{s}_p + \mathbf{v}_p \quad (2.1)$$

where \mathbf{A}_p is the $M \times N$ ($M < N$) measurement matrix at the p -th node for $p = 1, \dots, P$. For each p , the entries of \mathbf{A}_p are assumed to be drawn from a Gaussian ensemble with mean zero. The sparse signal vector of interest, \mathbf{s}_p for $p = 1, \dots, P$, has only $K (\ll N)$ nonzero elements with the same support and N is the signal dimension. The measurement noise vector, \mathbf{v}_p , at the p -th node, is assumed to be independent and identically distributed (i.i.d.) Gaussian with mean vector $\mathbf{0}$ and covariance matrix $\sigma_v^2 \mathbf{I}_M$. We assume that \mathbf{y}_p is quantized element-wise into one of the two levels resulting in one-bit per element:

$$r_{ip} = \begin{cases} -1, & \text{if } \tau_0 < y_{ip}^r < \tau_1 \\ +1, & \text{if } \tau_1 \leq y_{ip}^r < \tau_2 \end{cases} \quad (2.2)$$

where y_{ip}^r is the (i) -th element of \mathbf{y}_p^r , for $i = 1, 2, \dots, M$. In particular, we take sign measurements of the signal and, therefore, zero acts as the quantizer threshold, i.e., $\tau_0 = -\infty, \tau_1 = 0$ and $\tau_2 = \infty$. We assume that, in a centralized setting, all the distributed nodes send their one-bit compressive measurements to a centralized FC. Let \mathbf{y} denote the matrix that includes the multiple measurement vectors obtained from distributed nodes. The matrix is of size $M \times P$ in which the p -th column is \mathbf{y}_p for $p = 1, 2, \dots, P$. Similarly, let \mathbf{S} be the $N \times P$ matrix which contains \mathbf{s}_p as its p -th column for $p = 1, 2, \dots, P$. We assume that, in a centralized setting, all the distributed nodes send their one-bit compressive measurements to a

centralized FC. Let \mathbf{R} be the matrix received at the FC (which is also an $M \times P$ matrix).

$$\mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1P} \\ \vdots & \ddots & \vdots \\ y_{M1} & \cdots & y_{MP} \end{bmatrix} \quad (2.3)$$

where $y_{ip} = r_{ip} + w_{ip}$, for $i = 1, 2, \dots, M$ and $p = 1, 2, \dots, P$; and w_{ip} is the noise at the decoder which is assumed to be an i.i.d. Gaussian with mean zero and variance σ_w^2 .

The problem of joint support recovery based on one-bit CS reduces to the detection of the nonzero rows of \mathbf{S} based on \mathbf{R} in (2.3). First, we obtain the performance bounds when we assume that support recovery is performed using the ML decoder. It is noted that the implementation of the ML decoder becomes computationally intractable as N increases. However, the performance bounds obtained via the ML algorithm can be considered as a benchmark while comparing suboptimal computationally tractable algorithms for sparsity recovery with one-bit CS. Next, we develop computationally tractable algorithms to estimate the common sparse support in centralized as well as decentralized settings.

2.3 Joint Sparse Support Recovery with One-Bit Compressed Sensing via the Maximum Likelihood Decoder

2.3.1 Probability of Error and Maximum Likelihood Decoder

Define the support set of the signal \mathbf{s}_p to be $\mathcal{U} := \{i \in \{1, 2, \dots, N\} \mid \mathbf{s}_p(i) \neq 0, p \in 1, 2, \dots, P\}$. The support of the sparse signals is one of the possible subsets of size K from $\binom{N}{K}$ possible subsets, where $\binom{N}{K}$ number of combinations of K elements when chosen from N . Assuming that each subset occurs with equal probability, the probability of error for any

decoder θ that maps the quantized observation matrix \mathbf{Y} to an estimated support $\hat{\mathcal{U}}$ is given by,

$$P_{err} = \frac{1}{N_0} \sum_{k=1}^{N_0} Pr(\theta(\mathbf{Y}) \neq \mathcal{U}_k | \mathcal{U}_k)$$

where $N_0 = \binom{N}{K}$.

Let $\tilde{\mathbf{S}}$ denote the $K \times P$ matrix which contains the non-zero elements of \mathbf{S} . We assume that each column of $\tilde{\mathbf{S}}$ is distributed as $\mathbf{s}_p \sim \mathcal{N}(\boldsymbol{\mu}, \sigma_s^2 \mathbf{I}_K)$. The ML decoder selects the true support of the signal \mathbf{S} as,

$$\hat{\mathcal{U}} = \underset{\mathcal{U}_k, k=1, \dots, N_0}{\operatorname{argmax}} p(\mathbf{Y} | \mathcal{U}_k) \quad (2.4)$$

where $p(\mathbf{Y} | \mathcal{U}_k)$ is the joint probability density function (pdf) of the observation matrix \mathbf{Y} given the support \mathcal{U}_k where $\mathcal{U}_k \subset \{1, 2, \dots, N\}$ with $|\mathcal{U}_k| = K$.

2.3.2 Sufficient Conditions for Support Recovery

Since the evaluation of the exact probability of error of the ML decoder is difficult in general, we calculate an upper bound on the probability of error based on union and Chernoff bounds. The probability of error of the ML decoder is upper bounded by,

$$\begin{aligned} P_{err} &= \frac{1}{N_0} \sum_{k=0}^{N_0-1} Pr(\hat{\mathcal{U}} \neq \mathcal{U}_k | \mathcal{U}_k) \\ &\leq \frac{1}{2N_0} \sum_{k=0}^{N_0-1} \sum_{j=0, j \neq k}^{N_0-1} \exp(-\hat{\mathcal{C}}(\alpha_0; p_k, p_j)) \end{aligned}$$

where $\hat{\mathcal{C}}(\alpha_0; p_k, p_j) = \max_{0 \leq \alpha \leq 1} \mathcal{C}(\alpha; p_k, p_j)$ with $\mathcal{C}(\alpha; p_k, p_j)$ being the Chernoff distance between the two distributions $p(\mathbf{R} | \mathcal{U}_k)$ and $p(\mathbf{R} | \mathcal{U}_j)$ which is defined as,

$$\mathcal{C}(\alpha; p_k, p_j) = -\log \left\{ \tilde{\mathcal{C}}(\alpha; p_k, p_j) \right\} \quad (2.5)$$

where $\tilde{\mathcal{C}}(\alpha; p_k, p_j) = \int p(\mathbf{R}|\mathcal{U}_k)^{1-\alpha} p(\mathbf{R}|\mathcal{U}_j)^\alpha dr$. In a manner similar to that in [94], it can be shown that with N, K , and M sufficiently large P_{err} is upper bounded by

$$P_{err} \leq \frac{1}{2} \sum_{t=0}^{K-1} \binom{K}{t} \binom{N-K}{K-t} \prod_{p=1}^P (\bar{g}_{pt})^M \quad (2.6)$$

where $\bar{g}_{pt} = \mathbb{E}\{g_{p1}^{\frac{1}{2}}(\mathcal{U}_j, \mathcal{U}_k) \mid (|\mathcal{U}_j \cap \mathcal{U}_k| = t)\}$,

$$g_{pi}^{\frac{1}{2}}(\mathcal{U}_j, \mathcal{U}_k) = \sum_{l=\{0,1\}} \int \mathcal{N}(r_{ip}; l, \sigma_w^2) [\lambda_{pi}^l - \lambda_{pi}^{l+1}] \left(\frac{\sum_{m=\{0,1\}} e^{\frac{mr_i - m^2/2}{\sigma_w^2}} [\lambda_{pij}^m - \lambda_{pij}^{m+1}]}{\sum_{n=\{0,1\}} e^{\frac{nr_i - n^2/2}{\sigma_w^2}} [\lambda_{pij}^n - \lambda_{pij}^{n+1}]} \right)^{\frac{1}{2}} dr_{ip}, \quad (2.7)$$

$\lambda_{pi}^l = Q \left(\frac{\tau_l - \sum_{j=1}^K (\tilde{\mathbf{A}}_{\mathcal{U}_k}^p)_{ij} \mu_j^p}{\sqrt{\sigma_v^2 + \sigma_s^2 \sum_{j=1}^K (\tilde{\mathbf{A}}_{\mathcal{U}_k}^p)_{ij}^2}} \right)$, $\tilde{\mathbf{A}}_{\mathcal{U}_k}^p$ is a $M \times K$ submatrix of \mathbf{A}_p at the p -th node such that $\tilde{\mathbf{A}}_{\mathcal{U}_k}^p \tilde{\mathbf{s}}_p = \mathbf{A}_p \mathbf{s}_p$ when the support of the signal \mathbf{s}_p is \mathcal{U}_k , and $Q(p) = \int_p^{+\infty} \mathcal{N}(x; 0, 1) dx$.

When $\sigma_w^2 \rightarrow 0$, and when parameters N, M , and K are sufficiently large, the right hand side of (2.6) is upper bounded by

$$P_{err} \leq \frac{1}{2} \sum_{t=0}^{K-1} \binom{K}{t} \binom{N-K}{K-t} \prod_{p=1}^P (\bar{a}_{pt})^M \quad (2.8)$$

where $\bar{a}_{pt} = \sum_{l=\{0,1\}} \mathbb{E}\{\tilde{a}_{pt}^l\}$ and $\mathbb{E}\{\tilde{a}_{pt}^l\} = \mathbb{E}\{(\lambda_{p1j}^l - \lambda_{p1j}^{l+1})^{\frac{1}{2}} (\lambda_{p1k}^l - \lambda_{p1k}^{l+1})^{\frac{1}{2}} \mid (|\mathcal{U}_j \cap \mathcal{U}_k| = t)\}$ and λ_{p1k} is given by

$$\lambda_{p1k} = Q \left(\frac{-\sum_{i=1}^K (\tilde{\mathbf{A}}_{\mathcal{U}_k}^p)_{1i} \mu_i}{\sigma_v^2 + \sigma_s^2 \sum_{i=1}^K (\tilde{\mathbf{A}}_{\mathcal{U}_k}^p)_{1i}^2} \right). \quad (2.9)$$

The sufficient conditions for reliable recovery of the sparsity pattern with one-bit quantized observations (5.8) when $\sigma_w^2 \rightarrow 0$ are stated in the following theorem:

Theorem 2.1. In the high dimensional setting (such that the parameters N, K, M are suf-

ficiently large), for one-bit quantized compressive measurements, the sufficient condition to have a vanishing probability of error asymptotically when using the ML decoder for support recovery of the sparse signal \mathbf{s} is:

$$M \geq \max \{f_0(N, K, P, \gamma), \dots, f_{K-1}(N, K, P, \gamma)\} \quad (2.10)$$

where

$$f_t(N, K, P, \gamma) = \frac{1}{\log \frac{1}{\prod_{p=1}^P \bar{a}_{pt}}} \times \left[(K-t) \left(2 + \log \frac{K}{K-t} + \log \frac{(N-K)}{K-t} \right) + \log \frac{1}{2} \right],$$

$\gamma = \frac{\boldsymbol{\mu}^T \boldsymbol{\mu} + K \sigma_s^2}{N \sigma_v^2}$ and \bar{a}_{pt} is as defined in Equation (2.8) for $t = 0, 1, \dots, K-1$.

Proof. The probability of error for sparse support recovery with one-bit quantized measurements is upper bound by

$$P_{err} \leq \frac{1}{2} \sum_{t=0}^{K-1} \binom{K}{t} \binom{N-K}{K-t} \prod_{p=1}^P (\bar{a}_{pt})^M \quad (2.11)$$

We want to find the conditions under which the bounds on probability of error in (2.11) tends to zero. Let lP_t be the logarithm of the t -th term in (2.11) which is given by

$$\begin{aligned} lP_t &= \log \binom{K}{t} + \log \binom{N-K}{K-t} + \log \frac{1}{2} - M \log \frac{1}{\prod_{p=1}^P (\bar{a}_{pt})} \\ &= \log \binom{K}{K-t} + \log \binom{N-K}{K-t} + \log \frac{1}{2} - M \log \frac{1}{\prod_{p=1}^P (\bar{a}_{pt})} \end{aligned} \quad (2.12)$$

Next, upper bounding $\binom{K}{K-t}$ by $\left(\frac{Ne}{K}\right)^K$, we get

$$lP_t \geq (K-t) \left(2 + \log \frac{K}{t} + \log \frac{N-K}{K-t} \right) + \log \frac{1}{2} - M \log \frac{1}{\prod_{p=1}^P (\bar{a}_{pt})} \quad (2.13)$$

For lP_t to be asymptotically negative,

$$M \geq \frac{1}{\log \frac{1}{\prod_{p=1}^P (\bar{a}_{pt})}} \left[(K-t) \left(2 + \log \frac{K}{t} + \log \frac{N-K}{K-t} \right) + \log \frac{1}{2} \right] \quad (2.14)$$

The maximum value of M for all t will tend P_{err} to zero.

□

The lower bound in (2.10) explicitly shows the minimum number of measurements required to recover the sparsity pattern of the sparse signal with one-bit CS using the ML decoder with multiple measurement vectors. Note that, the value of $\bar{a}_{pt}(\gamma, K)$ also depends on the sparsity index K and the measurement SNR γ . Further, using the definition of λ_{pik} and assuming finite σ_v^2 and σ_s^2 , it can be shown that $0 < \bar{a}_{pt} < 1$ which ensures that $\log \frac{1}{\prod_{p=1}^P \bar{a}_{pt}}$ is positive and its magnitude increases with an increase in P . Thus, the bound on M for asymptotically vanishing probability of error decreases with an increase in the number of sensors P .

Theorem 2.2. Let N, K , and M be sufficiently large and $\sigma_w^2 \rightarrow 0$. The lower bound on M for vanishing probability of error using the ML decoder for joint support recovery of the sparse signals with P one-bit MMVs is:

$$MP \geq C_K K \log \frac{N}{K} \quad (2.15)$$

where $C_k = \frac{1}{\log \frac{1}{a_k}}$, and $a_K = \max_{0 \leq t \leq K-1} \bar{a}_{pt,2}$.

Proof. See Appendix A.

□

- *Remark 1:* It is worth noting that the authors in [94] derived bounds on the number of measurements with a single measurement vector for vanishing probability of error. However, the authors had simplified the analysis under the assumption that \tilde{s} is drawn from a 1-st order Gaussian ($\sigma_s^2 = 0$) pdf. Here, we provide results without restricting to that assumption and in an extended setting when MMVs are available at a centralized FC.
- *Remark 2:* From Theorem 2.2, it is sufficient to have $\Omega(C_K \frac{K}{P} \log(\frac{N}{K}))$ measurements to recover the sparsity pattern with noisy corrupted one-bit information of the real valued compressed measurement vectors using the ML decoder. The number of compressed measurements per node M , required to recover the joint sparsity pattern with vanishing P_{err} has an inverse relation with the number of sensors P which is quite intuitive. When $P = 1$, it is sufficient to have $\Omega(C_K K \log(\frac{N}{K}))$ measurements to recover the sparsity pattern reliably which is compatible with the results on the number of measurements required for sparse support recovery obtained in [94] and [78]. When P is increased, M decreases. If P is large enough, say $P \gg N$, then $M < 1$. It means that only single measurements from some nodes of the network are sufficient to recover the joint sparsity pattern reliably. Thus, $M \rightarrow 0$, as $P \rightarrow \infty$.

2.4 Centralized Algorithms with One-Bit Compressed Sensing

In this section, we develop two computationally tractable algorithms for joint sparsity pattern recovery in a centralized setting. In the first algorithm, we use the regularized row l_1 norm minimization approach with the likelihood function as the cost function. In the second algorithm, we extend the BIHT algorithm to the MMV case.

2.4.1 Centralized $\ell_{1,\infty}$ Regularized Maximum Likelihood Based Algorithm

In this algorithm, the ML function is used as the cost function instead of the commonly used least squares function [87, 108]. Sparsity is imposed by using $\ell_{1,\infty}$ regularization on the signal matrix \mathbf{S} to obtain a sparse signal matrix estimate $\hat{\mathbf{S}}$ or the estimate of the support of \mathbf{S} . For the sake of tractability, we assume that the measurement matrix $\mathbf{A}_p = \mathbf{A}$ is the same for all $p = 1, 2, \dots, P$. Then, from (2.1), we have

$$y_{ip}^r = \mathbf{A}_i^T \mathbf{s}_p + v_{ip}, \quad (2.16)$$

for $i = 1, 2, \dots, M$ and $p = 1, 2, \dots, P$. In the rest of the chapter, \mathbf{A}_i denotes the i -th row of \mathbf{A} .

Let the matrix \mathbf{R} be obtained by element-wise quantization in (2.2) where r_{ip} is the (i, p) -th element of \mathbf{Y} . Next, we calculate the probabilities $\Pr(r_{ip} = 1)$ and $\Pr(r_{ip} = 0)$ which will later be used to write the expression of likelihood of \mathbf{R} given \mathbf{S} . We have,

$$\Pr(y_{ip}^r \geq 0) \Rightarrow \Pr(\mathbf{A}_i^T \mathbf{s}_p + v_{ip} \geq 0) = \phi(\mathbf{A}_i^T \mathbf{s}_p / \sigma_v).$$

Similarly,

$$\Pr(y_{ip}^r < 0) \Rightarrow \Pr(\mathbf{A}_i^T \mathbf{s}_p + v_{ip} < 0) = \phi(-\mathbf{A}_i^T \mathbf{s}_p / \sigma_v).$$

where $\phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^x e^{-t^2/2} dt$. The conditional probability of \mathbf{R} given \mathbf{S} , $\Pr(\mathbf{R}|\mathbf{S})$, is given by,

$$\Pr(\mathbf{R}|\mathbf{S}) = \prod_{p=1}^P \prod_{i=1}^M \Pr(r_{ip}|\mathbf{S}) = \prod_{p=1}^P \prod_{i=1}^M \left(\phi\left(\frac{\mathbf{A}_i^T \mathbf{s}_p}{\sigma_v}\right) \right)^{r_{ip}} \times \left(\phi\left(-\frac{\mathbf{A}_i^T \mathbf{s}_p}{\sigma_v}\right) \right)^{(1-r_{ip})}.$$

The negative log-likelihood of \mathbf{R} given \mathbf{S} , $f_{ml}(\mathbf{AS})$, is given by

$$f_{ml}(\mathbf{AS}) = -\log(\Pr(\mathbf{R}|\mathbf{S})) = -\sum_{p=1}^P \sum_{i=1}^M \left[r_{ip} \log \left(\phi \left(\frac{\mathbf{A}_i^T \mathbf{s}_p}{\sigma_v} \right) \right) + (1 - r_{ip}) \log \left(\phi \left(-\frac{\mathbf{A}_i^T \mathbf{s}_p}{\sigma_v} \right) \right) \right] \quad (2.17)$$

which can be rewritten as

$$f_{ml}(\mathbf{H}) = -\sum_{p=1}^P \sum_{i=1}^m \left[r_{ip} \log \left(\phi \left(\frac{h_{ip}}{\sigma_v} \right) \right) + (1 - r_{ip}) \log \left(\phi \left(\frac{-h_{ip}}{\sigma_v} \right) \right) \right], \quad (2.18)$$

and $\mathbf{H} = \mathbf{AS}$. In the following, we use \mathbf{H} and \mathbf{AS} interchangeably. We aim to minimize $f_{ml}(\mathbf{AS})$ as well as incorporate the sparsity condition of the signal matrix \mathbf{S} to obtain an estimated signal matrix $\hat{\mathbf{S}}$ or the support of \mathbf{S} . As the signals observed at all the nodes share the same support, the following matrix norm (as defined in [87] for real valued measurements) is appropriate. We define the row support of the coefficient matrix \mathbf{S} as [87]

$$\text{rowsupp}(\mathbf{S}) = \{w \in [1, N] : s_{wk} \neq 0 \text{ for some } k\},$$

and the row- l_0 quasi-norm of \mathbf{S} is given by,

$$\|\mathbf{S}\|_{\text{row-}l_0} = |\text{rowsupp}(\mathbf{S})|,$$

which is also known as the $l_{0,\infty}$ norm, where $|\cdot|$ denotes the cardinality. To compute \mathbf{S} , we aim to solve the following optimization problem:

$$\arg \min_{\mathbf{S}} \{f_{ml}(\mathbf{AS}) + \lambda \|\mathbf{S}\|_{0,\infty}\} \quad (2.19)$$

where λ is the penalty parameter. However, the problem (2.19) is not tractable in its current

form. The problem can be modified as

$$\arg \min_{\mathbf{S}} \{f_{ml}(\mathbf{A}\mathbf{S}) + \lambda \|\mathbf{S}\|_{1,\infty}\} \quad (2.20)$$

where $\|\mathbf{S}\|_{1,\infty} = \sum_{i=1}^N \max_{1 \leq j \leq P} |s_{ij}|$, i.e., $\|\mathbf{S}\|_{1,\infty}$ is the sum of all the elements with maximum absolute value in each row, also known as the $l_{1,\infty}$ quasi-norm of a matrix. It is noted that, $f_{ml}(\mathbf{A}\mathbf{S})$ and $\|\mathbf{S}\|_{1,\infty}$ are convex functions. The former is the ML function while the latter is a convex relaxation of the row- l_0 quasi-norm [87], therefore the expression in (2.20) is a convex optimization problem.

Algorithm for Solving the Optimization Problem in (2.20)

The goal is to solve the problem of the form

$$\arg \min_{\mathbf{S}} \{f(\mathbf{A}\mathbf{S}) + \lambda g(\mathbf{S})\} \quad (2.21)$$

where $f(\mathbf{A}\mathbf{S}) = f_{ml}(\mathbf{A}\mathbf{S})$ and $g(\mathbf{S})$ is the $l_{1,\infty}$ norm of \mathbf{S} .

The class of *iterative shrinkage-thresholding* algorithms (ISTA) provides one of the most popular methods for solving the problem as defined in (2.21). In ISTA, each iteration involves the solution of a simplified optimization problem, which in most of the cases can be easily solved using the proximal gradient method, followed by a shrinkage/soft-threshold step; for e.g., see [6, 16, 30]. It should be noted that in these papers, ISTA is applied to find an optimum vector which minimizes a given objective function and, therefore, cannot be applied here directly. We extend the idea to find an optimal matrix which is a minimizer of the expression $f(\mathbf{A}\mathbf{S}) + \lambda g(\mathbf{S})$.

From [6], at the k -th iteration we have

$$\mathbf{S}_k = P_{L_f}(\mathbf{S}_{k-1}) \quad (2.22)$$

where

$$P_{L_f}(\mathbf{T}) = \arg \min_{\hat{\mathbf{S}}} \lambda g(\mathbf{S}) + \frac{L_f}{2} \|\mathbf{S} - (\mathbf{T} - \frac{1}{L_f} \nabla f(\mathbf{T}))\|_F^2.$$

Inputs to the algorithm are L_f (the Lipschitz constant of ∇f) and \mathbf{S}_0 , the initialization for the iterative method, which could be assumed to be a null matrix or $\mathbf{A}^\dagger \mathbf{Z}$, where \mathbf{A}^\dagger is the pseudoinverse of \mathbf{A} . For our case, the gradient of $f_{ml}(\mathbf{H})$ w.r.t. matrix \mathbf{S} can be easily calculated as $\mathbf{A}^T \nabla f_{ml}(\mathbf{H})$, where $\mathbf{x} = \mathbf{A}\mathbf{S}$. Notice that, $\nabla f_{ml}(\mathbf{H})$ is the gradient of $f_{ml}(\mathbf{H})$ w.r.t. \mathbf{H} and is given by

$$\nabla f_{ml}(h_{ip}) = \frac{r_{ip} \exp(-\frac{\tilde{h}_{ip}^2}{2})}{\sqrt{2\pi}\sigma_v \phi(\tilde{h}_{ip})} - \frac{(1-r_{ip}) \exp(-\frac{\tilde{h}_{ip}^2}{2})}{\sqrt{2\pi}\sigma_v \phi(-\tilde{h}_{ip})}, \quad (2.23)$$

where $\tilde{h}_{ip} = h_{ip}/\sigma_v$.

The problem defined in (2.22) is row separable for each iteration. Therefore, to solve for \mathbf{S}_k , i.e., to find $P_L(\mathbf{S}_{k-1})$, we divide the problem into N subproblems, where N is the number of rows in \mathbf{S} . Next, we solve the following subproblem for each row of \mathbf{S}_k :

$$\arg \min_{\mathbf{s}^i} \lambda g(\mathbf{s}^i) + \frac{L_f}{2} \|\mathbf{s}^i - (\mathbf{t}^i - \frac{1}{L_f} \nabla f(\mathbf{t}^i))\|_2^2; \quad (2.24)$$

where \mathbf{s}^i , \mathbf{t}^i and $\nabla f(\mathbf{t}^i)$ are the i^{th} rows of \mathbf{S} , \mathbf{S}_{k-1} and $\nabla f(\mathbf{S}_{k-1})$ respectively, and $\|\mathbf{s}\|$ is the ℓ_2 norm of \mathbf{s} , i.e., $\|\mathbf{s}\| = (\sum_i |s_i|)^{\frac{1}{2}}$. Equation (2.24) is of the form:

$$\arg \min_{\mathbf{s}} \left\{ \lambda g(\mathbf{s}) + \frac{L_f}{2} \|\mathbf{s} - \mathbf{u}\|_2^2 \right\}; \quad (2.25)$$

where $g(\mathbf{s}) = \|\mathbf{s}^i\|_\infty$, i.e., the l_∞ norm of the i^{th} row of \mathbf{S} and constant vector \mathbf{u} is given by $\mathbf{u} = \mathbf{t}^i - \frac{1}{L_f} \nabla f(\mathbf{t}^i)$ (we avoid using superscript i on $g(\mathbf{s})$ and \mathbf{u} for brevity).

Introducing an auxiliary variable $t = g(\mathbf{s})$, the problem in (2.25) can be rewritten as

$$\arg \min_{\mathbf{s}} \left\{ t + \frac{1}{2\lambda} \|\mathbf{s} - \mathbf{u}\|_2^2 \right\}, \text{ s.t. } 0 \leq w_p s_p \leq t. \quad (2.26)$$

where $\bar{\lambda} = \frac{\lambda}{L_f}$, $w_p = \text{sign}(u_p)$ and u_p and s_p are the p -th elements in \mathbf{u} and \mathbf{s} respectively, for all $p = 1, 2, \dots, P$. The problem in (2.26) can be solved using Lagrangian based methods. The Lagrangian for (2.26) is defined as

$$L(\mathbf{s}, \alpha, \beta) = \frac{1}{2} \|\mathbf{s} - \mathbf{u}\|_2^2 + \bar{\lambda}t - \sum_i \alpha_i w_i s_i + \sum_i \beta_i (w_i s_i - t),$$

where α and β are the Lagrangian multipliers. Optimality conditions for $1 \leq p \leq P$ for the above equation are

$$\begin{aligned} (s_p - u_p) - \alpha_p w_p + \beta_p w_p &= 0, \\ \bar{\lambda} - \sum_p \beta_p &= 0, \\ \alpha_p (w_p s_p) &= 0, \\ \beta_p (w_p s_p - t) &= 0, \end{aligned} \tag{2.27}$$

for $\alpha_p, \beta_p \geq 0$.

Notice that, if the optimal $t = t^*$ was known, we could use the above conditions to compute the optimum x_p^* via

$$s_p^*(t^*) = \begin{cases} w_p t^* & \text{if } |u_p| \geq t^*; \\ u_p & \text{otherwise.} \end{cases} \tag{2.28}$$

The proof of the above statement follows from the following lemma [86].

Lemma 2.1. *The optimality conditions in (2.27) are satisfied by the solution in (2.28).*

Now, the problem reduces to finding the optimal t^* . To find the optimal t^* , we define the function

$$h(t) = \lambda - \sum_p \beta_p = \lambda + \sum_{p: |u_p| \geq t} w_p (s_p(t) - u_p), \tag{2.29}$$

where $s_p(t)$ is obtained by (2.28) with t instead of t^* . The optimal t^* can be found by solving

the following equation

$$h(t^*) = 0. \quad (2.30)$$

This can be easily solved for t^* by applying a bisection based method using the initial interval $[0, \|u\|_\infty]$. If there does not exist a solution in this interval, i.e., $h(0) \times g(\|u\|_\infty) \geq 0$, the trivial solution is given by $t^* = 0$.

Once we find t^* , the solution to (2.26) is given by (2.28). Each subproblem given by (2.24) can be solved in a similar way, the solution to each of which can be used to find \mathbf{S}_k through Equations (2.22) and (2.4.1). The summary of all the steps is provided in Algorithm 2.1, where $\|\cdot\|_F$ denotes the Frobenius norm. Algorithm 2.1 produces the matrix \mathbf{S}_k and locations of non-zero elements in \mathbf{S}_k yield the estimated support of original signal matrix \mathbf{S} .

Algorithm 2.1 Centralized $\ell_{1,\infty}$ Regularized ML Based Algorithm (MLA)

1. **Given** tolerance $\epsilon > 0$, parameters $\tilde{\lambda} > \lambda$, $0 < \alpha < 1$ and L_f
 2. **Initialize** $\mathbf{S}^0, \mathbf{S}^1$ such that $\mathbf{S}^0 \neq \mathbf{S}^1$, $\hat{\lambda} = \tilde{\lambda}$, $k = 1$
 3. **While** $\hat{\lambda} > \lambda$
 4. $\hat{\lambda} = \alpha \hat{\lambda}$
 5. **While** $\|\mathbf{S}^k - \mathbf{S}^{k-1}\|_F > \epsilon \|\mathbf{S}^{k-1}\|_F$
 6. $k = k + 1$
 7. Define matrix $\mathbf{U} = \mathbf{S}^{k-1} - \frac{1}{L_f} \nabla f(\mathbf{S}^{k-1})$ where $\nabla f(\mathbf{S}^{k-1})$ is computed as in (2.23)
 8. **For** each row of \mathbf{S}^k
 9. Update the p -th row element using (2.30) and (2.28) for $p = 1, 2, \dots, P$
 10. **End For**
 11. **End While**
 12. **End While**
-

2.4.2 Centralized-BIHT

In this section, we extend the BIHT [47] algorithm to the multiple sensor case and refer to it as the Centralized BIHT (C-BIHT) algorithm. For the sake of completeness, we first introduce the BIHT algorithm.

BIHT [47]

The BIHT algorithm is an iterative method that reconstructs a K -sparse signal \mathbf{s} from binary information of compressed measurements from a single sensor. The signal estimate at the k -th iteration with quantized measurements \mathbf{r} is given as

$$\mathbf{s}^k = \Theta_K \left(\mathbf{s}^{k-1} - \tau \mathbf{A}^T (\text{sign}(\mathbf{A} \mathbf{s}^{k-1}) - \mathbf{r}) \right),$$

where Θ_K is a K -ball projection operator which forces all the elements but K with largest magnitudes to zero and τ is the step size. This method is an iterative method where at each iteration an estimate for the support is computed. This estimate is improved in successive iterations.

Algorithm 2.2 Centralized BIHT algorithm (C-BIHT)

Inputs : $\mathbf{A}, K, \mathbf{Q}, \tau$

1. Initialize \mathbf{S}^0
 2. For iteration j until the stopping criteria
 3. $\mathbf{S}^j = \mathbf{S}^{j-1} + \tau \mathbf{A}^T (\mathbf{R} - \text{sign}(\mathbf{A} \mathbf{S}^{j-1}))$
 4. $\mathcal{U}^j = \text{DetectSupport}(\mathbf{S}^j, K)$
 5. $\mathbf{S}^j = \text{Threshold}(\mathbf{S}^j, \mathcal{U}^j)$
 6. End For
 7. $\hat{\mathbf{S}} = \mathbf{S}^{j^*}$ and $\hat{\mathcal{U}} = \mathcal{U}^{j^*}$ when stopping at iteration j^*
-

C-BIHT

In a centralized setting, \mathbf{R} is available at the fusion center. Merging BIHT [47] and the idea of Centralized IHT Algorithm [8], we propose the Centralized BIHT (C-BIHT) algorithm to estimate the joint sparsity pattern using multiple sensors. In Algorithm 2.2, we provide the pseudo code for C-BIHT.

We first initialize the signal estimate to \mathbf{S}^0 . Next, during the j -th iteration the gradient of the cost function is evaluated using \mathbf{S}^{j-1} and a step proportional to the gradient is taken in the negative direction. This step ensures that \mathbf{S}^j moves in the direction where the cost function is minimized. In Step 4, $\text{DetectSupport}(\mathbf{S}^j, K)$ is a function which computes the support of the K -sparse signal matrix. A simple implementation of this function is to select K rows with the largest l_2 norm as the support set. Next, in Step 5, the function $\text{Threshold}(\mathbf{S}^j, \mathcal{U}^j)$ forces all the rows of \mathbf{S}^j matrix to zero except for the indices in support \mathcal{U}^j . In other words, this step is a hard thresholding operation which forces matrix \mathbf{S}^j to be K -row sparse. These iterations are continued until the stopping criterion is satisfied (such as minimum squared error). The support estimate of the final iteration is the estimated support.

Since the algorithms developed in the centralized setting take into account all the measurements from multiple sensors, they expect to have better support recovery performance compared to algorithms that take measurements from a single sensor. However, a centralized system is not always feasible when the network is large or resource constrained. In the next section, we propose algorithms to solve the sparsity recovery problem in a decentralized manner.

Algorithm 2.3 Decentralized- Row- ℓ_1 Regularized ML Based Algorithm (DMLA)

1. **Given** tolerance $\epsilon > 0$, parameters $\tilde{\lambda} > \lambda$, $0 < \alpha < 1$ and L_f
 2. **Initialize** $\mathbf{S}_p^0, \mathbf{S}_p^1$ such that $\mathbf{S}_p^0 \neq \mathbf{S}_p^1$, $\hat{\lambda} = \tilde{\lambda}$, $k = 1$ for $p = 1, \dots, P$
 3. Local Communication at node p , for all $p \in \mathcal{V}$
 - Transmit \mathbf{r}_p to its one-hop neighborhood
 - Receive \mathbf{r}_r where $r \in \text{neigh}(p)$ and form \mathbf{r}_p
 4. $\mathcal{U}_p = \text{MLA}(\mathbf{r}_p)$ for $p = 1, \dots, P$
 5. Global Communication
 - (a) For all $p \in \mathcal{V}$, transmit $\hat{\mathcal{U}}_p$ to \mathcal{V}
 - (b) Receive $\hat{\mathcal{U}}_i$ for $\forall i \neq p$
 6. $\hat{\mathcal{U}} = \text{Majority}(\hat{\mathcal{U}}_1, \hat{\mathcal{U}}_2, \dots, \hat{\mathcal{U}}_P)$
-

2.5 Decentralized Algorithms with One-bit Compressed Sensing

In this section, the algorithms developed for the centralized case are extended to the decentralized settings. A decentralized network is modeled as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the node set $\{1, \dots, P\}$. The set of communication links in the network corresponds to the set of edges \mathcal{E} . An edge exists between the i -th node and the j -th node, if and only if there is a communication link between them (so that, they can directly communicate with each other). We also define by $\text{neigh}(i) = \{j \mid (i, j) \in \mathcal{E}\}$ the set of neighboring nodes of node i . The p -th node observes a sparse signal \mathbf{s}_p , compresses it, quantizes it to one-bit and shares measurements \mathbf{r}_p with its one-hop neighbors $\text{neigh}(p)$. The p -th node also receives the measurements from its neighboring nodes. Thus, the p -th node has access to the measurement matrix $\mathbf{R}_p = [\mathbf{R}_{\text{neigh}(p)}, \mathbf{r}_p]$, where $\mathbf{R}_{\text{neigh}(p)}$ is a matrix of \mathbf{r}_r for all $r \in \text{neigh}(p)$. Let the local support estimate at the p -th node based on the available information be $\hat{\mathcal{U}}_p$. Now, the goal of the p -th node is to collaborate with its one-hop neighbors to recover the common sparsity

pattern.

In this subsection, we extend Algorithm 2.1 to the decentralized setting for joint support recovery. In Algorithm 2.3, we provide the pseudo code for DMLA.

In Algorithm 2.3, each node first initializes the local signal estimate to the matrix \mathbf{S}_p^0 of size $N \times \text{card}(\text{neigh}(p)) + 1$ where $\text{card}(\cdot)$ denotes the cardinality of the corresponding set. There are two fusion stages in DMLA: 1) information fusion, and 2) index fusion.

Information Fusion Stage

The p -th node estimates the support set, $\hat{\mathcal{U}}_p$, based on \mathbf{R}_p , in a manner similar to FC in a centralized setting. However, the size of the problem each node solves is usually smaller than the one solved by FC in the centralized algorithm because measurements are available only from 1-hop neighbors. Note that, $\text{MLA}(\mathbf{R}_p)$ in the algorithm is a shorthand representation of the steps that the p -th node executes to estimate the support set of the signal. These steps are similar to steps 3 through 12 of Algorithm 2.1.

Index Fusion Stage

The p -th node forwards $\hat{\mathcal{U}}_p$ to all the nodes in the network. Similarly, p -th node receives $\hat{\mathcal{U}}_q, q \neq p$. Each node then decides the final sparsity pattern based on the Majority fusion rule implemented in the $\text{Majority}(\hat{\mathcal{U}}_1, \hat{\mathcal{U}}_2, \dots, \hat{\mathcal{U}}_P)$ function. A simple implementation of this function is to count the K -most repetitive indices among all the estimated support sets. The probability that two or more nodes estimate the same indices that do not belong to the sparsity pattern after solving step 5 is very small, especially when $N \gg K$.

2.5.1 Decentralized-BIHT

In this subsection, we propose a decentralized algorithm based on the BIHT algorithm for sparsity pattern recovery. In Algorithm 2.4, we provide the pseudo code for D-BIHT. This algorithm

also has information and index fusion stages as in DMLA.

Algorithm 2.4 Decentralized BIHT (D-BIHT)

Inputs : $\mathbf{A}, K, \tau, \text{neigh}(p)$ for all $p \in \mathcal{V}$

1. Initialize \mathbf{S}_p^0 for all $p \in \mathcal{V}$
 2. Local Communication at node p , for all $p \in \mathcal{V}$
 - Transmit \mathbf{r}_p to its one-hop neighborhood
 - Receive \mathbf{r}_r where $r \in \text{neigh}(p)$ and form \mathbf{R}_p
 3. For iteration j until the stopping criteria
 4. $\mathbf{S}_p^j = \mathbf{S}_p^{j-1} + \tau \mathbf{A}^T (\mathbf{R}_p - \text{sign}(\mathbf{A} \mathbf{S}_p^{j-1}))$
 5. $\mathcal{U}_p^j = \text{DetectSupport}(\mathbf{S}_p^j, K)$
 6. $\mathbf{S}_p^j = \text{Threshold}(\mathbf{S}_p^j, \mathcal{U}_p^j)$
 7. End For
 8. $\hat{\mathbf{S}}_p = \mathbf{S}_p^{j^*}$ and $\hat{\mathcal{U}}_p = \mathcal{U}_p^{j^*}$ when stopping at iteration j^*
 9. Global Communication
 - (a) For all $p \in \mathcal{V}$, transmit $\hat{\mathcal{U}}_p$ to \mathcal{V}
 - (b) Receive $\hat{\mathcal{U}}_i$ for $\forall i \neq p$
 10. $\hat{\mathcal{U}} = \text{Majority}(\hat{\mathcal{U}}_1, \hat{\mathcal{U}}_2, \dots, \hat{\mathcal{U}}_P)$
-

Information Fusion Stage

In this stage, the p -th node collects quantized compressed measurements from its one-hop neighbors and forms \mathbf{R}_p . The p -th node then uses the \mathbf{R}_p to estimate the support, $(\hat{\mathcal{U}}_p^j)$, and the signal matrix, \mathbf{S}_p^j , from steps 3 through 7.

Index Fusion Stage

In this stage the p -th node receives estimates of the support set from all the other nodes in the network. Each node decides on the sparsity pattern using the Majority fusion rule implemented

in majority function. This stage is the same as the index fusion stage of DMLA. The difference between the final performance of DMLA and D-BIHT is due to the difference in performance of their information fusion stage in estimating $\hat{\mathcal{U}}_p$.

Decentralized BIHT modified (D-BIHTm):

For a resource constrained network that has very severe restrictions on bandwidth usage and/or computation capacity (power constraint), we simplify Algorithm 2.4. In particular, Algorithm 2.4 is modified by omitting the Information Fusion stage. Each node obtains an estimate of the sparsity pattern, $\hat{\mathcal{U}}_p, p = 1, \dots, P$, based on only its information via the BIHT algorithm. This stage is referred to as the Self Decision Stage. Thus, the communication overhead/bandwidth of the network and the computational cost at each node is reduced. The next stage is the index fusion stage where the final estimate is obtained by global fusion as in Algorithm 2.4. This special case of D-BIHT is referred to as the Decentralized BIHT modified algorithm.

2.6 Simulation Results

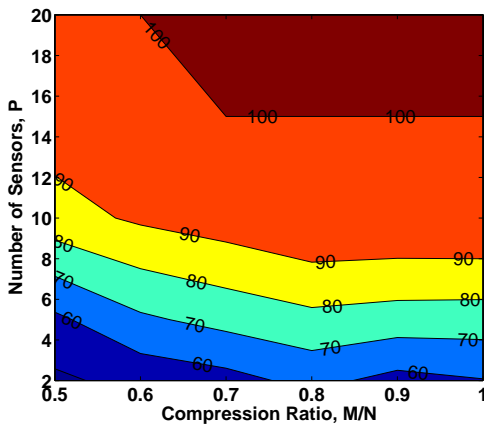


Fig. 2.1: PSPR for MLA when $\eta = -3.01$ dB, $N = 100$ and $K = 5$ in a network of 10 nodes.

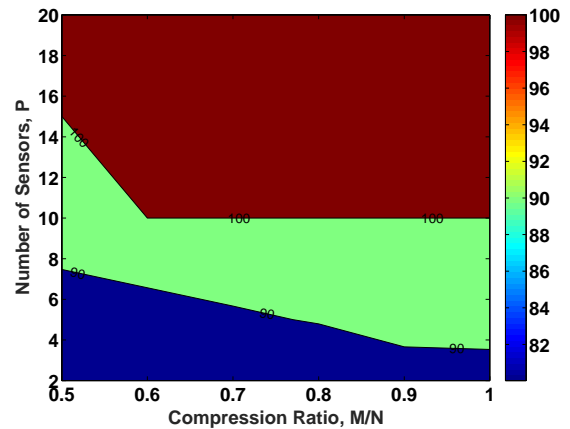


Fig. 2.2: PSPR for C-BIHT when $\eta = -3.01$ dB, $N = 100$ and $K = 5$ in a network of 10 nodes.

In this section, we evaluate the performances of the proposed centralized and decentralized algorithms through numerical simulations. We consider the percentage of sparsity pattern recovered (PSPR) correctly and the probability of exact sparsity pattern recovery (PESPR) as the performance metrics. They are defined as

$$\begin{aligned} \text{PSPR} &= \frac{\# \text{ of support elements out of } K \text{ that are recovered correctly}}{K} \times 100 \\ \text{PESPR} &= \frac{\# \text{ of Monte Carlo runs in which true support is recovered}}{\# \text{ of Monte Carlo runs}} \end{aligned}$$

respectively. We obtain the performances of the proposed algorithms for different values of compression ratios (M/N) and noise variances (σ_v^2). We generate a signal vector of length $N = 100$ with sparsity index, $K = 5$. For each M , we generate the elements of the $M \times N$ measurement matrix \mathbf{A} from a normal distribution with mean zero and unit variance. The sparse support set for the signal is selected from $[1, N]$ uniformly. The amplitudes of the signals in the support set are generated from an i.i.d. Gaussian distribution with zero mean and unit variance. The signals are assumed to remain constant over all Monte Carlo runs. The measurement noise at each node is i.i.d. Gaussian with zero mean. The variance of the noise vector, \mathbf{v} , is set such that $\mathbb{E}(\|\mathbf{v}\|_2^2)$ is a constant. We define the total noise power as $\eta = 10 \log_{10}(\mathbb{E}(\|\mathbf{v}\|_2^2))$ dB. We compare our proposed algorithms with algorithms proposed in [104] and [52] and refer to them as ImpNoise and 1bitGAMP, respectively. We ran C-BIHT, MLA, D-BIHT, DMLA, D-BIHTm, ImpNoise, 1bitGAMP, and BIHT algorithms for 1000 Monte Carlo runs. In simulations, we use step size $\tau = 1$ for C-BIHT, D-BIHT and D-BIHTm algorithms. We ran all the proposed decentralized algorithms over a network with 10 nodes each of which

is assumed to be of degree 3.

We first compare the performances of the two proposed centralized algorithms, MLA and C-BIHT. Our results for MLA and C-BIHT are shown in Figures 2.1 and 2.2 respectively. We can see that, with an increase in M/N for a constant P , the PSPR of both the algorithms improves. We can also see similar improvement in PSPR, with an increase in P for a constant M/N . Further, C-BIHT achieves 100% in PSPR with less values of M/N and P than MLA. For example, when the M/N is 0.8, C-BIHT achieves 100% in PSPR with 10 sensors, while MLA requires 15 sensors to achieve the 100% performance for the same M/N . Next, we conduct

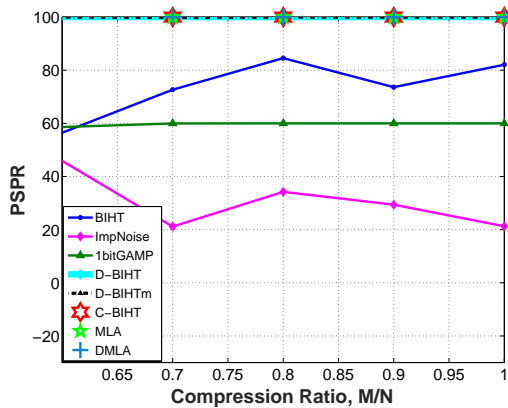


Fig. 2.3: PSPR when $\eta = -3.01$ dB for $N = 100$ and $K = 5$

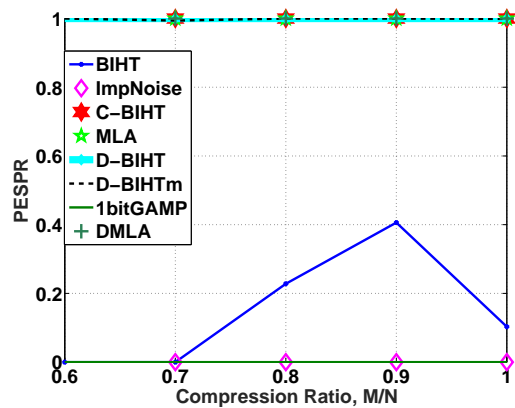


Fig. 2.4: PESPR when $\eta = -3.01$ dB for $N = 100$ and $K = 5$

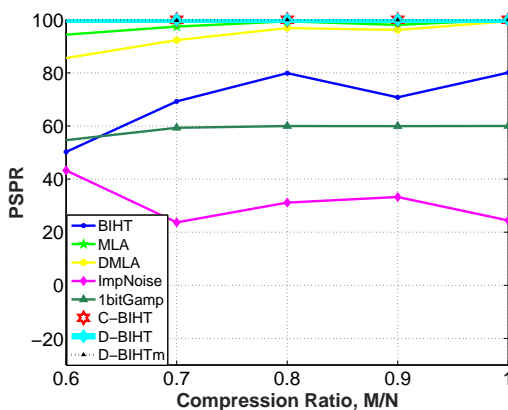


Fig. 2.5: PSPR when $\eta = 6.99$ dB for $N = 100$ and $K = 5$

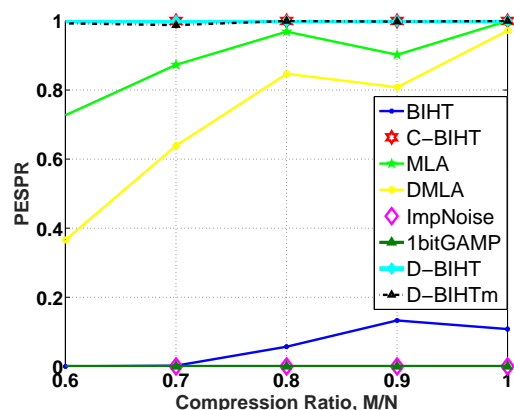


Fig. 2.6: PESPR when $\eta = 6.99$ dB for $N = 100$ and $K = 5$

two experiments where we compare PSPR and PESPR of all the proposed algorithms when η

= -3.01 dB and $\eta = 6.99$ dB respectively. In Figures 2.3 and 2.4, we present our results when $\eta = -3.01$ dB where the values of PSPR and PESPR are shown as a function of M/N . The results show that, when $\eta = -3.01$ dB, all of the proposed centralized and decentralized algorithms perform well with near 100% sparsity pattern recovery. All the curves corresponding to the proposed algorithms, in Figures 2.3 and 2.4 overlap and are not distinguishable. As expected, SMV based algorithms, BIHT and 1bitGAMP, perform quite poorly. It should be noted that ImpNoise, which is a distributed one-bit CS algorithm, also performs quite poorly. For the case when $\eta = 6.99$ dB, we present our results in Figures 2.5 and 2.6. It can be seen that with increased noise power, there is degradation of performance of all of the algorithms. When M/N increases, the performance of all of these algorithms improves. It is seen that C-BIHT and D-BIHT have better performance in sparsity pattern recovery, i.e., almost 100% for all the measurements. MLA and DMLA performance is 90% or above only when M/N approaches 1. The performances of both decentralized algorithms are comparable to that of their corresponding centralized algorithms. However, the performance of D-BIHT is better than DMLA. All of these algorithms have much better performance compared to the BIHT and 1bitGAMP algorithms, i.e., the algorithms using single measurement vectors. The performance of ImpNoise algorithm is not as good as the proposed algorithms. It should be noted that D-BIHTm does not lose much in performance when it is compared with D-BIHT even though D-BIHTm uses less amount of information in estimating the joint sparse support set.

Next, we compare the performances of the proposed one-bit CS algorithms for joint support recovery with that of their real valued CS counterparts. Here we choose to compare C-BIHT, D-BIHT, D-BIHTm with the simultaneous iterative hard thresholding (SIHT) algorithm [8]. Figure 2.9 shows the PSPR of C-BIHT, D-BIHT and SIHT when $\eta = 16.99$ dB as a function of M/N . It is quite interesting to see that, even with an increase in η , the PSPR of C-BIHT and D-BIHT is almost 100% for all the values of M/N . Next, we analyze the sensitivity of all of these algorithms with respect to η . In this experiment, we choose $M/N = 0.6$ and vary η .

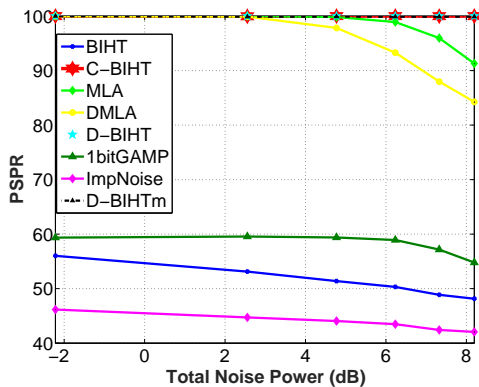


Fig. 2.7: PSPR for D-BIHT, D-BIHTm, DMLA, C-BIHT, MLA, BIHT (SMV) as a function of η when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.

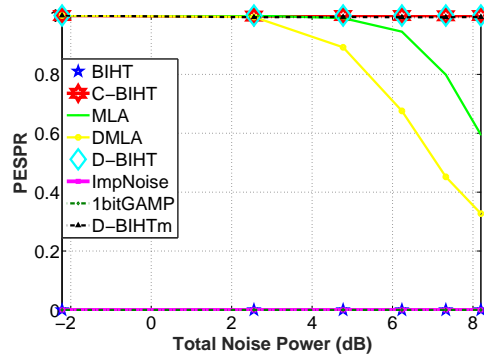


Fig. 2.8: PESPR for D-BIHT, D-BIHTm, DMLA, C-BIHT, MLA, BIHT (SMV) as a function of η when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.

Figures 2.10 shows PSPR values of the proposed algorithms and SIHT as a function of η . It is seen that C-BIHT and D-BIHT perform similar to SIHT when η is 16 dB or less. The result is quite promising even when $M/N = 0.6$. However, when $\eta > 16$ dB, the rate of degradation in the performances of C-BIHT and D-BIHT is higher than SIHT. Degradation in the performance of one-bit CS algorithms with an increase in η is expected. It is noted that one-bit CS yields huge saving in the number of bits required to store and/or transmit compressed measurements. For each signal vector with one-bit CS measurements, \mathbf{z}_p , requires only M bits. However, approximation of a real valued CS with L level of quantization requires $M \log_2(L)$ bits. Thus, one-bit CS saves $M \log_2(L) - M$ bits. The saving increases by a factor of P in a sensor network with P sensors. The performances of these one-bit CS algorithms, except in very low SNR regimes, are comparable to their real valued counterparts. Thus, the proposed algorithms provide a promising alternatives to real valued CS based algorithms, especially in resource constrained networks except when the total noise power is large.

Next, we numerically estimate the total number of one-bit compressed measurements required by the C-BIHT algorithm on an average for joint support recovery with minimum error. Here, for each value of signal SNR, $\gamma = \frac{K\sigma_s^2 + \mu^T \mu}{K\sigma_v^2}$, we run the C-BIHT algorithm for different

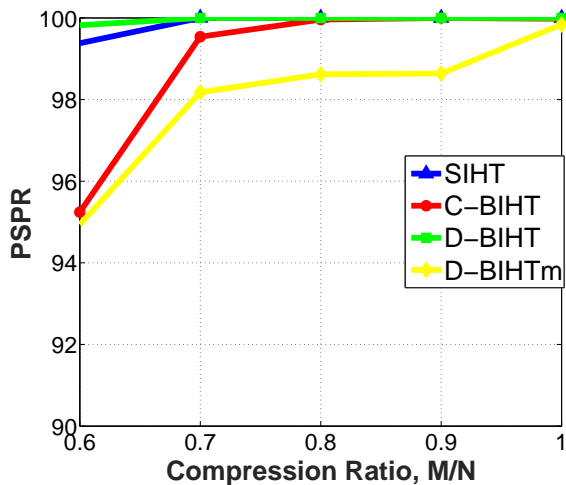


Fig. 2.9: Comparison of SIHT with C-BIHT, D-BIHT, and D-BIHTm when $N = 100$, $P = 10$, $K = 5$, and $\eta = 16.99$ dB

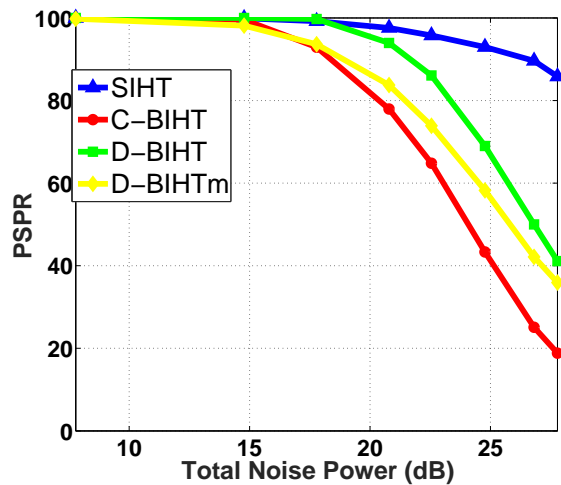


Fig. 2.10: Comparison of SIHT with C-BIHT, D-BIHT, and D-BIHTm when $N = 100$, $P = 10$, $K = 5$, and $M = 60$

number of measurements, say M' , starting from 1. We estimate the probability of joint support recovery from 100 runs of the algorithm for M' . If the probability of joint support recovery is less than 0.95, we increase the value of M' by one and repeat the experiment. The first M' with the probability of joint support recovery of 0.95 or more is considered as the lower bound, M_{min} . For each value of γ , we repeat the experiment for 20 instances. The average of all M_{min} s for a γ is considered the desired lower bound for C-BIHT, M_{CBIHT} , for the γ . We compare this M_{CBIHT} with the bound obtained in (2.15) which we label as M_{ML} . Figure 2.11 shows the comparison of M_{CBIHT} with M_{ML} for $K = 20$ and 40. It can be seen that the M_{CBIHT} is lower-bounded by M_{ML} . When the γ is small, the difference between M_{CBIHT} and M_{ML} is high. When γ increases, the difference decreases. However, any further increase in the value of γ does not change M_{CBIHT} by much and M_{CBIHT} tends to stabilize. Next, we compare the computational complexities of the proposed algorithms in a centralized setting. Addition and multiplication are considered as the basic operations in evaluating complexity. We assume that the number of operations in the evaluation of a gradient and a function is considered to be equal to the dimensions of the gradient and the variable returned by the function,

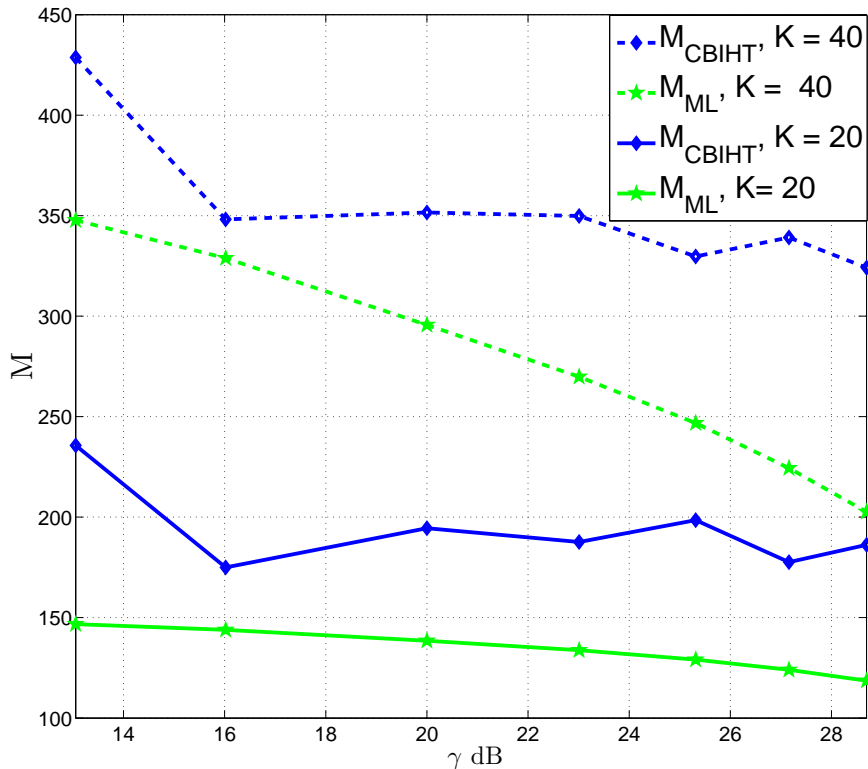


Fig. 2.11: Minimum number of measurements for support recovery vs. Signal SNR, γ when $N = 1000$, $K = 20$, $\sigma_v^2 = 0.01$, $\mu_j = 1$, for $j \in \mathcal{U}$, and $P = 5$

respectively. Based on these assumptions, the total number of operations carried out by the MLA can be shown to be of order $\mathcal{O}(\text{ceil}(\log_\alpha \frac{\tilde{\lambda}}{\lambda})(1 + T'(3NP + NP N_E + 1)))$, where T' is the number of times the inner while loop executes and N_E denotes the total number of operations required to update the elements of \mathbf{S}^k using (25) and (27). Similarly, the total number of operations required by C-BIHT is of order $\mathcal{O}(T(4NP + KP))$. As the exact analysis of computational complexity of MLA depends on the algorithm parameters $\tilde{\lambda}$, λ , and T' , it is difficult to provide a fair comparison with the computational complexity of C-BIHT. Therefore, we employ the run times of centralized algorithms as a measure of computational complexity of these algorithms. The analysis of time complexities of the decentralized algorithms is similar to their respective centralized counterparts. This is because time complexity at each node in the

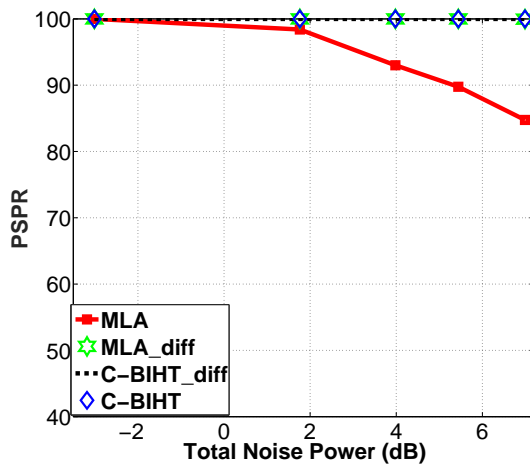


Fig. 2.12: PSNR values of C-BIHT and MLA algorithms as a function of η when different random Gaussian \mathbf{A}_p are used when $N = 100$, $M = 60$, $K = 5$, and $P = 10$.

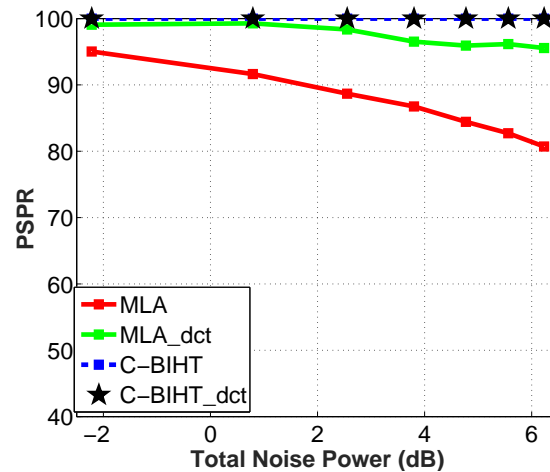


Fig. 2.13: PSNR values of C-BIHT and MLA algorithms as a function of η when random partial DCT matrix is used when $N = 100$, $M = 60$, $K = 5$, and $P = 10$.

decentralized algorithms is dominated by the time complexity of the smaller instance of their corresponding centralized algorithms. Table 2.1 gives a summary of run times required by the C-BIHT and MLA algorithms to estimate the sparsity pattern from given one-bit compressive MMV for different values of N , P and M . The experiment is carried out in Matlab 2015b using processor Intel Xenon(R). The values in the table show the average times required by the centralized algorithms in seconds to obtain the sparsity pattern which is obtained by averaging the total time required for 20 runs. For both the algorithms, the time required increases when one or more of N , P and M increase. The run times of C-BIHT and MLA have a 5-fold increase when the problem size, N , have a 5-fold increase, i.e., from 100 to 500. For the same values of N , P and M , C-BIHT is around 100 times faster than MLA and hence is a clear winner in terms of time complexity.

Finally, we evaluate the performance of the C-BIHT and MLA algorithms when we relax our assumptions on the measurement matrices. In the first case, we consider the case when the measurement matrices \mathbf{A}_p for $p = 1, \dots, P$ are different and compare the performance with the case when $\mathbf{A}_p = \mathbf{A}$ for all p . Figure 2.12 shows the PSNR values of the C-BIHT algo-

Table 2.1: Comparison of run times of C-BIHT and MLA in seconds to obtain the sparsity pattern when $N=100$ and $N=500$

C-BIHT when $N = 100$ and $K = 5$					
M \rightarrow	60	70	80	90	100
P = 3	0.0285	0.030	0.0307	0.034	0.035
P = 5	0.067	0.069	0.069	0.072	0.118
P = 10	0.090	0.092	0.095	0.111	0.160
MLA when $N = 100$ and $K = 5$					
M \rightarrow	60	70	80	90	100
P = 3	2.2676	2.5566	2.9729	3.3396	4.5327
P = 5	4.5370	5.1878	6.0173	6.6772	7.5497
P = 10	9.3963	10.9925	12.6525	14.0072	15.7242
C-BIHT when $N = 500$ and $K = 25$					
M \rightarrow	300	350	400	450	500
P = 3	0.1757	0.1757	0.1695	0.1942	0.1857
P = 5	0.1764	0.1832	0.1890	0.1970	0.2127
P = 10	0.2170	0.2398	0.2594	0.2694	0.3382
MLA when $N = 500$ and $K = 25$					
M \rightarrow	300	350	400	450	500
P = 3	11.8385	13.7213	15.8544	17.6784	19.6593
P = 5	19.7641	23.1694	26.4415	29.6953	32.8875
P = 10	39.7087	46.1982	52.7405	59.1511	65.5580

rithm and MLA algorithm as a function of η . `MLA_diff` and `C-BIHT_diff` refer to results when different measurement matrices are used at different sensors. The performance of the C-BIHT algorithm using different \mathbf{A}_p is comparable to when using \mathbf{A} at all nodes. However, Figure 2.12 shows that the MLA algorithm has improvement in performance when different \mathbf{A}_p are used at different sensors, especially when the total noise power is high. The improvement in the performance is due to the diversity of the measurement matrices. Previous works [43, 64] have also shown improvement in the signal recovery performance theoretically and through numerical experiments when different measurement matrices are used instead of single measurement matrix. These works, however, assumed η compressed measurements to be real-valued. The numerical results in Figure 2.12 show similar results when we have multiple one-bit compressed measurements from different measurement matrices. We can also see that the performance of

the MLA algorithm improves and becomes comparable to that of the C-BIHT algorithm by using different A_p . We should note that the FC is required to know the measurement matrices used by each sensor in the network. Similarly, in a decentralized system, each sensor is required to know the measurement matrices of all of its one-hop neighbors. Hence, in both of these cases, prior communication between sensors and the FC or among sensors are required for sharing measurement matrices. Further, the space complexity of the FC in a centralized setting and each of the nodes in a decentralized setting increases with the increase in the size of the network and increase in connectivity among nodes respectively.

In the second case, PSPR values of C-BIHT and MLA algorithms are evaluated when a random partial DCT matrix, which is obtained by picking M rows uniformly at random from N rows of an $N \times N$ DCT matrix, is used as the measurement matrix and is compared to the case when random Gaussian measurement matrix is used. The numerical results of the experiment are shown in Figure 2.13. Curves represented by legends C-MLA_dct and C-BIHT_dct refer to the results when partial DCT matrices are used instead of a random Gaussian matrices. We can see a similar performance of the C-BIHT algorithm and improvement of the MLA algorithm with random partial DCT matrix when compared to a random Gaussian matrix.

We have studied the advantages of using MMVs in centralized and decentralized settings over SMV. Next, we study the sparsity pattern recovery performance of the SMV-based algorithm with MMV-based algorithms when we put a constraint on the total number of bits that can be used by algorithms to estimate support of sparse signal(s). Let N_B be the total number of bits that can be used in total to estimate the support of the sparse signals. one-bit SMV-based CS algorithm makes all N_B measurements to estimate the support of the sparse signal, whereas each sensor in MMV based CS algorithms makes N_B/P one-bit CS measurements. In this setting, i.e., when the total number of bits is constant, we compare the performance of MMV based algorithms (both centralized and decentralized) with SMV-based algorithms. Here, we chose to compare the performance of C-BIHT, D-BIHT, D-BIHTm with BIHT. We should note

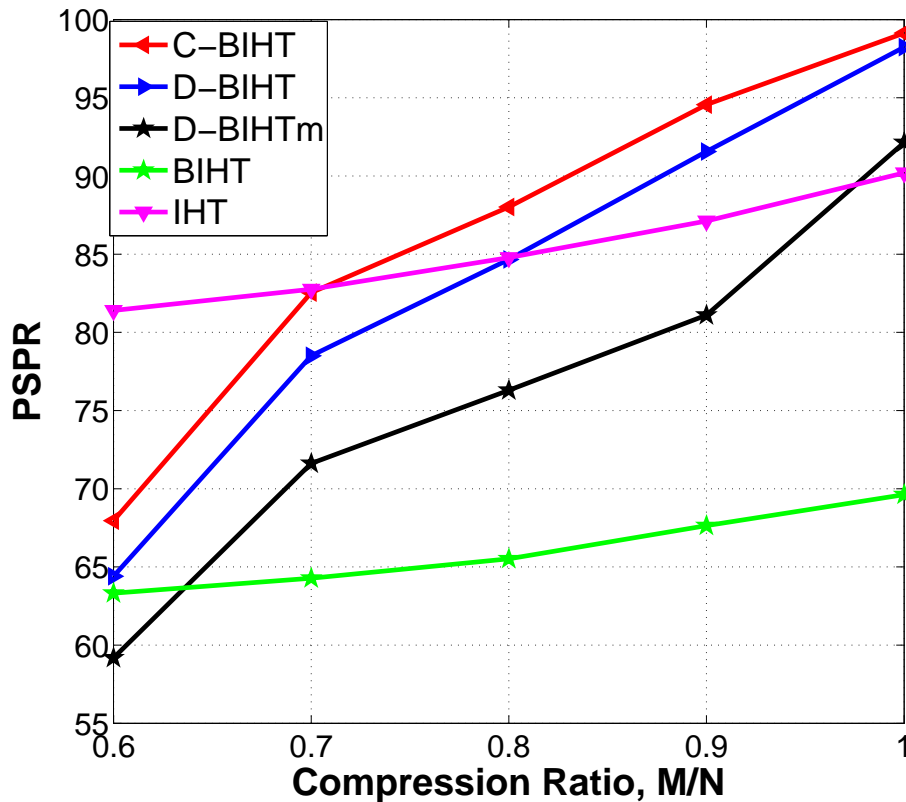


Fig. 2.14: Percentage of Sparsity Pattern Recovery for D-BIHT, D-BIHTm, C-BIHT, BIHT, and IHT as a function of M/N when $N = 100$ and $K = 5$ in a network of 10 nodes each of which has degree 3.

that in this setting BIHT uses an overcomplete dictionary whereas C-BIHT, D-BIHT, D-BIHTm uses an under-complete dictionary to obtain measurements. Hence, we compare the support recovery performance when SMV is available from the over-determined dictionary with the joint support recovery with MMV from the under-determined dictionary. Figure 2.14 shows the PSPR values of these algorithms for different values of M/N . It should be noted that M/N refers to the compression ratio of the measurement vectors of each sensor in centralized and decentralized settings. BIHT is performed when the compression ratio is equal to PM/N .

2.7 Summary

In this chapter, we considered the problem of joint sparsity pattern recovery with one-bit quantized compressed measurements. We determined the performance bounds for joint support recovery of sparse signals when one-bit quantized measurements from the distributed sensors are available at the FC. We showed that the number of compressive measurements required to recover the joint sparsity pattern with vanishing probability of error has an inverse relation with the number of sensors in the sensor network. We also developed two computationally tractable centralized algorithms, namely MLA and C-BIHT for sparsity pattern recovery with one-bit quantized measurements. Further, we extended the proposed centralized algorithms to decentralized settings. We showed that the performance of these decentralized algorithms is comparable to the centralized algorithms. Through numerical simulations, we showed that one-bit CS algorithms have comparable performance to real-valued CS algorithms except in cases when the total noise power is large. The proposed one-bit CS algorithms are promising for resource-constrained networks as they provide a significant saving in the number of bits required to store and/or transmit with performance comparable to their real-valued CS counterparts.

CHAPTER 3

ONE-BIT COMPRESSED SENSING WITH HOMOGENEOUS SIDE-INFORMATION

3.1 Introduction

In general, the performance of sparse signal reconstruction or parameter estimation from one-bit measurements is susceptible to noise. In the previous chapter, we looked into the problem of sparse support estimation in a centralized and decentralized setup. We provided empirical results that demonstrated that the proposed algorithms improved the sparse support estimation performance. In this chapter, we consider the task of sparse signal reconstruction from noisy one-bit compressed measurements. We aim to improve the reconstruction performance by using a statistically dependent signal that the receiver has as an aid. We refer to this signal as side-information (SI).

The concept of using SI is not new. For example, SI is available in certain applications, including in the reconstruction of sequences of signals such as in dynamic MRI reconstruction [68], video signal reconstruction [53], and sequential estimation [17]. Several authors [19, 69, 72, 89, 91, 107] have shown improved reconstruction performance from com-

pressed measurements by incorporating SI. In [89], the authors assume that the receiver uses the partial support set of the sparse signal as SI. In [72, 91], the authors assume that SI is a noisy version of the actual compressed signal. All of these works have exploited SI with real-valued measurements. Further, in the channel estimation problem in wireless communication, the channel estimate at the previous time instant can be used as SI to estimate channel at the current time instant [69]. In the case of one-bit CS, a few works [76, 79] have exploited SI in the signal recovery task. In [76], the authors assume that the receiver has access to partial support information as SI, and in [79], the authors assume the complete knowledge of the support set as SI. In [76] and [79], however, the authors do not account for noise in one-bit measurements and SI.

In this work, we assume that the SI is noisy. We consider two different scenarios of SI -a) SI consists of support information only, and b) SI consists of both support and amplitude information. We approach the problem from the Bayesian perspective. This allows us to incorporate the sparse structure on signals, noise in the SI and one-bit measurements and model dependence between the signal and the SI with ease. In this problem, we use Bernoulli-Gaussian distribution as a prior on the signal to impose sparse structure. Note that the SI is usually the signal reconstructed at the previous time instant. Based on the temporal dynamics of the observed phenomenon, the support and the amplitude of the sparse signal might change over time. Further, due to noise in the compressed measurement process, the reconstructed signals might have some incorrect support and amplitude information. Hence, we model the SI as the signal corrupted with additive noise to account for the discrepancies between the SI and the signal. For this setup, we develop an algorithm when we model the additive noise in SI using Laplacian distribution. Second, we assume that the receiver has access to the support information as SI. When SI consists of support information only, we use Bernoulli distribution to model the noise of the support. We recover the signal as a minimum mean square error (MMSE) estimate of the posterior pdf of the signal. As the computation of the MMSE estimator is in-

tractable in a higher dimension, we propose a generalized approximate message passing-based algorithm to approximate the MMSE estimator. The Expectation-Maximization algorithm is used to estimate the noise parameter of SI. We show that the signal recovery performance can be significantly improved by exploiting available side information at the receiver.

3.2 Signal and Measurement Models

In the following, we introduce our signal and measurement models.

3.2.1 Signal Model

We consider the input signal $\mathbf{x} \in \mathbb{R}^N$ to be random with elements having identical and independent (i.i.d.) distribution

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{n=1}^N p_{x_n}(x_n), \quad (3.1)$$

where each component x_i is a Bernoulli-Gaussian distributed random variable with pdf

$$p_{x_n}(x_n) = (1 - \lambda)\delta(x_n) + \lambda\mathcal{N}(x_n; 0, v_x), \quad (3.2)$$

where $\delta(x)$ is the Dirac-delta function, and λ is the probability of having non-zero values. \mathbf{x} is a sparse signal. λ controls the sparsity of the signal. Smaller the value of λ , sparser the signal.

3.2.2 Measurement Model

Figure 3.1 shows the transmission chain of the measurement model of the problem considered in this work. The sparse signal $\mathbf{x} \in \mathbb{R}^N$ is linearly transformed to a vector $\mathbf{z} \in \mathbb{R}^M$ using the random measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. The transformed vector, \mathbf{z} , is assumed to be corrupted by additive i.i.d Gaussian noise vector with mean zero and variance v , i.e., $n_m \sim \mathcal{N}(0, v)$. This corrupted compressed vector is quantized element-wise to $+1$ or -1 based on

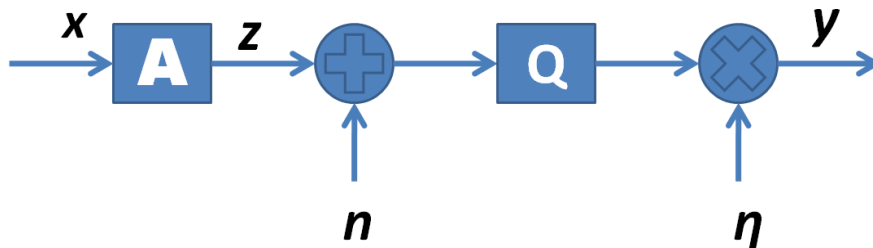


Fig. 3.1: One-bit CS with pre-quantization and post-quantization noise.

the sign of the signal. We assume a noisy channel between the quantizer and the receiver where the quantized measurements are corrupted by multiplicative noise which takes values either $+1$ or -1 . Formally, the quantized measurement model can be written as

$$\mathbf{y} = \boldsymbol{\eta} \odot Q(\mathbf{A}\mathbf{x} + \mathbf{n}), \quad (3.3)$$

where the quantizer $Q : \mathbb{R}^M \rightarrow \{-1, +1\}^M$ is the element-wise sign quantizer. The m -th element at the output of the quantizer is

$$[Q(\zeta)]_m = \begin{cases} +1, & \text{if } \zeta_m > 0, \\ -1, & \text{if } \zeta_m \leq 0, \end{cases} \quad (3.4)$$

and $\boldsymbol{\eta} \in \{-1, +1\}^M$ is the i.i.d. post-quantization noise. η_m is assumed to follow Bernoulli distribution with $Pr(\eta_m = 1) = \gamma$. We define the inverse of quantization function, $Q^{-1}(\cdot)$, as

$$[Q^{-1}(\mathbf{y})]_m = \begin{cases} (-\infty, 0], & \text{if } y_m \leq 0, \\ (0, \infty), & \text{if } y_m > 0, \end{cases} \quad (3.5)$$

where y_m is the m -th element of \mathbf{y} .

3.2.3 Noisy One-Bit Compressed Sensing

As in [52], the posterior distribution of the signal, \mathbf{x} , given the quantized and noisy measurements, \mathbf{y} , at the receiver is

$$p_{\mathcal{X}|\mathcal{Y}}(\mathbf{x}|\mathbf{y}) \propto p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x})p_{\mathcal{X}}(\mathbf{x}) \propto \prod_{m=1}^M \mathbb{I}_{z_m \in \{Q^{-1}(\eta_m y_m)\}} \prod_{n=1}^N p_{\mathcal{X}_n}(x_n), \quad (3.6)$$

where $\mathbb{I}_{(\cdot)}$ represents the indicator function, and \propto represents equality upto a proportional constant. The minimum mean square error (MMSE) estimator of \mathbf{x} is the mean of the posterior distribution, i.e., $\mathbb{E}_{\mathcal{X}|\mathcal{Y}}[\mathbf{x}|\mathbf{y}]$. Next, we assume that the receiver has access to side-information which is related to the signal of interest. The side-information is imposed as probability distribution. Let $\tilde{\mathbf{x}}$ represent the side-information of signal \mathbf{x} . Here, we construct the posterior distribution of signal, \mathbf{x} , given the noisy one-bit compressed measurements \mathbf{y} and side-information $\tilde{\mathbf{x}}$ as

$$p_{\mathcal{X}|\mathcal{Y},\tilde{\mathcal{X}}}(\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}}) \propto p_{\mathcal{Y}|\mathcal{X}}(\mathbf{y}|\mathbf{x})p_{\tilde{\mathcal{X}}|\mathcal{X}}(\tilde{\mathbf{x}}|\mathbf{x}) \propto \prod_{m=1}^M \mathbb{I}_{z_m \in \{Q^{-1}(\eta_m y_m)\}} \prod_{n=1}^N p_{\mathcal{X}_n|\tilde{\mathcal{X}}_n}(x_n|\tilde{x}_n), \quad (3.7)$$

where $p_{\mathcal{X}_n|\tilde{\mathcal{X}}_n}(x_n|\tilde{x}_n)$ is the conditional density function that gives the statistical characterization of the sparse signal when the side-information is given. The MMSE estimator of \mathbf{x} with SI at the receiver is $\mathbb{E}_{\mathcal{X}|\mathcal{Y},\tilde{\mathcal{X}}}[\mathbf{x}|\mathbf{y},\tilde{\mathbf{x}}]$. We note that the derivation of the MMSE estimators (3.6) and (3.7) is intractable in direct form. Therefore, we develop GAMP-based algorithms to approximate the MMSE estimator.

3.3 Noisy One-Bit Compressed Sensing Algorithm

In this section, we begin with an introduction to the GAMP algorithm. GAMP algorithm [81] is a generalization of the AMP algorithm [27]. Both AMP and GAMP algorithms apply loopy belief propagation in the bipartite graph under the Gaussian approximation for the involved messages.

These methods fall under the Bayesian framework which assume a prior distribution, $p_{\mathcal{X}}(\mathbf{x})$, on \mathbf{x} . The key idea in the Bayesian approach is to find the marginal posterior distributions $p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y})$ which could be used in minimum mean square error (MMSE) or maximum a posteriori (MAP) estimation of each x_n as:

$$\begin{aligned}\hat{x}_n^{\text{MAP}} &= \arg \max_{x_n} p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y}), \\ \hat{x}_n^{\text{MMSE}} &= \arg \min_{\hat{x}_n} \mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}\{(x_n - \hat{x}_n)^2\} \\ &= \mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}\{x_n|\mathbf{y}\}.\end{aligned}$$

AMP inherently assumes the prior of a signal to be Gaussian whereas GAMP offers the systematic approach of taking any prior of the signal into account during the denoising step. However, the evaluation of the true marginal distributions, $p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y})$, of a high-dimensional vector, \mathbf{x} , is analytically intractable and computationally prohibitive. The GAMP algorithm implements loopy belief propagation and uses the central limit theorem with quadratic approximations to approximate $p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y})$ to improve computational performance. The GAMP algorithm uses the sum-product and max-sum belief propagation algorithms to compute MMSE and MAP estimators respectively. In the next section, we focus on the MMSE estimation problem corresponding to the posterior densities (3.6) and (3.7). For detailed expositions on AMP, and GAMP, we refer the readers to [27] and [81]. In this work, we consider the sum-product version of the GAMP algorithm where we find the MMSE estimator of \mathbf{x} corresponding to the posterior densities (3.6) and (3.7).

3.3.1 Noisy One-Bit Compressed Sensing (Noisy1bG)

In this subsection, we develop a GAMP based algorithm that reconstructs a sparse signal from its noisy one-bit compressed measurements. Define $\mathbf{z} \triangleq \mathbf{A}\mathbf{x}$ as the linear transformation of \mathbf{x} . The transformed signal, \mathbf{z} , is corrupted by i.i.d. Gaussian noise which is quantized to

one-bit as defined in (3.4). The one-bit quantized signal is transmitted over a channel with probability of sign-flip $1 - \gamma$. We represent the entire effect of additive white Gaussian noise (measurement noise), one-bit quantization and sign-flip error (channel noise) by a probabilistic channel, $p_{\mathcal{Y}|\mathcal{Z}}(\mathbf{y}|\mathbf{z}; \sigma_w^2)$. Since we assume that the measurement noise and the channel noise are i.i.d., the channel is represented as

$$p_{\mathcal{Y}|\mathcal{Z}}(\mathbf{y}|\mathbf{z}; \sigma_w^2, \gamma) = \prod_{m=1}^M p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m; \sigma_w^2, \gamma). \quad (3.8)$$

In Algorithm 3.4.1, we summarize the steps of the GAMP algorithm for sparse signal reconstruction from one-bit noisy compressed measurements. We refer to this algorithm as `Noisy1bG`. This Algorithm requires the computations of non-linear functions $F_1(\cdot)$, $F_2(\cdot)$, $G_1(\cdot)$, and $G_2(\cdot)$ as defined in (3.10) and (3.12).

Evaluation of $F_1(\cdot)$ and $F_2(\cdot)$

First, we evaluate the channel, $p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m)$, based on our system model as

$$\begin{aligned} p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m) &= \sum_{y_{qm}} p(y_{qm}|z_m) p(y_m|y_{qm}, z_m) \\ &= \gamma p(y_{qm} : y_{mq} = y_m|z_m) + (1 - \gamma) p(y_{qm} : y_{qm} \neq y_m|z_m), \end{aligned} \quad (3.13)$$

where y_{qm} is the m -th element of the output of the quantizer \mathbf{y}_q . Let $\Phi(p) = \int_{-\infty}^p \mathcal{N}(x; 0, 1) dx$, $\delta_m^+ = \delta(y_m + 1)$, and $\delta_m^- = \delta(y_m - 1)$. It is noted that $p(y_{qm} : y_{qm} = y_m|z_m)$ is given by

$$\begin{aligned} p(y_{qm} : y_{qm} = y_m|z_m) &= p(z_m + n_m \geq 0|z_m) \delta_m^- + p(z_m + n_m \leq 0|z_m) \delta_m^+ \\ &= \Phi\left(\frac{z_m}{\sqrt{v}}\right) \delta_m^- + (1 - \Phi\left(\frac{z_m}{\sqrt{v}}\right)) \delta_m^+. \end{aligned} \quad (3.14)$$

Algorithm 3.1 Noisy1bG Algorithm

1. Initialization: Set $t = 0$ and initialize $\hat{\mathbf{x}}^t$, τ_x^t , and $\hat{\mathbf{s}}^t$ as $\hat{\mathbf{x}}^t = \mathbb{E}[\mathbf{x}]$, $\tau_x^t = \text{var}[\mathbf{x}]$, $\hat{\mathbf{s}}^t = \mathbf{0}$, where the expectation and variance of \mathbf{x} are with respect to p_x

2. **Measurement Update**

- Linear Step

$$\tau_p^{t+1} = (\mathbf{A} \odot \mathbf{A}) \tau_x^t, \quad \hat{\mathbf{p}}^{t+1} = \mathbf{A} \hat{\mathbf{x}}^t - \tau^{p,t+1} \odot \hat{\mathbf{s}}^t,$$

- Non-Linear Step

$$\begin{aligned} \hat{\mathbf{s}}^{t+1} &= F_1(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau^{p,t+1}), \\ \tau_s^{t+1} &= F_2(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau^{p,t+1}), \end{aligned} \quad (3.9)$$

where F_1 and F_2 are applied element-wise and are defined as

$$\begin{aligned} F_1(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau^{p,t+1}) &= \frac{1}{\tau^{p,t+1}} \left(\mathbb{E}[z|\mathbf{y}] - \hat{\mathbf{p}}^{t+1} \right), \\ F_2(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau^{p,t+1}) &= \frac{1}{\tau^{p,t+1}} \left(1 - \frac{\text{var}[z|\mathbf{y}]}{\tau^p} \right). \end{aligned} \quad (3.10)$$

The expectation and variance are evaluated with respect to $z \sim \mathcal{N}(\hat{p}, \tau^p)$.

3. **Estimation Update**

Linear Step

$$\tau^{r,t+1} = ((\mathbf{A} \odot \mathbf{A})^T \tau_s^{t+1})^{-1}, \quad \hat{\mathbf{r}}^{t+1} = \hat{\mathbf{x}}^t + \tau^{r,t+1} \odot (\mathbf{A}^T \hat{\mathbf{s}}^{t+1}),$$

where the inversion is performed element-wise

Non-linear Step

$$\begin{aligned} \hat{\mathbf{x}}^{t+1} &= G_1(\hat{\mathbf{r}}^{t+1}, \tau^{r,t+1}; p_{\mathcal{X}}), \\ \tau_x^{t+1} &= G_2(\hat{\mathbf{r}}^{t+1}, \tau^{r,t+1}; p_{\mathcal{X}}), \end{aligned} \quad (3.11)$$

where G_1 and G_2 are applied element-wise and are defined as

$$\begin{aligned} G_1(\hat{r}_n, \tau_n^r; p_{x_n}) &= \mathbb{E}_{\mathcal{X}_n|\mathbf{y}}[x_n | \mathbf{y}; \hat{r}_n, \tau_n^r], \\ G_2(\hat{r}_n, \tau_n^r; p_{x_n}) &= \text{var}_{\mathcal{X}_n|\mathbf{y}}[x_n | \mathbf{y}; \hat{r}_n, \tau_n^r]. \end{aligned} \quad (3.12)$$

The expectation and variance are evaluated with respect to $p_{\mathcal{X}_n|\mathbf{y}} \propto \mathcal{N}(\cdot; \hat{r}_n, \tau_n^r) p_{\mathcal{X}_n}(\cdot)$.

Set $t = t+1$ and return to step 2 until $t < T$.

Similarly, we evaluate $p(y_{qm} : y_{qm} \neq y_m | z_m)$ as

$$p(y_{qm} : y_{qm} \neq y_m | z_m) = (1 - \Phi(\frac{z_m}{\sqrt{v}}))\delta_m^- + \Phi(\frac{z_m}{\sqrt{v}})\delta_m^+. \quad (3.15)$$

Using central limit theorem arguments, GAMP approximates the distribution of random variable \mathcal{Z} as Gaussian with mean \hat{p} and variance τ^p , i.e., $\mathcal{Z} \sim \mathcal{N}(\hat{p}, \tau^p)$. The posterior marginal distribution, $p_{\mathcal{Y}_m | \mathcal{Z}_m}(y_m | z_m)$, can be evaluated as

$$p_{\mathcal{Z}_m | \mathcal{Y}}(z_m | \mathbf{y}; \hat{p}_m, \tau_m^p) = \frac{p_{\mathcal{Y}_m | \mathcal{Z}_m}(y_m | z_m) \mathcal{N}(z_m; \hat{p}_m, \tau_m^p)}{\int_{z_m} p_{\mathcal{Y} | \mathcal{Z}_m}(y_m | z_m) \mathcal{N}(z_m; \hat{p}_m, \tau_m^p)}.$$

The term in the denominator is the normalization constant. In the following, we evaluate the normalization constant, Z_m^p , the posterior mean $\mathbb{E}_{\mathcal{Z}_m | \mathcal{Y}}[z_m | \mathbf{y}]$ and the posterior variance $\text{var}_{\mathcal{Z}_m | \mathcal{Y}}[z_m | \mathbf{y}]$. Define $PI_m^q = \int z_m^q \Phi(z_m / \sqrt{v}) \mathcal{N}(z; \hat{p}_m, \tau_m^p) dz$ for $q = 0, 1$, and 2. Using the definition of $p_{\mathcal{Y}_m | \mathcal{Z}_m}(y_m | z_m)$ from (3.14), and (3.15), the normalization constant can be derived as

$$\begin{aligned} Z_m^p &= \int p_{\mathcal{Y}_m | \mathcal{Z}_m}(y_m | z_m) \mathcal{N}_{\mathcal{Z}_m}(z_m; \hat{p}_m, \tau_m^p) dz \\ &= \gamma \left(PI_m^0 \delta_m^- + (1 - PI_m^0) \delta_m^+ \right) + (1 - \gamma) \left((1 - PI_m^0) \delta_m^- + PI_m^0 \delta_m^+ \right). \end{aligned}$$

Next, we evaluate the posterior mean of z_m as

$$\mathbb{E}_{\mathcal{Z}_m | \mathcal{Y}}[z_m | \mathbf{y}; \hat{p}_m, \tau_m^p] = \left[\gamma \left(PI_m^1 \delta_m^- + (\hat{p}_m - PI_m^1) \delta_m^+ \right) + (1 - \gamma) \left((\hat{p}_m - PI_m^1) \delta_m^- + PI_m^1 \delta_m^+ \right) \right] \frac{1}{Z_m^p}.$$

Similarly, we can evaluate $\mathbb{E}_{\mathcal{Z}_m | \mathcal{Y}_m}[z_m^2 | y_m]$ as,

$$\begin{aligned} \mathbb{E}_{\mathcal{Z}_m | \mathcal{Y}}[z_m^2 | \mathbf{y}; \hat{p}_m, \tau_m^p] &= \left[\gamma \left(PI_m^2 \delta_m^- + (\hat{p}_m^2 + \tau_m^p - PI_m^2) \delta_m^+ \right) \right. \\ &\quad \left. + (1 - \gamma) \left((\hat{p}_m^2 + \tau_m^p - PI_m^2) \delta_m^- + PI_m^2 \delta_m^+ \right) \right] \frac{1}{Z_m^p}. \end{aligned}$$

For the evaluation of Z_m^p , $\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|\mathbf{y}; \hat{p}_m, \tau_m^p]$, and $\mathbb{E}_{\mathcal{Z}|\mathcal{Y}}[z_m^2|\mathbf{y}; \hat{p}_m, \tau_m^p]$, we need to evaluate integrals PI_m^0 , PI_m^1 , and PI_m^2 . Integrals PI_m^q for $q = 0, 1$, and 2 can be evaluated in closed-form as

$$\begin{aligned} PI_m^0 &= \Phi\left(\frac{\hat{p}_m}{\sqrt{v + \tau_m^p}}\right), \\ PI_m^1 &= \hat{p}_m PI_m^0 + \frac{\tau_m^p \mathcal{N}\left(\frac{\hat{p}_m}{\sqrt{v + \tau_m^p}}\right)}{\sqrt{v + \tau_m^p}}, \\ PI_m^2 &= \tau_m^p PI_m^0 + \hat{p}_m PI_m^1 + \frac{\tau_m^p \hat{p}_m v \mathcal{N}\left(\frac{\hat{p}_m}{\sqrt{v + \tau_m^p}}\right)}{(v + \tau_m^p)^{1.5}}. \end{aligned} \quad (3.16)$$

The derivations of the closed-form expressions of the integrals are provided in Appendix B. The posterior variance can be computed as $\text{var}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|\mathbf{y}] = \mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m^2|\mathbf{y}] - (\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|\mathbf{y}])^2$. With $\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|\mathbf{y}]$ and $\text{var}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|\mathbf{y}]$, non-linear functions $F_1(\cdot)$, and $F_2(\cdot)$ can be computed as defined in (3.10).

Next, we derive the analytical expressions for $G_1(\cdot)$ and $G_2(\cdot)$, i.e., expressions for $\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}[x_n|\mathbf{y}; \hat{r}_n, \tau_n^r]$ and $\text{var}_{\mathcal{X}_n|\mathcal{Y}}[x_n|\mathbf{y}; \hat{r}_n, \tau_n^r]$. The expectation is carried out with respect to the random variable \mathcal{X}_n given $\hat{\mathcal{R}}_n = \hat{r}_n$ for random variables

$$\hat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n,$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent. Therefore, the marginal posterior distribution can be approximated as

$$p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y}; \hat{r}_n, \tau_n^r) = \frac{p_{\mathcal{X}}(x_n)\mathcal{N}(x_n; \hat{r}_n, \tau_n^r)}{\int_{x_n} p_{\mathcal{X}_n}(x_n)\mathcal{N}(x_n; \hat{r}_n, \tau_n^r)}. \quad (3.17)$$

For Bernoulli-Gaussian distribution, the first-order moment can be computed as

$$\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}[x_n|\mathbf{y}; \hat{r}_n, \tau_n^r] = \frac{1}{Z_n^r} \int x_n p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y}; \hat{r}_n, \tau_n^r) dx_n.$$

Using (3.2) and (3.17), and some algebra, we can show that the approximate posterior mean can be expressed as

$$\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}[x_n|\mathbf{y}; \hat{r}_n, \tau_n^r] = Z'_n \exp\left(-\frac{\hat{r}_n^2}{2(v_x + \tau_n^r)}\right) \hat{r}_n, \quad (3.18)$$

where $Z'_n = \frac{1}{Z_n^r} \frac{\lambda}{\sqrt{2\pi}} \frac{v_x}{(v_x + \tau_n^r)^{1.5}}$, and Z_n^r is the normalization constant which is evaluated as

$$\begin{aligned} Z_n^r &= \int x_n p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y}; \hat{r}_n, \tau_n^r) dx_n \\ &= \frac{1 - \lambda}{\sqrt{2\pi\tau_n^r}} \exp\left(\frac{-\hat{r}_n^2}{2\tau_n^r}\right) + \frac{\lambda}{\sqrt{2\pi(v_x + \tau_n^r)}} \exp\left(\frac{-\hat{r}_n^2}{2(\tau_n^r + v_x)}\right). \end{aligned}$$

Similarly, we can evaluate the second-order moment as

$$\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}}[x_n^2|\mathbf{y}; \hat{r}_n, \tau_n^r] = Z'_n \exp\left(\frac{-\hat{r}_n^2}{2(v_x + \tau_n^r)}\right) \left(\frac{\hat{r}_n^2 v_x}{v_x + \tau_n^r} + \tau_n^r\right). \quad (3.19)$$

Using (3.18) and (3.19), the non-linear functions $G_1(\cdot)$ and $G_2(\cdot)$ in (3.12) can be evaluated and hence we can carry out the update in (3.11) of Noisy1bG. Thus, we have derived all the statistical quantities required to implement one-bit CS with pre- and post-quantization noise. Accounting for the noise leads to an improved signal reconstruction performance. However, we emphasize that there are applications where the receiver has access to SI which can be used to further improve signal reconstruction performance. In the next section, we look into how we can model SI in the sparse signal reconstruction problem and exploit it for better reconstruction performance.

3.4 Noisy One-bit Compressed Sensing with Side-Information

In this section, we study the problem of signal reconstruction from noisy one-bit compressed measurements when the receiver has access to SI, $\tilde{\mathbf{x}}$, which has both support and amplitude

information. We design a GAMP based sparse signal reconstruction algorithm taking SI into account. We assume that the SI is erroneous. The error in SI can either be in the amplitude or in the support set of the signal. We assume that the signal has a small fraction of support that is not in the support set of the SI. These errors are random, and hence, we model side information as a noisy version of the signal, i.e.,

$$\tilde{\mathcal{X}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad n = 1, 2, \dots, N \quad (3.20)$$

where \mathcal{V}_n is an additive noise. Note that the magnitude of noise v_n for $n \in \{n' : x_{n'} \neq 0 \text{ and } \tilde{x}_{n'} \neq 0\} \cup \{n' : x_{n'} = 0 \text{ and } \tilde{x}_{n'} = 0\}$ is relatively small and close to zero. But for the indices $n \in \{n' : x_{n'} \neq 0 \text{ and } \tilde{x}_{n'} = 0\} \cup \{n' : x_{n'} = 0 \text{ and } \tilde{x}_{n'} \neq 0\}$, the magnitude of v_n is quite large. This nature of the error vector suggests that only a small fraction of the error vector has significant values, while most of them are close to zero. Since the noise vector is sparse, we model the noise distribution in (3.20) by a Laplace distribution as it forces most of its coefficients to be very small, allowing some occasional large values, i.e., it promotes sparsity on the noise vector [3]. We then use Gaussian distribution to model the noise distribution and develop algorithms for both of these two cases. Through numerical simulations, we will study the gain in reconstruction performances by the algorithms when the noise, \mathcal{V}_n , is modeled by the sparsity promoting distribution, i.e., Laplace distribution.

3.4.1 Noisy one-bit Compressed Sensing with Laplacian Noise (laplacianSI)

In this subsection, we model the noise in SI as a Laplacian distributed random variable. Thus, we choose $p_{\tilde{\mathbf{x}}|\mathbf{x}}(\tilde{\mathbf{x}}|\mathbf{x})$ as

$$p_{\tilde{\mathbf{x}}|\mathbf{x}}(\tilde{\mathbf{x}}|\mathbf{x}) = \left(\frac{1}{4v_s}\right)^N \exp\left(-\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|_1}{2v_s}\right), \quad (3.24)$$

Algorithm 3.2 GAMP Algorithm for noisy one-bit CS with SI

1. **Initialization:** Set $t=0$ and initialize $\hat{\mathbf{x}}^t, \tau_x^t, \hat{\mathbf{s}}^t$ and \mathbf{v}_s as $\hat{\mathbf{x}}^t = \mathbb{E}[\mathbf{x}]$, $\tau_x^t = \text{var}[\mathbf{x}]$, $\hat{\mathbf{s}}^t = 0$, and $\mathbf{v}_s = 0$ where the expectation and variance of \mathbf{x} are with respect to p_x

2. **While** loop $l < L$

3. **While** loop $t < T$

4. **Measurement Update**

Same as in Algorithm

5. **Estimation Update**

Linear Step

$$\begin{aligned}\boldsymbol{\tau}^{r,t+1} &= ((\mathbf{A} \odot \mathbf{A})^T \boldsymbol{\tau}_s^t)^{-1}, \\ \hat{\mathbf{r}}^{t+1} &= \hat{\mathbf{x}}^t + \boldsymbol{\tau}^{r,t} \odot (\mathbf{A}^T \hat{\mathbf{s}}^{t+1}),\end{aligned}$$

where the inversion is performed element-wise

Non-linear Step

$$\hat{\mathbf{x}}^{t+1} = G_1(\hat{\mathbf{r}}^{t+1}, \boldsymbol{\tau}^{r,t+1}; p_{\mathcal{X}|\mathbf{Y}, \tilde{\mathcal{X}}}) \quad (3.21)$$

$$\boldsymbol{\tau}_x^{t+1} = G_2(\hat{\mathbf{r}}^{t+1}, \boldsymbol{\tau}^{r,t+1}; p_{\mathcal{X}|\mathbf{Y}, \tilde{\mathcal{X}}}), \quad (3.22)$$

where G_1 and G_2 are applied element-wise and are defined as

$$\begin{aligned}G_1(\hat{r}_n, \tau_n^r; p_{\mathcal{X}|\mathbf{Y}, \tilde{\mathcal{X}}}) &= \mathbb{E}_{\mathcal{X}_n|\mathbf{Y}, \tilde{\mathcal{X}}_n} [x_n | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r], \\ G_2(\hat{r}_n, \tau_n^r; p_{\mathcal{X}|\mathbf{Y}, \tilde{\mathcal{X}}}) &= \text{var}_{\mathcal{X}_n|\mathbf{Y}, \tilde{\mathcal{X}}_n} [x_n | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r].\end{aligned} \quad (3.23)$$

The expectation and variance are evaluated with respect to $p_{\mathcal{X}_n|\tilde{\mathcal{X}}_n, \mathbf{Y}} \propto \mathcal{N}(\cdot; \hat{r}_n, \tau_n^r) p_{\mathcal{X}}(\cdot) p_{\tilde{\mathcal{X}}_n|\mathcal{X}_n}(\cdot)$, and can be computed by using (3.25)

Set $t = t+1$

6. **End While**

7. Update v_s using (8.16)

8. $l = l + 1$

9. **End While**

Table 3.1: GAMP Equations for Side-Information

$$\begin{aligned}
Z_n^l &= \frac{1 - \lambda}{4v_s \sqrt{2\pi\tau_n^r}} \exp\left(-\frac{\hat{r}_n^2}{2\tau_n^r} - \frac{|\tilde{x}_n|}{2v_s}\right) + \lambda \phi\left(\frac{\hat{r}_n}{\sqrt{\tau_n^r + v_x}}\right) \left(C_{1,n} \Phi\left(\frac{m_n^l - (m_n^g + \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right) + \right. \\
&\quad \left. C_{2,n} \left(1 - \Phi\left(\frac{m_n^l - (m_n^g - \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right)\right)\right) \\
\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{\mathcal{X}}_n}[x_n|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] &= \frac{\lambda \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(C_{1,n} I_1\left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g\right) \right. \\
&\quad \left. + C_{2,n} \left(m_n^g - \frac{v_n^g}{2v_n^l} - I_1\left(m_n^l; m_n^g - \frac{v_n^g}{2v_n^l}, v_n^g\right)\right)\right) \\
\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{\mathcal{X}}_n}[x_n^2|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] &= \frac{\lambda \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(C_{1,n} I_2\left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g\right) + \right. \\
&\quad \left. C_{2,n} \left(\left(m_n^g - \frac{v_n^g}{2v_n^l}\right)^2 + v_n^g - I_2\left(m_n^l; m_n^g - \frac{v_n^g}{2v_n^l}, v_n^g\right)\right)\right). \tag{3.25}
\end{aligned}$$

where v_s is a constant that determines the variance of the distribution and it captures the confidence that the receiver has on how close SI is to the sparse signal.

Next, we develop a GAMP-based algorithm for one-bit CS with side-information. Note that the evaluation of $\mathbb{E}_{\mathcal{Z}_m|\mathbf{y}}[z_m|\mathbf{y}]$ and $\mathbb{E}_{\mathcal{Z}_m|\mathbf{y}}[z_m^2|\mathbf{y}]$ depends only on the distribution of the channel and hence is the same as in Algorithm . Next, we derive expressions for $G_1(\cdot)$ and $G_2(\cdot)$ when the receiver has access to SI. Here, we assume that the noise is Laplacian. The expectation is carried out with respect to random variable \mathcal{X}_n given $\hat{\mathcal{R}}_n = \hat{r}_n$, and $\tilde{\mathcal{X}}_n = \tilde{x}_n$ for random variables

$$\hat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad \tilde{\mathcal{X}}_n = \mathcal{X}_n + \mathcal{W}_n,$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_r^n)$, $\mathcal{W} \sim \mathcal{L}(0, 2v_s)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent. Therefore, the marginal posterior distribution can be approximated as

$$p_{\mathcal{X}_n|\mathbf{y}}(x_n|\mathbf{y}; \hat{r}_n, \tau_n^r) = \frac{p_{\mathcal{X}_n}(x_n)\mathcal{N}(x_n; \hat{r}_n, \tau_n^r)\mathcal{L}(x_n; \tilde{x}_n, 2v_s)}{\int_{x_n} p_{\mathcal{X}_n}(x_n)\mathcal{N}(x; \hat{r}_n, \tau_n^r)\mathcal{L}(x_n; \tilde{x}_n, 2v_s)}.$$

Using the approximated posterior density function, $p_{\mathcal{X}_n|\mathbf{y}, \tilde{x}_n}(x_n|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r)$, we evaluate the first-order moment, $\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{x}_n}[x_n|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$ and second-order moment, $\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{x}_n}[x_n^2|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$.

Result 3.1. Define $m_n^g \triangleq \frac{v_x \hat{r}_n}{v_x + \tau_n^r}$, $v_n^g \triangleq \frac{v_x \tau_n^r}{v_x + \tau_n^r}$, $m_n^l \triangleq \tilde{x}_n$, $v_n^l \triangleq v_s$, $C_{1,n} \triangleq \frac{1}{4v_n^l} \exp(-\frac{1}{2v_n^l}(m_n^l - m_n^g - \frac{v_n^g}{4v_n^l}))$, and $C_{2,n} \triangleq \frac{1}{4v_n^l} \exp(-\frac{1}{2v_n^l}(-m_n^l + m_n^g - \frac{v_n^g}{4v_n^l}))$. The posterior first-order and second-order moments are listed in (3.25).

The sketch of the derivations is provided in Appendix C. The first-order and second-order moments require evaluation of integrals $I_0(\cdot)$, $I_1(\cdot)$ and $I_2(\cdot)$. We have the following results on the closed-form expressions of these integrals.

Result 3.2. With $I_q(\tilde{x}_n; m_n, \tau_n^r) \triangleq \int_{-\infty}^{\tilde{x}_n} x_n^q \mathcal{N}(x_n|m_n, \tau_n^r) dx_n$, the analytical expressions of I_n^1 and I_n^2 are

$$\begin{aligned} I_1(\tilde{x}_n; m_n, \tau_n^r) &= m_n \Phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) - \sqrt{\tau_n^r} \phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) \\ I_2(\tilde{x}_n; m_n, \tau_n^r) &= m_n I_1(\tilde{x}_n; m_n, \tau_n^r) + \tau_n^r \Phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) - \tilde{x}_n \sqrt{\tau_n^r} \phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right). \end{aligned} \quad (3.26)$$

Proof. We know that $I_0(\tilde{x}; m, \tau) = \int_{-\infty}^{\tilde{x}} \mathcal{N}(x; m, \tau) dx = \Phi(\frac{\tilde{x}-m}{\sqrt{\tau}})$. Differentiating $I_0(\tilde{x}; m, \tau)$ with respect to m , we get

$$\begin{aligned} \frac{\partial I_0(\tilde{x}; m, \tau)}{\partial m} &= \int_{-\infty}^{\tilde{x}} \frac{x - m}{\tau} \mathcal{N}(x; m, \tau) dx \\ &= \frac{1}{\tau} \left\{ \int_{-\infty}^{\tilde{x}} x \mathcal{N}(x; m, \tau) dx - m \Phi\left(\frac{\tilde{x} - m}{\sqrt{\tau}}\right) \right\} \\ &\Rightarrow I_1(\tilde{x}; m, \tau) = m \Phi\left(\frac{\tilde{x} - m}{\sqrt{\tau}}\right) - \sqrt{\tau} \phi\left(\frac{\tilde{x} - m}{\sqrt{\tau}}\right). \end{aligned} \quad (3.27)$$

Differentiating $I_0(\tilde{x}; m, \tau)$ twice with respect to m , and following steps similar to those in (3.27), we get

$$I_2(\tilde{x}; m, \tau) = mI_1(\tilde{x}; m, \tau) + \tau \Phi\left(\frac{\tilde{x} - m}{\sqrt{\tau}}\right) - \tilde{x}\sqrt{\tau}\phi\left(\frac{\tilde{x} - m}{\sqrt{\tau}}\right). \quad (3.28)$$

□

With posterior first-order moments and second-order moments, we have all the statistical quantities required to implement Algorithm .

3.4.2 Estimation of the v_s

In the following, we employ the Expectation-Maximization (EM) algorithm to estimate the side-information parameter, v_s . The EM algorithm is an iterative technique that increases the lower bound on the likelihood $p(\mathbf{y}; v_s)$ at each iteration, which guarantees that the likelihood converges to a local maximum, or at least to a saddle point. Specifically, the EM algorithms iterates over two steps: 1) *Expectation step*: choosing distribution to maximize the lower bound for fixed $v_s = v_s^k$, and 2) *Maximization step*: choosing v_s to maximize the lower bound for the fixed distribution from Step 1. We emphasize that the maximizing pdf is the true posterior under the prior parameter, v_s . Since, it is very difficult to compute the true posterior, we use the posterior approximated by the GAMP algorithm in the evaluation of the expectation. The EM algorithm is summarized as

$$v_s^{k+1} = \operatorname{argmin}_{v_s} \mathbb{E}_{\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}; v_s^k} [-\log p(\mathbf{y}, \mathbf{x}, \tilde{\mathbf{x}}; v_s)], \quad (3.29)$$

where $p(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{x}}; v_s)$ is the joint probability distribution of the complete data and $p(\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}; v_s^k)$ is the approximated posterior density given the side-information which is parameterized by the

previous iteration estimate of v_s^k . We first carry out the expectation step as

$$\begin{aligned} \mathbb{E}_{\mathcal{X}|\mathcal{Y}, \tilde{\mathcal{X}}; v_s}[-\log p(\mathbf{y}, \mathbf{x}, \tilde{\mathbf{x}}, v_s)] = \\ \mathbb{E}_{\mathcal{X}|\mathcal{Y}, \tilde{\mathcal{X}}; v_s}[-\log p(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{x}}, v_s^t) - \log p(\mathbf{x}, \tilde{\mathbf{x}}|v_s) - \log p(v_s)]. \end{aligned}$$

We note that, the expectation step is followed by the maximization step, and all the terms that do not involve v_s eventually go to zero. Since $\log p(\mathbf{y}|\mathbf{x}, \tilde{\mathbf{x}})$ does not depend on v_s , we drop the term. Similarly, we will drop all the terms that do not depend on v_s in the subsequent steps.

$$\begin{aligned} \mathbb{E}_{\mathcal{X}|\mathcal{Y}, \tilde{\mathcal{X}}; v_s}[-\log p(\mathbf{x}, \tilde{\mathbf{x}}|v_s) - \log p(v_s)] = \\ \sum_{n=1}^N \left(\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}, \tilde{\mathcal{X}}; v_s} \left(\frac{|x_n - \tilde{x}_n|}{2v_s} \right) + \log(v_s) - \log p(v_s) \right), \end{aligned}$$

where the summation over indices is due to the fact that the posterior density, $p_{\mathcal{X}|\mathcal{Y}}$ is approximated as $p_{\mathcal{X}|\mathcal{Y}} = \prod_{n=1}^N p_{\mathcal{X}_n|\mathcal{Y}}$. From (3.29), the estimation of v_s can be written as

$$v_s^{k+1} = \underset{v_s}{\operatorname{argmin}} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_n|\mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} \frac{|x_n - \tilde{x}_n|}{2v_s} + \log(v_s) - \log p(v_s).$$

We assume a non-informative prior on the parameter v_s . Hence, we drop the $\log p(v_s)$ term and find the maximum likelihood estimate of v_s as

$$v_s^{k+1} = \frac{1}{2N} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_n|\mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} |x_n - \tilde{x}_n| \tag{3.30}$$

With the notations as defined in Result 3.1, we can evaluate the expectation in (3.30) in closed-form as

$$\begin{aligned}
\mathbb{E}_{\mathcal{X}_n|\mathcal{Y},\tilde{\mathcal{X}};v_s^k} |x_n - \tilde{x}_n| &= - \int_{-\infty}^{\tilde{x}} (x_n - \tilde{x}_n) \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) p_{\mathcal{X}_n}(x_n) \exp\left(\frac{x_n - \tilde{x}_n}{2v_s^k}\right) dx_n \\
&\quad + \int_{\tilde{x}_n}^{\infty} (x_n - \tilde{x}_n) \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) p_{\mathcal{X}_n}(x_n) \exp\left(-\frac{x_n - \tilde{x}_n}{2v_s^k}\right) dx_n \\
&= \frac{1}{Z_n^l} \left(\lambda \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) C_{2,n} \left(m_n^g - \frac{v_n^g}{2v_n^l} - I_1(m_n^l; m'_G - \frac{v_n^g}{2v_n^l}, v_n^g) \right. \right. \\
&\quad \left. \left. - \tilde{x}_n \left(1 - \Phi\left(\frac{m_n^l - (m_n^g - \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right) \right) \right) - \lambda C_{1,n} \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \right. \\
&\quad \left. \left(I_1(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g) - \tilde{x}_n \Phi\left(\frac{m_n^l - (m_n^g + \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right) \right) \right. \\
&\quad \left. + |\tilde{x}_n| (1 - \lambda) \mathcal{N}(0; \hat{r}_n, \tau_n^r) \mathcal{L}(0; \tilde{x}_n, 2v_s^k) \right)
\end{aligned} \tag{3.31}$$

Using (3.31) in (3.30), we find the estimate of the v_s using the EM algorithm.

Hence, we have derived all the expressions required for signal reconstruction from one-bit measurements with side-information. In Algorithm , we summarize the steps for signal reconstruction for one-bit compressed sensing with side-information with parameter estimation.

3.4.3 Noisy one-bit Compressed Sensing with Gaussian Noise (GaussianSI)

Next, we list the steps for the estimation of the sparse signals when the side-information is assumed to be the actual signal corrupted by Gaussian noise.

$$\hat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad \tilde{\mathcal{X}}_n = \mathcal{X}_n + \mathcal{W}_n,$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, $\mathcal{W}_n \sim \mathcal{N}(0, v_s)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent. Next, we state the results for the first and second order moments for this setup.

Result 3.3. *The posterior first-order and second-order moments of the signal given side-*

information, \tilde{x}_n are

$$\begin{aligned}\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r] &= \pi_n^g \frac{\hat{r}_n v_s v_x + v_x \tau_n^r \tilde{x}}{v_x \tau_n^r + \tau_n^r v_s + v_s v_x} \triangleq \pi_n^g m_n^g \\ \mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n^2|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r] &= \pi_n^g \left(\frac{v_s \tau_n^r v_x}{v_x \tau_n^r + \tau_n^r v_s + v_s v_x} + (m_n^g)^2 \right),\end{aligned}\tag{3.32}$$

where $\pi_n^g = \frac{\lambda}{\lambda + (1-\lambda)Z_n}$ and $Z_n = \frac{\mathcal{N}(0;\hat{x},\tau_n^r)\mathcal{N}(0;\tilde{x}_n,v_s)}{\mathcal{N}(0;\hat{r}_n,v_x+\tau_n^r)\mathcal{N}(0;\frac{\hat{r}_n v_x}{v_x+\tau_n^r}-\tilde{x},\frac{\tau_n^r v_x}{v_x+\tau_n^r}+v_s)}$. The sketch of derivations is provided in Appendix D. Next, we estimate the side-information parameter, v_s using the EM algorithm. Following the steps as in the Laplacian noise case, we can show that the maximum likelihood estimator of the v_s is,

$$\begin{aligned}v_s^{k+1} &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n;v_s^k}(x_n - \tilde{x}_n)^2 \\ &= \frac{1}{N} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n;v_s^k}(x_n^2) - 2\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n;v_s^k}(x_n) + \tilde{x}_n^2 \\ &= \frac{1}{N} \sum_{n=1}^N \pi_n^g \left(\frac{v_s^k \tau_n^r v_x}{v_x \tau_n^r + \tau_n^r v_s^k + v_s^k v_x} - \left(\pi_n^g \frac{\hat{r}_n v_s^k v_x + v_x \tau_n^r \tilde{x}_n}{v_x \tau_n^r + \tau_n^r v_s^k + v_s^k v_x} \right)^2 - 2\tilde{x}_n m_n^g + \tilde{x}_n^2 \right)\end{aligned}\tag{3.33}$$

where the equality is obtained by replacing $\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n;v_s^k}(x_n^2)$ and $\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n;v_s^k}(x_n)$ from (3.32).

With (3.32) and (3.33), we have evaluated all the expressions required for implementing the one-bit compressed sensing algorithm with Gaussian side-information. In the simulation section, we will discuss that modeling noise with Laplacian distribution in noise makes the proposed algorithm more robust when the side-information has partial support information or when the support in the side-information is erroneous .

3.5 Noisy One-bit Compressed Sensing with support as side-information

In this section, we investigate the problem of sparse signal reconstruction from noisy one-bit compressed measurements when the receiver has access to only support-information as SI. We develop a GAMP based algorithm by taking support information as SI into account. We assume that there are some discrepancies between the support of the signal and the SI. We model these discrepancies using multiplicative noise. Formally, let \mathcal{S}_n be a random variable that represents the support of the n -th index of the sparse signal, \mathcal{X}_n , for $n = 1, \dots, N$. \mathcal{S}_n takes values 1 and -1 depending on whether or not the signal index is in the support set of \mathcal{X} , i.e.,

$$s_n = \begin{cases} +1, & \text{if } x_n \neq 0, \\ -1, & \text{if } x_n = 0, \end{cases} \quad (3.34)$$

Let $\tilde{\mathcal{X}}_n$ be the n -th element of SI which is the noisy version of the actual support of the signal, i.e., \mathcal{S}_n . We assume that a small fraction of the support set is different (erroneous) in the SI from that of the signal. We model this relationship between SI and the actual support-set of the signal by

$$\tilde{\mathcal{X}}_n = \zeta_n \mathcal{S}_n,$$

where ζ_n is the multiplicative noise which can take values 1 or -1 , and is assumed to be a Bernoulli distributed random variable with probability β for event $\zeta_n = 1$ and probability $1 - \beta$ for event $\zeta_n = -1$, respectively. Thus

$$\begin{aligned} p(\tilde{x}_n = 1 | s_n = -1) &= p(\tilde{x}_n = -1 | s_n = 1) = 1 - \beta, \\ p(\tilde{x}_n = 1 | s_n = 1) &= p(\tilde{x}_n = -1 | s_n = -1) = \beta \end{aligned}$$

Next, we develop a GAMP-based algorithm for one-bit CS with erroneous support information as SI. As the evaluation of $\mathbb{E}_{\mathcal{Z}|Y}[z|y]$ and $\mathbb{E}_{\mathcal{Z}|Y}[z^2|y]$ depends only on the distribution of the channel (3.8), $F_1(\cdot)$ and $F_2(\cdot)$ are essentially the same as in subsection 3.3.1. Next, we derive expressions for $\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r]$ and $\mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n^2|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r]$ when the receiver has access to noisy support information as SI. The expectation is carried out with respect to the random variable \mathcal{X}_n given $\hat{\mathcal{R}}_n = \hat{r}_n$, and $\tilde{\mathcal{X}}_n = \tilde{x}_n$ for random variables

$$\hat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad \tilde{\mathcal{X}}_n = \zeta_n \mathcal{S}_n$$

where \mathcal{V}_n , ζ_n and \mathcal{X}_n are independent. Therefore, the marginal posterior distribution can be approximated as

$$p_{\mathcal{X}_n|\mathbf{y}}(x_n|\mathbf{y};\hat{r}_n,\tau_n^r) = \frac{p_{\mathcal{X}_n}(x_n)\mathcal{N}(x_n;\hat{r}_n,\tau_n^r)p_{\zeta_n}(\zeta_n = \frac{\tilde{x}_n}{s_n})}{\int_{x_n} p_{\mathcal{X}_n}(x_n)\mathcal{N}(x_n;\hat{r}_n,\tau_n^r)p_{\zeta_n}(\zeta_n = \frac{\tilde{x}_n}{s_n})}. \quad (3.35)$$

With (3.35) as the approximated marginal posterior density function, we express the analytical expression for posterior first-order and second-order moments as

Result 3.4. *Let π_n be the posterior probability of x_n being a non-zero element. Then*

$$\pi_n = \frac{\lambda p(\tilde{x}_n|s_n = 1)}{\lambda p(\tilde{x}_n|s_n = 1) + (1 - \lambda)p(\tilde{x}_n|s_n = 0)Z_n}$$

where $Z_n = \frac{\mathcal{N}(0;\hat{r}_n,\tau_n^r)}{\mathcal{N}(0;\hat{r}_n,v_x+\tau_n^r)}$. *The posterior first-order and second-order moments of the sparse signal given noisy support-information, \tilde{x}_n , are*

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r] &= \pi_n \frac{\hat{r}_n v_x}{v_x + \tau_n^r} \\ \mathbb{E}_{\mathcal{X}_n|\mathbf{y},\tilde{\mathcal{X}}_n}[x_n^2|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r] &= \pi_n \left(\frac{\tau_n^r v_x}{v_x + \tau_n^r} + \left(\frac{\hat{r}_n v_x}{v_x + \tau_n^r} \right)^2 \right), \end{aligned} \quad (3.36)$$

where $Z_n = \frac{\mathcal{N}(0; \hat{x}_n, \tau_n^r)}{\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}$. We can obtain the above results by the substitution of $p_{\mathcal{X}_n}(x_n)$ from (3.2) in (3.35), followed by representing the posterior density, $p_{\mathcal{X}_n|\mathbf{y}}$, as a Bernoulli-Gaussian pdf: $(1 - \pi_n)\delta(x) + \pi_n\mathcal{N}(x; m, v)$, and using the definition of first-order and second-order moments. Since the derivation of the first-order and the second-order moments is similar to the case when the noise is assumed to Laplacian, we omit the actual derivations. Note that, we assumed noisy support-information in the problem statement. Next, we estimate the noise parameter using the EM algorithm. Following the EM algorithm based approach in the previous section, the maximum likelihood estimate of β is

$$\begin{aligned}\beta^{t+1} &= \underset{\beta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}; \beta}[-\log p(\mathbf{y}, \mathbf{x}, \tilde{\mathbf{x}}; \beta)] \\ &= \underset{\beta}{\operatorname{argmin}} \mathbb{E}_{\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}; \beta}[-\log p(\tilde{\mathbf{x}}|\mathbf{x}; \beta)]\end{aligned}$$

With π_n as the posterior probability of n -th element of \mathbf{x} being non-zero, the expectation can be evaluated as

$$\begin{aligned}\mathbb{E}_{\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}; \beta}[-\log p(\tilde{\mathbf{x}}|\mathbf{x}; \beta)] &= \\ &\sum_{\{n: \tilde{x}_n=1\}} \log(1 - \beta)(1 - \pi_n) + \log(\beta)\pi_n \\ &+ \sum_{\{n: \tilde{x}_n=-1\}} \log(1 - \beta)\pi_n + \log(\beta)(1 - \pi_n),\end{aligned}$$

Next, we estimate the value of β that maximizes the expectation. Differentiating the expectation with respect to β and equating to zero, we get

$$\beta = \frac{\sum_{\{n: \tilde{x}_n=1\}} \pi_n + \sum_{\{n: \tilde{x}_n=-1\}} (1 - \pi_n)}{N} \quad (3.37)$$

With the results in (3.36) and (3.37), we can use Algorithm for estimating sparse signals from their one-bit compressed measurements with erroneous support information as the SI. Next, we provide simulation results for the proposed algorithms.

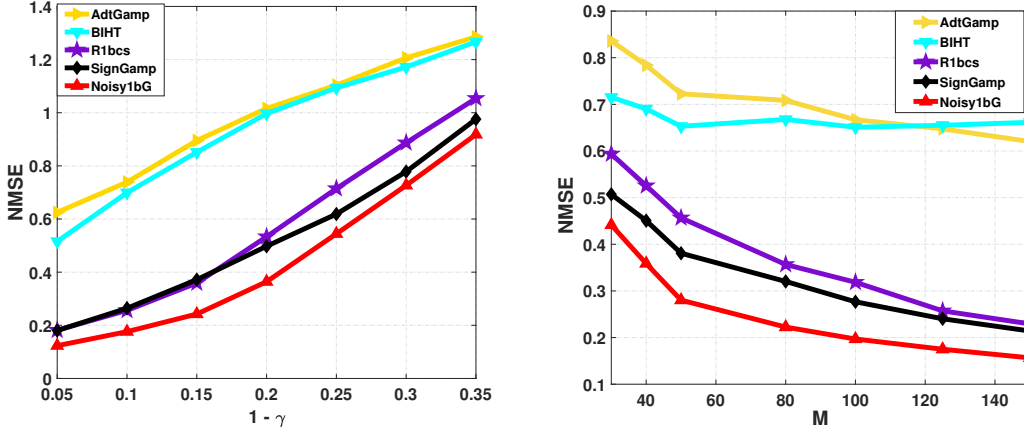


Fig. 3.2: Comparison of reconstruction performance of the proposed method as a function of sign-flip, $1 - \gamma$ and number of measurements, M when $N = 50$, $\lambda = 0.15$.

3.6 Simulation Results

In this section, we evaluate the signal reconstruction performance of the proposed sparse signal reconstruction algorithms from noisy one-bit measurements with the state-of-the-art algorithms. We consider the problem of reconstructing a sparse signal of dimension N from M noisy one-bit measurements. The measurement matrix, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is drawn from an i.i.d. Gaussian distribution with zero-mean and unit variance. We consider real-valued compressed measurements that are corrupted by AWGN noise before quantization and the sign-flip noise (Bernoulli) after quantization. We employ normalized mean square error (NMSE) as the performance metric which is defined as

$$\text{NMSE} = \sqrt{\left\| \left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} \right\|_2 \right\|_2^2}$$

where \mathbf{x} and $\hat{\mathbf{x}}$ are the actual signal and the reconstructed signal, respectively. We generate a sparse signal vector from the Bernoulli-Gaussian distribution with signal sparsity parameter $\lambda = 0.1$, mean zero and variance 5.5. We assume that the signal is corrupted by additive white Gaussian noise before quantization with mean zero and covariance $v\mathbb{I}_N$. After quantization,

the one-bit quantized measurements are corrupted by sign-flip noise generated from Bernoulli distribution with probability of sign flip $1 - \gamma$. We ran the algorithm for 500 Monte-Carlo runs.

In the first experiment, we evaluate the performance of the proposed one-bit CS algorithm with the state-of-the-art algorithms. In this experiment, we compare the performance of the proposed algorithm, Noisy1bG, with algorithms proposed in [47], [52], and [62] respectively and refer to these algorithms as BIHT, AdtGamp, and R1bcs. SignGAMP refers to the one-bit GAMP algorithm that does not take noise into account. In Figure 3.2, we summarize the NMSE performance of the one-bit algorithms as a function of $1 - \gamma$, and M . In Figure 3.2, we see that the proposed algorithm has superior performance compared to R1bcs, SignGamp, BIHT, and AdtGamp. BIHT and AdtGAMP perform the worst. Further, the BIHT algorithm does not account for the noise, which leads to poor performance. We note that the proposed algorithm performs better than the R1bcs algorithm, which is a Bayesian algorithm that is robust to sign-flip noise. Moreover, the R1bcs algorithm requires matrix inversion in the algorithm and is computationally expensive than the proposed algorithm. From the first experiment, we conclude that accounting for both pre-quantization and post-quantization noise leads to improved reconstruction performance. In the following experiments, we consider the performance of Noisy1bG as the baseline and compare the performance of the SI based algorithms.

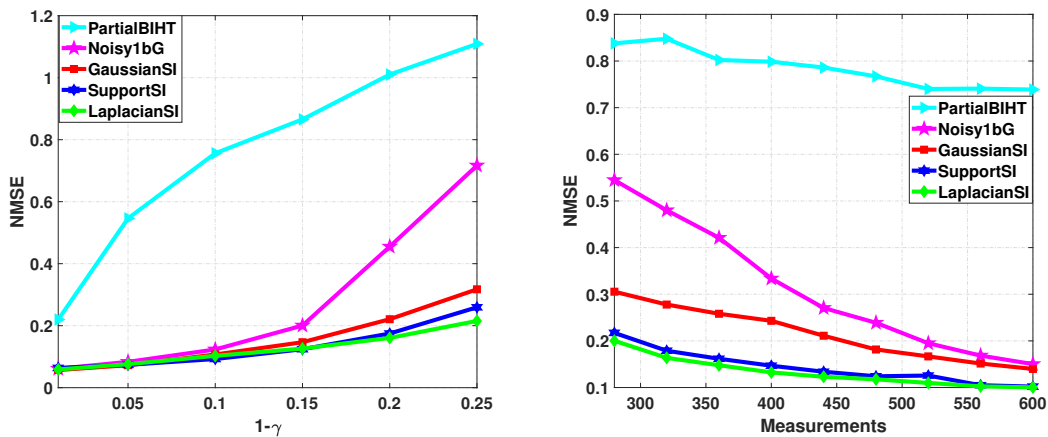


Fig. 3.3: Comparison of reconstruction performance of the proposed methods as a function of sign-flip probability and measurements in presence of SI when $N = 200$, $\lambda = 0.15$.

In the second experiment, we study the reconstruction performance of sparse signals from their noisy one-bit compressed measurements when the receiver has access to some SI. We assume that the SI is erroneous. A small fraction of the elements in the support set of SI do not lie in the support set of the compressed signal. Further, we assume some additive noise present in the SI. The additive noise and the change of support are modeled by the Laplacian noise and the Gaussian noise in the proposed algorithms `LaplacianSI`, and `GaussianSI`, respectively. With the noisy SI at the receiver, figure 3.3 demonstrates the reconstruction performance of the proposed algorithms. The first plot shows the performance of the proposed algorithms against sign-flip probability, and the second plot shows the performance of the proposed algorithms as a function of M . From both of these results, we conclude that all the proposed algorithms with SI perform better than the case when we do not have side-information. We emphasize that the `LaplacianSI` algorithm outperforms the `GaussianSI` algorithm. The error in support with the amplitude information between SI and the compressed signal can be modeled better by the Laplacian distribution than the Gaussian distribution. We further emphasize that the `SupportSI` algorithm only considers the support information as the side-information. We see that `SupportSI` performs better than the `GaussianSI` algorithm. As the change in support is difficult to model by Gaussian noise, we claim that the poor performance of `GaussianSI` is due to the modeling error.

Third, we consider the effect of noise in SI on the reconstruction performance from one-bit measurements. Like in the second experiment, the SI at the receiver has a fraction of elements in its support set, which are not in the support of the compressed signals. Further, the amplitudes of the SI are corrupted by additive noise. In the experiment, 10% of the elements in the support set of SI are not in the support set of the compressed signal. Further, we use Gaussian noise as additive noise in the SI. In Figure 3.4, we plot the results of the experiment. It is evident that the performance of algorithms `Noisy1bG` and `SupportSI` is relatively constant for different values of the variance of additive noise. For the `SupportSI` algorithm, we assume that

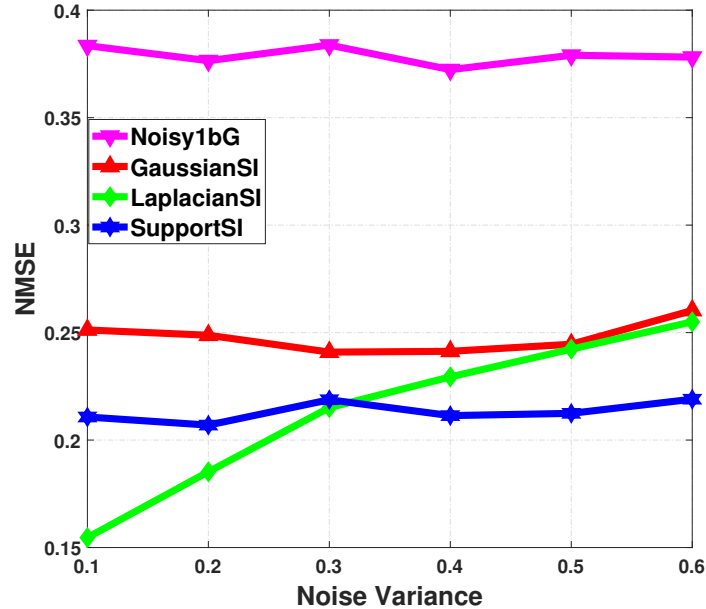


Fig. 3.4: Comparison of the effect of noise in SI on the reconstruction performance of proposed algorithms when $\lambda = 0.1$, $1 - \gamma = 0.15$, and $v = 0.15$,

the knowledge of support does not change with the additive noise; hence it does not affect the performance of the algorithm. Since the Gaussian density could not model the sparse nature of the noise vector well, the performance of the GaussianSI algorithm is worse than SupportSI algorithm for all values of the noise variance. The performance of the LaplacianSI degrades with the increase in the noise in SI. Note that, the performance of LaplacianSI is worse than SupportSI when the noise in the SI is above a certain level. Hence, using support information, if available, is better than using the entire SI signal when the signal to noise ratio of SI is small.

In the final experiment, we consider the case where the support of the observed sparse signal changes slowly over time. In the simulation, we generate a sequence of sparse signals such that 10% of the support changes between two consecutive time instants. For the first time instant, the non-zero elements are generated from an i.i.d. Gaussian distribution with mean zero and variance 5.5. We then obtain the amplitudes of the indices that continue to be in the support set of the signal by adding a random vector with zero mean and a small

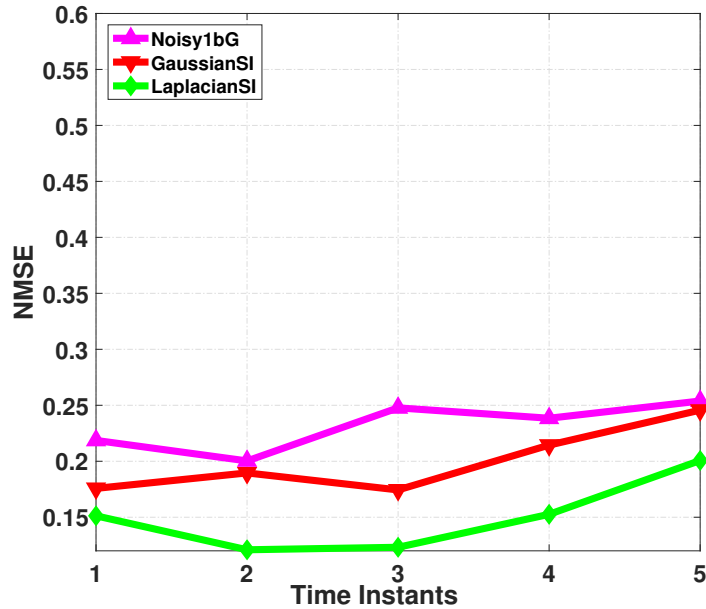


Fig. 3.5: traction performance of the proposed algorithms when $\lambda = 0.1$, $1 - \gamma = 0.15$, and $v = 0.15$, $N = 200$, and $M = 600$

variance generated from an i.i.d. Gaussian distribution. For the indices that are not in the support set of the signal at the previous time instant, the amplitudes are generated from an i.i.d. Gaussian distribution with mean zero and variance 5.5. The receiver has access to noisy one-bit measurements of these signals. The receiver estimates the sparse signal at the first time instant using the Noisy1bG algorithm. This estimate of the sparse signal is now fed to the GaussianSI and the LaplacianSI algorithms as the SI. Using this SI, the proposed algorithms estimate the compressed signal. In the next iteration, GaussianSI and LaplacianSI use their estimates of the previous time instant signal as the SI and estimate the compressed signal. In Figure 3.5, we show the NMSE performance of the proposed algorithms. We can see that LaplacianSI performs better than the GaussianSI and Noisy1bG algorithms. The GaussianSI algorithm, though worse than the LaplacianSI algorithm, performs better than the Noisy1bG algorithm. Hence we conclude that when the support of the signal changes slowly over time, using the signal reconstructed at the previous time instant as SI leads to improved performance than just

using a one-bit reconstruction algorithm.

3.7 Conclusion

This chapter developed signal reconstruction algorithms from one-bit measurements using the generalized approximate message passing (GAMP) framework considering a generalized noisy measurement process. We then considered the scenario when SI is available at the receiver. We developed two different algorithms that consider SI, which has either support information only or both support and amplitude information. We derived closed-form expressions for GAMP estimation functions for all the proposed algorithms. We showed that by incorporating SI, we could improve the reconstruction performance in terms of NMSE. Further, we showed that the noise in the SI is better modeled by the Laplacian noise than Gaussian noise. We used the EM algorithm to estimate the noise parameter that governs our SI model adaptively from one-bit measurements and the side-information. Future work can consider extending the given algorithms to centralized and decentralized settings, especially when different nodes in a network have access to SI.

CHAPTER 4

ONE-BIT COMPRESSED SENSING WITH SIDE-INFORMATION IN TIME-VARYING ENVIRONMENTS

4.1 Introduction

In the previous chapter, we assumed that the SI is a noisy version of the sparse signal. The proposed algorithm could not take advantage of any specific dependence between the signals and the SI. There are scenarios where a sequence of signals has temporal dependence. In this chapter, we consider the problem of signal reconstruction of time-varying sparse signals from noisy one-bit compressed measurements with side-information (SI). We model the time-varying nature of the signal using a birth-death-drift (BDD) model. We assume that one-bit compressed measurements are corrupted by additive white Gaussian noise before quantization and sign-flip error after quantization. The exact computation of the MMSE estimator requires evaluation of high-dimensional integrals that is computationally infeasible. We develop a computationally tractable algorithm to approximate the MMSE estimator of the signal, which takes advantage

of the underlying time-varying model and SI to improve reconstruction performance. SI is assumed to be noisy with independent distributions over signal indices. Laplace distribution, which is a sparsity promoting distribution, is chosen to model the noise in SI. We demonstrate that the proposed algorithm greatly improves signal reconstruction performance by exploiting the underlying dynamic model while being robust to noise in SI.

4.2 Signal and Measurement Models

4.2.1 Signal Model

We consider the case where the support of sparse signals changes slowly over time and there is a strong correlation between the non-zero elements over successive time instants. We assume that the signal continues to be sparse. We model the stochastic dependence of the time varying signals by the Birth-Death-Drift (BDD) [69,83] model. Let x_p and x_c be the previous time and the current time instant signals, respectively. For each index n , when the sparse signal changes slowly over time, there are four possible events:

- **Event 1:** $x_{pn} \propto \delta(\cdot)$ and $x_{cn} \propto \delta(\cdot)$, i.e., the n -th element which was zero at the previous time instant continues to be zero. Let λ_1 be the probability of this event.
- **Event 2: Death** $x_{pn} \propto \mathcal{N}(\cdot; 0, v_x)$ and $x_{cn} \propto \delta(\cdot)$, i.e., the n -th element that was non-zero at the previous time instant is zero at the current time instant. Let λ_2 be the probability of this event.
- **Event 3: Birth** $x_{pn} \propto \delta(\cdot)$ and $x_{cn} \propto \mathcal{N}(\cdot; 0, v_x)$, i.e., the n -th element that was zero at the previous time instant is non-zero at the current time instant. Let λ_3 be the probability of this event.
- **Event 4: Drift** $x_{pn}, x_{cn} \propto f(x_{pn}, x_{cn})$, i.e., the n -th element that was non-zero at the previous time instant continues to be non-zero. Let λ_4 be the probability of this event.

The joint probability density function (pdf) of \mathbf{x}_p and \mathbf{x}_c can be modeled by the following distribution.

$$\begin{aligned}
p(\mathbf{x}_c, \mathbf{x}_p) &= \prod_{n=1}^N (\lambda_1 \delta(x_{pn}) \delta(x_{cn}) + \lambda_2 \delta(x_{cn}) \mathcal{N}(x_{pn}; 0, v_x) \\
&\quad + \lambda_3 \delta(x_{pn}) \mathcal{N}(x_{cn}; 0, v_x)) + \lambda_4 f(x_{cn}, x_{pn}) \\
&= \prod_{n=1}^N \sum_{i=1}^4 \lambda_i p(x_{pn}, x_{cn} | i)
\end{aligned} \tag{4.1}$$

where we assume that the joint pdf is independent over indices n . In this work, for event 4, we consider a fixed dependence model between x_{pn} and x_{cn} as $x_{cn} = \rho x_{pn} + u$, where $x_{pn} \propto \mathcal{N}(0, v_x)$, and $u \propto \mathcal{N}(0, v)$. The signal model (4.1) finds one of its many applications in the channel estimation problem for wireless communication [69]. The channel estimation task is equivalent to the problem of sparse signal reconstruction. The channel is known to be slowly time-varying and exhibits strong dependencies between channel responses in adjacent time instants. The random variable, u , represents short-term fading due to multipath and oscillator drift, and ρ represents correlations or drift between non-zero elements of \mathbf{x} , and is inversely correlated to the amount of fading in a wireless channel. We emphasize that, though the algorithm we develop focuses on a specific dependence model, this approach is general and can be extended to any dependence model for event 4 in the BDD model or for any joint pdf of \mathbf{x}_p and \mathbf{x}_c that are independent over indices, i.e., $p(\mathbf{x}_c, \mathbf{x}_p) = \prod_{n=1}^N p(x_{cn}, x_{pn})$.

4.2.2 Measurement Model

Figure 4.1 shows the transmission chain of the measurement model of the problem considered in this work. The sparse signal $\mathbf{x}_c \in \mathbb{R}^N$ is linearly transformed to $\mathbf{z} \in \mathbb{R}^M$ using the random projection matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. The measurement vector, \mathbf{z} , is assumed to be corrupted by additive i.i.d. Gaussian noise vector with mean zero and variance σ^2 , i.e., $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbb{I}_M)$. This corrupted signal is quantized to +1 or -1 using sign quantizer. We assume that the

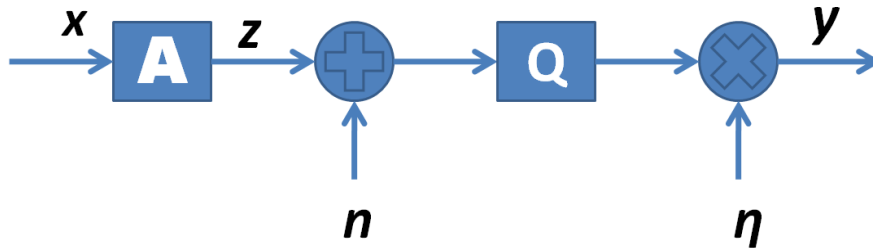


Fig. 4.1: 1-bit CS with pre-quantization and post-quantization noise.

quantized measurements are corrupted by multiplicative noise which takes value either $+1$ or -1 . Formally, the quantized measurement model can be written as

$$\mathbf{y} = \boldsymbol{\eta} \odot Q(\mathbf{A}\mathbf{x} + \mathbf{n}), \quad (4.2)$$

where the quantizer $Q : \mathbb{R}^M \rightarrow \{-1, +1\}^M$ is the element-wise sign quantizer. The m -th element at the output of the quantizer is

$$[Q(\zeta)]_m = \begin{cases} +1, & \text{if } \zeta_m > 0, \\ -1, & \text{if } \zeta_m \leq 0, \end{cases} \quad (4.3)$$

and $\boldsymbol{\eta} \in \{-1, +1\}^M$ is the post-quantization noise. $\boldsymbol{\eta}$ is assumed to follow Bernoulli distribution with $Pr(\eta_m = 1) = \gamma$. We define the inverse of quantization function, $Q^{-1}(\cdot)$, as

$$[Q^{-1}(\mathbf{y})]_m = \begin{cases} (-\infty, 0], & \text{if } y_m \leq 0, \\ (0, \infty), & \text{if } y_m > 0, \end{cases} \quad (4.4)$$

where y_m is the m -th element of \mathbf{y} .

4.2.3 Noisy One-Bit Compressed Sensing

Let $\tilde{\mathbf{x}}$ represent the SI of the signal \mathbf{x}_c . Here, we construct the joint posterior density of the signal at the current time instant signal, \mathbf{x}_c , and at the previous time instant signal, \mathbf{x}_p , given the noisy one-bit compressed measurements \mathbf{y} and SI $\tilde{\mathbf{x}}$ as

$$p(\mathbf{x}_p, \mathbf{x}_c | \mathbf{y}, \tilde{\mathbf{x}}) \propto p(\mathbf{y} | \mathbf{x}_c) p(\mathbf{x}_p, \mathbf{x}_c | \tilde{\mathbf{x}}) \propto \prod_{m=1}^M \mathbb{I}_{z_m \in \{Q^{-1}(\eta_m y_m)\}} \prod_{n=1}^N p(x_{pn}, x_{cn} | \tilde{x}_n), \quad (4.5)$$

where $p(x_{pn}, x_{cn})$ is the joint density function that gives the joint statistical characterization of the sparse signal at the previous time instant and at the current time instant. The minimum mean square error (MMSE) estimator of \mathbf{x}_c given SI is $\mathbb{E}[\mathbf{x}_c | \mathbf{y}, \tilde{\mathbf{x}}]$. We note that the derivation of the MMSE estimator (4.5) is intractable in direct form. So, we develop a GAMP-based algorithm to approximate the MMSE estimator.

4.3 Noisy One-Bit Compressed Sensing Algorithm with SI in Time-varying Environments

In this section, we develop a one-bit CS signal recovery algorithm based on GAMP that exploits the available SI at the receiver.

In the subsequent sections, we use GAMP algorithm based on the sum-product belief propagation algorithm to compute the MMSE estimator of \mathbf{x}_c corresponding to the posterior density (4.3).

4.3.1 Noisy One-Bit Compressed Sensing in a Time-Varying Environment

The sparse signal \mathbf{x}_c is linearly transformed to \mathbf{z} which is corrupted by i.i.d. Gaussian noise. This corrupted measurement is quantized to one-bit as defined in (4.18) and is propagated over

Algorithm 4.1 GAMP Algorithm for noisy one-bit CS with SI (1bitdyn)

1. Initialization: Set $t=0$ and initialize $\hat{\mathbf{x}}^t = \mathbb{E}_{\mathcal{X}}[\mathbf{x}]$, $\boldsymbol{\tau}_x^t = \text{var}_{\mathcal{X}}[\mathbf{x}]$, $\hat{\mathbf{s}}^t = 0$,
2. **While** loop $l < L_{max}$
3. **While** loop $t < T_{max}$
4. **Measurement Update**

- Linear Step

$$\boldsymbol{\tau}^{p,t+1} = (\mathbf{A} \odot \mathbf{A})\boldsymbol{\tau}_x^t, \quad \hat{\mathbf{p}}^{t+1} = \mathbf{A}\hat{\mathbf{x}}^t - \boldsymbol{\tau}^{p,t+1} \odot \hat{\mathbf{s}}^t,$$

- Non-Linear Step

$$\begin{aligned} \hat{\mathbf{s}}^{t+1} &= F_1(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \boldsymbol{\tau}^{p,t+1}), \\ \boldsymbol{\tau}_s^{t+1} &= F_2(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \boldsymbol{\tau}^{p,t+1}), \end{aligned} \quad (4.6)$$

where F_1 and F_2 are applied element-wise and are defined as

$$\begin{aligned} F_1(y_m, \hat{p}_m^{t+1}, \tau_m^{p,t+1}) &= \frac{1}{\tau_m^{p,t+1}} \left(\mathbb{E}_{\mathcal{Z}_m | \mathcal{Y}_m} [z_m | y_m] - \hat{p}_m \right), \\ F_2(y_m, \hat{p}_m^{t+1}, \tau_m^{p,t+1}) &= \frac{1}{\tau_m^{p,t+1}} \left(1 - \frac{\text{var}_{\mathcal{Z}_m | \mathcal{Y}_m} [z_m | y_m]}{\tau_m^p} \right). \end{aligned} \quad (4.7)$$

The expectation and the variance are evaluated with respect to $z \sim \mathcal{N}(\hat{p}, \tau^p)$.

5. Estimation Update

Linear Step

$$\begin{aligned} \boldsymbol{\tau}^{r,t+1} &= ((\mathbf{A} \odot \mathbf{A})^T \boldsymbol{\tau}_s^{t+1})^{-1}, \\ \hat{\mathbf{r}}^{t+1} &= \hat{\mathbf{x}}^t + \boldsymbol{\tau}^{r,t+1} \odot (\mathbf{A}^T \hat{\mathbf{s}}^{t+1}), \end{aligned}$$

where the inversion is performed element-wise

Non-linear Step

$$\hat{\mathbf{x}}^{t+1} = G_1(\hat{\mathbf{r}}^{t+1}, \boldsymbol{\tau}^{r,t+1}), \quad (4.8)$$

$$\boldsymbol{\tau}_x^{t+1} = G_2(\hat{\mathbf{r}}^{t+1}, \boldsymbol{\tau}^{r,t+1}), \quad (4.9)$$

where G_1 and G_2 are applied element-wise and are defined as

$$\begin{aligned} G_1(\hat{r}_n, \tau_n^r) &= \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n} [x_{cn} | \mathbf{y}, \tilde{\mathbf{x}}_n; \hat{r}_n, \tau_n^r], \\ G_2(\hat{r}_n, \tau_n^r) &= \text{var}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n} [x_{cn} | \mathbf{y}, \tilde{\mathbf{x}}_n; \hat{r}_n, \tau_n^r]. \end{aligned} \quad (4.10)$$

The expectation and the variance are evaluated with respect to $p_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n}$, and can be computed by using (4.13).

Set $t = t+1$

6. **End While**
 7. Update v_s using (4.16)
 8. $l = l + 1$
 9. **End While**
-

a noisy binary symmetric channel with probability of sign-flip $1 - \gamma$. The entire effect of additive white Gaussian noise (measurement noise), one-bit quantization and sign-flip error (channel noise) is modeled by a probabilistic channel, $p_{\mathbf{y}|\mathbf{z}}(\mathbf{y}|\mathbf{z}; \sigma^2, \gamma)$. Since we assume that the measurement noise and the channel noise are i.i.d., the channel is represented as

$$p_{\mathbf{y}|\mathbf{z}}(\mathbf{y}|\mathbf{z}; \sigma^2, \gamma) = \prod_{m=1}^M p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m; \sigma^2, \gamma). \quad (4.11)$$

In Algorithm 4.1, we summarize the steps of the GAMP algorithm for noisy one-bit CS with SI for a dynamic time-varying sequence of sparse signals. We refer to this algorithm as 1bdyn. The algorithm has the following three major steps

Evaluation of $F_1(\cdot)$ and $F_2(\cdot)$

In Step 4 of Algorithm 4.1, we need to evaluate the non-linear functions $F_1(\cdot)$ and $F_2(\cdot)$. As defined in Equation (4.7), we can evaluate $F_1(\cdot)$ and $F_2(\cdot)$ by computing the posterior mean and variance of \mathcal{Z}_m , i.e., $\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}_m}(z_m|y_m)$ and $\text{var}_{\mathcal{Z}_m|\mathcal{Y}_m}(z_m|y_m)$. GAMP uses the central limit theorem to approximate the distribution of random variable \mathcal{Z}_m to be Gaussian with mean \hat{p}_m and variance τ_m^p , i.e., $\mathcal{Z}_m \sim \mathcal{N}(\hat{p}_m, \tau_m^p)$. These parameters are computed in Step 4 of Algorithm 4.1. Using Bayes rule, the posterior marginal density, $p_{\mathcal{Z}_m|\mathcal{Y}_m}(z_m|y_m)$, can be approximated as

$$p_{\mathcal{Z}_m|\mathcal{Y}_m}(z_m|\mathbf{y}; \hat{p}_m, \tau_m^p) = \frac{p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m)\mathcal{N}(z_m; \hat{p}_m, \tau_m^p)}{\int_z p_{\mathcal{Y}_m|\mathcal{Z}_m}(y_m|z_m)\mathcal{N}(z_m; \hat{p}_m, \tau_m^p)}.$$

In Appendix B, we provide exact expressions to evaluate $\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m|y_m]$ and $\mathbb{E}_{\mathcal{Z}_m|\mathcal{Y}}[z_m^2|y_m]$ which are used to evaluate $F_1(\cdot)$ and $F_2(\cdot)$.

Evaluation of $G_1(\cdot)$ and $G_2(\cdot)$

$G_1(\cdot)$ and $G_2(\cdot)$ are the functions of one-bit measurements and SI at the receiver. The side information \tilde{x}_n is considered to be the noisy estimate of the actual signal x_{pn} at the previous time-instant. The estimated signal can have errors either in amplitudes or both the supports and the amplitudes of the signal. The error in the estimate can be modeled as an additive noise which has a sparse structure. We use the Laplacian distribution to model the sparse structure of the noise. Similarly, the GAMP algorithm estimates, x_{cn} with \hat{r}_n . At every iteration, the algorithm has access to $\hat{\mathcal{R}}_n = \hat{r}_n$, and $\tilde{\mathcal{X}}_n = \tilde{x}_n$ which are defined as

$$\hat{\mathcal{R}}_n = \mathcal{X}_{cn} + \mathcal{V}_n, \quad \tilde{\mathcal{X}}_n = \mathcal{X}_{pn} + \mathcal{W}_n \quad (4.12)$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$ and $\mathcal{W}_n \sim \mathcal{L}(0, 2v_s)$ are independent additive noises. Note that \mathcal{X}_{pn} and \mathcal{X}_{cn} follows joint pdf in (4.1) and are independent with additive noises. The non-linear functions $G_1(\cdot)$, and $G_2(\cdot)$ defined in Equation (4.10) are the posterior mean and the variance of the signal given the SI. Next, we present results for evaluating the posterior first-order moment and second-order moments in Result 4.1.

Result 4.1. Define $I_q(a; m, v) = \int_{-\infty}^a x_n^q \mathcal{N}(x_n; m, v) dx_n$, and

$$PI_q(\mu, \tau, m, v) = \int x_n^q \Phi\left(\frac{x_n - \mu}{\sqrt{\tau}}\right) \mathcal{N}(x_n; m, v) dx_n.$$

$$\text{Let } PI_{qn}^{1,2} = PI_q(m_{2n}, v_{2n}, m_{1n}, v_{1n}), PI_{qn}^{3,4} = PI_q(m_{4n}, v_{4n}, m_{3n}, v_{3n}),$$

$$C_{1n} = \frac{1}{4v_s} \exp\left(-\frac{1}{2v_s}\left(\tilde{x}_n - m_n^g - \frac{v_n^g}{4v_s}\right)\right),$$

$$C_{2n} = \frac{1}{4v_s} \exp\left(-\frac{1}{2v_s}\left(-\tilde{x}_n + m_n^g - \frac{v_n^g}{4v_s}\right)\right),$$

$$c'_{1n} = \frac{1}{4v_s} \exp\left(-u_{cn}\left(\left(\tilde{x}_n - \frac{v_{pn}}{4v_s}\right)\left(\rho + \frac{\tau_n^r}{\rho v_x}\right) - \hat{r}_n\right)\right) \exp\left(-\frac{\hat{r}_n v}{v_x + \tau_n^r} u_{cn} + \frac{v_{1n} u_{cn}^2}{2}\right),$$

$$c'_{2n} = \frac{1}{4v_s} \exp\left(-u_{cn}\left(\left(-\tilde{x}_n - \frac{v_{pn}}{4v_s}\right)\left(\rho + \frac{\tau_n^r}{\rho v_x}\right) + \hat{r}_n\right)\right) \exp\left(\frac{\hat{r}_n v}{v_x + \tau_n^r} u_{cn} + \frac{v_{1n} u_{cn}^2}{2}\right),$$

$$u_{cn} = \left(2v_s\left(\rho + \frac{\tau_n^r}{\rho v_x}\right)\right)^{-1},$$

$$m_n^l = \tilde{x}_n, \quad m_n^g = \frac{\rho \hat{r}_n v_x}{v_x + \tau_n^r}, \quad v_n^g = \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}, \quad v_n^l = v_s,$$

$$m_{1n} = \frac{\hat{r}_n v}{v_x + \tau_n^r} - u_{cn} v_{1n}, \quad m_{2n} = \hat{r}_n - \left(\rho + \frac{\tau_n^r}{\rho v_x}\right)\left(\tilde{x}_n - \frac{v_{pn}}{2v_n^l}\right),$$

$m_{3n} = \frac{\hat{r}_n v}{v_x + \tau_n^r} + u_{cn} v_{1n}$, $m_{4n} = \hat{r}_n - (\rho + \frac{\tau_n^r}{\rho v_x})(\tilde{x}_n + \frac{v_{pn}}{2v_x^l})$, $m_{pn} = \frac{\rho v_x (\hat{r}_n - u_n)}{\rho^2 v_x + \tau_n^r}$,
 $v_{1n} = \frac{(\rho^2 v_x + \tau_n^r)v}{v_x + \tau_n^r}$, $v_{2n} = (\rho + \frac{\tau_n^r}{\rho v_x})^2 v_{pn}$, $v_{3n} = \frac{(\rho^2 v_x + \tau_n^r)v}{v_x + \tau_n^r}$, $v_{4n} = (\rho + \frac{\tau_n^r}{\rho v_x})^2 v_{pn}$,
 and $v_{pn} = \frac{v_x \tau_n^r}{\rho^2 v_x + \tau_n^r}$. The closed-form expressions for $\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathbf{y}, \tilde{x}_n} [x_{cn} | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$, and $\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathbf{y}, \tilde{x}_n} [x_{cn}^2 | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$ are presented in (4.13). The details of the derivations are provided in Appendix E.

Note that the first-order and second-order moments require the evaluation of integrals $I_q(\cdot)$, and $PI_q(\cdot)$ for $q = 1$ and 2 . We have stated the closed-form expressions for $PI_q(\cdot)$ in the evaluation of $F_1(\cdot)$ and $F_2(\cdot)$ in (4.13). Next, we state results on the closed-form expression of integrals $I_q(\cdot)$ for $q = 1$ and 2 .

Result 4.2. With $I_q(\tilde{x}_n; m_n, \tau_n^r) \triangleq \int_{-\infty}^{\tilde{x}_n} x_n^q \mathcal{N}(x_n; m_n, \tau_n^r) dx_n$, the analytical expressions of I_1 and I_2 are

$$\begin{aligned}
 I_1(\tilde{x}_n; m_n, \tau_n^r) &= m_n \Phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) - \sqrt{\tau_n^r} \phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) \\
 I_2(\tilde{x}_n; m_n, \tau_n^r) &= m_n I_1(\tilde{x}_n; m_n, \tau_n^r) + \tau_n^r \Phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right) \\
 &\quad - \tilde{x}_n \sqrt{\tau_n^r} \phi\left(\frac{\tilde{x}_n - m_n}{\sqrt{\tau_n^r}}\right).
 \end{aligned}$$

Using Result 4.1 and Result 4.2 we can evaluate functions $G_1(\cdot)$ and $G_2(\cdot)$ required in Step 5 of the Algorithm 4.1.

SI Parameter Estimation

SI is considered to be a noisy version of the signal. The noise distribution is assumed to be the Laplacian distribution. In the following, we resort to the expectation-maximization (EM) algorithm, a popular tool in maximum-likelihood parameter estimation, to estimate the SI parameter, v_s as defined in (4.12). The EM algorithm is an iterative technique that increases a lower bound on the likelihood $p(\mathbf{y}; v_s)$ at each iteration, which guarantees that the likelihood converges to a local maximum, or at least to a saddle point. Specifically, the EM algorithm

Table 4.1: GAMP Equations for SI in a dynamic setting

$$\begin{aligned}
Z_n^l &= \int p_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathcal{X}}_n}(x_c, x_p | \hat{r}_n, \tilde{x}_n) = \lambda_1 \mathcal{L}(0; |\tilde{x}_n, 2v_s) \mathcal{N}(0; \hat{r}_n, \tau_n^r) + \lambda_3 \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \mathcal{L}(0; \tilde{x}_n, 2v_s) \\
&\quad + \lambda_2 \mathcal{N}(0; \hat{r}_n, \tau_n^r) \int \mathcal{L}(x; \tilde{x}_n, 2v_s) \mathcal{N}(x; 0, v_x) dx \\
&\quad + \lambda_4 \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \int \mathcal{N}(x; \frac{\rho \hat{r}_n v_x}{v_x + \tau_n^r}, \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}) \mathcal{L}(x; \tilde{x}_n, 2v_s) dx \\
\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathcal{X}}_n}[x_{cn} | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] &= \frac{\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(\lambda_3 \mathcal{L}(0; \tilde{x}_n, 2v_s) \frac{v_x \hat{r}_n}{v_x + \tau_n^r} + \lambda_4 \left[c'_{1n} P I_{1n}^{1,2} \right. \right. \\
&\quad \left. \left. + c'_{2n} (m_{3n} - P I_{1n}^{3,4}) + \rho \left(C_{1n} I_1 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) + C_{2n} \left(m_n^g - \frac{v_n^g}{4v_n^l} - I_1 \left(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g \right) \right) \right] \right) \\
\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathcal{X}}_n}[x_{cn}^2 | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] &= \frac{\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(\lambda_3 \mathcal{L}(0; \tilde{x}_n, 2v_s) \left(\frac{v_x \tau_n^r}{v_x + \tau_n^r} + \left(\frac{v_x \hat{r}_n}{v_x + \tau_n^r} \right)^2 \right) \right. \\
&\quad \left. + \lambda_4 \left[\left(\rho^2 - \frac{2\rho^2 v}{\tau_n^r + v} \right) \left(C_{1n} I_2 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) + C_{2n} \left(v_n^g + \left(m_n^g - \frac{v_n^g}{4v_n^l} \right)^2 - I_2 \left(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g \right) \right) \right] \right. \\
&\quad \left. + c'_{1n} P I_{2n}^{1,2} + c'_{2n} \left((m_{3n})^2 + v_{3n} - P I_{2n}^{3,4} \right) + \frac{2\rho \hat{r}_n v}{\tau_n^r + v} \left(C_{1n} I_1 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) \right. \right. \\
&\quad \left. \left. + C_{2n} \left(m_n^g - \frac{v_n^g}{4v_n^l} - I_1 \left(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g \right) \right) \right] \right). \tag{4.13}
\end{aligned}$$

starts with an initial guess, v_s^0 . Then, the EM algorithm alternates between the expectation and the maximization steps in the following manner 1) *Expectation step*: choosing distribution to maximize the lower bound for fixed $v_s = v_s^k$, and 2) *Maximization step*: choosing v_s to maximize the lower bound for the fixed distribution from Step 1. The algorithm continues until the convergence criterion is met, or the maximum number of iterations is reached. We emphasize that maximizing the pdf in the maximization step is the true posterior under the prior parameter, v_s . Since it is computationally prohibitive to compute the true posterior, we use the posterior approximated by the GAMP algorithm in the evaluation of the expectation. The EM algorithm is summarized as

$$v_s^{k+1} = \underset{v_s}{\operatorname{argmin}} \mathbb{E}_{\mathcal{X}_c, \mathcal{X}_p | \mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} [-\log p(\mathbf{y}, \mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}}; v_s)], \quad (4.14)$$

where $p(\mathbf{y}, \mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}}; v_s)$ is the joint probability density of the complete data and $p(\mathbf{x}_c, \mathbf{x}_p | \mathbf{y}, \tilde{\mathbf{x}}, v_s^k)$ is the approximated posterior joint density given the SI which is parameterized by the previous iteration estimate of v_s^k . We first carry out the expectation step as

$$\begin{aligned} & \mathbb{E}_{\mathcal{X}_c, \mathcal{X}_p | \mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} [-\log p(\mathbf{y}, \mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}}; v_s)] \\ &= \mathbb{E}_{\mathcal{X}_c, \mathcal{X}_p | \mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} [-\log p(\mathbf{y} | \mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}}, v_s) - \log p(\mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}} | v_s) - \log p(v_s)]. \end{aligned}$$

Since the maximization steps follow the expectation step, we drop all the terms that do not involve v_s as they go to zero.

$$\begin{aligned} & \mathbb{E}_{\mathcal{X}_c, \mathcal{X}_p | \mathcal{Y}, \tilde{\mathcal{X}}; v_s^k} [-\log p(\mathbf{x}_c, \mathbf{x}_p, \tilde{\mathbf{x}} | v_s) - \log p(v_s)] = \\ & \sum_{n=1}^N \left(\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathcal{X}}_n; v_s^k} \left[\frac{|x_{pn} - \tilde{x}_n|}{2v_s} \right] + \log(v_s) - \log p(v_s) \right), \end{aligned}$$

We assume the non-informative prior on the parameter v_s . Hence, we drop $\log p(v_s)$ term and find the maximum likelihood estimate of v_s as

$$v_s^{k+1} = \frac{1}{2N} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathbf{y}, \tilde{\mathcal{X}}_n; v_s^k} |x_{pn} - \tilde{x}_n| \quad (4.15)$$

While evaluating the expectation, it is required to evaluate the integrals for all the four events in (4.1). For simplicity, define $\tilde{\mathbb{E}}_{i,k} |x_{pn} - \tilde{x}_n| \triangleq p(\hat{r}_n, \tilde{x}_n | i) \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathbf{y}, \tilde{\mathcal{X}}_n; v_s^k} [|x_{pn} - \tilde{x}_n| | \mathbf{y}, \tilde{x}_n, i; \hat{r}_n, \tau_n^r]$.

1) *Event 1:* $x_{pn} \propto \delta(\cdot)$, and $x_{cn} \propto \delta(\cdot)$

$$\tilde{\mathbb{E}}_{i=1,k} |x_{pn} - \tilde{x}_n| = |\tilde{x}_n| \mathcal{N}(0; \hat{r}_n; \tau_n^r) \mathcal{L}(0; \tilde{x}_n, 2v_s^k)$$

2) *Event 2:* $x_{pn} \propto \mathcal{N}(\cdot; 0, v_x)$, and $x_{cn} \propto \delta(\cdot)$

$$\begin{aligned} \tilde{\mathbb{E}}_{i=2,k} |x_{pn} - \tilde{x}_n| &= \mathcal{N}(0; \hat{r}_n; \tau_n^r) \left\{ -C_n^{1'} I_1\left(\tilde{x}_n; \frac{v_x}{2v_s^k}, v_x\right) \right. \\ &\quad + C_n^{2'} \left(-\frac{v_x}{2v_s} - I_1\left(\tilde{x}_n; -\frac{v_x}{2v_s^k}, v_x\right) \right) \\ &\quad \left. + \tilde{x}_n \left(C_n^{1'} \Phi\left(\frac{\tilde{x}_n - \frac{v_x}{2v_s^k}}{\sqrt{v_x}}\right) - C_n^{2'} \left(1 - \Phi\left(\frac{\tilde{x}_n + \frac{v_x}{2v_s^k}}{\sqrt{v_x}}\right)\right) \right) \right\}, \end{aligned}$$

where $C_n^{1'}$ and $C_n^{2'}$ follow the definition of C_1 and C_2 of Appendix A with $m_L = \tilde{x}_n$, $m_G = 0$, $v_L = v_s^k$, and $v_G = v_x$.

3) *Event 3:* $x_{pn} \propto \delta(\cdot)$ and $x_{cn} \propto \mathcal{N}(\cdot; 0, v_x)$

$$\tilde{\mathbb{E}}_{i=3,k} |x_{pn} - \tilde{x}_n| = |\tilde{x}_n| \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \mathcal{L}(0; \tilde{x}_n, 2v_s^k)$$

4) *Event 4:* $x_{pn} \neq 0, x_{cn} \neq 0$:

$$\begin{aligned} \tilde{\mathbb{E}}_{i=4,k} |x_{pn} - \tilde{x}_n| &= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \\ &\int_{-\infty}^{\tilde{x}_n} -(x_{pn} - \tilde{x}_{pn}) \mathcal{N}(x_{pn}; m'_{1n}, v'_{1n}) \mathcal{L}(x_{pn}; \tilde{x}_n, 2v_s^k) dx_{pn} \\ &+ \int_{-\infty}^{\tilde{x}_n} (x_{pn} - \tilde{x}_n) \mathcal{N}(x_{pn}; m'_{1n}, v'_{1n}) \mathcal{L}(x_{pn}; \tilde{x}_n, 2v_s^k) dx_{pn} \end{aligned}$$

where $m'_{1n} = \frac{\rho \hat{r}_n v_x}{v_x + \tau_n^r}$, and $v'_{1n} = \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}$. Therefore, following the intermediate steps similar to those in Appendix C, we get

$$\begin{aligned} &\tilde{\mathbb{E}}_{i=4,k} |x_{pn} - \tilde{x}_n| \\ &= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \left\{ -C''_{1n} I_1(\tilde{x}_n; m'_{1n} + \frac{v'_{1n}}{2v_s}, v'_{1n}) \right. \\ &+ C''_{2n} \left(m'_{1n} - \frac{v_1}{2v_s} - I_1(\tilde{x}_n; m'_{1n} - \frac{v_1}{2v_s}, v'_{1n}) \right) \\ &\left. - \tilde{x}_n \left(C''_{1n} \Phi\left(\frac{\tilde{x}_n - (m'_{1n} + \frac{v'_{1n}}{2v_s})}{\sqrt{v'_{1n}}}\right) + C''_{2n} \Phi\left(-\frac{\tilde{x}_n - (m'_{1n} - \frac{v'_{1n}}{2v_s})}{\sqrt{v'_{1n}}}\right) \right) \right\} \end{aligned}$$

Note that C''_{1n} and C''_{2n} are the functions of $m'_{1n}, v'_{1n}, \tilde{x}_n$ and v_s^k . The ML estimate of v_s is

$$v_s^{k+1} = \frac{1}{2N} \sum_{n=1}^N \sum_{i=1}^4 \frac{\lambda_i \tilde{\mathbb{E}}_{i,k} |x_{pn} - \tilde{x}_n|}{Z_n^l} \quad (4.16)$$

Hence, we have derived all the statistical quantities required for signal recovery from one-bit measurements with SI. We summarize all the steps of the algorithm in Algorithm 4.1.

4.3.2 Variations of Algorithm 4.1

In this subsection, we provide two different algorithms that we used to compare the performance of the proposed algorithm. We briefly state these algorithms and provide the details of the update equations of GAMP based algorithms in Appendix F.

1bitdynG

This algorithm assumes noise in SI to be additive white Gaussian. At every iteration, the algorithm has access to $\widehat{\mathcal{R}}_n = \widehat{r}_n$, and $\widetilde{\mathcal{X}}_n = \widetilde{x}_n$ which are defined as

$$\widehat{\mathcal{R}}_n = \mathcal{X}_{cn} + \mathcal{N}(0, \tau_n^r) \quad \widetilde{\mathcal{X}}_n = \mathcal{X}_{pn} + \mathcal{W}_n \quad (4.17)$$

where $\mathcal{W}_n \sim \mathcal{N}(0, v_s)$ and $\mathcal{N}(0, \tau_n^r)$ are independent. We refer to this algorithm as 1bitdynG. Through numerical simulations, we compare the signal recovery performance with 1bdyn.

1bdynS

In this algorithm, we assume the receiver has access to the noisy support-information of the sparse signal at the previous instant as SI. Let \mathcal{S}_n be a random variable that represents the support of the n -th index of the sparse signal, \mathcal{X}_{pn} , for $n = 1, \dots, N$ which can take values $+1$ and -1 according to

$$s_n = \begin{cases} +1, & \text{if } x_{pn} \neq 0, \\ -1, & \text{if } x_{pn} = 0, \end{cases} \quad (4.18)$$

Let $\widetilde{\mathcal{X}}_n$ be the n -th element of SI which is the noisy version of \mathcal{S}_n . We assume that a small fraction of the support set is different (erroneous) in the SI from that of x_p . We model this relationship between SI and the actual support-set of the signal by

$$\widetilde{\mathcal{X}}_n = \zeta_n \mathcal{S}_n,$$

where ζ_n is the multiplicative noise which can take values 1 or -1 , and is assumed to be a Bernoulli distributed random variable with probability β for event $\zeta_n = 1$ and probability $1 - \beta$

for event $\zeta_n = -1$, respectively.

At every iteration, the algorithm has access to $\widehat{\mathcal{R}}_n = \widehat{r}_n$, and $\widetilde{\mathcal{X}}_n = \widetilde{x}_n$ which are defined as

$$\widehat{\mathcal{R}}_n = \mathcal{X}_{cn} + \mathcal{V}_n, \quad \widetilde{\mathcal{X}}_n = \zeta_n \mathcal{S}_n, \quad (4.19)$$

For this setup, we develop a GAMP based signal recovery algorithm and refer to it as 1bdynS. The comparison of signal recovery performance of 1bdyn with 1bdynS gives us the gain in performance by using both support and amplitude information as SI.

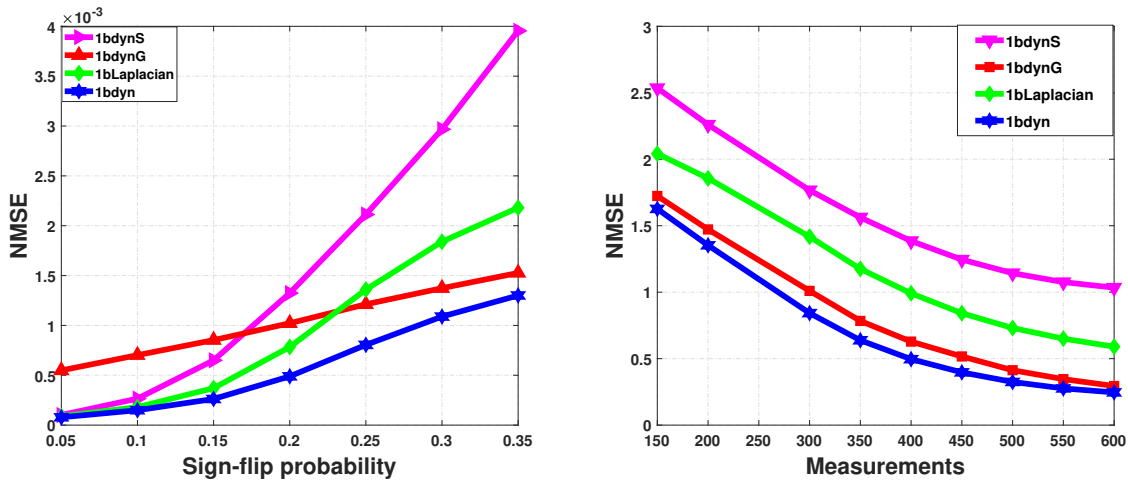


Fig. 4.2: Comparison of reconstruction performance of the proposed method with 1bdynG, 1bdyn and 1bLaplacian as a function of sign-flip probability and number of measurements, respectively when $N = 200$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.1$. In first plot $M = 200$, and in second plot $\gamma = 0.8$.

4.4 Simulation Results

In this section, we evaluate the performance of the proposed sparse signal reconstruction algorithm with the state-of-the-art algorithms. We consider the normalized mean square error

(NMSE) as the performance metric which is defined as

$$\text{NMSE} = \sqrt{\left\| \frac{\mathbf{x}}{\|\mathbf{x}\|_2} - \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} \right\|_2^2}$$

where \mathbf{x} , and $\hat{\mathbf{x}}$ are the actual signal and the reconstructed signal respectively. Consider a sparse signal of dimension $N = 200$. Unless specified otherwise, we consider $\rho = 0.9$ and $v_x = 5.0$. We generate a pair of signals \mathbf{x}_p and $\mathbf{x}_c \in \mathbb{R}^N$ following distribution (4.1) with parameters $\lambda_1 = 0.86, \lambda_2 = 0.02, \lambda_3 = 0.02$, and $\lambda_4 = 0.1$. The sparse signal is linearly transformed using the measurement matrix, $\mathbf{A} \in \mathbb{R}^{M \times N}$ which is corrupted by the noise vector assumed to be Gaussian distributed with zero mean and $v_n \mathbb{I}_M$ co-variance. The elements of \mathbf{A} are drawn from i.i.d. Gaussian distribution with zero-mean and unit variance. These noisy measurements are quantized to one-bit and transmitted through the noisy channel where the measurements are corrupted by sign-flip noise which is generated from Bernoulli distribution with the probability of sign flip $1 - \gamma$. We employ 500 Monte-Carlo runs to generate the results. In the first experiment, we evaluate the performance of the proposed one-bit CS algorithm and compare it with the state-of-the-art algorithms when the receiver has access to noisy SI. Figure 4.2 summarizes the results of the experiment where we plot the NMSE performance of the proposed algorithm, 1bdyn with 1bLaplacian, 1bitdynG and 1bitdynS. Note that 1bLaplacian does not consider the underlying temporal dependence between the previous time instant signal and the current time instant signal. 1bdynG uses the same dependence between the previous-instant signal and the current time-instant signal as in 1bitdyn. However, this algorithm assumes the SI at the receiver to be the previous time instant signal corrupted by the additive Gaussian noise. Similarly, 1bitdynS assumes the support of the previous time-instant signal as the SI and reconstructs the signal from the one-bit measurements. In Figure 4.2, we compare the NMSE performance of the proposed algorithm as a function of M , and $1 - \gamma$. From the plots of Figure 4.2, we see superior performance of the proposed 1bdyn algorithm compared to 1bdynG, 1bdynS, and 1bitdynS. The proposed method is expected to perform better than 1bLaplacian as

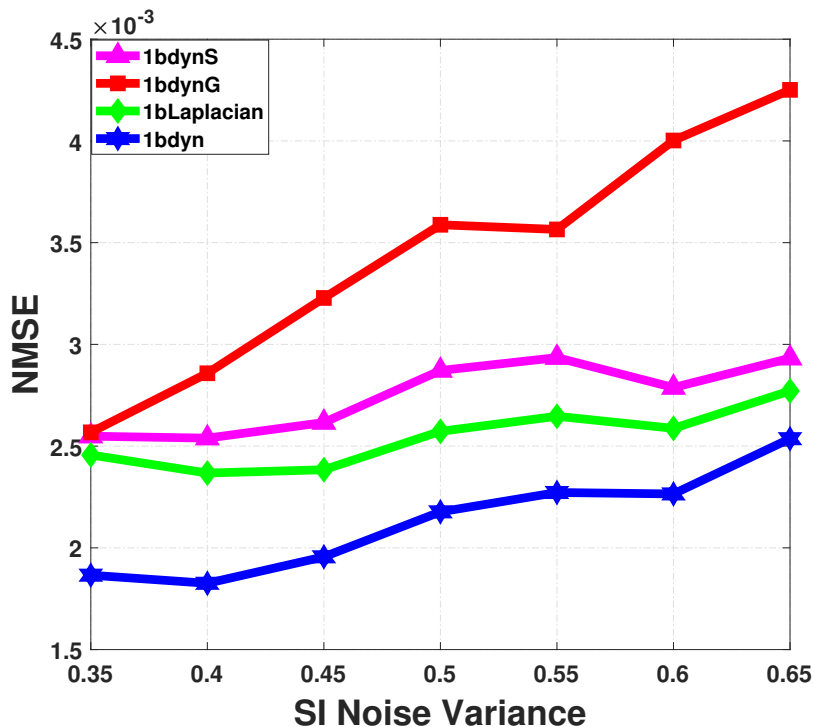


Fig. 4.3: Comparison of the reconstruction performance of the proposed method with 1bdynG, 1bdyn and 1bLaplacian as a function of noise-variance in SI when $N = 200$, $M = 200$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$

the proposed method incorporates the temporal dependence between the previous time-instant signal and the current time-instant signal while reconstructing the signals. 1bdynS does not perform as well as the 1bdyn because 1bdyn uses both the support and the amplitude information while 1bdynS uses only the support information as the SI. 1bdyn uses the Laplacian distribution to model the noise and is better than the Gaussian distribution when there is an error in the support of the SI compared to the actual support of the previous-time instant signal. So, the proposed method has better reconstruction performance than 1bdynG. Next, we study the effect of noise in SI in the reconstruction performance of the compressed signal. The SI at the receiver is assumed to be corrupted by additive i.i.d. Gaussian noise. We show the NMSE of the proposed algorithm compared with the state-of-the-art algorithms as a function of noise variance. In Figure 4.4, we summarize the results of the experiment. It is seen that the per-

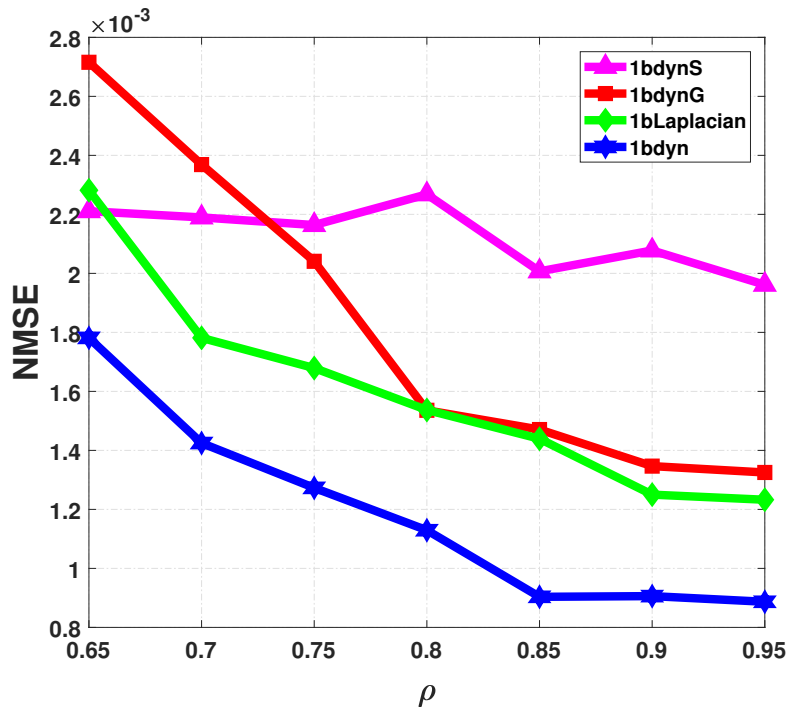


Fig. 4.4: NMSE performance as a function of ρ when $N = 200$, $M = 200$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$

formance of 1bdyn is better than all the other algorithms for all values of the noise variance. The proposed algorithm is more robust to noise in the SI. This result is particularly important because the SI at the receiver is the signal reconstructed from the previous time instant. Note that we need to estimate the signal using one-bit CS algorithms at the first time instant, which gives us noisy estimates. From the given results, we can see that even when the noise in the SI is large, the proposed algorithm is better at estimating the signal. Note that 1bdynG has the worst performance in the presence of noise in the SI.

In the third experiment, we compare the performance of the proposed algorithm with the other algorithms as a function of ρ . A higher value of ρ means that the correlation between the previous time-instant signal, and the current time-instant signals is higher. The result of the experiment is summarized in Figure 4.4. As expected, the performance of the proposed algorithms and all the other algorithms degrades for an increased value of ρ . Like in previous

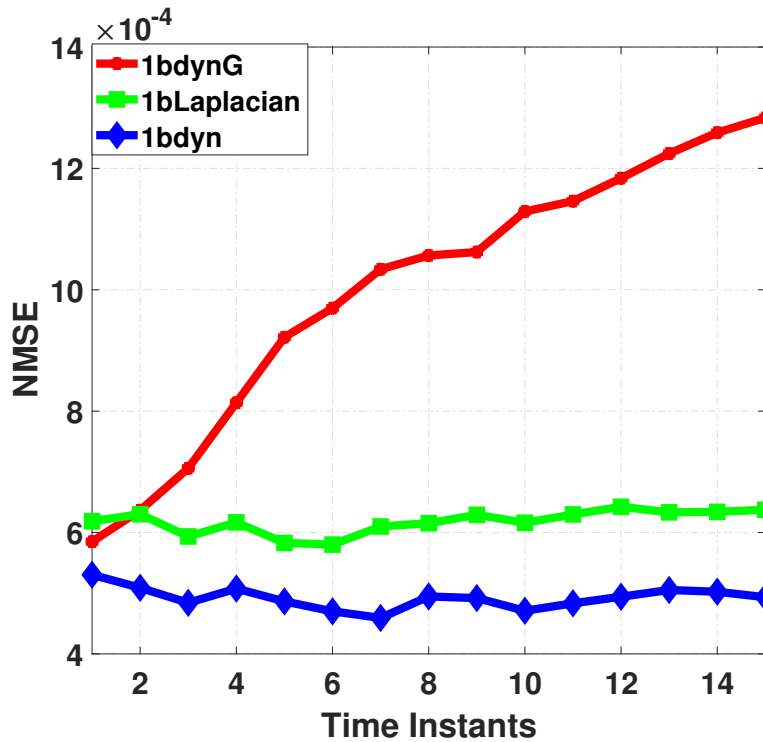


Fig. 4.5: NMSE performance as a function of time instants when $N = 200$, $M = 400$, $\rho = 0.9$, $v_x = 5.0$, $\sigma^2 = 1.0$, $\gamma = 0.9$, $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, and $\lambda_4 = 0.10$ of a sequence of sparse signal with slow time varying dynamics

experiments, the proposed algorithm performs better than all the algorithms considered. In the final experiment, we consider 50 different sequences of signals that follow the model described in (4.1). In the experiment, we consider $\lambda_1 = 0.86$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$ and $\lambda_4 = 0.10$. We fix $M = 400$, $N = 200$, $\rho = 0.90$, $\gamma = 0.90$, and $v_n = 0.2$. At the first time instant, we reconstruct the signal from the one-bit CS algorithm proposed in Chapter 3. At the second time instant, we consider this reconstructed signal as the SI to reconstruct the signal. In this experiment, we compare the performance of 1bdyn with 1bdynG and 1bLaplacian. For any other time instant t , all of these algorithms use the signal reconstructed at the previous time instant as SI to reconstruct the signal. For each realization of the sequence of sparse signals, we ran simulations for 500 iterations. We then averaged the results over 50 different sequences of sparse signals. We summarize the results of the experiment in Figure 4.5. We see that the proposed algorithm

has better NMSE performance than 1bdynG and 1bLaplacian. Similar to Figure 4.3, the noisy SI leads to poorer reconstruction performance for 1bdynG. So, the noise in reconstructed signal is accumulated over time instants and that lead to increase in reconstruction error with time instants. Figure 5 shows that the noise due to signal reconstruction does not accumulate over time for the proposed algorithm like 1bdynG and is better than 1bLaplacian.

4.5 Conclusion

In this chapter, we developed a new signal reconstruction algorithm from noisy one-bit compressed measurements using the GAMP framework exploiting the temporal dynamics of the signal and dependence between the non-zero elements. We used the Laplace distribution to model the noise in the side-information (SI) to account for the noise due to error in the amplitudes or error in both the amplitudes and the support in SI. We used Birth-Death-Drift (BDD) model to capture the dependence among consecutive time signals. We provided all closed-form expressions required to evaluate all the statistical quantities required for the GAMP algorithm. We showed that when time dynamics is taken into account, the algorithm performs better than when the time dynamics is ignored. Though the algorithm developed and results provided focused on the BDD signal model, the approach considered in this chapter is general and can be extended to any joint pdf of two consecutive time instant signals that are independent over indices.

CHAPTER 5

ONE-BIT CS WITH HETEROGENEOUS SI

5.1 Introduction

In the previous chapter, we implicitly assumed that the SI and the signal follow the same distribution and have joint sparse representation. This might not always be the case, especially when a system has measurements from multiple sensors of different modalities. In this chapter, we extend the results of the previous chapter to incorporate a statistically dependent signal from a different modality and possibly without joint sparse representation with the sparse signal as a SI to improve sparse signal reconstruction performance. We refer to this SI as a heterogeneous SI (HSI). We assume a general measurement model where compressed measurements are corrupted by additive white Gaussian noise before quantization and sign-flip errors after quantization. We propose a generalized approximate message passing-based algorithm for signal reconstruction from noisy one-bit compressed measurements, which leverages the dependence between the signal and the heterogeneous side-information. We model the dependence between signal and heterogeneous side-information using copula functions and show, through numerical experiments, that the proposed algorithm yields a better reconstruction performance than one-bit CS-based recovery algorithms that do not exploit the side-information..

5.1.1 Signal and Measurement Model

5.1.2 Signal Model

We consider the elements of the input signal $\mathbf{x} \in \mathbb{R}^n$ to be random and i.i.d. with joint pdf

$$p_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^N p(x_i), \quad (5.1)$$

where each component x_i is a Gaussian Mixture

$$p_{x_i} = (1 - \lambda)\mathcal{N}(x_i|0, v_1) + \lambda\mathcal{N}(x_i|0, v_2), \quad (5.2)$$

where λ is the probability of having non-zero values, and $v_1 \ll v_2$ with v_1 close to zero to impose sparse structure on the signal.

5.1.3 Copula Functions

Copula functions [75] allow the signals produced from different modalities to have arbitrary marginal distributions, while merging them into a joint multivariate probability distribution function.

Theorem 5.1 (Sklar's Theorem). *The joint distribution function F of random variables x_1, \dots, x_d with continuous marginal distribution functions F_1, \dots, F_d can be cast as*

$$F(x_1, x_2, \dots, x_d) = C(F_1(x_1), F_2(x_2), \dots, F_d(x_d)), \quad (5.3)$$

where C is a unique standard d -dimensional copula. Conversely, given a copula C and univariate Cumulative Distribution Functions (CDFs) F_1, \dots, F_d , F in (5.3) is a valid multivariate CDF with marginals F_1, \dots, F_d .

For absolutely continuous distributions F and F_1, \dots, F_d , the joint Probability Density

Function (PDF) of random variables x_1, \dots, x_d can be obtained by differentiating both sides of (5.3):

$$f(x_1, \dots, x_d) = \left(\prod_{m=1}^d f_m(x_m) \right) c(F_1(x_1), \dots, F_d(x_d)), \quad (5.4)$$

where f_1, \dots, f_d are the marginal densities and c is referred to as the density of standard multivariate copula C that is given by

$$c(\mathbf{u}) = \frac{\partial^L (C(u_1, \dots, u_d))}{\partial u_1, \dots, \partial u_d}, \quad (5.5)$$

where $u_m = F_m(x_m)$ and $\mathbf{u} = [u_1, \dots, u_d]$.

Thus, given specified univariate marginal distributions F_1, \dots, F_d and copula model C , the joint distribution function F can be constructed by

$$F(F_1^{-1}(u_1), F_2^{-1}(u_2), \dots, F_d^{-1}(u_d)) = C(u_1, u_2, \dots, u_d), \quad (5.6)$$

where $u_m = F_m(x_m)$ and $F_m^{-1}(u_m)$ are the inverse distribution functions of the marginals, $m = 1, 2, \dots, d$.

Note that $C(\cdot)$ is a valid CDF and $c(\cdot)$ is a valid PDF for uniformly distributed random variables u_m , $m = 1, 2, \dots, d$. Since the random variable u_m represents the CDF of x_m , the CDF of u_m naturally follows a uniform distribution over $[0, 1]$.

5.1.4 Measurement Model

Figure 5.1 shows the measurement model of noisy one-bit CS with pre- and post-quantization noise. The sparse signal $\mathbf{x} \in \mathbb{R}^N$ is compressed to a lower-dimensional vector $\mathbf{z} \in \mathbb{R}^M$ using the random measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. The vector, \mathbf{z} , is assumed to be corrupted by additive i.i.d. Gaussian noise vector, \mathbf{n} , with mean vector zero and covariance matrix $v_n \mathbf{I}_M$, where \mathbf{I}_M is the $M \times M$ identity matrix. This corrupted compressed vector is quantized

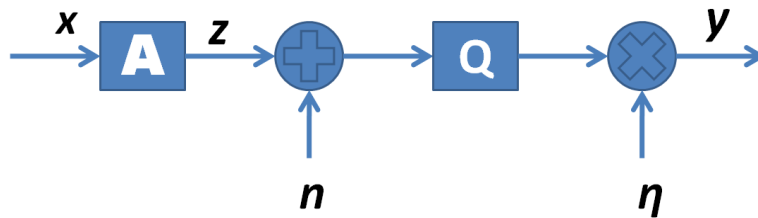


Fig. 5.1: one-bit CS with pre-quantization and post-quantization noise.

element-wise to $+1$ or -1 based on the sign of the signal.

We assume that the channel between the one-bit quantizer and the receiver is noisy. The quantized measurements are corrupted by sign-flip noise. Formally, the noisy one-bit quantized measurement model can be written as

$$y_j = \eta_j Q(\mathbf{A}_j^T \mathbf{x} + n_j), \quad j = 1, \dots, M \quad (5.7)$$

where \mathbf{A}_j^T is the j -th row of \mathbf{A} , n_j is the j -th element of \mathbf{n} , the quantizer $Q : \mathbb{R} \rightarrow \{-1, +1\}$ is the sign quantizer

$$[Q(\zeta)]_j = \begin{cases} +1, & \text{if } \zeta_j > 0, \\ -1, & \text{if } \zeta_j \leq 0, \end{cases} \quad (5.8)$$

and $\eta_j \in \{-1, +1\}$ is the post-quantization noise. η_j is assumed to follow i.i.d. Bernoulli distribution with $Pr(\eta_j = 1) = \gamma$. We define the inverse of the quantization function, $Q^{-1}(\cdot)$, as

$$[Q^{-1}(\mathbf{y})]_j = \begin{cases} (-\infty, 0], & \text{if } y_j \leq 0, \\ (0, \infty), & \text{if } y_j > 0, \end{cases} \quad (5.9)$$

where y_j is the j -th element of \mathbf{y} .

5.1.5 Bayesian Formulation

We assume that HSI, $\tilde{\mathbf{x}}$, is available at the receiver. HSI is assumed to be dissimilar to but statistically dependent with the sparse signal. Using Bayesian rule, $p(\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}) \propto p(\mathbf{x})p(\tilde{\mathbf{x}}, \mathbf{y}|\mathbf{x}) = p(\mathbf{x})p(\tilde{\mathbf{x}}|\mathbf{x})p(\mathbf{y}|\mathbf{x})$, the posterior distribution of \mathbf{x} given the noisy one-bit quantized measurements, \mathbf{y} , and HSI, $\tilde{\mathbf{x}}$, at the receiver is

$$p(\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}, \tilde{\mathbf{x}}) \propto \prod_{j=1}^M \mathbb{I}_{z_j \in \{Q^{-1}(\eta_j y_j)\}} \prod_{i=1}^N p(x_i, \tilde{x}_i), \quad (5.10)$$

where $\mathbb{I}_{(\cdot)}$ represents the indicator function, and \propto represents equality up to a proportional constant, and $p(x_i, \tilde{x}_i)$ is the joint density function that gives the joint statistical characterization of the sparse signal and the HSI. The minimum mean square error (MMSE) estimator of \mathbf{x} is the mean of the posterior distribution (5.10), i.e., $\mathbb{E}[p(\mathbf{x}|\mathbf{y}, \tilde{\mathbf{x}})]$. As the evaluation of the MMSE estimator of (5.10) is intractable, we develop a message passing-based algorithm to approximate the MMSE estimator.

5.2 GAMP Algorithm Update Equations

In this section, we develop a GAMP-based algorithm which reconstructs the sparse signal from its noisy one-bit compressed measurements when the receiver has access to HSI.

5.2.1 Heterogeneous Side-Information (HSI)

HSI is a signal which is assumed to be of different modality than the compressed signal and hence may not share joint-sparse representation as considered in the literature previously [69, 72, 93]. So, we cannot impose the Laplace distribution between HSI and the signal during reconstruction as in the Algorithm 3.5. Instead, we assume that the HSI is dependent with the sparse signal, and use the Copula function [75] to model the statistical dependence between the

signal and the HSI.

Algorithm 5.1 GAMP Algorithm for noisy one-bit CS with heterogeneous SI (HSI-GAMP)

1. Initialization: Set $t=0$ and initialize $\hat{\mathbf{x}}^t, \tau_x^t$, and $\hat{\mathbf{s}}^t$ as $\hat{\mathbf{x}}^t = \mathbb{E}[\mathbf{x}]$, $\tau_x^t = \text{var}[\mathbf{x}]$, $\hat{\mathbf{s}}^t = 0$, where the expectation and variance of \mathbf{x} are with respect to p_x .

2. **Measurement Update**

- Linear Step

$$\tau_p^{t+1} = (\mathbf{A} \odot \mathbf{A}) \tau_x^t, \quad \hat{\mathbf{p}}^{t+1} = \mathbf{A} \hat{\mathbf{x}}^t - \tau_p^{t+1} \odot \hat{\mathbf{s}}^t,$$

- Non-Linear Step

$$\hat{\mathbf{s}}^{t+1} = F_1(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau_p^{t+1}), \quad (5.11)$$

$$\tau_s^{t+1} = F_2(\mathbf{y}, \hat{\mathbf{p}}^{t+1}, \tau_p^{t+1}), \quad (5.12)$$

where F_1 and F_2 are applied element-wise and are defined as

$$F_1(y, \hat{p}, \tau_p) = \frac{1}{\tau_p} \left(\mathbb{E}[z|y] - \hat{p} \right), \quad (5.13)$$

$$F_2(y, \hat{p}, \tau_p) = \frac{1}{\tau_p} \left(1 - \frac{\text{var}[z|y]}{\tau_p} \right).$$

The expectation and variance are evaluated with respect to $z \sim \mathcal{N}(\hat{p}, \tau_p)$.

3. **Estimation Update**

- Linear Step

$$\tau_r^{t+1} = ((\mathbf{A} \odot \mathbf{A})^T \tau_s^t)^{-1}, \quad \hat{\mathbf{r}}^{t+1} = \hat{\mathbf{x}}^t + \tau_r^{t+1} \odot (\mathbf{A}^T \hat{\mathbf{s}}^{t+1}),$$

where the inversion is performed element-wise

- Non-linear Step

$$\hat{\mathbf{x}}^{t+1} = G_1(\hat{\mathbf{r}}^{t+1}, \tau_r^{t+1}), \quad (5.14)$$

$$\tau_x^{t+1} = G_2(\hat{\mathbf{r}}^{t+1}, \tau_r^{t+1}), \quad (5.15)$$

where G_1 and G_2 are applied element-wise and are defined as

$$G_1(\hat{r}, \tau_r) = \mathbb{E}[x|\hat{r}, \tilde{x}], \quad G_2(\hat{r}, \tau_r) = \text{var}[x|\hat{r}, \tilde{x}]. \quad (5.16)$$

The expectation and variance are evaluated with respect to $p_{x|\hat{r}} \propto \mathcal{N}(\cdot; \hat{r}, \tau_r) p_x(\cdot) p_{\tilde{x}|\hat{r}}(\cdot)$.

Set $t = t+1$ and return to step 2.

5.2.2 Copula Functions

Let $F_{\mathcal{X}_i}$ and $F_{\tilde{\mathcal{X}}_i}$ represent the marginal cumulative distribution of the i -th signal and HSI coefficients, respectively. Using copula function, the joint probability density function of the signal and its HSI is

$$p(x_i, \tilde{x}_i) = c[F_{\mathcal{X}_i}, F_{\tilde{\mathcal{X}}_i}] p(x_i) p(\tilde{x}_i), \quad (5.17)$$

where $c[F_{\mathcal{X}_i}, F_{\tilde{\mathcal{X}}_i}]$ represents the bivariate copula density function. There are several copula densities such as Gaussian copula, Clayton, and Frank copula which represent different correlation structure between random variables. Among several copula functions, the one that captures the dependencies between the signal and the HSI should be selected [75]. For simplicity of exposition, we assume that the Gaussian copula models the dependence between the signal and the HSI. The distribution of Gaussian copula is defined as

$$c_g(\mathbf{w}) = |\mathbf{R}_g|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} \mathbf{w}_i^T (\mathbf{R}_g^{-1} - \mathbf{I}) \mathbf{w}_i \right], \quad (5.18)$$

where $\mathbf{w}_i = [\Phi^{-1}(F_{\mathcal{X}_i}), \Phi^{-1}(F_{\tilde{\mathcal{X}}_i})]^T$, \mathbf{I} is the identity matrix, and $\mathbf{R}_g = [1 \ \rho; \ \rho \ 1]$ is the Gaussian copula parameter.

We assume that the receiver knows the copula function that captures the dependence between the signal and the HSI.

5.2.3 Update Equations

In this subsection, we develop a GAMP based algorithm that reconstructs the sparse signal from noisy one-bit compressed measurements when the dependence between the signal and the HSI is modeled by copula functions. In Algorithm 5.1, we summarize the steps required for the estimation of the sparse signal and refer to it as HSI-GAMP. The algorithm requires the evaluation of F_1, F_2, G_1 , and G_2 as defined in (5.13), and (5.16). For F_1 and F_2 , we require the evaluation of $\mathbb{E}[z|y]$ and $\text{var}[z|y]$. Following the steps as in Chapter 3, we obtain the following

expressions for $\mathbb{E}[z|y]$ and $\mathbb{E}[z^2|y]$

$$\begin{aligned}\mathbb{E}[z|y] &= \frac{1}{C_1} \left[\gamma \left(PI_1(v_n, \hat{p}, \tau_p) \delta(y-1) + (\hat{p} - PI_1(v_n, \hat{p}, \tau_p)) \delta(y+1) \right) \right. \\ &\quad \left. + (1-\gamma) \left((\hat{p} - PI_1(v_n, \hat{p}, \tau_p)) \delta(y-1) + PI_1(v_n, \hat{p}, \tau_p) \delta(y+1) \right) \right], \\ \mathbb{E}[z^2|y] &= \frac{1}{C_1} \left[\gamma \left(PI_2(v_n, \hat{p}, \tau_p) \delta(y-1) + (\hat{p}^2 + \tau_p - PI_2(v_n, \hat{p}, \tau_p)) \delta(y+1) \right) \right. \\ &\quad \left. + (1-\gamma) \left((\hat{p}^2 + \tau_p - PI_2(v_n, \hat{p}, \tau_p)) \delta(y-1) + PI_2(v_n, \hat{p}, \tau_p) \delta(y+1) \right) \right], \\ C_1 &= \gamma \left(PI_0(v_n, \hat{p}, \tau_p) \delta(y-1) + \left(1 - PI_0(v_n, \hat{p}, \tau_p) \right) \delta(y+1) \right) \\ &\quad + (1-\gamma) \left(\left(\hat{p} - PI_0(v_n, \hat{p}, \tau_p) \right) \delta(y-1) + PI_0(v_n, \hat{p}, \tau_p) \delta(y+1) \right).\end{aligned}$$

As $\text{var}[z|y] = \mathbb{E}[z^2|y] - (\mathbb{E}[z|y])^2$, F_1 and F_2 can now be evaluated. Next, we evaluate non-linear function G_1 and G_2 required in the estimation update in Algorithm 5.1.

$$\mathbb{E}[x|\hat{r}, \tilde{x}] = \frac{1}{C_2} \int x \mathcal{N}(x|\hat{r}, \tau_r) c[F_{X_i}, F_{\tilde{X}_i}] p(x_i) dx, \quad (5.19)$$

where C_2 is a normalization constant and is given by

$$C_2 = \int \mathcal{N}(x|\hat{r}, \tau_r) c[F_{X_i}, F_{\tilde{X}_i}] p(x_i) dx. \quad (5.20)$$

Similarly,

$$\mathbb{E}[x^2|\hat{r}, \tilde{x}] = \frac{1}{C_2} \int x^2 \mathcal{N}(x|\hat{r}, \tau_r) c[F_{X_i}, F_{\tilde{X}_i}] p(x_i) dx. \quad (5.21)$$

From Equations (5.19) and (5.21), we can clearly see that the method considered is general and be used for any bivariate coupla function. The evaluation of the closed-form expressions for (5.19), and (5.21) is possible if the copula density is Gaussian and the signal \mathbf{x} follows Gaussian distribution. In other cases, we may need to resort to numerical integrations. Note that, the evaluation of the mean and the variance in (5.19) and (5.21), respectively requires unidimen-

sional integrations and hence are computationally feasible. Note that, in both of the Measurement Update and the Estimation Update sections of Algorithm 5.1, per iteration computation is dominated by matrix multiplication. Hence, the per-iteration computation complexity of the algorithm is $O(MN)$.

5.3 Simulation Results

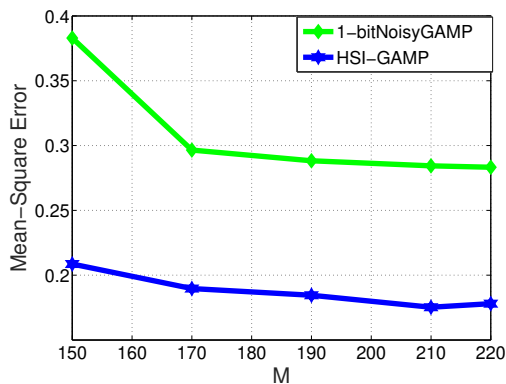


Fig. 5.2: Reconstruction performance of the proposed method when $\rho = 0.7, \gamma = 0.10, v_n = 0.1, \lambda = 0.1$

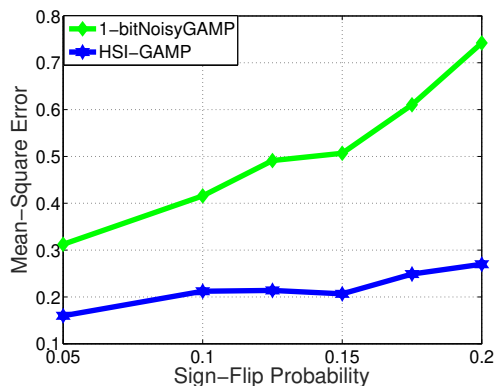


Fig. 5.3: Reconstruction performance of the proposed method when $M/N = 2, v_n = 0.1, \lambda = 0.1, \rho = 0.7$

We consider the problem of reconstructing a sparse signal of dimension N from M noisy one-bit compressed measurements. The measurement matrix, \mathbf{A} , is drawn from an i.i.d. Gaussian distribution with zero-mean and $\frac{1}{M}$ variance. We assume that the side-information at the receiver is heterogeneous, which may or may not be sparse. We use the Gaussian copula, with copula parameter \mathbf{R}_g , to generate the sparse signal and the HSI. The signal, \mathbf{x} , follows the Gaussian mixture distribution in (5.2) with $\lambda = 0.1, v_1 = 0.1, v_2 = 5$, and $N = 100$. We perform 100 Monte Carlo runs and compare the mean square errors (MSE) of HSI-GAMP with one-bit noisyGAMP proposed in Chapter 3. In several experiments, we study the reconstruction performance of the proposed algorithm. In the first experiment, we study the reconstruction performance of the proposed algorithm as a function of M . In Figure 5.2, we

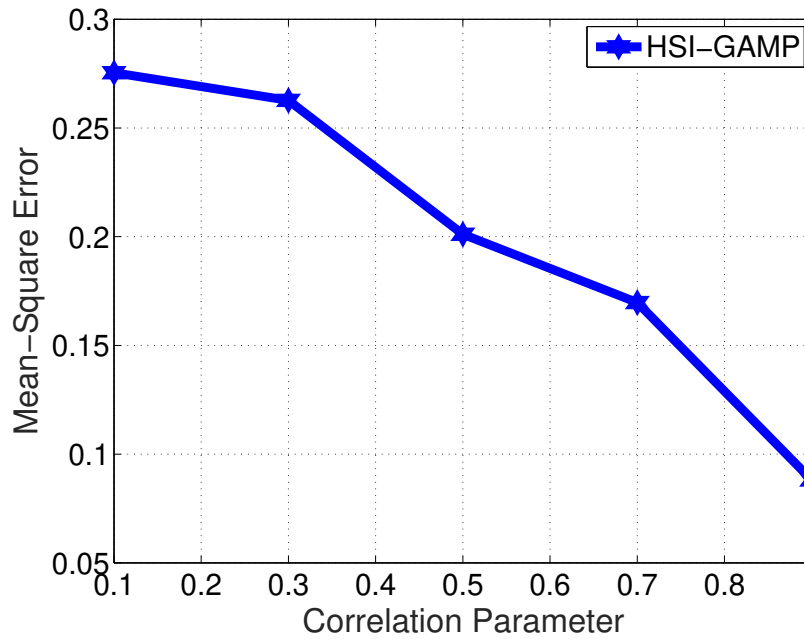


Fig. 5.4: Reconstruction performance of the proposed method. $M/N = 2$, $v_n = 0.1$, $\lambda = .1$, $\gamma = 0.1$

plot the MSE values of the HSI-GAMP algorithm and the 1-bitNoisyGAMP against M . In the second experiment, we study the effect of the sign-flip noise on the reconstruction performance of the proposed algorithm. Figure 5.3 shows the results of the second experiment. In the third experiment, we study the effect of the correlation parameter of the Gaussian copula, ρ , on the reconstruction performance of HSI-GAMP. Figure 5.3 shows the results of the third experiment. From Figure 5.2, we can see that incorporating HSI at the receiver yields improved reconstruction performance when compared to 1-bitnoisyGAMP. From Figure 5.3, we can see that the reconstruction performance of the proposed algorithm is more robust to the sign-flip noise when compared to 1-bitNoisyGAMP. HSI-GAMP exploits the dependence between the signal and the HSI to get better MSE performance. From Figure 5.4, we can see significant improvement in the reconstruction performance when the dependence between the signal and the HSI is large. When the dependence is large, the receiver has more information regarding the sparse signal through its statistical characterization which leads to improved reconstruction performance. In the third experiment, we obtain the average MSE value of 0.292

for 1-bitNoisyGAMP and is always larger than HSI-GAMP.

5.4 Conclusion

In this chapter, we proposed an algorithm for sparse signal recovery from noisy one-bit measurements when the receiver has access to heterogeneous side-information. We used copula functions to capture the dependence between the side-information and the signal. We showed that by taking the heterogeneous side-information into account during signal reconstruction yields improved performance. We translate the problem of evaluating a very high dimensional integration into several signal dimension integration which is computationally very efficient. We showed that the proposed algorithm is robust to sign-flip noise and can significantly reduce the reconstruction error when the signal and the HSI are highly dependent.

CHAPTER 6

ONE-BIT COMPRESSED SENSING USING NETWORK PRIOR

6.1 Introduction

In this chapter, we address the classical problem of one-bit compressed sensing using deep learning-based (DL) approaches. We consider two different setups. We present a deep learning-based reconstruction algorithm that relies on a trained generative model in the first setup. The generator, a deep neural network, learns a mapping from a low dimensional space to a higher dimensional set comprising sparse vectors. Generative models such as generative adversarial networks (GANs) [38] and variational autoencoders (VAEs) [56] are trained on data to learn the data distribution. The sparse vectors are reconstructed from their one-bit measurements by searching over the range of this trained generator. A well-trained generative network learns the distribution of the signal, which includes all possible structural information in the signal, including sparsity. Because of the use of this trained network, the proposed algorithm provides excellent reconstruction performance compared to the traditional methods that account for just sparsity. Importantly, we provide theoretical guarantees on the reconstruction accuracy and

sample complexity of the presented algorithm. In the second setup, we consider an untrained neural network as a prior on the signal. The use of an untrained network-based approach makes sense when it is not possible to train a generative model before reconstruction. We choose the architecture of the untrained network such that it acts as a good prior on natural signals such as images. We provide a reconstruction algorithm to estimate signals from the one-bit measurements using an untrained network as the prior. In both setups, numerical results demonstrate the efficacy of our algorithm compared to other existing algorithms. We also show that, in contrast to existing algorithms, the proposed algorithms can recover both the amplitude and the direction of the signal from one-bit measurements.

6.2 System Model and Algorithm

We consider the problem of recovering an unknown sparse vector $\mathbf{x}^* \in \mathbb{R}^N$ from a set of one-bit measurements $\mathbf{y} \in \{\pm 1\}^M$. The i^{th} measurement y_i is modeled as follows:

$$\mathbf{y}_i = \text{sign}(\langle \mathbf{A}_i, \mathbf{x}^* \rangle) \in \{\pm 1\}, i = 1, 2, \dots, M \quad (6.1)$$

where $\mathbf{A}_i \in \mathbb{R}^N$ is the i^{th} row of the known measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. Our goal is to find a reconstruction $\hat{\mathbf{x}}$ such that it is close to the ground truth \mathbf{x}^* .

We compute \mathbf{x}^* from one-bit measurements \mathbf{y} by optimizing a loss function $\mathcal{L}(\mathbf{A}\mathbf{x}; \mathbf{y})$, between \mathbf{y} and the corresponding linear measurements $\mathbf{A}\mathbf{x}$. Then, the corresponding optimization problem is given by

$$\min_{\mathbf{x} \in \mathcal{S}} \mathcal{L}(\mathbf{A}\mathbf{x}, \mathbf{y}), \quad (6.2)$$

where $\mathcal{S} \subseteq \mathbb{R}^N$ represents the solution space of the signal. The design specifications of the optimization problem in (6.2) include two parts: one, an accurate representation of the solution space, and two, the form of the loss function $\mathcal{L}(\cdot)$. The same structure can be observed in a

traditional one-bit CS problem as well. For example

$$\begin{aligned} \hat{\mathbf{x}} = \min_{\mathbf{x}} \quad & -\mathbf{y}^\top \mathbf{A}\mathbf{x} \\ \text{subject to} \quad & \|\mathbf{x}\|_1 \leq \sqrt{K} \text{ and } \|\mathbf{x}\| \leq 1 \end{aligned}$$

In this problem, the loss function when minimized enforces consistency between \mathbf{y} and $\mathbf{A}\mathbf{x}$ which lies on the space $\mathcal{S} = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^N, \|\mathbf{x}\|_0 \leq K, \text{ and } \|\mathbf{x}\| = 1\}$. The K -sparse structure of the solution set ($\|\mathbf{x}\|_0 \leq K$) is imposed through inequality involving ℓ_1 norm which is obtained as

$$|\tilde{\mathbf{1}}^\top \mathbf{x}| \leq \|\tilde{\mathbf{1}}\| \|\mathbf{x}\| \leq \sqrt{K},$$

where $\tilde{\mathbf{1}} \in \mathbb{R}^N$ is a vector with all ones at indices where \mathbf{x} is non-zero and zeroes otherwise. The first inequality is due to the Cauchy-Schwartz inequality while the second is due to the K -sparsity. Next, we introduce two ideas of compressed sensing from one-bit measurements using a neural network as a prior. In the following sections, we provide the details of the solution space, and the optimization problem of the two proposed algorithms and the detailed empirical results to demonstrate the superior performance.

6.3 Compressed Sensing using Trained Network as a Prior

In this section, we provide the details of the solution space, optimization problem, theoretical results of the proposed algorithm when a trained generative model is used as the prior.

6.3.1 Solution Space

In this approach, we assume that the set \mathcal{S} represents the range space of a trained neural network $G(\mathbf{w}; \mathbf{g})$. Here, \mathbf{w} denotes the weights (parameters) of the network, and $\mathbf{g} \in \mathbb{R}^k$ is the latent

(input) vector. Thus, any sparse vector \mathbf{x} can be represented as $\mathbf{x} = G(\mathbf{w}; \mathbf{g})$ for some $\mathbf{g} \in \mathbb{R}^k$ and \mathbf{w} . In general, to solve (6.2), we need to estimate \mathbf{g} , or \mathbf{w} , or both. For a trained network, \mathbf{w} is fixed as it is learned during the training process. So the problem of estimating $\hat{\mathbf{x}}$ translates to evaluating an estimate $\hat{\mathbf{g}}$. Therefore, the optimization problem in (6.2) can be rewritten as

$$\min_{\mathbf{g}} \mathcal{L}(\mathbf{A}G(\mathbf{w}; \mathbf{g}), \mathbf{y}). \quad (6.3)$$

Our approach is to use a trained generative model such as VAEs and GANs trained on some datasets which maps input drawn from a distribution p_G over \mathbb{R}^k to desired signal by a deterministic function: $G : \mathbb{R}^k \rightarrow \mathbb{R}^n$. Typically, $k \ll N$, i.e., G is a mapping from a low dimensional representation space ($\subset \mathbb{R}^k$) to a high dimensional sample space ($\subset \mathbb{R}^N$), learned by the model. In the training phase, the algorithm learns the function G that can map the distribution p_G to the distribution of the data from training samples. To be specific, we train the generative model using sparse vectors so that the range of the generator mapping is close to the desired set of sparse vectors. As \mathbf{w} is fixed for trained network, we represent a trained network by $G(\mathbf{g})$.

During signal reconstruction, we minimize the following objective function:

$$\text{loss}(\mathbf{g}) = \|G(\mathbf{g})\|_2^2 - \frac{\sqrt{2\pi}}{M} \mathbf{y}^\top \mathbf{A}G(\mathbf{g}). \quad (6.4)$$

The second term of the objective function maximizes the correlation between the one-bit measurements \mathbf{y} and the corresponding linear measurements. For a fixed l_2 norm of $G(\mathbf{g})$, the term is maximized when $\text{sign}(\mathbf{A}G(\mathbf{g})) = \mathbf{y}$. Therefore, the second term ensures the match between $\mathbf{A}G(\mathbf{g})$ and \mathbf{y} . However, the second term decreases as the l_2 norm of $G(\mathbf{g})$ increases, and therefore, we use the first term to control the norm. Hence, the two terms of the objective function jointly optimize the representation error.

Any optimization procedure can be used to minimize the loss function, and if the generative

model is differentiable, we can use the standard back-propagation learning. Let $\hat{\mathbf{g}}$ denotes the optimization procedure output. Our reconstructed signal is given by $\hat{\mathbf{x}} = G(\hat{\mathbf{g}})$. We note that although the objective function to be optimized is non-convex, the optimization problem can be solved using gradient descent. Next, we discuss some theoretical guarantees for the above algorithm, assuming that the gradient descent finds a good approximate solution to the above non-convex optimization problem.

6.3.2 Theoretical Analysis

Let G be a d -layer neural network with at most n nodes per layer, all weights are upper bounded by w_{\max} in absolute value, and the non-linearity after each layer is L -Lipschitz. Further, let the range of the generator be denoted by \mathcal{S} . The main result of this section is as follows:

Theorem 6.1. *Let the input to the model G have independent entries drawn from a uniform distribution over $\left[-\frac{r}{\sqrt{k}}, \frac{r}{\sqrt{k}}\right]$ during the training phase, and $\mathbf{A} \in \mathbb{R}^{M \times N}$ be a random Gaussian matrix, scaled so that $\mathbf{A}_{i,j} \sim \mathcal{N}(0, 1/M)$. Fix \mathbf{x}^* satisfying $\|\mathbf{x}^*\|_2 = 1$. Assume that the measurement vector \mathbf{y} follows the model given by (6.1). Suppose $\hat{\mathbf{g}}$ minimizes the cost function in (6.11) to within additive δ of the optimum over the vectors with $\|\mathbf{g}\|_2 \leq r$. Then, for any $\epsilon > 0$ there exists universal constants $C, c > 0$ such that if*

$$M \geq C\epsilon^{-2}k(r^2 + d \log Lnw_{\max}), \quad (6.5)$$

the following holds with probability at least $1 - 4 \exp(-c\epsilon^2 m)$

$$\|G(\hat{\mathbf{g}}) - \mathbf{x}^*\|_2^2 \leq \min_{\substack{\mathbf{g} \in \mathbb{R}^k \\ \|\mathbf{g}\| \leq r}} \|G(\mathbf{g}) - \mathbf{x}^*\|_2 + \delta + \epsilon. \quad (6.6)$$

The details of the proof is provided in Appendix G.

Discussion

Optimality of the error bound: The first term of the error bound is also known as representation error. It arises because we search over the range of the generator to find the unknown sparse vector. The second term δ accounts for the fact that the gradient descent does not necessarily converge to the global optimum. Empirically, we see that these error terms converge to zero. These error terms are the minimum possible error terms for the presented technique, and therefore, they seem to be optimal for our algorithm.

Dependence on r : We see that as r increases, the number of measurements increases and the estimation error decreases. This is intuitive as r increases, the domain of the generator expands and thus, we need more measurements to train the generator. Further, as the domain expands, the range also becomes larger which results in improved accuracy.

Dependence on network parameters: As the network parameters, n , L and w_{\max} increases, the number of measurements increases. This is because, as these parameters increase, the number of unknowns to be learned by the network or their respective ranges increase, and therefore, more number of measurements are required. On the other hand, this also increases the flexibility of network and thus, the range of $G(\mathbf{g})$ can become larger, and therefore, the first term in the error bound decreases. Hence, an increase in the number of measurements results in an improved error bound, as expected.

Distribution of input: Although the statement of the theorem specifies the distribution of the input to the generator as a uniform distribution parametrized by r , the proof only requires to assume that the norm of the input is bounded. Therefore, the result applies to an input \mathbf{g} drawn from independent Gaussian or any other heavy-tailed distributions as this only prunes an exponentially unlikely fraction of the possible outputs.

6.3.3 Simulation Results

In this section, we evaluate the signal reconstruction performance of the proposed algorithm. Let \mathbf{x}^* and $\hat{\mathbf{x}}$ represent the true signal and the estimated signal, respectively. We evaluate the recovery performance using following two metrics:

- Mean Square Error (MSE):

$$\text{MSE} \triangleq \|\mathbf{x}^* - \hat{\mathbf{x}}\|^2 \quad (6.7)$$

- Normalized Mean Square Error (NMSE):

$$\text{NMSE} \triangleq \left\| \frac{\mathbf{x}^*}{\|\mathbf{x}^*\|} - \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|} \right\|^2 \quad (6.8)$$

We consider the MNIST handwritten digit dataset [59], the Fashion MNIST dataset [96], and the Omniglot dataset [58] for training the generative models. We chose these image datasets because they are sparse in pixel intensities and hence, we can compare the performance with traditional compressed sensing algorithms. We compare the performance of our algorithm with two traditional one-bit CS algorithms: convex optimization-based algorithm in (labeled as YP) [78], and binary iterative hard thresholding (BIHT) [48] algorithm. The experimental setup is as follows:

Generative model: We adopt the setup from [11] and train variational autoencoders (VAEs) [56] as the generative model using training images from the MNIST, Fashion-MNIST, and Omniglot datasets. The image size is 28×28 , and thus, the input dimension is $N = 784$. We choose the size of the input to the generator as $k = 40$. The generator is a fully connected 40-500-500-746 neural network, and the encoder is also a fully connected 784-500-500-40 neural network. For each of the datasets, VAE is trained for 200 epochs with a mini-batch size of 64 using the Adam optimizer [54] with a learning rate of 0.001.

Measurement model: The columns of measurement matrix \mathbf{A} are drawn uniformly from the surface of the m -dimensional unit hypersphere [73]. We use noisy one-bit compressed measurements

$$y_i = \eta_i \text{sign}(\langle \mathbf{A}_i \mathbf{x}^* + n_i \rangle) \in \{\pm 1\}, i = 1, 2, \dots, M, \quad (6.9)$$

where n_i is the additive Gaussian noise with mean zero and variance v_n . Similarly, η_i is Rademacher distributed random variable that takes value 1 and -1 with probability α and $1 - \alpha$, respectively.

We use 10 different images from the testing set and generate one-bit compressed measurements. As the optimization problem we consider is a non-convex problem, we do 10 random restarts with 100 gradient descent steps per restart for signal reconstruction and report the result with the least error. For all the experiments where noisy one-bit measurements are used, we average the results over 50 Monte Carlo runs for each of ten images.

Noiseless Setting

In Figures 6.1 and 6.2, we provide the recovery performance of the proposed algorithm (labeled as GenModel), YP [78] and BIHT [47] in a noiseless scenarios, i.e., when $v_n = 0$ and $\alpha = 1$. The performance of all the algorithms improves with the number of measurements M , as more information about the sparse vector is available. However, when we look at MSE performance and NMSE performance, we can see two different performance traits of the proposed algorithm.

MSE Performance: The MSE performance of the proposed algorithm is less by one order of magnitude for same M . Generative models, such as VAEs and GANs, when pre-trained on a dataset, learn the distribution of the compressed signal. Hence, a well-trained generative model has information on both the magnitude and direction of the

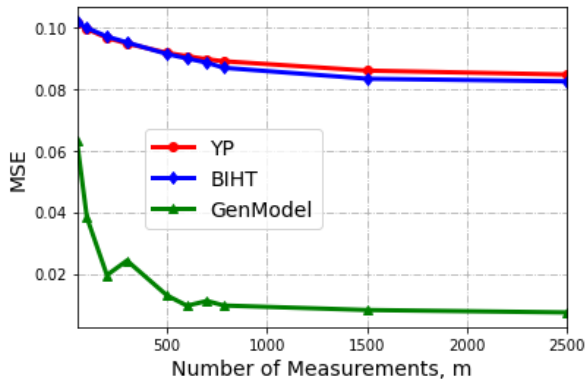


Fig. 6.1: MSE of the proposed algorithm compared with BIHT and YP as a function of number of measurements M in a noiseless setting.

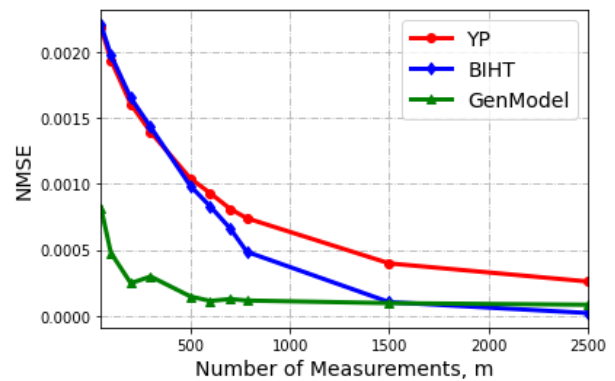


Fig. 6.2: NMSE of the proposed algorithm compared with BIHT and YP as a function of number of measurements M in a noiseless setting.

compressed signal. The proposed algorithm uses this well-trained generative model, and recovers both the magnitude and the direction of the compressed signal. As BIHT and YP estimate signals on the unit ball, the proposed method have superior MSE performance, especially when the sparse signal does not lie on the unit ball.

NMSE Performance: The NMSE performance of the proposed algorithm depends on the value of M and can be divided into following two regions in Figure 6.2:

$M < 1500$: In this regime, the proposed algorithm has better reconstruction performance. Usually, the range space of a well-trained generative model has more structural information on signal than just sparsity. Further, the range space is smaller subspace than entire signal space, i.e., \mathbb{R}^N . Hence, the proposed algorithm evaluates a better estimate of the direction of the compressed signal than traditional algorithm with a smaller number of one-bit measurements (M).

$M \geq 1500$: When the number of measurements is sufficiently large, the traditional algorithms either match or better the performance of the proposed algorithm. In Figure 6.2, BIHT algorithm outperforms the proposed algorithm when $M \geq 1500$. It can be observed that the NMSE value of the proposed algorithm stagnates after

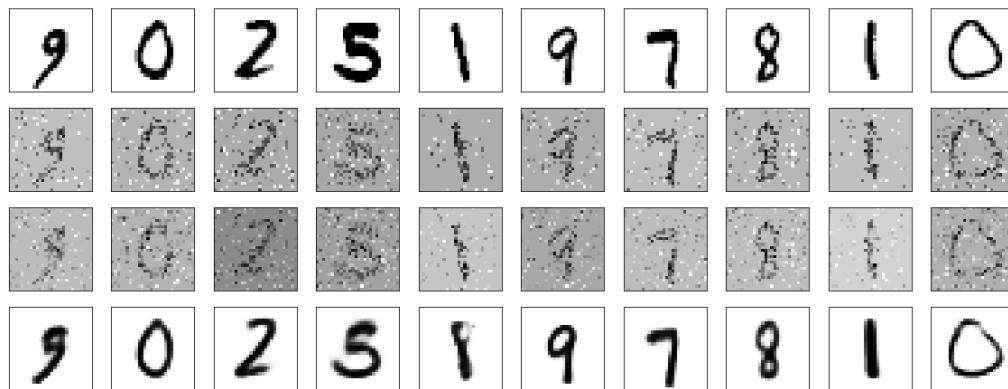


Fig. 6.3: The first rows show the original images, the second, third and fourth rows are the reconstruction images using BIHT, YP and proposed algorithms, respectively when $m = 784$ in a noiseless setting for MNIST dataset.

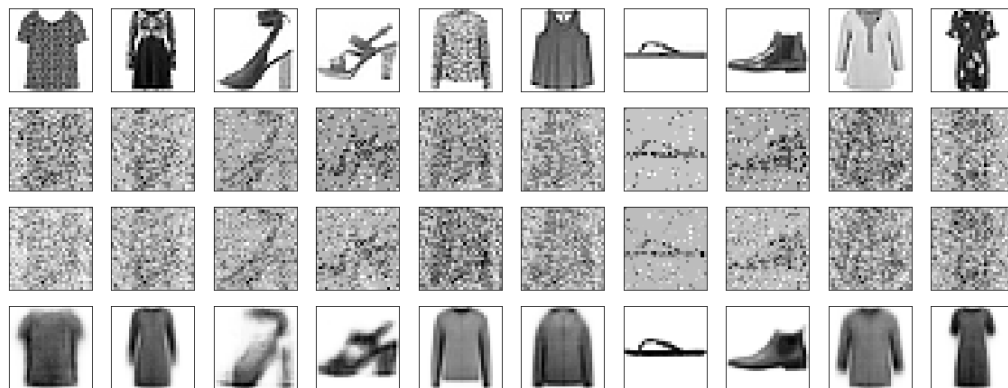


Fig. 6.4: The first rows show the original images, the second, third and fourth rows are the reconstruction images using BIHT, YP and proposed algorithms, respectively when $m = 784$ in a noiseless setting for FashionMNIST dataset.

certain value of M . This is because the proposed algorithm approximates the sparse signal by some signal in the range space of the generator that minimizes the loss, i.e., the NMSE of the proposed algorithm is always lower-bounded by the representation error. However, with the traditional algorithms, an increase in M will always provide extra information and improves the reconstruction performance. Hence, we see a monotonic decrease of NMSE values of the traditional algorithms with M .

Next, we study the effect of choice of the architecture of generative models on signal reconstruction performance. To investigate the effect of choice of architecture, we study the re-

construction performance of MNIST and FashionMNIST dataset with same VAE architecture. In [96], the data distribution of Fashion-MNIST is shown to be more complicated compared to MNIST through tSNE visualization [70]. We train VAEs for these two datasets with the same neural network structure, over the same number of epochs with the same learning rate, and with the same optimizer. The data distribution of the testing set for MNIST and Fashion-MNIST is similar to their training sets. In this setup, we consider noiseless one-bit measurements for signal reconstruction. In Figure 6.3, we plot the reconstructed images of one-bit CS with generative models for images from MNIST and Fashion-MNIST datasets. The reconstructed image shows that with the MNIST dataset, the visual quality is better than that of the Fashion-MNIST dataset. We can see that for the Fashion-MNIST dataset, the generative model learned the distribution of the shape of the clothing images, but it did not learn the details in images such as textures. Though the generative model learned the distribution of the MNIST dataset, the architecture was not complex enough to learn the entire distribution of the Fashion-MNIST dataset. So, the reconstruction performance with the images from Fashion-MNIST dataset is not as good as of MNIST dataset. So, we should note that the neural network architecture should be chosen appropriately to ensure better signal reconstruction performance.

Noisy setting

In our next experiment, we study the reconstruction performance in the presence of noise. In Figures 6.5 and 6.6, we provide the recovery performance of our algorithm with YP and BIHT in a noisy scenario, i.e., when $v_n = 0.1$ and $\alpha = 0.85$. We can see similar trends in the recovery performance of the proposed algorithm, BIHT and YP with a few notable differences. First, the reconstruction performance of all these algorithms has decreased with noise compared to the noiseless scenario. Second, unlike in Figure 6.2, there does not exist a point or region in Figure 6.6 where the NMSE performance of the BIHT algorithm or the YP algorithm is equal to or better than the performance of the proposed algorithm for any value of M . This shows

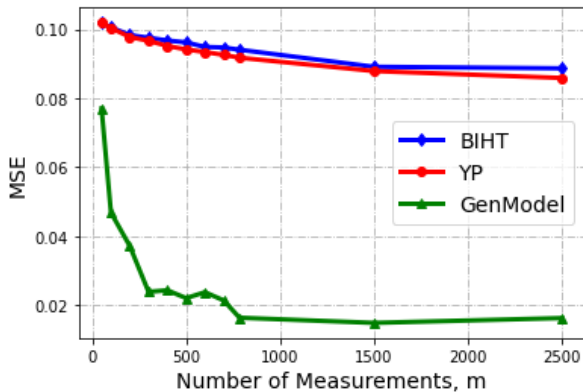


Fig. 6.5: MSE of the proposed algorithm compared with BIHT, and YP as a function of number of measurements m in a noisy setup ($v_n = 0.1$ and $\alpha = 0.85$).

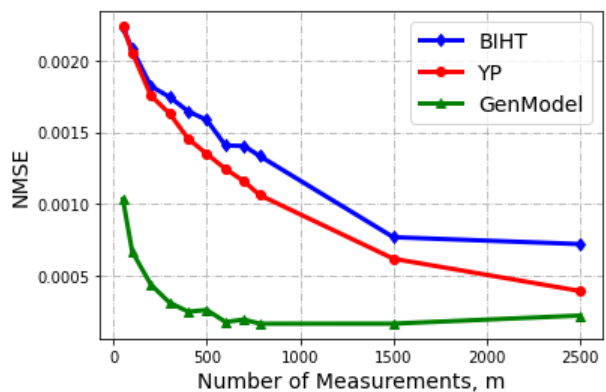


Fig. 6.6: NMSE of the proposed algorithm compared with BIHT, and YP as a function of number of measurements m in a noisy setup ($v_n = 0.1$ and $\alpha = 0.85$).

that the performance of the traditional algorithms is more sensitive to noise. Also, the loss in performance because of the representation error is minimal in comparison to the performance degradation in recovery performance in the presence of noise. Hence, the proposed algorithm outperforms the traditional algorithms in both MSE and NMSE metrics for the noisy setup. Similarly, in Figures 6.7 and 6.8, we plot MSE and NMSE values for the proposed algorithm, BIHT and YP as a function of sign-flip probability $1 - \alpha$ when $M = 784$. These results shows that the proposed method has better reconstruction performance in presence of sign-flip noise compared to BIHT, and YP.

Next, we compare the robustness of the proposed algorithm and traditional algorithms with respect to the additive measurement matrix uncertainties. The receiver has noiseless one-bit measurements, i.e., $v_n = 0$ and $\alpha = 1$. Let \mathbf{A}' be the measurement matrix at the receiver, which is defined as

$$\mathbf{A}' = \mathbf{A} + \Delta,$$

where Δ is the unknown perturbation in the measurement matrix, \mathbf{A} . We draw each element of the perturbation, Δ , from i.i.d. Gaussian distribution with zero mean and v_Δ variance. In Figure 6.9, we plot the NMSE performance as the function of the uncertainties, i.e., v_Δ , when

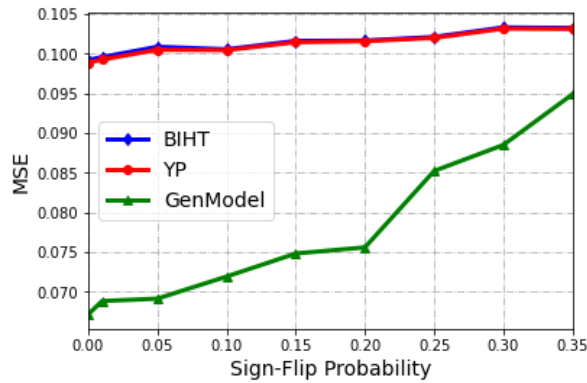


Fig. 6.7: MSE of the proposed algorithm compared with BIHT, and YP as a function of number of sign-flip probability when $M = 784$ and $v_n = 0.1$.

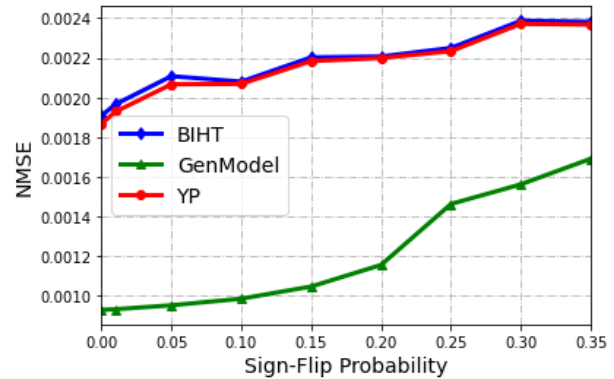


Fig. 6.8: NMSE of the proposed algorithm compared with BIHT, and YP as a function of sign-flip probability when $M = 784$ and $v_n = 0.1$.

$M = 1500$. From the simulation result, we can see that the proposed algorithm has the least rate of increase of NMSE with respect to v_Δ which shows the proposed algorithm is more robust to the additive measurement matrix uncertainties.

Limitation

In this subsection we study the limitation of using generative models to the task of compressed sensing. Specifically, we look into the performance of the proposed algorithm when the range space of the generative models does not faithfully represent the distribution of the compressed signal. In this experiment, we use the Omniglot dataset, where the testing data does not follow the exact distribution of the training data. So, the range space of the generative models does not include the testing signals that we desire to reconstruct. The NMSE values of this experiment is plotted in Figure 6.10. For Omniglot dataset, we can see that the NMSE value is higher with the generative model compared to the MNIST, which is because of the high representation error. Further, because of high representation error, BIHT outperforms Genmodel with a smaller value of M , i.e., when M slightly greater than 300. However, for MNIST, the BIHT algorithm matches up the performance of GenModel when $M = 1500$. Hence, it is crucial that the trained

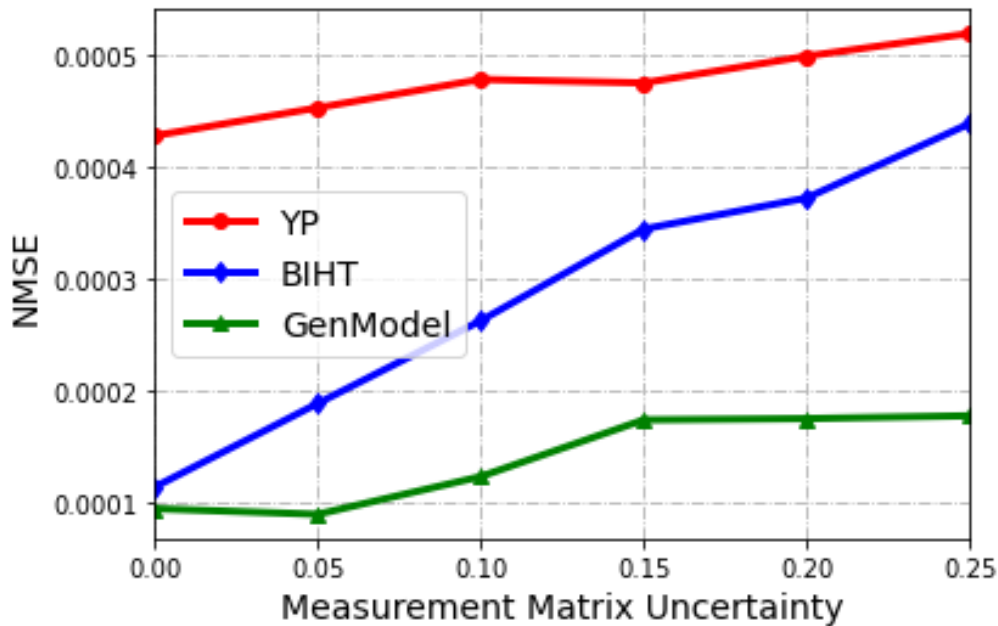


Fig. 6.9: Reconstruction performance of the proposed algorithm, BIHT and YP as a function of measurement matrix uncertainty, v_{Δ} , when $M = 1500$ in a noiseless setting ($\alpha = 1$ and, $v_n = 0$).

generative model represents the signal distribution of the compressed signal. Any change in the signal distribution over time may lead to degradation of reconstruction performance.

Comparison with Generative model based algorithm:

In this subsection, we compare the recovery performance of the proposed algorithm with generative model-based one-bit CS algorithm in the literature (labeled as projGen). From the simulation result we can observe that the proposed algorithm has better MSE and NMSE reconstruction performance in a noiseless setting, i.e., when $v_n = 0$ and $\alpha = 1$. From MSE and NMSE results in Figures 6.11 and 6.11, respectively, we can conclude that the proposed algorithm has better estimation of the signal direction and amplitude than GenModel_pg. Hence, the proposed algorithm is better than the model-based and the competing generative model-based CS algorithms from one-bit measurements.

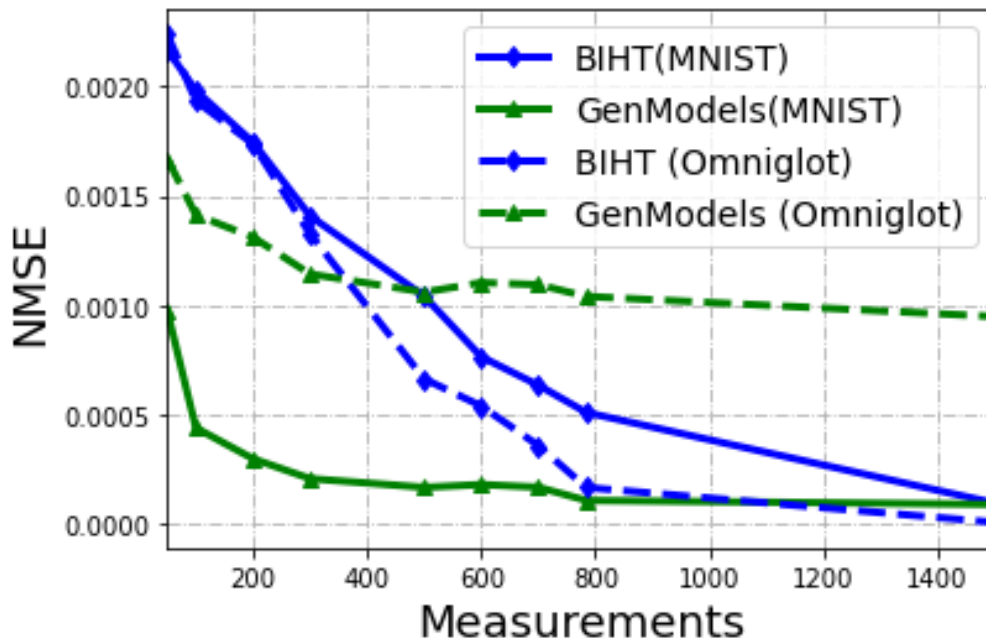


Fig. 6.10: NMSE values of proposed algorithm and BIHT for MNIST and Omniglot dataset as a function of number of measurements

Application: Channel State Estimation Using One-bit Measurements

In this subsection, we investigate a potential application of one-bit CS with generative models to estimate channel state information. Consider an FDD MIMO system. The downlink channel from the base station to the mobile station is not symmetric with the uplink channel as different spectral bands are used for the uplink and downlink communication. Hence, the base station transmits a pilot sequence to the mobile user to estimate the downlink channel. The mobile station estimates the channel and feedbacks the CSI to the base station. The channel in massive MIMO is known to be sparse and hence can be compressed by a mobile station using the compressed sensing paradigm. To decrease the uplink bandwidth resource, the CSI is compressed and each element of the compressed vector is quantized to one-bit. These one-bit compressed measurements could be further superimposed on the uplink user sequence to reduce overhead on the uplink bandwidth resource. The process of superposition of one-bit measurements in uplink user sequence and the recovery of one-bit compressed channel state information is dis-

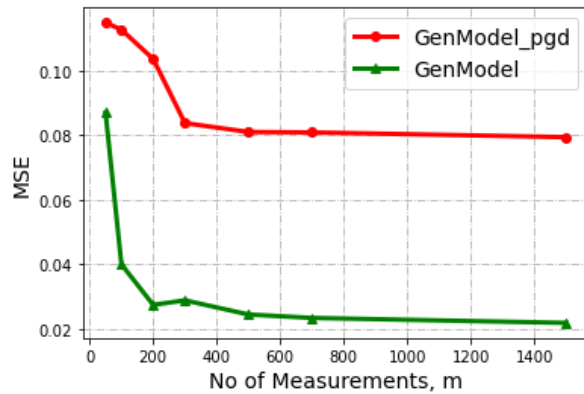


Fig. 6.11: MSE of the proposed algorithm compared with GenModel_pgd as a function of number of Measurements when $v_n = 0.1$.

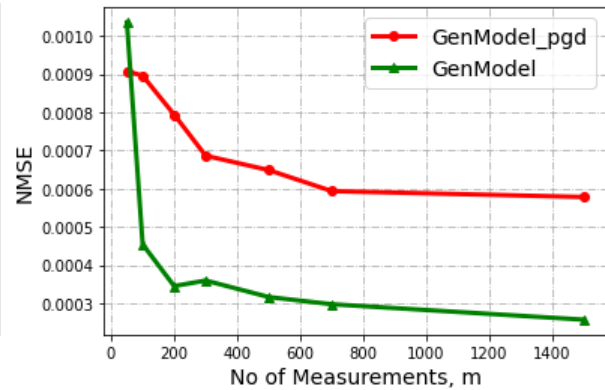


Fig. 6.12: NMSE of the proposed algorithm compared with GenModel_pgd as a function of number of Measurements when $v_n = 0.1$.

cussed in detail in [79, 98]. Next, the base station recovers one-bit measurements from the uplink user sequence. The final goal at the base station is to estimate the downlink CSI from the recovered one-bit measurements.

In this experiment, we compare the work in [79] with the proposed algorithm after one-bit compressed measurements have been recovered at the BS. We train a generative adversarial network to learn the downlink channel distribution. The downlink channel is assumed to be Rayleigh distributed. The channel vector is generated following the 3GPP protocol. We consider 60000 channel vectors as the training set and 10000 channel vectors as the testing set. We consider a generative adversarial network with a fully connected neural network as the generator and a fully connected neural network as the discriminator. We train such generative adversarial networks with the Adam optimizer with a learning rate of 0.001. We pick 100 data points from the testing set and take one-bit compressed measurements. We consider a noisy compressed sensing model for the generation of one-bit measurements with $\alpha = 0.90$ and $v_n = 0.05$. The Gaussian noise represents the error in channel estimation by the mobile station, and the sign-flip noise represents the uplink channel noise. The base station reconstructs CSI from noisy one-bit measurements. In Figure 7, we plot the reconstruction performance of channel estimation of the proposed algorithm with the algorithm in [79]. As the algorithm

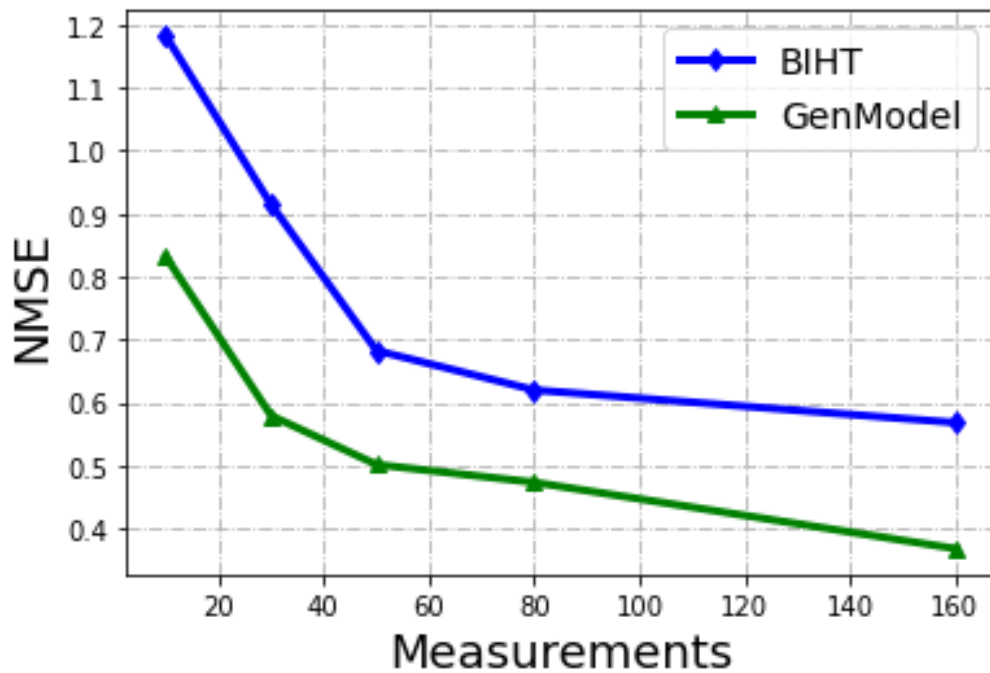


Fig. 6.13: Comparison of CSI estimation of the proposed algorithm with the state-of-the-art algorithm from one-bit compressed measurements when $\alpha = 0.9$ and $v_n = 0.05$

in [79] is inspired by BIHT, we refer to this algorithm as BIHT in the figure. The NMSE performance shows the proposed algorithm outperforms the BIHT algorithm. In addition to the gain in performance, the proposed algorithm provides a saving in the number of bits required to transmit from the mobile station to the BS compared to [79]. For the same number of one-bit compressed measurements, [79] requires additional N bits to transmit support information to the BS. Even with additional N bits, the existing algorithm performs worse compared to the proposed algorithm.

6.4 Compressed Sensing using Untrained Network as a Prior

In this section, we study the second approach to CS that uses an untrained network as a prior on the signal. In the following subsections, we provide the details of the solution space, loss

function, solution methodology and detailed experimental results.

6.4.1 Solution Space

In the approach, we assume that the set \mathcal{S} represents the range space of a neural network $G(\mathbf{w}; \mathbf{g})$. Here, \mathbf{w} denotes the weights (parameters) of the network, and $\mathbf{g} \in \mathbb{R}^k$ is the latent (input) vector. Thus, any sparse vector \mathbf{x} can be represented as $\mathbf{x} = G(\mathbf{w}; \mathbf{g})$ for some $\mathbf{g} \in \mathbb{R}^k$ and \mathbf{w} . Our approach considers an untrained network $G(\mathbf{w}; \mathbf{g})$ with a fixed latent vector \mathbf{g} . So the problem of estimating $\hat{\mathbf{x}}$ translates to evaluating an estimate $\hat{\mathbf{w}}$ of the neural network weights. Therefore, the optimization problem in (6.2) can be rewritten as

$$\min_{\mathbf{w}} \mathcal{L}(\mathbf{A}G(\mathbf{w}; \mathbf{g}), \mathbf{y}). \quad (6.10)$$

6.4.2 Design of the Loss Function \mathcal{L}

We consider the following objective function for signal recovery from one-bit measurements:

$$\mathcal{L}(\mathbf{A}G(\mathbf{w}; \mathbf{g}), \mathbf{y}) = -\frac{\mathbf{y}^\top \mathbf{A}G(\mathbf{w}; \mathbf{g})}{\|G(\mathbf{w}; \mathbf{g})\|_2^2}. \quad (6.11)$$

The numerator of the objective function measures the consistency between the sign measurements \mathbf{y} and the corresponding linear measurements. For a fixed l_2 norm of $G(\mathbf{w}; \mathbf{g})$, the term is maximized when $\text{sign}(\mathbf{A}G(\mathbf{w}; \mathbf{g})) = \mathbf{y}$. Therefore, the numerator ensures the consistency between $\mathbf{A}G(\mathbf{g})$ and \mathbf{y} . However, the numerator decreases as the l_2 norm of $G(\mathbf{w}; \mathbf{g})$ increases, and therefore, we use the term in the denominator to control the norm. Hence, the two terms of the objective function jointly optimize the representation error.

Network Architecture

We consider an untrained network $G(\mathbf{w}; \mathbf{g})$ that takes the form of an expansive neural network as in [44] as the prior. The network transforms a randomly chosen fixed latent vector to output signal by applying 1×1 convolution, applying rectified linear units (ReLU), upsampling operations and normalizing the channels. Let $\mathbf{Z}_1 \in \mathbb{R}^{n_0 \times k_1}$ be input of the network. Specifically, the tensor output of d -th layer is given by

$$\mathbf{Z}_d = bn(\mathbf{U}_{d-1}\sigma_r(\mathbf{Z}_{d-1}\mathbf{W}_{d-1})), \quad d = 0, \dots, D-1, \quad (6.12)$$

where $\mathbf{W}_d \in \mathbb{R}^{(d-1) \times d}$ is 1×1 convolution layer, \mathbf{U}_d is an upsampling operator that perform bi-linear upsampling, $\sigma_r(\cdot)$ is a ReLU activation function, and $bn(\cdot)$ is a batch normalization layer. Finally the output of the untrained network is formed as

$$\begin{aligned} \mathbf{x} &= G(\mathbf{w}; \mathbf{g}) = \sigma_s(\mathbf{U}_{D-1}\sigma(\mathbf{Z}_{D-1}\mathbf{W}_{D-1})\mathbf{W}_D) \\ &= \sigma_s(\mathbf{Z}_D\mathbf{W}_D), \end{aligned}$$

where $\sigma_s(\cdot)$ is a sigmoid function. We call any untrained network as a prior to signal $\mathbf{x} \in \mathbb{R}^n$ if the signal belongs to a set \mathcal{S} defined as:

$$\mathcal{S} = \{x | x = G(\mathbf{w}; \mathbf{g})\}, \quad (6.13)$$

where $\mathbf{g} := \text{vec}(\mathbf{Z}_1)$ is a randomly chosen but fixed latent vector and $G(\mathbf{w}; \mathbf{g})$ is the untrained neural network. The set \mathcal{S} is called as the range space of the network. The untrained network architecture $G(\mathbf{w}; \mathbf{g})$ should be chosen based on the signal \mathbf{x} such that the signal lies in the range space or has small representation error.

Algorithm 6.1 Projection based gradient descent method for one-bit compressed sensing

- 1: **Input:** Data samples \mathbf{y} , \mathbf{A} , η .
- 2: **Output:** $G(\mathbf{w}; \mathbf{g})$.

- 3: **for** $t = 0$ to $T - 1$,
- 4: $\mathbf{s}^t = \mathbf{x}^t - \eta \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{w})$,
- 5: $\mathbf{w}^t = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{s}^t - G(\mathbf{w}^t; \mathbf{g})\|_2$,
- 6: $\mathbf{x}^{t+1} = G(\mathbf{w}^t; \mathbf{g})$
- 7: **end for**
- 8: Output the $\hat{\mathbf{x}} = G(\mathbf{w}^T; \mathbf{g})$.

6.5 Solution Methodology

We minimize the loss function in (6.11) using a projected gradient descent based algorithm. The steps of the algorithms are outlined in Algorithm 1, which has three major steps. In the first step, we evaluate the gradient of the loss function, $\mathcal{L}(\cdot)$ with respect to \mathbf{x} , *i.e.*, $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x})$, and minimize the objective by taking one step of gradient descent step in Step 4 of Algorithm 1. The updated variable g^t may not lie on the range space of the untrained network, \mathcal{S} . Next, we project the signal, g^t in the range space of the untrained network. The optimization problem for the projection takes the form

$$\mathbf{w}^t = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{s}^t - G(\mathbf{w}; \mathbf{g})\|_2. \quad (6.14)$$

Note that the optimization problem is non-convex due to the structure of $G(\mathbf{w}; \mathbf{g})$. Hence, the minimization of the loss function does not guarantee convergence to the globally optimal solution. The minimization task is carried out by backpropagation. This minimization task can be carried out using standard computational frameworks such as Tensorflow and Pytorch. Finally, we compute the signal in \mathcal{S} by the action of $G(\mathbf{w}; \mathbf{g})$ with updated weight, \mathbf{w}^t on the fixed latent vector \mathbf{g} , *i.e.* $\mathbf{x}^{t+1} = G(\mathbf{w}^t; \mathbf{g})$. We repeat the algorithm until the stopping condition is reached.

Initialization: The latent vector \mathbf{g} is drawn from the uniform distribution and kept constant

over entire minimization task.

6.5.1 Simulation Results

In this section, we evaluate the signal reconstruction performance of our algorithm. Let \mathbf{x}^* and $\hat{\mathbf{x}}$ represent the true signal and the estimated signal respectively. We evaluate performances using following two metrics

- Mean Square Error (MSE):

$$\text{MSE} \triangleq \|\mathbf{x}^* - \hat{\mathbf{x}}\|_2^2 \quad (6.15)$$

- Normalized Mean Square Error (NMSE):

$$\text{NMSE} \triangleq \left\| \frac{\mathbf{x}}{\|\mathbf{x}^*\|_2} - \frac{\hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|_2} \right\|_2^2 \quad (6.16)$$

Dataset: We consider the MNIST handwritten digit dataset [59] for experimental validation of the proposed algorithm. The images in the MNIST dataset are sparse in pixel values.

Baseline Algorithms: We compare the performance of our algorithm with two traditional one-bit CS algorithms that requires sparsity on some basis: convex optimization-based algorithm (labeled as Convex), and binary iterative hard thresholding (labeled as BIHT) algorithm. Further, we compare the performance with [50] (labeled as TrainedNwk), which uses a trained Generative model as the prior.

Untrained Network: The untrained network $G(\mathbf{w}; \mathbf{g})$ takes the form of an expansive network. We adopt the setup from [49] where we fix the architecture to have 2 layer configuration with $k_1 = 15, k_2 = 15$ and $k_3 = 10$ and use bilinear upsampling operations. Both trained and untrained network are implemented in Pytorch with GPU support. We use the ADAM optimizer to minimize the cost function with respect to model parameters. The latent variable, \mathbf{g} , is drawn

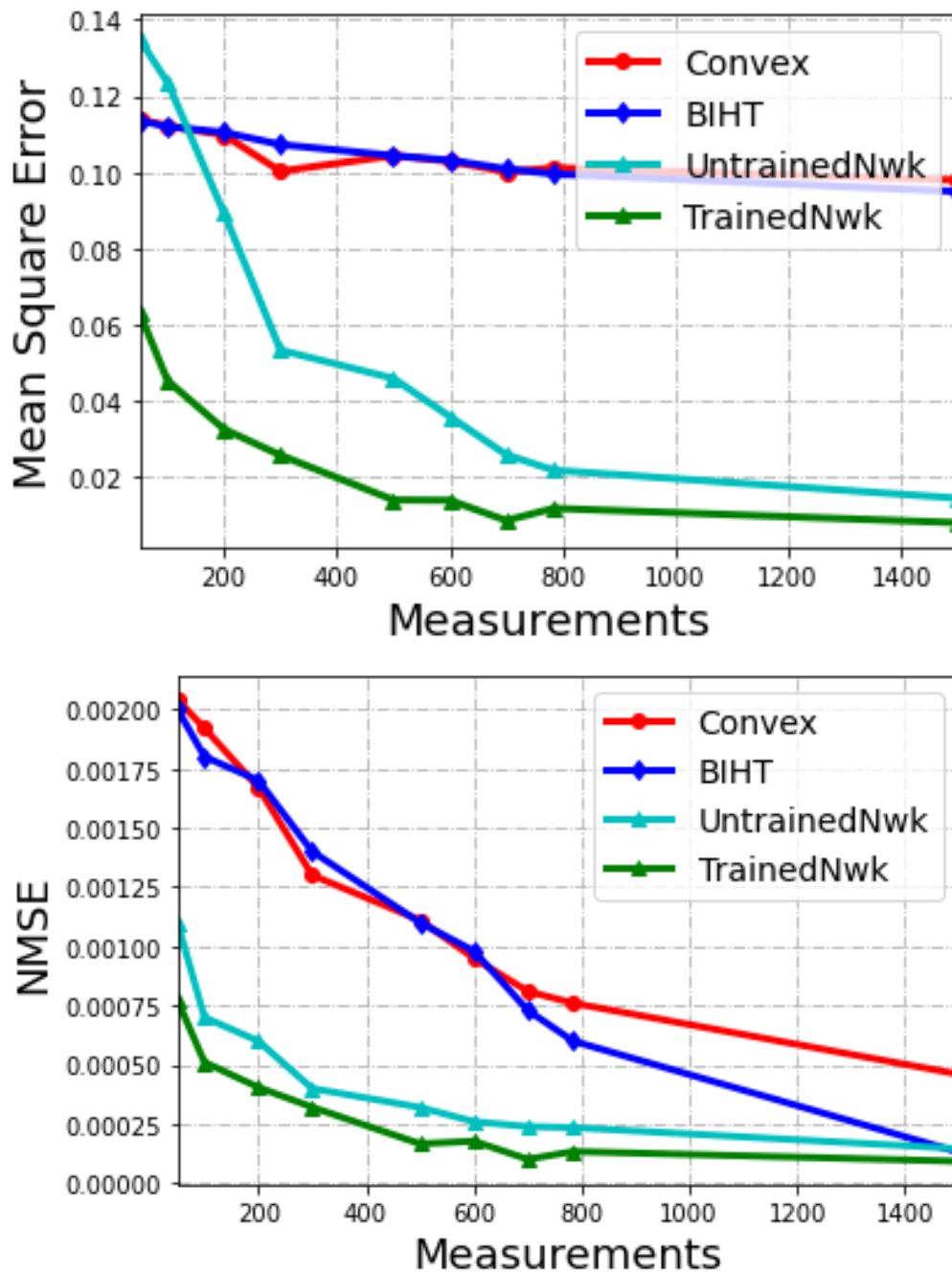


Fig. 6.14: Reconstruction performance of our algorithm compared with the state-of-the-art algorithm.

from uniform distribution.

Measurement model: The columns of measurement matrix \mathbf{A} are drawn uniformly from the surface of the m -dimensional unit hypersphere [73]. We use one-bit compressed measurements $y_i = \text{sign}(\langle \mathbf{A}_i, \mathbf{x}^* \rangle) \in \{\pm 1\}, i = 1, 2, \dots, M$.

We average the results over ten different images from the testing set of MNIST dataset. As the considered algorithm is not convex, we do three random restarts for each image and report the average loss as the reconstruction loss.

In Figure 6.14, we provide the signal reconstruction performance of our algorithm (labeled as `UntrainedNwk`) with `Convex`, `BIHT`, and `TrainedNwk` in a noiseless scenario. The key observations are as follows:

- The MSE performance of our algorithm is comparable to `BIHT` and `Convex` and worse than `TrainedNwk` for small number of measurements ($M < 200$). This behavior is because `TrainedNwk` uses a generative model that is well-trained on a huge data set. On the other hand, the other three algorithms have no prior information regarding the compressed signal and require more measurements to ensure a similar accuracy level.
- For a slightly larger number of measurements ($M > 200$), the performance of our algorithm becomes comparable to that of `TrainedNwk`, whereas the other two algorithms continue to offer poor MSE performance. In other words, our algorithm recovers both magnitude and direction information without pretraining the network on a huge dataset. Thus, our method eliminates the need for pretraining without significantly compromising the recovery performance.
- The NMSE performance shows that our algorithm estimates the direction of the sparse vector with a similar accuracy level as that of `TrainedNwk`, when M is comparable to N or larger. Note that `TrainedNwk` and `UntrainedNwk` output an estimate from their respective range spaces that minimizes the loss. Thus, a limitation of these algorithms is that if the compressed

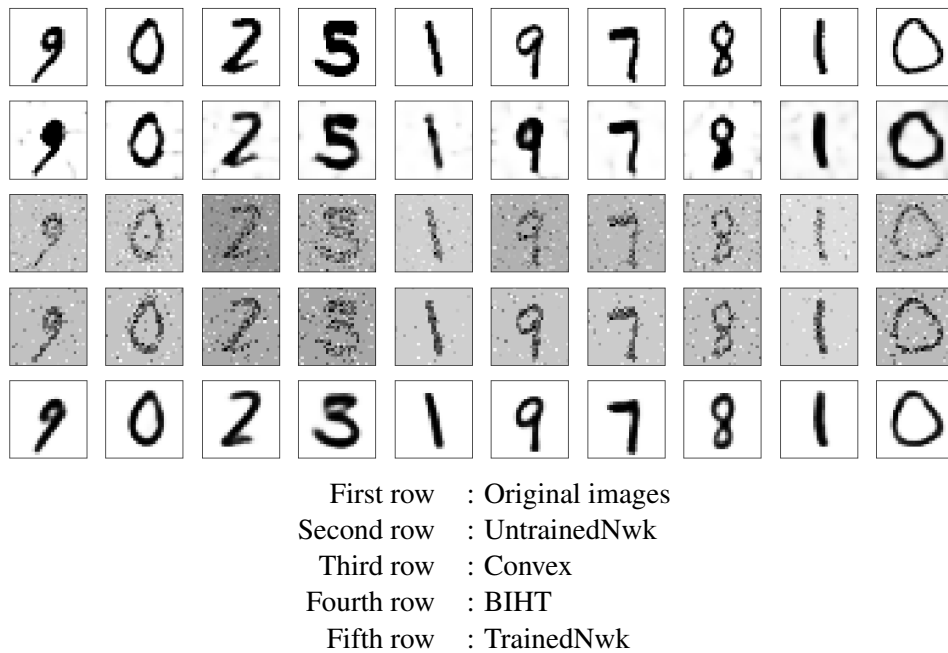


Fig. 6.15: Reconstruction when $M = 1500$.

signal lies outside \mathcal{S} , they output a vector from the range space which is closest to the ground truth. So the NMSE is always lower bounded by the distance between the true vector and the range space, irrespective of the value of M . The NMSE value of our algorithm saturates at $M = 784$ and does not significantly improve beyond that. However, the NMSE of BIHT monotonically decreases with M .

- The NMSE of BIHT becomes similar to that of the proposed algorithm when $M = 1500$. However, the proposed algorithm arrives at a reliable estimate of the sparse vector with almost half of the measurement compared to that required by BIHT algorithm. We also note that the MSE of BIHT is high for a large value of M , even when the NMSE is low. This observation indicates that the direction information of the sparse vector is recovered with good accuracy, but the magnitude is not accurate. This behavior is because BIHT estimates the signals on the unit ball, and thus, it is unable to recover the magnitude of the sparse vector.

In the next experiment, we compare the images reconstructed by the proposed algorithm

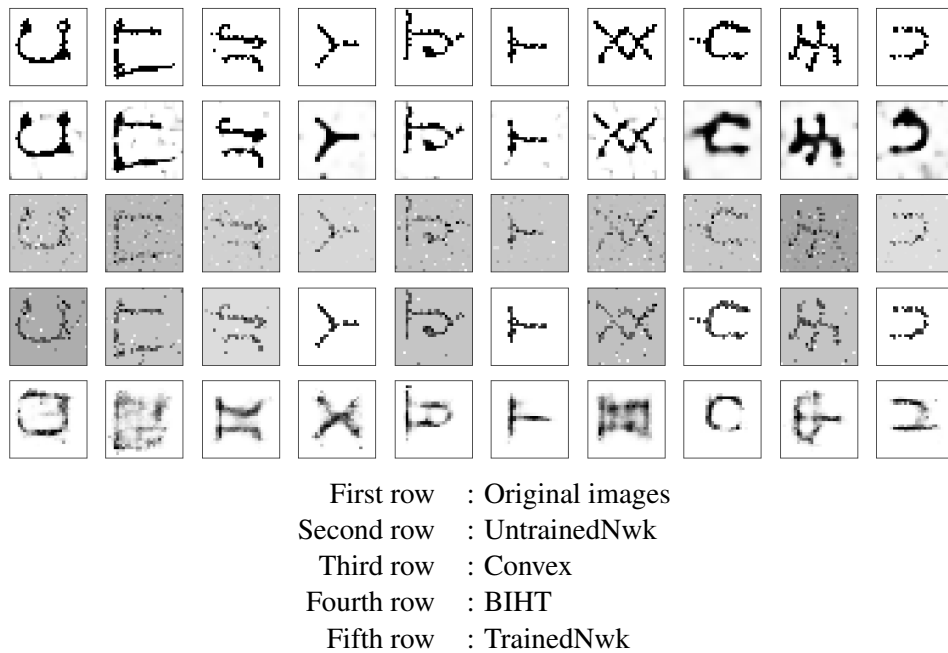


Fig. 6.16: Reconstruction when $M = 1500$.

with the competing algorithm. Figures 6.15 and 6.16 compare the images reconstructed by the above four algorithms from MNIST and Omniglot datasets using 1500 one-bit measurements. For the untrained and trained network-based algorithm, we use the same network architecture for both datasets. UntrainedNwk algorithm offers good reconstruction quality for both datasets. Further, we observe that the visual quality of images reconstructed by BIHT and Convex algorithms are worse compared to the other two algorithms. This is because these model-based algorithms always output an estimate lying in the unit norm ball, and they provide inaccurate estimates when the norm of the unknown vector (image in our case) is not unity. On the contrary, the deep learning-based algorithms recover both the magnitude and the direction of the unknown vector, and thus, offer a better visual quality. However, the visual quality of TrainedNwk is better than that of UntrainedNwk for the MNIST dataset, and it is worse for the Omniglot dataset. This is because the performance of TrainedNwk depends on the training data, and for the Omniglot dataset, the distribution of test data is slightly different from that of the training data. This observation confirms that if the distribution of data changes over time,

the recovery performance of the TrainedNwk algorithm degrades while the performance of the UntrainedNwk algorithm is relatively constant.

6.6 Summary

In this chapter, we proposed deep learning-based approaches for the problem of one-bit compressed sensing using trained and untrained feed-forward neural networks as priors. We provided a lower bound on the sample complexity and quantified the reconstruction performance of the one-bit compressed sensing algorithm when a trained network is used as a prior. We provided detailed numerical results to demonstrate the superior performance compared to traditional algorithms such as BIHT. We also showed that the recovery performance of the proposed algorithm is robust to noise. We further investigated one-bit compressed sensing with an untrained neural network as a prior. We formulated an optimization problem and solved it using the projected gradient descent-based method. We provided sufficient numerical experiments to demonstrate the superior performance of the proposed algorithm compared to the existing model-based approach. In contrast to the existing model-based algorithm, we showed that both the proposed algorithms estimated both the magnitude and the direction of the signal from one-bit measurements.

CHAPTER 7

SUMMARY AND FUTURE DIRECTIONS

7.1 Summary

In this dissertation, we investigated the performance of several algorithms for sparse signal estimation or its parameter estimation from noisy one-bit measurements. We studied the signal recovery performance using both model-based sparse signal recovery algorithms and deep learning-based signal recovery algorithm. We provided computationally tractable algorithms in all of these problems and showed superior performance compared to the competing algorithms.

We studied the problem of joint sparse support estimation from one-bit noisy measurements. We analyzed the error performance of ML-based support estimation and provided the lower bounds on the number of one-bit measurements per node for vanishing probability of error. As this ML-based method is computationally intractable, we proposed tractable centralized and decentralized algorithms for joint sparse support estimation. We showed that the proposed algorithms are better than the competing algorithms and have good performance even in the presence of noise.

We then considered sparse signal reconstruction problems from noisy one-bit measurements. First, we developed an algorithm for sparse signal reconstruction from noisy one-bit

measurements. We then extended this algorithm to the setup when the receiver has access to some side-information. In the first problem, we considered the scenario where we do not assume temporal dependence in sparse signals. We developed a GAMP-based algorithm to approximate the MMSE estimator of the signal. We provided closed-form expressions for all the update equations, and hence the algorithms are computationally efficient. In the second problem, we extended the setup of the first problem to the case where the receiver has access to SI and inherent time-varying dynamics of the signal. We used the BDD model to capture the dependence between the signals at consecutive time instances. We developed a GAMP-based algorithm for this setup. In the third problem, we studied a case when the receiver has access to heterogeneous SI. These signals and the heterogeneous SI are from different modalities and may not share joint sparse representation. We used Copula functions to incorporate the dependence information between the SI and the signal. We then developed a GAMP-based algorithm to compute the MMSE estimator of the signal.

Finally, we studied data-driven-based approaches for one-bit compressed sensing. We developed algorithms for signal reconstruction from one-bit measurements using a trained network and an untrained network as priors on the signal. We provided a lower bound on sample complexity and an upper bound on the reconstruction performance when the trained network was used as a prior. In contrast to the model-based algorithms such as BIHT, we showed that the proposed algorithms estimate both amplitude and direction of sparse signals from the one-bit measurements.

7.2 Future Directions

Some promising directions for future work are listed in the following:

1. **Decentralized one-bit compressed sensing:** The one-bit compressed sensing problem has not been investigated much in a decentralized setup. In Chapter 2, we have proposed

decentralized one-bit compressed sensing algorithms. The work assumes a joint-sparse structure and independence among signal amplitudes which is not always the case. Following problems can be studied in the future for a better understanding of the decentralized one-bit CS:

- The theoretical analysis of the algorithms in Chapter 2 for quantifying error performance and sample complexity needs to be developed.
- The assumptions in Chapter 2 need to be extended to a more complex setting where signals have joint support with some innovations and higher-order dependence among the amplitudes.
- In Chapter 5, we developed a signal reconstruction algorithm from one-bit measurements when we have heterogeneous side-information at the receiver. In this dissertation, we assumed that the signal and the side-information are of the same dimension. However, in several practical scenarios, the signal and heterogeneous SI are likely to be of different dimensions. One should extend the work such that the algorithm generalizes to this setup.

2. **One-bit compressed sensing with network prior:** A few works have scratched the surface of using the deep learning-based method for one-bit compressed sensing. A detailed understanding of the signal reconstruction from one-bit compressed measurements will translate into some of the following problems:

- The theoretical results in Chapter 6 need to be extended to a setup when the measurement matrix is deterministic or is realized from other distributions such as a Sub-Gaussian distribution.
- In Chapter 6, we studied the empirical performance of the one-bit compressed sensing using an untrained network prior. Theoretical performance guarantees of the

proposed algorithm, i.e., the sample complexity for desired signal reconstruction performance, need to be developed.

- In Chapter 6, we proposed two different algorithms, one for one-bit CS with a trained prior and the other for one-bit CS with an untrained prior. In this setup, even a slow change in signal distribution can cause the trained network-based method to have increased error during signal reconstruction. In the future, one should investigate possible neural network architectures and corresponding optimization problems that can track the slow change of distribution of the signal and study the reconstruction performance.

APPENDICES

A Proof of Theorem 2.2

Proof of Theorem 2.2 First, we compute $\mathbb{E}\{\tilde{a}_{pt}^0\}$ and $\mathbb{E}\{\tilde{a}_{pt}^1\}$ and use it to derive the desired lower bound.

Let $u_{pk} = \sum_{i=1}^K \left(\tilde{\mathbf{A}}_{\mathcal{U}_k}^p \right)_{1i} \mu_i$. When the entries of the projection matrix \mathbf{A} are i.i.d. Gaussian with mean zero and variance $\frac{1}{N}$, it can be shown that u_{pk} is a Gaussian random variable with mean zero and variance $\frac{1}{N} \|\boldsymbol{\mu}\|_2^2$. Then we have,

$$\begin{aligned} \mathbb{E}\{\tilde{a}_{pt}^0\} &= \mathbb{E}\{(1 - \lambda_{pj})^{\frac{1}{2}} (1 - \lambda_{pk})^{\frac{1}{2}} | |\mathcal{U}_j \cap \mathcal{U}_k| = t\} \\ &= \int \int (1 - \lambda_{pj}(u_{pj}))^{\frac{1}{2}} (1 - \lambda_{pk}(u_{pk}))^{\frac{1}{2}} f_{U_k U_j}(u_{pk}, u_{pj}) du_{pk} du_{pj} \end{aligned}$$

which can be found by a 2-fold integration, where we write $\lambda_{pj}(u_{pj}) = Q\left(\frac{-u_{pj}}{\sigma_v^2 + \frac{K}{N}\sigma_s^2}\right)$. In the high dimensional setting, given that $|\mathcal{U}_j \cap \mathcal{U}_k| = t$, the joint pdf of (u_{pk}, u_{pj}) , $f_{U_k U_j}(u_{pk}, u_{pj})$ tends to be a bi-variate Gaussian with mean $\mathbf{0}$ and covariance matrix Σ_t [94] given by

$$\Sigma_t = \frac{\boldsymbol{\mu}^T \boldsymbol{\mu}}{N} \begin{bmatrix} 1 & \rho_t \\ \rho_t & 1 \end{bmatrix}$$

where $\rho_0 = 0$, $\rho_t = \frac{\sum_{i=1}^t \tilde{\mu}_i}{\boldsymbol{\mu}^T \boldsymbol{\mu}}$ for $t = 1, \dots, K-1$, $\tilde{\boldsymbol{\mu}}_t = \{\mu_m \mu_n; \text{ if } \left(\tilde{\mathbf{A}}_{\mathcal{U}_k}^p \right)_{1m} = \left(\tilde{\mathbf{A}}_{\mathcal{U}_j}^p \right)_{1n} \text{ for } m, n = 1, 2, \dots, K\}$ for $|\mathcal{U}_j \cap \mathcal{U}_k| = t$ and $\tilde{\boldsymbol{\mu}}_t = [\tilde{\mu}_1, \dots, \tilde{\mu}_t]^T$.

In the following, we evaluate $\mathbb{E}\{\tilde{a}_{pt}^0\}$ and $\mathbb{E}\{\tilde{a}_{pt}^1\}$, and upper bound P_{err} . In contrast to [94], where $\tilde{\mathbf{s}}$ was assumed to be first order Gaussian, we do not make any assumptions on $\tilde{\mathbf{s}}$. Let

$u_{pk} = x$, $u_{pj} = y$ and $\sigma_p^2 = \sigma_v^2 + \frac{K}{N}\sigma_s^2$ and $\sigma^2 = \frac{1}{N}\|\boldsymbol{\mu}\|_2^2$. From Equations (2) and (14) in [20], we have the following approximation

$$Q(x) \approx \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) \quad (8.1)$$

for small x , where $Q(x) = \int_x^\infty \sqrt{\frac{1}{2\pi}} \exp(-\frac{z^2}{2}) dz$. In [20], the authors have claimed that the right hand side (R.H.S.) of (8.1) also acts as a tight upper bound for $x > 0.5$.

Accordingly, we also have

$$\lambda_{pj}(x) \approx \begin{cases} \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) & \text{if } x < 0 \\ 1 - \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) & \text{if } x > 0 \end{cases}$$

and similar is the case with $\lambda_{pk}(y)$. Then, we have,

$$\begin{aligned} \mathbb{E}\{\tilde{a}_{pt}^0\} &= \mathbb{E}\{(1 - \lambda_{pj})^{\frac{1}{2}}(1 - \lambda_{pk})^{\frac{1}{2}} \mid |\mathcal{U}_j \cap \mathcal{U}_k| = t\} \\ &= (1 - \lambda_{pj}(x < 0))(1 - \lambda_{pk}(y < 0)) \times Pr(x < 0 \& y < 0 \mid |\mathcal{U}_j \cap \mathcal{U}_k| = t) \\ &\quad + (1 - \lambda_{pj}(x < 0))(1 - \lambda_{pk}(y > 0)) \times Pr(x < 0 \& y > 0 \mid |\mathcal{U}_j \cap \mathcal{U}_k| = t) \\ &\quad + (1 - \lambda_{pj}(x > 0))(1 - \lambda_{pk}(y < 0)) \times Pr(x > 0 \& y < 0 \mid |\mathcal{U}_j \cap \mathcal{U}_k| = t) \\ &\quad + (1 - \lambda_{pj}(x > 0))(1 - \lambda_{pk}(y > 0)) \times Pr(x > 0 \& y > 0 \mid |\mathcal{U}_j \cap \mathcal{U}_k| = t). \end{aligned}$$

We need to find the following expression to calculate the upper bound on P_{err} .

$$\mathbb{E}\{\tilde{a}_{pt}^0\} = \mathbb{E}\{\tilde{a}_{pt}^1\} = 2(I_{p1} + I_{p2} - I_{p3} - I_{p4}) + \frac{1}{4} + \frac{1}{2\pi} \sin^{-1} \rho_t \quad (8.2)$$

where

$$I_{p1} = \int_{-\infty}^0 \int_{-\infty}^0 \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) \\ \times \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{y^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2y^2}{3\sigma_p^2}\right) \right) \times f(x, y) dx dy$$

$$I_{p2} = \int_0^{\infty} \int_{-\infty}^0 \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{y^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2y^2}{3\sigma_p^2}\right) \right) \times f(x, y) dx dy$$

$$I_{p3} = \int_0^{\infty} \int_{-\infty}^0 \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) \\ \times \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{y^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2y^2}{3\sigma_p^2}\right) \right) \times f(x, y) dx dy$$

$$I_{p4} = \int_0^{\infty} \int_0^{\infty} \frac{1}{4} \left(\frac{1}{3} \exp\left(-\frac{x^2}{2\sigma_p^2}\right) + \exp\left(-\frac{2x^2}{3\sigma_p^2}\right) \right) \times f(x, y) dx dy$$

and $f_{U_k U_j}(x, y)$ is bi-variate Gaussian with mean zero and the covariance matrix Σ_t

$$f(x, y) = \frac{1}{2\pi\sigma^2\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 + y^2 - 2\rho_t xy}{2\sigma^2(1-\rho_t^2)}\right).$$

Also, $\sigma^2 = \frac{1}{N} \|\boldsymbol{\mu}\|_2^2$. The solution to the above integral can easily be found using change of variables to simplify it to the form

$$\int_{-\infty}^0 \int_{-\infty}^0 f_{U_k U_j}(u_k, u_j) du_k du_j = \frac{1}{4} + \frac{\arcsin(\rho_t)}{2\pi}$$

and

$$\int_0^\infty \int_{-\infty}^0 f_{U_k U_j}(u_k, u_j) du_k du_j = \frac{1}{4} - \frac{\arcsin(\rho_t)}{2\pi}.$$

The solution to (8.2) is

$$\begin{aligned} \mathbb{E}\{\tilde{a}_{pt}^0\} &= \mathbb{E}\{\tilde{a}_{pt}^1\} = 2(I_{p1} + I_{p2} - I_{p3} - I_{p4}) + \frac{1}{4} + \frac{1}{2\pi} \sin^{-1} \rho_t \\ &= \frac{\sigma_p^2 \sqrt{1 - \rho_t^2}}{2\pi} \left[\frac{1}{4} \left(\frac{\sin^{-1} \rho_{t1}}{9\sqrt{1 - \rho_{t1}^2}} + \frac{2\sin^{-1} \rho_{t2}}{\sqrt{1 - \rho_{t2}^2}} + \frac{3\sin^{-1} \rho_{t3}}{\sqrt{1 - \rho_{t3}^2}} \right) \right. \\ &\quad \left. - \left(\frac{\sin^{-1} \rho_{t4}}{3\sqrt{1 - \rho_{t4}^2}} + \frac{\sin^{-1} \rho_{t5}}{\sqrt{1 - \rho_{t5}^2}} \right) \right] + \frac{1}{4} + \frac{1}{2\pi} \sin^{-1} \rho_t \quad (8.3) \end{aligned}$$

Here,

$$\begin{aligned} \rho_{pt1} &= \frac{\rho_t \sigma_p^2}{\sigma^2(1 - \rho_t^2) + \sigma_p^2}; \\ \rho_{pt2} &= \frac{\rho_t \sigma_p^2}{\sqrt{\sigma^2(1 - \rho_t^2) + \sigma_p^2} \sqrt{\frac{4}{3}\sigma^2(1 - \rho_t^2) + \sigma_p^2}}; \\ \rho_{pt3} &= \frac{\rho_t \sigma_p^2}{\frac{4}{3}\sigma^2(1 - \rho_t^2) + \sigma_p^2}; \rho_{pt4} = \frac{\rho_t \sigma_p}{\sqrt{\sigma^2(1 - \rho_t^2) + \sigma_p^2}}; \end{aligned}$$

and

$$\rho_{pt5} = \frac{\rho_t \sigma_p}{\sqrt{\frac{4}{3}\sigma^2(1 - \rho_t^2) + \sigma_p^2}}.$$

If we assume the noise variances to be independent of p , we have the above solution for $\bar{a}_{pt,2}(\gamma, K) = \mathbb{E}\{\tilde{a}_{pt}^0\} + \mathbb{E}\{\tilde{a}_{pt}^1\}$ independent of p . With $a_{pt,2}(\gamma, K)$, we upper bound P_{err} in (2.6). Let $\max_{0 \leq t \leq K-1} \bar{a}_{pt,2}(\gamma, K) = a_K(\gamma)$ where $0 < a_K < 1$. Then P_{err} with one-bit quanti-

zation can be upper bounded as,

$$\begin{aligned}
P_{err} &\leq \frac{1}{2} \sum_{t=0}^{K-1} \binom{K}{t} \binom{N-K}{K-t} \prod_{p=1}^P \left(\max_{1 \leq t \leq K} \bar{a}_{pt,2}(\gamma, K) \right)^M \\
&= \frac{1}{2} (a_K(\gamma))^{MP} \left[\sum_{t=0}^K \binom{K}{t} \binom{N-K}{K-t} - 1 \right] \\
&= \frac{1}{2} (a_K(\gamma))^{MP} \left[\binom{N}{K} - 1 \right] < \frac{1}{2} (a_K(\gamma))^{MP} \binom{N}{K}
\end{aligned} \tag{8.4}$$

Thus, to have a vanishing probability of error it is required that, $MP \geq C_K K \log \frac{N}{K}$, where

$C_K = \frac{1}{\log \frac{1}{a_K(\gamma)}}$ only depends on K .

B Derivation of (3.16)

$$\begin{aligned} PI_0(v, \hat{p}, \tau^p) &= \int \Phi(x/\sqrt{v})\mathcal{N}(x; \hat{p}, \tau^p)dx \\ &= \int_{-\infty}^{\infty} \left[\int_{-\infty}^x N(t|0, v)dt \right] \mathcal{N}(x; \hat{p}, \tau^p)dx \end{aligned}$$

Using change of variable as $u = t - x + \hat{p}$ and $w = x - \hat{p}$, and changing the order of the integration, we get

$$\begin{aligned} PI_0(v, \hat{p}, \tau^p) &= \frac{1}{2\pi\sqrt{v\tau^p}} \int_{-\infty}^{\tau^p} \int_{-\infty}^{\infty} \exp \left\{ - \left[\frac{(u+w)^2}{2v} + \frac{u^2}{2\tau^p} \right] \right\} dw du \\ &= \frac{1}{2\pi\sqrt{v\tau^p}} \\ &\quad \int_{-\infty}^{\tau^p} \int_{-\infty}^{\infty} \exp \left\{ - \frac{1}{2} \begin{bmatrix} w \\ u \end{bmatrix}^T \begin{bmatrix} \frac{1}{2v} + \frac{1}{2\tau^p} & \frac{1}{2v\tau^p} \\ \frac{1}{2v\tau^p} & \frac{1}{2\tau^p} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \right\} dw du \\ &= \int_{-\infty}^{\tau^p} \int_{-\infty}^{\infty} \mathcal{N} \left(\begin{bmatrix} w \\ u \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \tau^p & -\tau^p \\ -\tau^p & v + \tau^p \end{bmatrix} \right) dw du \\ &= \int_{-\infty}^{\mu} \mathcal{N}(u; 0, v + \tau^p) du \\ &= \Phi \left(\frac{\hat{p}}{\sqrt{v + \tau^p}} \right) \end{aligned}$$

The above expression represents marginalization of w in the bi-variate normal density which is followed by the integration over $(-\infty, \tau^p]$. From the property of bivariate Gaussian distribution, marginalization of the bivariate normal density results in normal distribution. The mean and variance can be shown to be zero and $v + \tau$. Next, consider the equality

$$\int \Phi(x/\sqrt{v})\mathcal{N}(x; \hat{p}, \tau^p)dx = \Phi \left(\frac{\hat{p}}{\sqrt{v + \tau^p}} \right) \quad (8.5)$$

Differentiating both side of (8.5) with respect to \hat{p} , we get,

$$\int \frac{x - \hat{p}}{\tau^p} \Phi(x/\sqrt{v}) \mathcal{N}(x; \hat{p}, \tau^p) dx = \frac{\mathcal{N}(\frac{\hat{p}}{\sqrt{v+\tau^p}})}{\sqrt{v + \tau^p}}$$

$$PI_1 = \hat{p} \Phi\left(\frac{\hat{p}}{\sqrt{v + \tau^p}}\right) + \tau^p \frac{\mathcal{N}(\frac{\hat{p}}{\sqrt{v+\tau^p}})}{\sqrt{v + \tau^p}}$$
(8.6)

Finally, differentiating both side of (8.6) with respect to \hat{p} , we get

$$\int x \frac{x - \hat{p}}{\tau^p} \Phi(x/\sqrt{v}) \mathcal{N}(x; \hat{p}, \tau^p)$$

$$= \Phi\left(\frac{\hat{p}}{\sqrt{v + \tau^p}}\right) + \frac{\hat{p} \mathcal{N}(\frac{\hat{p}}{\sqrt{v+\tau^p}})}{\sqrt{v + \tau^p}} + \tau^p \hat{p} \frac{\mathcal{N}(\frac{\hat{p}}{\sqrt{v+\tau^p}})}{(v + \tau^p)^{1.5}}$$

$$\implies PI_2 = \tau^p \Phi\left(\frac{\hat{p}}{\sqrt{v + \tau^p}}\right) + \hat{p} PI_1 + \frac{\tau^p \hat{p} v \mathcal{N}(\frac{\hat{p}}{\sqrt{v+\tau^p}})}{(v + \tau^p)^{1.5}}$$

C Derivation of Result 3.1:

Let $P_{\mathcal{X}}(x) \propto \mathcal{N}(x; m_G, v_G) \times \mathcal{L}(x; m_L, v_L)$ be a probability density function. We compute mean of x as

$$\begin{aligned} \mathbb{E}[x] &= \int x P_{\mathcal{X}}(x) dx = \frac{1}{Z} \int x \mathcal{N}(x; m_G, v_G) \times \mathcal{L}(x; m_L, v_L) \\ &= \frac{1}{Z} \left[\int_{-\infty}^{m_L} x \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) dx + \int_{m_L}^{\infty} x \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) dx \right] \end{aligned}$$

After some algebraic steps,

$$\begin{aligned} \frac{1}{Z} \left[\int_{-\infty}^{m_L} x \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) \right] &= \frac{C_1}{Z} \int_{-\infty}^{m_L} x \mathcal{N}(x; m_G + \frac{v_G}{2v_L}, v_G) dx \\ &= \frac{C_1}{Z} I_1(m_L; m_G + \frac{v_G}{2v_L}, v_G), \end{aligned}$$

where $C_1 = \frac{1}{4v_L} \exp(-\frac{1}{2v_L}(m_L - m_G - \frac{v_G}{4v_L}))$. Similarly,

$$\begin{aligned} \frac{1}{Z} \left[\int_{-m_L}^{\infty} x \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) \right] &= \frac{C_2}{Z} \int_{-m_L}^{\infty} x \mathcal{N}(x; m_G - \frac{v_G}{2v_L}, v_G) dx \\ &= \frac{C_2}{Z} \left(m_G - \frac{v_G}{2v_L} - I_1\left(m_L; m_G - \frac{v_G}{2v_L}, v_G\right) \right), \end{aligned}$$

where $C_2 = \frac{1}{4v_L} \exp(-\frac{1}{2v_L}(-m_L + m_G - \frac{v_G}{4v_L}))$. The normalization constant, Z , can be evaluated as

$$\begin{aligned} Z &= \int \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) dx \\ &= C_1 \int_{-\infty}^{m_L} \mathcal{N}(x; m_G + \frac{v_G}{2v_L}, v_G) + C_2 \int_{m_L}^{\infty} \mathcal{N}(x; m_G - \frac{v_G}{2v_L}, v_G) \quad (8.7) \\ &= C_1 \Phi\left(\frac{m_L - (m_G + \frac{v_G}{2v_L})}{\sqrt{v_G}}\right) + C_2 \left(\Phi\left(-\frac{m_L - (m_G - \frac{v_G}{2v_L})}{\sqrt{v_G}}\right)\right) \end{aligned}$$

Next, we compute $\mathbb{E}[x^2]$ using the definition

$$\begin{aligned}
\mathbb{E}[x^2] &= \int x^2 \mathcal{N}(x; m_G, v_G) \mathcal{L}(x; m_L, v_L) dx \\
&= C_1 \int_{-\infty}^{m_L} x^2 \mathcal{N}(x; m_G + \frac{v_G}{2v_L}, v_G) + C_2 \int_{-m_L}^{\infty} x^2 \mathcal{N}(x; m_G - \frac{v_G}{2v_L}, v_G) \\
&= \frac{C_1}{Z} I_2(m_L; m_G + \frac{v_G}{2v_L}, v_G) + \frac{C_2}{Z} \left(v_G + \left(m_G - \frac{v_G}{2v_L} \right)^2 - I_2(m_L; m_G - \frac{v_G}{2v_L}, v_G) \right).
\end{aligned} \tag{8.8}$$

Using these results, we derive $\mathbb{E}_{\mathcal{X}_n | \mathbf{y}, \tilde{\mathcal{X}}} [x_n | \mathbf{y}, \tilde{x}_n, \hat{r}_n, \tau_n^r]$, and $\mathbb{E}_{\mathcal{X}_n | \mathbf{y}, \tilde{\mathcal{X}}} [x_n^2 | \mathbf{y}, \tilde{x}_n, \hat{r}_n, \tau_n^r]$. Note that, the receiver has access to side-information which is assumed to be the actual signal corrupted by Laplacian noise.

$$\hat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad \tilde{\mathcal{X}}_n = \mathcal{X}_n + \mathcal{W}_n$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, $\mathcal{W}_n \sim \mathcal{N}(0, v_s)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent. The GAMP algorithm approximates the marginal posterior distribution as

$$p_{\mathcal{X}_n | \mathbf{y}}(x_n | \mathbf{y}; \hat{r}_n, \tau_n^r) = \frac{p_{\mathcal{X}_n}(x_n) \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{L}(x_n; \tilde{x}_n, 2v_s)}{\int_{x_n} p_{\mathcal{X}_n}(x_n) \mathcal{N}(x; \hat{r}_n, \tau_n^r) \mathcal{L}(x_n; \tilde{x}_n, 2v_s)}.$$

Normalization Constant:

$$\begin{aligned}
Z_n^l &= \int_{-\infty}^{\tilde{x}_n} \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) p_{\mathcal{X}_n}(x_n) \frac{1}{4v_s} \exp\left(-\frac{|x_n - \tilde{x}_n|}{2v_s}\right) dx \\
&= \int_{-\infty}^{\infty} \frac{1-\lambda}{4v_s} \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \exp\left(\frac{|x_n - \tilde{x}_n|}{2v_s}\right) \delta(x_n) dx_n \\
&\quad + \int_{-\infty}^{\infty} \frac{\lambda}{4v_s} \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x_n; 0, v_x) \exp\left(-\frac{|x_n - \tilde{x}_n|}{2v_s}\right) dx_n
\end{aligned}$$

Using Gaussian product rule, $\mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x_n; 0, v_x) = \mathcal{N}(0; \hat{r}_n, \tau_n^r + v_x) \mathcal{N}(x; \frac{v_x \hat{r}_n}{v_x + \tau_n^r}, \frac{v_x \tau_n^r}{v_x + \tau_n^r})$,

and (8.7), we get

$$\begin{aligned}
&= \frac{1 - \lambda}{4v_s\sqrt{2\pi\tau_n^r}} \exp\left(-\frac{\hat{r}_n^2}{2\tau_n^r} - \frac{|\tilde{x}_n|}{2v_s}\right) + \lambda\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \\
&\left(C_{1,n}\Phi\left(\frac{m_n^l - (m_n^g + \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right) + C_{2,n}\left(1 - \Phi\left(\frac{m_n^l - (m_n^g - \frac{v_n^g}{2v_n^l})}{\sqrt{v_n^g}}\right)\right)\right)
\end{aligned}$$

where $m_n^g = \frac{v_x\hat{r}}{v_x + \tau_n^r}$, $v_n^g = \frac{v_x\tau_n^r}{v_x + \tau_n^r}$, $m_n^l = \tilde{x}_n$, $v_n^l = v_s$. $C_{1,n}$ and $C_{2,n}$ depend on parameters m_n^g, v_n^g, m_n^l , and v_n^l .

Derivation of $\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{X}_n}[x_n|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{X}_n}[x_n|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] \\
&= \frac{1}{Z_n^l} \int x_n \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x_n; 0, v_x) \mathcal{L}(x_n; \tilde{x}_n, 2v_s) dx_n \\
&= \frac{\lambda\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(C_{1,n} I_1\left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g\right) + C_{2,n} \left(m_n^g - \frac{v_n^g}{2v_n^l} - I_1\left(m_n^l; m_n^g - \frac{v_n^g}{2v_n^l}, v_n^g\right)\right) \right)
\end{aligned}$$

Derivation of $\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{X}_n}[x_n^2|\mathbf{y}, \tilde{x}_n, \hat{r}_n, \tau_n^r]$:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{X}_n|\mathbf{y}, \tilde{X}_n}[x_n^2|\mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] \\
&= \frac{\lambda}{Z_n^l} \int x_n^2 \mathcal{N}(x|\hat{r}, \tau_n^r) \mathcal{N}(x; 0, v_x) \mathcal{L}(x_n; \tilde{x}_n, 2v_s) dx_n \\
&= \frac{\lambda\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r)}{Z_n^l} \left(C_{1,n} I_2\left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g\right) + C_{2,n} \left(\left(m_n^g - \frac{v_n^g}{2v_n^l}\right)^2 + v_n^g - I_2\left(m_n^l; m_n^g - \frac{v_n^g}{2v_n^l}, v_n^g\right) \right) \right)
\end{aligned}$$

D Derivation of Result 3.3

The receiver has access to side-information which is assumed to be the actual signal corrupted by Gaussian noise.

$$\widehat{\mathcal{R}}_n = \mathcal{X}_n + \mathcal{V}_n, \quad \widetilde{\mathcal{X}}_n = \mathcal{X}_n + \mathcal{W}_n$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, $\mathcal{W}_n \sim \mathcal{N}(0, v_s)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent.

The GAMP algorithm approximates the marginal posterior distribution as

$$p_{\mathcal{X}_n|\mathcal{Y}}(x_n|\mathbf{y}; \widehat{r}_n, \tau_n^r) = \frac{p_{\mathcal{X}_n}(x_n)\mathcal{N}(x_n; \widehat{r}_n, \tau_n^r)\mathcal{N}(x_n; \widetilde{x}_n, v_s)}{\int_{x_n} p_{\mathcal{X}_n}(x_n)\mathcal{N}(x; \widehat{r}_n, \tau_n^r)\mathcal{N}(x_n; \widetilde{x}_n, v_s)}.$$

Next, we derive the posterior mean and variance.

Derivation of $\mathbb{E}_{\mathcal{X}_n|\mathcal{Y}, \widetilde{\mathcal{X}}_n}[x_n|\mathbf{y}, \widetilde{x}_n; \widehat{r}_n, \tau_n^r]$

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_n|\mathcal{Y}, \widetilde{\mathcal{X}}_n}[x_n|\mathbf{y}, \widetilde{x}_n; \widehat{r}_n, \tau_n^r] &= \frac{1}{Z_n^g} \int x_n \mathcal{N}(x_n|\widehat{r}_n, \tau_n^r) \mathcal{N}(x_n|0, v_x) \mathcal{N}(x_n; \widetilde{x}_n, v_s) dx_n \\ &= \frac{\lambda \phi\left(\frac{\widehat{r}_n}{\sqrt{\tau_n^r + v_x}}\right)}{Z_n^g} \int x_n \mathcal{N}\left(x_n; \frac{\widehat{r}_n v_x}{v_x + \widehat{r}_n}, \frac{v_x \tau_n^r}{v_x + \tau_n^r}\right) \mathcal{N}(x_n; \widetilde{x}_n, v_s) dx_n \\ &= \frac{\lambda \phi\left(\frac{\widehat{r}_n}{\sqrt{\tau_n^r + v_x}}\right)}{Z_n^g} \mathcal{N}\left(x_n; \frac{\widehat{r}_n v_x}{v_x + \widehat{r}_n}, \frac{v_x \tau_n^r}{v_x + \tau_n^r} + v_s\right) \\ &\quad \int x \mathcal{N}\left(x_n; \frac{\widehat{r}_n v_x v_s + v_x \tau_n^r \widetilde{x}_n}{v_x \tau_n^r + v_x v_s + v_s \tau_n^r}, \frac{v_s v_x \tau_n^r}{v_x \tau_n^r + v_x v_s + v_s \tau_n^r}\right) dx_n \\ &= \frac{\lambda \phi\left(\frac{\widehat{r}_n}{\sqrt{\tau_n^r + v_x}}\right)}{Z_n^g} \mathcal{N}\left(0; \frac{\widehat{r}_n v_x}{v_x + \widehat{r}_n}, \frac{v_x \tau_n^r}{v_x + \tau_n^r} + v_s\right) \frac{\widehat{r}_n v_x v_s + v_x \tau_n^r \widetilde{x}_n}{v_x \tau_n^r + v_x v_s + v_s \tau_n^r}, \end{aligned} \tag{8.9}$$

where Z_n^g is the normalization constant. The normalization constant is evaluated as

$$\begin{aligned}
Z_n^g &= \int p_{\mathcal{X}_n}(x_n) \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x; \tilde{x}_n, v_x) dx \\
&= \int (1 - \lambda) \mathcal{N}(x_n; \tilde{x}_n, \tau_n^r) \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \delta(x) + \\
&\quad \lambda \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x; \tilde{x}_n, v_x) \mathcal{N}(x_n; 0, v_x) dx \\
&= (1 - \lambda) \mathcal{N}(0; \tilde{x}_n, \tau_n^r) \mathcal{N}(0; \hat{r}_n, \tau_n^r) + \lambda \mathcal{N}\left(0; \frac{\hat{r}_n v_x}{v_x + \hat{r}_n}, \frac{v_x \tau_n^r}{v_x + \tau_n^r} + v_s\right) \phi\left(\frac{\hat{r}_n}{\sqrt{\tau_n^r + v_x}}\right)
\end{aligned} \tag{8.10}$$

Replacing (8.10) in (8.9), and with some algebraic steps, we can show that

$$\mathbb{E}_{\mathcal{X}_n | \mathbf{y}, \tilde{\mathcal{X}}} [x_n | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] = \pi_n^g \frac{\hat{r}_n v_s v_x + v_x \tau_n^r \tilde{x}}{v_x \tau_n^r + \tau_n^r v_s + v_s v_x} \triangleq \pi_n^g m_n^g$$

where $\pi_n^g = \frac{\lambda}{\lambda + (1-\lambda)Z_n}$ and $Z_n = \frac{\mathcal{N}(0; \hat{x}, \tau_n^r) \mathcal{N}(0; \hat{x}, \tau_n^r)}{\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \mathcal{N}(0; \frac{\hat{r}_n v_x}{v_x + \tau_n^r} - \tilde{x}, \frac{\tau_n^r v_x}{v_x + \tau_n^r} + v_s)}$.

Derivation of $\mathbb{E}_{\mathcal{X}_n | \mathbf{y}, \tilde{\mathcal{X}}} [x_n^2 | \mathbf{y}, \tilde{x}_n, \hat{r}_n, \tau_n^r]$

Using the definition of second-order moment and following the similar algebraic steps we can write

$$\begin{aligned}
&\mathbb{E}_{\mathcal{X}_n | \mathbf{y}, \tilde{\mathcal{X}}} [x_n^2 | \mathbf{y}, \tilde{x}_n, \hat{r}_n, \tau_n^r] \\
&= \frac{1}{Z_n^g} \int x_n^2 \mathcal{N}(x_n; \hat{r}_n, \tau_n^r) \mathcal{N}(x_n; 0, v_x) \mathcal{N}(x_n; \tilde{x}_n, v_s) dx_n \\
&= \pi_n^g \left(\frac{v_s v_x \tau_n^r}{v_x \tau_n^r + \tau_n^r v_s + v_s v_x} + (m_n^g)^2 \right)
\end{aligned} \tag{8.11}$$

Using the posterior first-order and second-order moments, the posterior variance can be

expressed as

$$\begin{aligned} \text{var}_{\mathcal{X}_n|\mathcal{Y},\tilde{X}_n}[x_n|\mathbf{y},\tilde{x}_n;\hat{r}_n,\tau_n^r] \\ = \pi_n^g \frac{v_s v_x \tau_n^r}{v_x \tau_n^r + \tau_n^r v_s + v_s v_x} + \pi_n^g (1 - \pi_n^g) (m_n^g)^2 \end{aligned}$$

E Evaluation of Result 4.1

At every iteration, the algorithm has access to $\widehat{\mathcal{R}}_n = \widehat{r}_n$, and $\widetilde{\mathcal{X}} = \widetilde{x}$ which are defined as

$$\widehat{\mathcal{R}}_n = \mathcal{X}_{cn} + \mathcal{V}_n, \quad \widetilde{\mathcal{X}}_n = \mathcal{X}_{pn} + \mathcal{W}_n$$

where the additive noises $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, and $\mathcal{W}_n \sim \mathcal{L}(0, 2v_s)$ are independent random variables. Remember \mathcal{X}_{cn} and \mathcal{X}_{pn} are random variables with joint distribution $p_{\mathcal{X}_{cn}, \mathcal{X}_{pn}}(\mathbf{x}_{pn}, \mathbf{x}_{cn})$ in (4.1). The posterior joint distribution can be approximated as

$$p_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \widetilde{\mathcal{X}}_n}(x_{cn}, x_{pn} | \mathbf{y}, \widetilde{x}_n) = \frac{\sum_i \lambda_i p_{\mathcal{X}_{cn}, \mathcal{X}_{pn}}(x_{cn}, x_{pn} | i) p_{\widehat{\mathcal{R}}_n, \widetilde{\mathcal{X}}_n}(\widehat{r}_n, \widetilde{x}_n | x_{cn}, x_{pn}, i)}{\int_{x_n} \sum_i \lambda_i p_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | i}(x_{cn}, x_{pn}) p_{\widehat{\mathcal{R}}_n, \widetilde{\mathcal{X}}_n}(\widehat{r}_n, \widetilde{x}_n | x_{cn}, x_{pn}, i)}$$

Derivation of $\mathbb{E}[x_{cn} | \widehat{r}_n, \widetilde{x}_n]$, and $\mathbb{E}[x_{cn}^2 | \widehat{r}_n, \widetilde{x}_n]$:

Note that $\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \widetilde{\mathcal{X}}_n}[(x_{cn})^q | \widehat{r}_n, \widetilde{x}_n, i] = 0$, and for $i = 1, 2$ and $q = 1, 2$ as $x_{cn} \propto \delta(\cdot)$. Next, we consider events 3 and 4.

Event 3

$x_{pn} \propto \delta(\cdot)$ and $x_{cn} \propto \mathcal{N}(x_{cn}; 0, v_x)$.

$$\widehat{\mathcal{R}}_n = \mathcal{X}_{cn} + \mathcal{V}_n, \quad \widetilde{\mathcal{X}}_n = \mathcal{X}_{pn} + \mathcal{W}_n$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, $\mathcal{W}_n \sim \mathcal{L}(0, 2v_s)$, $\mathcal{X}_{cn} \sim \mathcal{N}(0, v_x)$ and $\mathcal{X}_{pn} \sim \delta(\cdot)$ are independent random variables. For simplicity, define $\widetilde{\mathbb{E}}[x_{cn} | \widehat{r}_n, \widetilde{x}_n, i] \triangleq p(\widehat{r}_n, \widetilde{x}_n | i) \mathbb{E}_{\mathcal{X}_{cn} | \mathcal{Y}, \widetilde{\mathcal{X}}_n}[x_{cn} | \mathbf{y}, \widetilde{x}_n, i; \widehat{r}_n, \tau_n^r]$.

$$\begin{aligned}
\tilde{\mathbb{E}}[x_{cn}|\hat{r}_n, \tilde{x}_n, i = 3] &= p(\hat{r}_n, \tilde{x}_n|i = 3) \int x_{cn} p_{\mathcal{X}_{cn}|\mathcal{Y}, \tilde{x}_n}(x_{cn}|\mathbf{y}, \tilde{x}_n, i = 3; \hat{r}_n, \tau_n^r) \\
&= \int x_{cn} p_{\mathcal{X}_{cn}}(x_{cn}) p(\hat{r}_n|x_{cn}, i = 3) p(\tilde{x}_n|x_{cn}, i = 3) \\
&= \mathcal{L}(0; \tilde{x}_n, 2v_s) \int x_{cn} \mathcal{N}(x_{cn}; 0, v_x) \mathcal{N}(x_{cn}; \hat{r}_n, \tau_n^r) \\
&= \mathcal{L}(0; \tilde{x}_n, 2v_s) \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \frac{v_x \hat{r}_n}{v_x + \tau_n^r}
\end{aligned}$$

Similarly,

$$\begin{aligned}
\tilde{\mathbb{E}}[x_{cn}^2|\hat{r}_n, \tilde{x}_n, i = 3] &= p(\hat{r}_n, \tilde{x}_n|i = 3) \int (x_{cn})^2 p_{\mathcal{X}_{cn}|\hat{R}_n, \tilde{x}_n}(x_{cn}|\hat{r}_n, \tilde{x}_n, i = 3) \\
&= \mathcal{L}(0; \tilde{x}_n, 2v_s) \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \left(\frac{v_x \tau_n^r}{v_x + \tau_n^r} + \left(\frac{v_x \hat{r}_n}{v_x + \tau_n^r} \right)^2 \right).
\end{aligned}$$

Let Z_n^l be the normalization constant for n -th index. It is evaluated as

$$\begin{aligned}
Z_n^l &= \int p_{\mathcal{X}_{cn}, \mathcal{X}_{pn}|\mathcal{Y}, \tilde{x}_n}(x_{cn}, x_{pn}|\mathbf{y}, \tilde{x}_n, ; \hat{r}_n, \tau_n^r) \\
&= \lambda_1 \mathcal{L}(0; \tilde{x}_n, 2v_s) \mathcal{N}(0; \hat{r}_n, \tau_n^r) + \lambda_3 \tilde{\phi} \mathcal{L}(0; \tilde{x}_n, 2v_s) + \lambda_2 \mathcal{N}(0; \hat{r}_n, \tau_n^r) \int \mathcal{L}(x; \tilde{x}_n, 2v_s) \mathcal{N}(x; 0, v_x) dx \\
&+ \lambda_4 \tilde{\phi} \int \mathcal{N}(x; \frac{\rho \hat{r}_n v_x}{v_x + \tau_n^r}, \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}) \mathcal{L}(x; \tilde{x}_n, 2v_s) dx
\end{aligned}$$

where the integrals can be evaluated in closed-form using results from Appendix C. We avoided expressing here the closed-form expression to avoid extra notations without losing the essence of the task.

Event 4

When the signal is from event 4, we have $x_{pn} \neq 0$ and $x_{cn} = \rho x_{pn} + u$. The algorithm has access to $\widehat{\mathcal{R}}_n = \widehat{r}_n$, and $\widetilde{\mathcal{X}}_n = \widetilde{x}_n$ which are defined as

$$\widehat{\mathcal{R}}_n = \rho \mathcal{X}_{pn} + \mathcal{U}_n + \mathcal{V}_n, \quad \widetilde{\mathcal{X}}_n = \mathcal{X}_{pn} + \mathcal{W}_n$$

where $\mathcal{V}_n \sim \mathcal{N}(0, \tau_n^r)$, $\mathcal{U}_n \sim \mathcal{N}(0, v)$, $\mathcal{W}_n \sim \mathcal{L}(0, 2v_s)$ and $\mathcal{X}_n \sim p_{\mathcal{X}_n}(x_n)$ are independent random variables. Note that \widehat{r}_n and \widetilde{x}_n are dependent through \mathcal{X}_p . Using linearity of expectation,

$$\mathbb{E}_{\mathcal{X}_{cn}|\mathcal{Y}, \widetilde{\mathcal{X}}_n}[x_{cn}|\mathcal{Y}, \widetilde{x}_n; \widehat{r}_n, \tau_n^r] = \rho \mathbb{E}_{\mathcal{X}_{pn}|\mathcal{Y}, \widetilde{\mathcal{X}}_n}[x_{pn}|\mathcal{Y}, \widetilde{x}_n; \widehat{r}_n, \tau_n^r] + \mathbb{E}_{\mathcal{U}_n|\mathcal{Y}, \widetilde{\mathcal{X}}_n}[u_n|\mathcal{Y}, \widetilde{x}_n; \widehat{r}_n, \tau_n^r]$$

We compute the first term as

$$\begin{aligned} \widetilde{\mathbb{E}}[x_{pn}|\widehat{r}_n, \widetilde{x}_n, i=4] &= p(\widehat{r}_n, \widetilde{x}_n|i=4) \int x_{pn} p_{\mathcal{X}_p|\mathcal{Y}, \widetilde{\mathcal{X}}_n}(x_{pn}|\mathcal{Y}, \widetilde{x}_n, i=4; \widehat{r}_n, \tau_n^r) dx_{pn} \\ &= \int \frac{x_{pn}}{\rho} \mathcal{N}(x_{pn}; 0, v_x) \mathcal{N}(x_{pn}; \frac{\widehat{r}_n}{\rho}, \frac{\tau_n^r + v}{\rho^2}) \mathcal{L}(x_{pn}; \widetilde{x}_n, 2v_s) dx_{pn} \\ &= \widetilde{\phi} \int x_{pn} \mathcal{N}(x_{pn}; \frac{\rho \widehat{r}_n v_x}{v_x + \tau_n^r}, \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}) \mathcal{L}(x_{pn}; \widetilde{x}_n, 2v_s) dx_{pn} \\ &= \widetilde{\phi} \left(C_{1n} I_1 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) + C_{2n} \left(m_n^g - \frac{v_n^g}{4v_n^l} - I_1 \left(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g \right) \right) \right) \triangleq J_{1n}, \end{aligned}$$

where $\widetilde{\phi} = \mathcal{N}(0; \widehat{r}_n, v_x + \tau_n^r)$, $m_n^g = \frac{\rho \widehat{r}_n v_x}{v_x + \tau_n^r}$, $v_n^g = \frac{v_x(\tau_n^r + v)}{v_x + \tau_n^r}$, $m_n^l = \widetilde{x}_n$, and $v_n^l = v_s$. Note that, C_{1n} and C_{2n} are the function of m_n^l , v_n^l , m_n^g , and v_n^g . Similarly, we can show the second-order

moment to be

$$\begin{aligned}
\tilde{\mathbb{E}}[(x_{pn})^2 | \hat{r}_n, \tilde{x}_n, i = 4] &= p(\hat{r}_n, \tilde{x}_n | i = 4) \int (x_{pn})^2 p_{\mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathcal{X}}_n}(x_{pn} | \mathbf{y}, \tilde{x}_n, i = 4; \hat{r}_n, \tau_n^r) \\
&= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \\
&\quad \left(C_{1n} I_2 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) + C_{2n} \left(v_n^g + (m_n^g - \frac{v_n^g}{4v_n^l})^2 - I_2(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g) \right) \right) \triangleq J_{2n}.
\end{aligned} \tag{8.12}$$

Second,

$$\begin{aligned}
&\tilde{\mathbb{E}}[u_n | \mathbf{y}, \tilde{x}_n, i = 4; \hat{r}_n, \tau_n^r] \\
&= p(\hat{r}_n, \tilde{x}_n | i = 4) \int u_n p_{\mathcal{U}_n | \mathcal{Y}, \tilde{\mathcal{X}}_n}(u_n | \mathbf{y}, \tilde{x}_n, i = 4; \hat{r}_n, \tau_n^r) du_n \\
&= \int u_n p_{\mathcal{U}_n}(u_n) p(\hat{r}_n, \tilde{x}_n | u_n, i = 4) du_n \\
&= \int u_n p_{\mathcal{U}_n}(u_n) \int p_{\mathcal{X}_n}(x_n) p(\hat{r}_n, \tilde{x}_n | u_n, x, i = 4) dx_n du_n \\
&= \int \frac{u_n p_{\mathcal{U}_n}(u_n)}{\rho} \int \mathcal{N}(x_{pn}; 0, v_x) \mathcal{N}(x_{pn}; \frac{\hat{r}_n - u_n}{\rho}, \frac{\tau_n^r}{\rho^2}) \mathcal{L}(x_{pn}; \tilde{x}_{pn}, 2v_s) dx_n du_n \\
&= \int \frac{u_n p_{\mathcal{U}_n}(u_n) \mathcal{N}(0; \frac{\hat{r}_n - u_n}{\rho}, v_x + \frac{\tau_n^r}{\rho^2})}{\rho} \int \mathcal{N}(x_{pn}; m_{pn}, v_{pn}) \mathcal{L}(x; \tilde{x}_n, 2v_s) dx_n du_n \\
&= \int u_n p_{\mathcal{U}_n}(u_n) \mathcal{N}(u_n; \hat{r}_n, \rho^2 v_x + \tau_n^r) du_n \int \mathcal{N}(x_{pn}; m_{pn}, v_{pn}) \mathcal{L}(x_{pn}; \tilde{x}_n, 2v_s) dx_{pn}
\end{aligned}$$

where $m_{pn} = \frac{\rho v_x (\hat{r}_n - u_n)}{\rho^2 v_x + \tau_n^r}$, and $v_{pn} = \frac{v_x \tau_n^r}{\rho^2 v_x + \tau_n^r}$. Using result of Appendix A,

$$\begin{aligned}
&= \int du_n u_n \mathcal{N}(u_n; 0, v) \mathcal{N}(u_n; \hat{r}_n, \rho^2 v_x + \tau_n^r) \\
&C_{p1n} \Phi\left(\frac{\tilde{x}_n - (m_{pn} + \frac{v_{pn}}{2v_l})}{\sqrt{v_{pn}}}\right) + C_{p2n} \left(1 - \Phi\left(\tilde{x}_n - (m_{pn} - \frac{v_{pn}}{2v_l})\right)\right) \\
&= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \int du_n u_n \mathcal{N}(u_n; \frac{\hat{r}_n v}{v_x + \tau_n^r}, \frac{(\rho^2 v_x + \tau_n^r)v}{v_x + \tau_n^r}) \\
&C_{p1n} \Phi\left(\frac{\tilde{x}_n - (m_{pn} + \frac{v_{pn}}{2v_l})}{\sqrt{v_{pn}}}\right) + C_{p2n} \left(1 - \Phi\left(\tilde{x}_n - (m_{pn} - \frac{v_{pn}}{2v_l})\right)\right),
\end{aligned}$$

where $C_{p1n} = \frac{1}{4v_s} \exp\left(-\frac{1}{2v_s}(\tilde{x}_n - m_{pn} - \frac{v_{pn}}{4v_s})\right)$. Note that m_{pn} and C_{p1n} are the functions of variable u_n . After some algebraic steps, we rearrange the first term as

$$\begin{aligned} & \int du_n u_n \mathcal{N}(u_n; m'_n, v_{1n}) C_{p1n} \Phi\left(\frac{\tilde{x}_n - (m_{pn} + \frac{v_{pn}}{2v_l})}{\sqrt{v_{pn}}}\right) \\ &= c'_1 \int du_n u_n \mathcal{N}(u_n; m_{1n}, v_{1n}) \Phi\left(\frac{u_n - m_{2n}}{\sqrt{v_{2n}}}\right) \\ &= c'_{1n} PI_1(m_{2n}, v_{2n}, m_{1n}, v_{1n}), \end{aligned}$$

where $m_{1n} = m'_n - u_{cn}v_{1n}$, $m'_n = \frac{\hat{r}_n v}{v_x + \tau_n^r}$,

$$m_{2n} = \hat{r}_n - \left(\rho + \frac{\tau_n^r}{\rho v_x}\right) \left(\tilde{x}_n - \frac{v_{pn}}{2v_l}\right),$$

$$v_{1n} = \frac{(\rho^2 v_x + \tau_n^r)v}{v_x + \tau_n^r}, \quad v_{2n} = \left(\rho + \frac{\tau_n^r}{\rho v_x}\right)^2 v_{pn}, \quad u_{cn} = \left(2v_s \left(\rho + \frac{\tau_n^r}{\rho v_x}\right)\right)^{-1}, \text{ and}$$

$c'_{1n} = \frac{1}{4v_s} \exp\left(-u_{cn} \left(\tilde{x}_n - \frac{v_{pn}}{4v_s}\right) \left(\rho + \frac{\tau_n^r}{\rho v_x}\right) - \hat{r}_n\right) \exp\left(-m'_n u_{cn} + \frac{v_{1n} u_{cn}^2}{2}\right)$. The integral can be evaluated in closed-form using (??).

With similar algebraic steps, the second term is rearranged as

$$\begin{aligned} & \int du_n u_n \mathcal{N}(u_n; \hat{r}_n, \rho^2 v_x + \tau_n^r) C_{2n} \left(1 - \Phi\left(\frac{m_n^l - (m_n^g - \frac{v_g}{2v_l^l})}{\sqrt{v_g}}\right)\right) \\ &= c'_{2n} \int du_n u_n \mathcal{N}(u_n; m_{3n}, v_{3n}) \left(1 - \Phi\left(\frac{u_n - m_{4n}}{\sqrt{v_{4n}}}\right)\right) \\ &= c'_{2n} (m_{3n} - PI_1(m_{4n}, v_{4n}, m_{3n}, v_{3n})), \end{aligned}$$

where $m_{3n} = m'_n + u_{cn}v_{1n}$, $m_{4n} = \hat{r}_n - \left(\rho + \frac{\tau_n^r}{\rho v_x}\right) \left(\tilde{x}_n + \frac{v_{pn}}{2v_l}\right)$,

$$v_{3n} = \frac{(\rho^2 v_x + \tau_n^r)v}{v_x + \tau_n^r}, \quad v_{4n} = \left(\rho + \frac{\tau_n^r}{\rho v_x}\right)^2 v_{pn},$$

$$c'_{2n} = \frac{1}{4v_s} \exp\left(-u_{cn} \left(-\tilde{x}_n - \frac{v_{pn}}{4v_s}\right) \left(\rho + \frac{\tau_n^r}{\rho v_x}\right) + \hat{r}_n\right) \exp\left(m'_n u_{cn} + \frac{v_{1n} u_{cn}^2}{2}\right).$$

Thus,

$$\tilde{\mathbb{E}}[u_n | \hat{r}_n, \tilde{x}_n, i = 4] = \tilde{\phi} \left(c'_{1n} PI_1(m_{2n}, v_{2n}, m_{1n}, v_{1n}) + c'_{2n} (m_{3n} - PI_1(m_{3n}, v_{3n}, m_{4n}, v_{4n})) \right)$$

Similarly, we evaluate $\mathbb{E}_{u_n|\mathbf{y}, \tilde{x}_n} [u_n^2 | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r]$ as

$$\begin{aligned} \tilde{\mathbb{E}}[u_n^2 | \hat{r}_n, \tilde{x}_n] &= p(\hat{r}_n, \tilde{x}_n) \int u_n^2 p_{u_n|\mathbf{y}, \tilde{x}_n}(u_n | \hat{r}_n, \tilde{x}_n; \hat{r}_n, \tau_n^r) \\ &= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) (c'_{1n} PI_2(m_{2n}, v_{2n}, m_{1n}, v_{1n}) + c'_{2n} (m_{3n}^2 + v_{3n} - PI_2(m_{4n}, v_{4n}, m_{3n}, v_{3n}))) \end{aligned}$$

Combining all the results, we get

$$\begin{aligned} \tilde{\mathbb{E}}[x_{cn} | \hat{r}_n, \tilde{x}_n] &= \mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r) \left[\rho \left(C_{1n} I_1 \left(m_n^l; m_n^g + \frac{v_n^g}{2v_n^l}, v_n^g \right) \right. \right. \\ &\quad \left. \left. + C_{2n} \left(m_n^g - \frac{v_n^g}{4v_n^l} - I_1 \left(m_n^l; m_n^g - \frac{v_n^g}{4v_n^l}, v_n^g \right) \right) \right) + (c'_{1n} PI_1(m_{2n}, v_{2n}, m_{1n}, v_{1n}) \right. \\ &\quad \left. \left. + c'_{2n} (m_{3n} - PI_1(m_{4n}, v_{4n}, m_{3n}, v_{3n})) \right) \right] \end{aligned}$$

Next, we evaluate $\tilde{\mathbb{E}}[(x_{cn})^2 | \hat{r}_n, \tilde{x}_n, i = 4]$ as

$$\begin{aligned} \tilde{\mathbb{E}}[(x_{cn})^2 | \hat{r}_n, \tilde{x}_n, i = 4] &= \tilde{\mathbb{E}}[(\rho x_{pn} + u_n)^2 | \hat{r}_n, \tilde{x}_n, i = 4] \\ &= \rho^2 \tilde{\mathbb{E}}[(x_{pn})^2 | \hat{r}_n, \tilde{x}_n, i = 4] + \tilde{\mathbb{E}}[u_n^2 | \hat{r}_n, \tilde{x}_n, i = 4] + 2\rho \tilde{\mathbb{E}}[x_{pn} u_n | \hat{r}_n, \tilde{x}_n, i = 4] \end{aligned}$$

Since we already evaluated $\rho^2 \mathbb{E}[(x_{pn})^2 | \hat{r}_n, \tilde{x}_n, i = 4]$, and $\mathbb{E}[u_n^2 | \hat{r}_n, \tilde{x}_n, i = 4]$, we evaluate

$$\begin{aligned} &\tilde{\mathbb{E}}[x_{pn} u_n | \hat{r}_n, \tilde{x}_n, i = 4] \\ &= p(\hat{r}_n, \tilde{x}_n | i = 4) \int x_{pn} u_n p_{x_p, u_n | \mathbf{y}, \tilde{x}_n}(x_{pn}, u_n | \hat{r}_n, \tilde{x}_n, i = 4) \\ &= \int x_{pn} u_n p_{x_p, u}(x_{pn}, u_n | i = 4) p_{x_p, u_n | \mathbf{y}, \tilde{x}_n}(\hat{r}_n, \tilde{x}_n | x_{pn}, u_n, i = 4) \\ &= \int x_{pn} \mathcal{N}(x_{pn}; 0, v_x) \mathcal{L}(x_{pn}; \tilde{x}_n, 2v_s) \int u_n \mathcal{N}(u_n; 0, v) \mathcal{N}(u_n; \hat{r}_n - \rho x_{pn}, \tau_n^r) du_n dx_{pn} \\ &= \frac{1}{\rho} \int x_{pn} \mathcal{N}(x_{pn}; 0, v_x) \mathcal{L}(x_{pn}; \tilde{x}_n, 2v_s) \mathcal{N}\left(x_{pn}; \frac{\hat{r}_n}{\rho}, \frac{\tau_n^r + v}{\rho^2}\right) \left(\frac{(\hat{r}_n - \rho x_{pn})v}{\tau_n^r + v}\right) dx_{pn} \end{aligned}$$

Using the notations of (22) and (23), the expectation takes the form $\tilde{\mathbb{E}}[x_{pn} u_n | \hat{r}_n, \tilde{x}_n] =$

$$J_{1n} \frac{\widehat{r}_n v}{\tau_n^{r+v}} - J_{2n} \frac{\rho v}{\tau_n^{r+v}}.$$

Finally, we obtain (4.13) by combining all these results as

$$\begin{aligned} \mathbb{E}[x_{cn} | \widehat{r}_n, \widetilde{x}_n] &= \frac{1}{Z_n^l} (\lambda_3 \widetilde{\mathbb{E}}[x_{cn} | \widehat{r}_n, \widetilde{x}_n, 3] + \lambda_4 \widetilde{\mathbb{E}}[x_{cn} | \widehat{r}_n, \widetilde{x}_n, 4]), \\ \mathbb{E}[x_{cn}^2 | \widehat{r}_n, \widetilde{x}_n] &= \frac{1}{Z_n^l} (\lambda_3 \widetilde{\mathbb{E}}[x_{cn}^2 | \widehat{r}_n, \widetilde{x}_n, 3] + \lambda_4 \widetilde{\mathbb{E}}[x_{cn}^2 | \widehat{r}_n, \widetilde{x}_n, 4]). \end{aligned} \tag{8.13}$$

F Results for 1bitdynG

Result 8.1. Let $Z_{n,1}^g = \mathcal{N}(0; \tilde{x}_n, v_s)\mathcal{N}(0; \hat{r}_n, \tau_n^r)$,

$$Z_{n,2}^g = \mathcal{N}(0; \hat{r}_n, \tau_n^r)\mathcal{N}(0; \tilde{x}_n, v_s + v_x),$$

$$Z_{n,3}^g = \mathcal{N}(0; \tilde{x}_n, v_s)\mathcal{N}(0; \hat{r}_n, v_x + \tau_n^r),$$

and $Z_{n,4}^g = \mathcal{N}(0; \tilde{x}_n, v_x + v_s)\mathcal{N}(\hat{r}_n; \frac{\rho\tilde{x}_nv_x}{v_x+v_s}, \frac{v_x(v_s+v)}{v_x+v_s} + \tau_n^r)$.

The posterior first-order and second-order moments of the signal given side-information,

$\tilde{\mathbf{x}}$, are

$$\mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n} [x_{cn} | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] = \frac{1}{Z_n^g} \left\{ \lambda_3 Z_{n,3}^g \frac{\hat{r}_n v_x}{v_x + \tau_n^r} + \lambda_4 Z_{n,4}^g \frac{\hat{r}_n v_x (v_s + v) + \rho \tilde{x}_n v_x \tau_n^r}{(v_s + \tau_n^r + v) v_x + v_s \tau_n^r} \right\}$$

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n} [x_{cn}^2 | \mathbf{y}, \tilde{x}_n; \hat{r}_n, \tau_n^r] &= \frac{1}{Z_n^g} \left\{ \lambda_3 Z_{n,3}^g \left(\left(\frac{\hat{r}_n v_x}{v_x + \tau_n^r} \right)^2 + \frac{v_x \tau_n^r}{v_x + \tau_n^r} \right) + \lambda_4 Z_{n,4}^g \left(\rho^2 \left(\frac{\rho v_x v_s \hat{r}_n + \tilde{x}_n v_x (\tau_n^r + v)}{(v_s + \tau_n^r + v) v_x + v_s \tau_n^r} \right) \right. \right. \\ &+ \left. \frac{v_s v_x (\tau_n^r + v)}{(v_s + \tau_n^r + v) v_x + v_s \tau_n^r} \right) + \left(\frac{v (\hat{r}_n (v_x + v_s) - \rho v_x \tilde{x}_n)}{v_x (v_s + \tau_n^r + v) + v_s \tau_n^r} \right)^2 + \frac{v (\rho^2 v_x v_s + \tau_n^r v_s + \tau_n^r v_x)}{v_x (v_s + \tau_n^r + v) + v_s \tau_n^r} \\ &+ \left. 2\rho \frac{(\hat{r}_n m_n' - \rho ((m_n^{g'})^2 + v_n^{g'}))}{\tau_n^r + v} \right\}, \end{aligned}$$

(8.14)

where $Z_n^g = \lambda_1 Z_{n,1}^g + \lambda_2 Z_{n,2}^g + \lambda_3 Z_{n,3}^g + \lambda_4 Z_{n,4}^g$, $m_n^{g'} = \frac{\rho \hat{r}_n v_x v_s + \tilde{x}_n v_s (v + \tau_n^r)}{v_x (v_s + v + \tau_n^r) + \tau_n^r v_s}$, and $v_n^{g'} = \frac{v_s v_x (v + \tau_n^r)}{v_x (v_s + v + \tau_n^r) + \tau_n^r v_s}$.

Since the derivation of the first-order and the second-order moment is similar to that of the proposed algorithm, we omit the derivation. Next, we estimate the side-information parameter v_s using the expectation-maximization (EM) algorithm. Following the steps as in the case of proposed algorithm, we can show that

$$v_s^{k+1} = \frac{1}{2N} \sum_{n=1}^N \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{\mathbf{x}}_n; v_s^k} (x_{pn} - \tilde{x}_n)^2 \quad (8.15)$$

The expectation can be evaluated in closed-form as

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{X}_n; v_s^k} (x_{pn} - \tilde{x}_n)^2 &= \\ \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{X}_n; v_s^k} (x_{pn})^2 - 2\tilde{x}_n \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{X}_n; v_s^k} x_{pn} + \tilde{x}_n^2 \end{aligned} \quad (8.16)$$

While evaluating the expectation, we require to evaluate integral for all four cases: (1) $x_p = 0$, $x_c = 0$, (2) $x_p \neq 0$, $x_c = 0$, (3) $x_p = 0$, $x_c \neq 0$, and (4) $x_p \neq 0$, $x_c \neq 0$. Using notations from Result 1, we can express the expectation as

$$\begin{aligned} \mathbb{E}_{\mathcal{X}_{cn}, \mathcal{X}_{pn} | \mathcal{Y}, \tilde{X}_n; v_s^k} (x_{pn} - \tilde{x}_n)^2 &= \frac{1}{Z_n^g} \left(\lambda_1 Z_{n,1}^g \tilde{x}_n^2 + \lambda_3 Z_{n,3}^g \tilde{x}_n^2 + \lambda_2 Z_{n,2}^g \left(\left(\frac{\hat{r}_n v_s^k v_x + \tilde{x} v_x \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} \right)^2 \right. \right. \\ &+ \left. \frac{v_x v_s^k \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} \right) - 2\tilde{x}_n \frac{\hat{r}_n v_s^k v_x + \tilde{x} v_x \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} + \tilde{x}_n^2) + \lambda_4 Z_{n,4}^g \left(\left(\frac{\hat{r}_n v_s^k v_x + \tilde{x} v_x \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} \right)^2 \right. \\ &+ \left. \frac{v_x v_s^k \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} \right) - 2\tilde{x}_n \frac{\hat{r}_n v_s^k v_x + \tilde{x} v_x \tau_n^r}{v_x v_s^k + v_s^k \tau_n^r + \tau_n^r v_x} + \tilde{x}_n^2) \end{aligned} \quad (8.17)$$

Using (8.17) in (8.15), we can evaluate the estimate of v_s analytically.

G Proof of Theorem 6.1

Proof. Let $\bar{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathcal{S}} \|\mathbf{x}^* - \mathbf{x}\|_2$. At a high level, the main steps of the proof are as follows:

A. We first prove that, for any $\beta > 0$, the following holds with probability at least $1 - 4 \exp(-2\beta^2)$,

$$\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 \leq \|\bar{\mathbf{x}} - \mathbf{x}^*\|_2^2 + \delta + 4\sqrt{\frac{2\pi}{M}} (\mathscr{W}(\mathcal{S}) + \beta), \quad (8.18)$$

where $\mathscr{W}(\mathcal{S})$ is the Gaussian mean width (See [78, Section 1.3] for the definition) of the range \mathcal{S} .

B. Next, we show that there exists a constant $C' > 0$ such that $\mathscr{W}(\mathcal{S})$ satisfies the following for any $r > 0$:

$$\mathscr{W}(\mathcal{S}) \leq 8r\sqrt{k} + C'\sqrt{kd \log(LNw_{\max})}. \quad (8.19)$$

C. Finally, we combine the above two steps to bound $\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2$ using an appropriately choice of β .

Step A: The proof of Step A is based on the concentration of the random function $f(\mathbf{x}) \triangleq \frac{1}{M} \sum_{i=1}^M \mathbf{y}_i \mathbf{A}_i^\top \mathbf{x}$ around its expectation, which is given in [78, Proposition 4.2]. By assumption, $\hat{\mathbf{x}} = G(\hat{\mathbf{g}})$ minimizes the cost function in (6.11) over \mathcal{S} to within additive δ of the optimum.

Thus, we get that

$$\begin{aligned}
-\delta &\leq \min_{\mathbf{x} \in \mathcal{S}} \left(\|\mathbf{x}\|_2^2 - \frac{\sqrt{2\pi}}{M} \mathbf{y}^\top \mathbf{A} \mathbf{x} \right) - \left(\|\hat{\mathbf{x}}\|_2^2 - \frac{\sqrt{2\pi}}{M} \mathbf{y}^\top \mathbf{A} \hat{\mathbf{x}} \right) \\
&\leq \left(\|\bar{\mathbf{x}}\|_2^2 - \frac{\sqrt{2\pi}}{M} \mathbf{y}^\top \mathbf{A} \bar{\mathbf{x}} \right) - \left(\|\hat{\mathbf{x}}\|_2^2 - \frac{\sqrt{2\pi}}{M} \mathbf{y}^\top \mathbf{A} \hat{\mathbf{x}} \right) \\
&\leq \|\bar{\mathbf{x}}\|_2^2 - \|\hat{\mathbf{x}}\|_2^2 + \sqrt{2\pi} f(\hat{\mathbf{x}} - \bar{\mathbf{x}}) \tag{8.20}
\end{aligned}$$

$$\begin{aligned}
&\leq \|\bar{\mathbf{x}}\|_2^2 - \|\hat{\mathbf{x}}\|_2^2 + \sqrt{2\pi} \mathbb{E} \{ f(\hat{\mathbf{x}} - \bar{\mathbf{x}}) \} \\
&\quad + 4\sqrt{\frac{2\pi}{M}} (\mathcal{W}(\mathcal{S}) + \beta) \tag{8.21}
\end{aligned}$$

$$\begin{aligned}
&= \|\bar{\mathbf{x}}\|_2^2 - \|\hat{\mathbf{x}}\|_2^2 + 2(\hat{\mathbf{x}} - \bar{\mathbf{x}})^\top \mathbf{x}^* + 4\sqrt{\frac{2\pi}{M}} (\mathcal{W}(\mathcal{S}) + \beta) \tag{8.22} \\
&= \|\bar{\mathbf{x}} - \mathbf{x}^*\|_2^2 - \|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 + 4\sqrt{\frac{2\pi}{M}} (\mathcal{W}(\mathcal{S}) + \beta),
\end{aligned}$$

with probability at least $1 - 4 \exp(-2\beta^2)$. Also, we use [78, Proposition 4.2] with parameter $t = \frac{4\beta}{\sqrt{M}}$, and [78, Lemma 4.1] to get (8.21) and (8.22), respectively. On rearranging the terms, we get the desired result.

Step B: The input to the generator \mathbf{g} follows a uniform distribution and therefore, we get that $\|\mathbf{g}\|_2 \leq r$. This in turn implies that $\mathcal{S} \subseteq G(\mathcal{B}_r^k)$, where $\mathcal{B}_r^k = \{\mathbf{g} \in \mathbb{R}^k : \|\mathbf{g}\|_2 \leq r\}$ denote a ball of radius r . Next, we construct a $\frac{t}{(LNw_{\max})^d}$ -cover \mathcal{T} of \mathcal{B}_r^k such that its cardinality is upper bounded by $\left(\frac{4r(LNw_{\max})^d}{t}\right)^k$ [90, Section 4.2.1]. Further, we use [11, Lemma 8.5.] to assert that G is $(LNw_{\max})^d$ -Lipschitz. Hence, $G(\mathcal{T})$ is a t -cover of $G(\mathcal{B}_r^k)$, and thus, $G(\mathcal{T})$ is a t -cover of \mathcal{S} . Thus,

$$\|G(\mathcal{T})\|_2 \leq |\mathcal{T}|_2 \leq \left(\frac{4r(LNw_{\max})^d}{t}\right)^k \tag{8.23}$$

Therefore, for any $\mathbf{x} \in \mathcal{S}$, there exists a point $T(\mathbf{x}) = \arg \min_{\mathbf{t} \in G(\mathcal{T})} \|\mathbf{t} - \mathbf{x}\|_2$ such that $\|\mathbf{x} - T(\mathbf{x})\|_2 \leq t$.

Having constructed a finite cover $G(\mathcal{T})$, we next bound the Gaussian mean width of \mathcal{S} . For

any vector $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

$$\mathscr{W}(\mathcal{S}) \triangleq \mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \langle \mathbf{b}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle \right\} \quad (8.24)$$

$$\begin{aligned} &\leq \mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \langle \mathbf{b}, (\mathbf{x}_1 - T(\mathbf{x}_1) + T(\mathbf{x}_2) - \mathbf{x}_2) \rangle \right\} \\ &\quad + \mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \langle \mathbf{b}, (T(\mathbf{x}_1) - T(\mathbf{x}_2)) \rangle \right\}. \end{aligned} \quad (8.25)$$

We further simplify the first term of the inequality as follows:

$$\begin{aligned} &\mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \langle \mathbf{b}, (\mathbf{x}_1 - T(\mathbf{x}_1) + T(\mathbf{x}_2) - \mathbf{x}_2) \rangle \right\} \\ &\leq \mathbb{E} \{ \|\mathbf{b}\|_2 \} \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \|\mathbf{x}_1 - T(\mathbf{x}_1) + T(\mathbf{x}_2) - \mathbf{x}_2\|_2 \end{aligned} \quad (8.26)$$

$$\leq 2\sqrt{s} \sup_{\mathbf{x} \in \mathcal{S}} \|\mathbf{x} - T(\mathbf{x})\|_2 \leq 2t\sqrt{s}. \quad (8.27)$$

Here, (8.26) follows from Cauchy-Schwarz inequality, and (8.27) uses the fact that $\mathbb{E} \{ \|\mathbf{b}\|_2 \} \leq \sqrt{\mathbb{E} \{ \|\mathbf{b}\|_2^2 \}} = \sqrt{k}$. Similarly, simplifying the second term of (8.25),

$$\begin{aligned} &\mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}} \langle \mathbf{b}, (T(\mathbf{x}_1) - T(\mathbf{x}_2)) \rangle \right\} \\ &\leq \mathbb{E} \left\{ \sup_{\mathbf{x}_1, \mathbf{x}_2 \in G(\mathcal{B}_r^k)} \langle \mathbf{b}, (\mathbf{x}_1 - \mathbf{x}_2) \rangle \right\} \end{aligned} \quad (8.28)$$

$$\leq \mathscr{W}(G(\mathcal{T})) \quad (8.29)$$

$$= C' \sqrt{2k \log \left(\frac{4r(Lnw_{\max})^d}{t} \right)}. \quad (8.30)$$

Here, (8.29) follows because $T(\mathbf{x}_1), T(\mathbf{x}_2) \in G(\mathcal{B}_r^k)$, and thus, supremum in (8.29) is over a larger set. Also, (8.30) follows from (8.23) and [78, Section 2.1]. Finally, using (8.25), (8.27),

(8.30), we get the following:

$$\mathcal{W}(\mathcal{S}) \leq 2t\sqrt{k} + C' \sqrt{2k \log \left(\frac{4r(Lnw_{\max})^d}{t} \right)}. \quad (8.31)$$

Finally, we choose $t = 4r$ to complete Step B.

Step C: Combining Steps A and B, we get that with probability at least $1 - 4 \exp(-2\beta^2)$

$$\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 \leq \|\bar{\mathbf{x}} - \mathbf{x}^*\|_2^2 + \delta + 4\sqrt{\frac{2\pi}{m}} \left(8r\sqrt{r} + C' \sqrt{kd \log LNw_{\max}} + \beta \right). \quad (8.32)$$

As given in the statement of the theorem, let the following lower bound on M holds for $C_1 > 64\pi$,

$$M \geq C_1 \epsilon^{-2} k (8r^2 + C' d \log LNw_{\max}) \quad (8.33)$$

$$\geq \frac{C_1}{2\epsilon^2} \left(8r\sqrt{k} + C' \sqrt{kd \log LNw_{\max}} \right)^2 \quad (8.34)$$

If we choose $\beta = C_2 \epsilon \sqrt{M}$ with $C_2 = \frac{1}{4\sqrt{2\pi}} - \sqrt{\frac{2}{C_1}} > 0$,

$$\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 \leq \|\bar{\mathbf{x}} - \mathbf{x}^*\|_2^2 + \epsilon + \delta. \quad (8.35)$$

with probability at least $1 - 4 \exp(-c\epsilon^2 M)$. Finally, we also have

$$\|\bar{\mathbf{x}} - \mathbf{x}^*\|_2^2 = \min_{\substack{\mathbf{g} \in \mathbb{R}^k \\ \|\mathbf{g}\|_\infty \leq r}} \|G(\mathbf{g}) - \mathbf{x}^*\|_2^2 \leq \min_{\substack{\mathbf{g} \in \mathbb{R}^k \\ \|\mathbf{g}\|_2 \leq r}} \|G(\mathbf{g}) - \mathbf{x}^*\|_2^2.$$

Thus, the proof is complete. □

REFERENCES

- [1] A. Ai, A. Lapanowski, Y. Plan, and R. Vershynin, “One-bit compressed sensing with non-gaussian measurements,” *Linear Algebra Appl.*, vol. 441, pp. 222–239, 2014.
- [2] E. G. Allstot, A. Y. Chen, A. M. Dixon, D. Gangopadhyay, H. Mitsuda, and D. J. Allstot, “Compressed sensing of ecg bio-signals using one-bit measurement matrices,” in *IEEE 9th Int. New Circuits Syst. conf.* IEEE, 2011, pp. 213–216.
- [3] S. D. Babacan, R. Molina, and A. K. Katsaggelos, “Bayesian compressive sensing using Laplace priors,” *IEEE Trans. on Image Process.*, vol. 19, no. 1, pp. 53–63, 2010.
- [4] S. Bahmani, P. T. Boufounos, and B. Raj, “Robust 1-bit compressive sensing via gradient support pursuit,” *arXiv preprint arXiv:1304.6627*, 2013.
- [5] R. G. Baraniuk, S. Foucart, D. Needell, Y. Plan, and M. Wootters, “Exponential decay of reconstruction error from binary measurements of sparse signals,” *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3368–3385, 2017.
- [6] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009. [Online]. Available: <http://dx.doi.org/10.1137/080716542>
- [7] R. Berinde, P. Indyk, and M. Ruzic, “Practical near-optimal sparse recovery in the l_1 norm,” in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.* IEEE, 2008, pp. 198–205.

- [8] J. D. Blanchard, M. Cermak, D. Hanle, and Y. Jing, “Greedy algorithms for joint sparse recovery,” *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1694–1704, 2014.
- [9] T. Blumensath and M. E. Davies, “Gradient pursuits,” *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2370–2382, 2008.
- [10] ———, “Iterative hard thresholding for compressed sensing,” *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, 2009.
- [11] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, “Compressed sensing using generative models,” in *ICML*, Aug. 2017, pp. 537–546.
- [12] P. T. Boufounos, “Greedy sparse signal reconstruction from sign measurements,” in *Proc. Asilomar Conf. Signal, Syst., Comput.*, 2009, pp. 1305–1309.
- [13] P. T. Boufounos and R. G. Baraniuk, “1-bit compressive sensing,” in *Proc. Annu. Conf. Inf. Sci. Syst.*, 2008, pp. 16–21.
- [14] 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.
- [15] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.
- [16] A. Chambolle, R. De Vore, N.-Y. Lee, and B. Lucier, “Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal through Wavelet Shrinkage,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 319–335, March 1998.
- [17] A. Charles, M. S. Asif, J. Romberg, and C. Rozell, “Sparsity penalties in dynamical system estimation,” in *Proc. Annu. Conf. Inf. Sci. Syst.* IEEE, 2011, pp. 1–6.

- [18] C.-H. Chen and J.-Y. Wu, "Amplitude-aided 1-bit compressive sensing over noisy wireless sensor networks," *IEEE Wireless Commun. Lett.*, vol. 4, no. 5, pp. 473–476, 2015.
- [19] M.-Y. Chen, F. Renna, and M. R. Rodrigues, "Compressive sensing with side information: How to optimally capture this extra information for gmm signals?" *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2314–2329, 2018.
- [20] M. Chiani, D. Dardari, and M. K. Simon, "New exponential bounds and approximations for the computation of error probability in fading channels," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 840–845, July 2003.
- [21] D.-Q. Dai, L. Shen, Y. Xu, and N. Zhang, "Noisy 1-bit compressive sensing: models and algorithms," *Appl. Comput. Harmon. Anal.*, vol. 40, no. 1, pp. 1–32, 2016.
- [22] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.
- [23] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 445–460, 2010.
- [24] S. Dirksen, H. C. Jung, and H. Rauhut, "One-bit compressed sensing with partial gaussian circulant matrices," *J. IMA Inf. Infer.*, vol. 9, no. 3, pp. 601–626, 2020.
- [25] X. Dong and Y. Zhang, "A map approach for 1-bit compressive sensing in synthetic aperture radar imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 6, pp. 1237–1241, 2015.
- [26] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

- [27] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proc. NAS*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- [28] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.
- [29] Y. C. Eldar, “Generalized sure for exponential families: Applications to regularization,” *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 471–481, 2009.
- [30] M. Figueiredo and R. Nowak, “An EM Algorithm for Wavelet-based Image Restoration,” *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug 2003.
- [31] L. Flodin, V. Gandikota, and A. Mazumdar, “Superset technique for approximate recovery in one-bit compressed sensing,” *arXiv preprint arXiv:1910.13971*, 2019.
- [32] N. Fu, L. Yang, and J. Zhang, “Sub-nyquist 1 bit sampling system for sparse multiband signals,” in *22nd Eur. Signal Process. Conf. IEEE*, 2014, pp. 736–740.
- [33] N. Galatsanos and A. Katsaggelos, “Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation,” *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 322–336, 1992.
- [34] C. Gianelli, L. Xu, J. Li, and P. Stoica, “One-bit compressive sampling with time-varying thresholds for sparse parameter estimation,” in *IEEE Sensor Array Multichannel Signal Process. Wrkshp. (SAM)*. IEEE, 2016, pp. 1–5.
- [35] ———, “One-bit compressive sampling with time-varying thresholds for multiple sinusoids,” in *IEEE Wrkshp. Comput. Adv. Multi-Sensor Adapt. Process.* IEEE, 2017, pp. 1–5.

- [36] G. H. Golub, M. Heath, and G. Wahba, “Generalized cross-validation as a method for choosing a good ridge parameter,” *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.
- [37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [38] ———, “Generative adversarial nets,” Dec. 2014, pp. 2672–2680.
- [39] S. Gopi, P. Netrapalli, P. Jain, and A. Nori, “One-bit compressed sensing: Provable support and vector recovery,” in *Int. Conf. Mach. Learn.* PMLR, 2013, pp. 154–162.
- [40] A. Gupta, R. Nowak, and B. Recht, “Sample complexity for 1-bit compressed sensing and sparse classification,” in *2010 IEEE Int. Symp. Inf. Theory*. IEEE, 2010, pp. 1553–1557.
- [41] J. Haboba, M. Mangia, R. Rovatti, and G. Setti, “An architecture for 1-bit localized compressive sensing with applications to eeg,” in *2011 IEEE Biomed. Circuits Syst. Conf.* IEEE, 2011, pp. 137–140.
- [42] J. Haupt and R. Nowak, “Compressive sampling for signal detection,” in *IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 3. IEEE, 2007, pp. III–1509.
- [43] R. Heckel and H. Bolcskei, “Joint sparsity with different measurement matrices,” in *50th Annu. Allerton Conf. Commun., Control, Comput.* IEEE, 2012, pp. 698–702.
- [44] R. Heckel and P. Hand, “Deep decoder: Concise image representations from untrained non-convolutional networks,” *arXiv preprint arXiv:1810.03982*, 2018.
- [45] S. Huang and T. D. Tran, “1-bit compressive sensing via approximate message passing with built-in parameter estimation,” *arXiv preprint arXiv:2007.07679*, 2020.

- [46] M. A. Iwen, F. Kraahmer, S. Krause-Solberg, and J. Maly, “On recovery guarantees for one-bit compressed sensing on manifolds,” *Discrete Comput. Geom.*, vol. 65, no. 4, pp. 953–998, 2021.
- [47] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, 2013.
- [48] ———, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2082–2102, 2013.
- [49] G. Jagatap and C. Hegde, “Algorithmic guarantees for inverse imaging with untrained network priors,” in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 14 832–14 842.
- [50] G. Joseph, S. Kafle, and P. K. Varshney, “One-bit compressed sensing using generative models,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2020, pp. 3437–3441.
- [51] S. Kafle, T. Wimalajeewa, and P. K. Varshney, “Bayesian sparse signal detection exploiting laplace prior,” 2018. [Online]. Available: <http://sigport.org/3081>
- [52] U. S. Kamilov, A. Bourquard, A. Amini, and M. Unser, “One-bit measurements with adaptive thresholds,” *IEEE Signal Process. Lett.*, vol. 19, no. 10, pp. 607–610, 2012.
- [53] L.-W. Kang and C.-S. Lu, “Distributed compressive video sensing,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2009, pp. 1169–1172.
- [54] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, Dec. 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [55] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

- [56] ———, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, Dec. 2013. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [57] K. Knudson, R. Saab, and R. Ward, “One-bit compressive sensing with norm estimation,” *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2748–2758, 2016.
- [58] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.
- [59] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” <http://yann.lecun.com/exdb/mnist/>, 2010.
- [60] D. Lee, T. Sasaki, T. Yamada, K. Akabane, Y. Yamaguchi, and K. Uehara, “Spectrum sensing for networked system using 1-bit compressed sensing with partial random circulant measurement matrices,” in *IEEE 75th Veh. Technol. Conf.* IEEE, 2012, pp. 1–5.
- [61] C. Li, Y. He, X. Wang, G. Li, and P. K. Varshney, “Distributed detection of sparse stochastic signals via fusion of 1-bit local likelihood ratios,” *IEEE Signal Process. Lett.*, vol. 26, no. 12, pp. 1738–1742, 2019.
- [62] F. Li, J. Fang, H. Li, and L. Huang, “Robust one-bit bayesian compressed sensing with sign-flip errors,” *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 857–861, 2014.
- [63] J. Li, M. M. Naghsh, S. J. Zahabi, and M. Modarres-Hashemi, “Compressive radar sensing via one-bit sampling with time-varying thresholds,” in *Proc. Asilomar Conf. Signal, Syst., Comput.* IEEE, 2016, pp. 1164–1168.
- [64] L. Li, X. Huang, and J. A. Suykens, “Signal recovery for jointly sparse vectors with different sensing matrices,” *Signal Process.*, vol. 108, pp. 451–458, 2015.

- [65] Q. Liu, W. Xu, Y. Dang, and X. Wang, "Compressive classification based on one-bit measurements," in *IEEE/CIC Int. Conf. Commun. China*. IEEE, 2019, pp. 101–105.
- [66] Z. Liu, S. Gomes, A. Tiwari, and J. Scarlett, "Sample complexity bounds for 1-bit compressive sensing and binary stable embeddings with generative priors," *arXiv preprint arXiv:2002.01697*, 2020.
- [67] W. Lu, B. Deng, W. Zhang, J. Wang, L. Zhong, and S. Peng, "Channel feedback based on complex 1-bit bayesian compressed sensing in fdd massive mimo systems," in *Proc. Int. Conf. on Arti. Int., Inf. Process. Cloud Comput.*, 2019, pp. 1–6.
- [68] W. Lu and N. Vaswani, "Regularized modified BPDN for noisy sparse reconstruction with partial erroneous support and signal value knowledge," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 182–196, 2012.
- [69] A. Ma, C. Rush, D. Baron, D. Needell *et al.*, "An approximate message passing framework for side information," *arXiv preprint arXiv:1807.04839*, 2018.
- [70] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *J Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [71] J. Mo, P. Schniter, N. G. Prelcic, and R. W. Heath, "Channel estimation in millimeter wave mimo systems with one-bit quantization," in *Proc. Asilomar Conf. Signal, Syst., Comput.* IEEE, 2014, pp. 957–961.
- [72] J. F. Mota, N. Deligiannis, and M. R. Rodrigues, "Compressed sensing with side information: Geometrical interpretation and performance bounds," in *IEEE Glob. Conf. Signal Inf. Process. (GlobalSIP)*, 2014, pp. 512–516.
- [73] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.

- [74] O. Musa, G. Hannak, and N. Goertz, “Generalized approximate message passing for one-bit compressed sensing with AWGN,” in *IEEE Glob. Conf. Signal Inf. Process.* IEEE, 2016, pp. 1428–1432.
- [75] R. B. Nelsen, *An Introduction to Copulas*. Springer Science & Business Media, 2007.
- [76] P. North and D. Needell, “One-bit compressive sensing with partial support,” in *IEEE Wrkshp. Comput. Adv. Multi-Sensor Adapt. Process.* IEEE, 2015, pp. 349–352.
- [77] Y. Plan and R. Vershynin, “One-Bit Compressed Sensing by Linear Programming,” *Commun. Pure Appl. Math.*, vol. 66, no. 8, pp. 1275–1297, 2013. [Online]. Available: <http://dx.doi.org/10.1002/cpa.21442>
- [78] —, “Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach,” *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 482–494, 2013.
- [79] C. Qing, Q. Yang, B. Cai, B. Pan, and J. Wang, “Superimposed coding based csi feedback using 1-bit compressed sensing,” *IEEE Commun. Lett.*, 2019.
- [80] S. Qiu, X. Wei, and Z. Yang, “Robust one-bit recovery via relu generative networks: Near-optimal statistical rate and global landscape analysis,” in *Int. Conf. Mach. Learn.* PMLR, 2020, pp. 7857–7866.
- [81] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*. IEEE, 2011, pp. 2168–2172.
- [82] Z. Sadeghigol, H. Zayyani, H. Abin, and F. Marvasti, “Multivariate copula spatial dependency in one bit compressed sensing,” *arXiv preprint arXiv:1711.09217*, 2017.
- [83] A. A. Saleh and R. Valenzuela, “A statistical model for indoor multipath propagation,” *IEEE J. Sel. Topics Signal Commun.*, vol. 5, no. 2, pp. 128–137, 1987.

- [84] Y. Seo and N. Y. Yu, "Optimum modulation orders for 1-bit compressively sampled signals in multicarrier transmission," in *IEEE Int. Wrkshp. Signal Process. Adv. Wireless Commun.* IEEE, 2019, pp. 1–5.
- [85] Y. Shen, J. Fang, and H. Li, "One-bit compressive sensing and source localization in wireless sensor networks," in *IEEE China Summit Int. Conf. Signal Inf. Process.* IEEE, 2013, pp. 379–383.
- [86] S. Sra., "Generalized Proximity and Projection with Norms and Mixed-norms." *Technical Report 192, Max Planck Institute for Biological Cybernetics*, May 2010.
- [87] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *EURASIP J. Signal Process.*, vol. 86, no. 3, pp. 589 – 602, 2006.
- [88] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [89] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, 2010.
- [90] R. Vershynin, *High-dimensional probability: An introduction with applications in data science.* Cambridge University Press, 2018.
- [91] X. Wang and J. Liang, "Side information-aided compressed sensing reconstruction via approximate message passing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2014, pp. 3330–3334.
- [92] X. Wang, G. Li, and P. K. Varshney, "Distributed detection of weak signals from one-bit measurements under observation model uncertainties," *IEEE Signal Process. Lett.*, vol. 26, no. 3, pp. 415–419, 2019.

- [93] L. Weizman, Y. C. Eldar, A. Eilam, S. Londner, M. Artzi, and D. B. Bashat, "Fast reference based MRI," in *IEEE 37th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2015, pp. 7486–7489.
- [94] T. Wimalajeewa and P. Varshney, "Performance Bounds for Sparsity Pattern Recovery With Quantized Noisy Random Projections," *IEEE J. Sel. Topics in Signal Process.*, vol. 6, no. 1, pp. 43–57, Feb 2012.
- [95] T. Wimalajeewa and P. K. Varshney, "Sparse signal detection with compressive measurements via partial support set estimation," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 46–60, 2017.
- [96] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
- [97] J. Xiong and Q. Tang, "1-bit compressive data gathering for wireless sensor networks," *J. Sensors*, vol. 2014, 2014.
- [98] D. Xu, Y. Huang, and L. Yang, "Feedback of downlink channel state information based on superimposed coding," *IEEE Commun. Lett.*, vol. 11, no. 3, pp. 240–242, 2007.
- [99] M. Yan, Y. Yang, and S. Osher, "Robust 1-bit compressive sensing using adaptive outlier pursuit," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3868–3875, 2012.
- [100] S. J. Zahabi, M. M. Naghsh, M. Modarres-Hashemi, and J. Li, "One-bit compressive radar sensing in the presence of clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 167–185, 2019.
- [101] H. Zayyani, M. Korke, and F. Marvasti, "Bayesian hypothesis testing for one bit compressed sensing with sensing matrix perturbation," *Scientia Iranica*, vol. 25, no. 6, pp. 3628–3633, 2018.

- [102] H. Zayyani, F. Haddadi, and M. Korki, “Double detector for sparse signal detection from one-bit compressed sensing measurements,” *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1637–1641, 2016.
- [103] H. Zayyani, M. Korki, and F. Marvasti, “Dictionary learning for blind one bit compressed sensing,” *IEEE Signal Process. Lett.*, vol. 23, no. 2, pp. 187–191, 2015.
- [104] —, “A distributed 1-bit compressed sensing algorithm robust to impulsive noise,” *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1132–1135, 2016.
- [105] J. Zhu, L. Han, X. Meng, and Z. Xu, “Binary sparse bayesian learning algorithm for one-bit compressed sensing,” *ArXiv e-prints*, May 2018. [Online]. Available: <https://arxiv.org/abs/1805.03043>
- [106] R. Zhu and Q. Gu, “Towards a lower sample complexity for robust one-bit compressed sensing,” in *Int. Conf. Mach. Learn.* PMLR, 2015, pp. 739–747.
- [107] E. Zimos, J. F. Mota, M. R. Rodrigues, and N. Deligiannis, “Bayesian compressed sensing with heterogeneous side information,” in *Data Compression Conference (DCC), 2016.* IEEE, 2016, pp. 191–200.
- [108] A. Zymnis, S. Boyd, and E. Candes, “Compressed Sensing With Quantized Measurements,” *Signal Process. Lett., IEEE*, vol. 17, no. 2, pp. 149–152, Feb 2010.

VITA

NAME OF AUTHOR: Swatantra Kafle

PLACE OF BIRTH: Gothadi, Palpa, Lumbini, Nepal

DATE OF BIRTH: May 12, 1989

EDUCATION: B.E., Nov 2011 Tribhuvan University, Nepal

PROFESSIONAL EXPERIENCE:

- Graduate Teaching Assistant, Department of Electrical Engineering and Computer Science, Syracuse University, Aug. 2014 - May. 2015
- Graduate Research Assistant, Sensor Fusion Laboratory, Department of Electrical Engineering and Computer Science, Syracuse University, May. 2015 - June. 2021

PUBLICATIONS:

1. Swatantra Kafle, Bhavya Khailkhura, Thakshila Wimalajeewa, and Pramod K. Varshney, "Decentralized joint sparsity pattern recovery using 1-bit compressed sensing," in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.
2. Swatantra Kafle, Vipul Gupta, Bhavya Khailkhura, Thakshila Wimalajeewa, and Pramod K. Varshney, "Joint sparsity pattern recovery with 1-b compressive sensing in distributed sensor networks," in *Proc. IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp.15-30, March 2019.

3. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Generalized approximate message passing for noisy 1-bit compressed sensing with side-information," in *Proc. 52nd IEEE Asilomar Conference on Signals Systems and Computer*, 2018.
4. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy 1-bit compressed sensing with heterogeneous side-information," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
5. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy one-bit compressed sensing with side-information," *IEEE Transactions on Signal Processing*, submitted, 2020.
6. Swatantra Kafle, Thakshila Wimalajeewa, and Pramod K. Varshney, "Noisy one-bit compressed sensing with side-information in Time-varying Environments," *IEEE Transactions on Signal Processing*, submitted, 2021.
7. Swatantra Kafle, Geethu Joseph, and Pramod K. Varshney, "One-bit Compressed Sensing with Untrained Prior", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021 Accepted
8. Swatantra Kafle, Geethu Joseph, and Pramod K. Varshney, "One-bit compressed sensing with Generative models", *IEEE Transactions on Signal Processing*, In preparation, 2021