COLLABORATIVE REGRESSION AND CLASSIFICATION VIA BOOTSTRAPPING

by
Luoluo Liu

A dissertation submitted to Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

Baltimore, Maryland
October, 2019

Abstract

In modern machine learning problems and applications, we deal with vast quantities of data that are often high dimensional, making data analysis time-consuming and computationally inefficient. Sparse recovery algorithms are developed to extract the underlining low dimensional structure from the data. Classical signal recovery based on ℓ_1 minimization solves the least squares problem with all available measurements via sparsity-promoting regularization. It has shown promising performances in regression and classification. Previous work on Compressed Sensing (CS) theory reveals that when the true solution is sparse and if the number of measurements is large enough, then solutions to ℓ_1 minimization converge to the ground truths. In practice, when the number of measurements is low, when the noise level is high, or when measurements arrive sequentially in streaming fashion, conventional ℓ_1 minimization algorithms tend to under-perform.

This research work aims at using multiple local measurements generated from resampling using bootstrap or sub-sampling to efficiently make global predictions to deal with the aforementioned challenging scenarios. We develop two main approaches – one extends the conventional bagging scheme in sparse regression from a fixed bootstrapping ratio whereas the other called JOBS applies

a support consistency among bootstrapped estimators in a collaborative fashion. We first derive rigorous theoretical guarantees for both proposed approaches and then carefully evaluate them with extensive simulations to quantify their performances. Our algorithms are quite robust compared to the conventional ℓ_1 minimization, especially in the scenarios with high measurements noise and low number of measurements. Our theoretical analysis also provides key guidance on how to choose optimal parameters, including bootstrapping ratios and number of collaborative estimates. Finally, we demonstrate that our proposed approaches yield significant performance gains in both sparse regression and classification, which are two crucial problems in the field of signal processing and machine learning.

Primary Reader and Advisor: Prof. Trac D. Tran

Secondary Reader: Prof. Vishal M. Patel

Thesis Committee

Prof. Trac D. Tran (Primary Reader, Advisor)
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Prof. Vishal M. Patel (Second Reader)
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Prof. Carey E. Priebe
Department of Applied Math and Statistics
Johns Hopkins Whiting School of Engineering

Prof. Sang Peter Chin (Advisor)
Department of Computer Science
& Hariri Institute of Computing
Boston University

Acknowledgments

I would like to thank to my advisor, Prof. Trac Tran for guiding me through this challenging journey. His kindness, wisdom, patience, and knowledge have always been greatly appreciated for people who work with him. It is my pleasure to be in his lab and work on interesting and challenging research problems with him.

I would also like to thank my co-advisor, Prof. Peter Chin. He has been very supportive of my work. Both Prof. Tran and Prof. Chin give me freedom in research and also give helpful and informative suggestions, which helps me become an independent and qualified researcher.

I want to thank my thesis committee members Prof. Carey Priebe, Prof. Vishal Patel for being willing to serve in my committee and for their valuable suggestions on my thesis work. In addition, I would like to thank my other two Graduate Board Oral Exam committee members: Prof. Raman Arora and Prof. Alan Yuille.

I would like to thank my collaborators: Ouyang Feng from Applied Physics Laboratory, Jeff Glaister, Aaron Carass, Prof. Jerry Prince from Electrical and Computer Engineering (ECE), Jasper Stroud, Prof. Mark Foster from ECE; Dung Tran and Arun Nair from my own group. It has been a pleasure to work with all these bright people.

I would like to thank my labmates at Digital Signal Processing lab: Shuai Huang, Tao Xiong, Xiaoxia Sun, Akshay Rangamani, Arun Nair, Dung Tran, Sonia Joy. My labmates have all been excellent and dedicated researchers, and it is my pleasure to be their collegiate, and I enjoy my discussions with them.

I would like to thank many professors at Hopkins: Prof. Raman Arora, Prof. Rene Vidal, Prof. Amitabh Basu, Prof. Daniel Robinson, Prof. Donniell Fishkind, Prof. Sanjeev Khudanpur and many more, forgiving me knowledge from their classes or discussions. I obtained necessary tools for my research work as well as gained the ability to learn new knowledge.

Since the numerical results are computed using the cluster service at the Maryland Advanced Research Computing Center (MARCC), I would like to express my thanks for MARCC and also Hopkins for providing such a important and efficient facility for research. Additionally, I would like to thank Dr. Kevin Manalo at MARCC to help optimizing the work flow.

I would like to thank staffs in ECE Department: Debbie Race, Cora Mayenschein, Belinda Blinkoff for their hard work. It is impossible for me (and also other graduates student) to have a joyful and productive time at ECE Department at Hopkins without your hard work.

I would like to thank my friends that I met during my time at Hopkins: Apurva Nakade, Chong You, Cindy Chen, Chin-Fu Liu, Dengrong Jiang, Daisuke Goto, George Betas, Hana Escovar, Kaylynn Sanders, Ke Li, Minhua Wu, Mary Yen, Liu Xu, Raymond Tuazon, Siddharth Mahendran, Sarah Kim, Stephanie Lau, Sen Lin, Shenwen Wang, Shuwen Wei, Ting Da, Paul Gorelick, Percy Li, Xilei Zhao, Xinlei Zhang, Yayun Wang, Zhenhui Liu, Zhiye Li and many more.

It is your friendship that make my life enjoyable at Hopkins. They are currently in different places and various roles: some work in industry; a few work in universities; some in graduate school. I wish all the best to all of you!

I would like to thank to my partner and "writing editor" Nicholas Huang, who has been always supportive during this journey.

Dedication

To my dearest parents- Mr. Wenge Liu and Ms. Qing Ye, who support me unconditionally over the years.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In compressed sensing (CS) and sparse recovery, solutions to the linear inverse problem in the form of least squares plus a sparsity-promoting penalty term have been extensively studied. Formally speaking, a measurements vector $\boldsymbol{y} \in \mathbb{R}^m$ is generated by the model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the sensing matrix, $\boldsymbol{x} \in \mathbb{R}^n$ contains the sparse codes with very few non-zero entries and $\boldsymbol{z}$ is noise vector with bounded energy. The problem of interest is to find the sparse vector $\boldsymbol{x}$ given the sensing matrix $\boldsymbol{A}$ as well as the measurement vector $\boldsymbol{y}$. Among many methods, the most common one is the ℓ_1 minimization, in which the regularization term is sum of the absolute values of the vector.

The performance of ℓ_1 minimization in recovering the true sparse solution has been thoroughly investigated in the CS literature (Cohen, Dahmen, and DeVore, 2009; Candes, 2008; Candes, Romberg, and Tao, 2006; Donoho, 2006; Candess and Romberg, 2007). Pioneer works study the correctness and robustness of ℓ_1 minimization. They establish conditions for successful recovery based on the Null Space Property (NSP) (Cohen, Dahmen, and DeVore, 2009) and the Restricted Isometry Property (RIP) as well as quantify the recovery performance via the

RIP constant (Candes, 2008; Candes, Romberg, and Tao, 2006). Additionally, under mild conditions on random sensing matrices, CS theory reveals that when the true solution is sparse and with enough measurements, then (2.8) recovers the ground truth and the solution to (2.9) is within a controllable neighborhood of the true solution with high probability (Candes, 2008). Unfortunately, in practice, measurements may not be available all at once. Moreover, certain parts of the data might be missing and/or severely corrupted. For example, in common streaming settings, measurements might be available sequentially or in small batches. Waiting for all measurements to be available wastes valuable processing time and buffering memory.

Alternatively, in sparse-representation-based classification, many schemes use local observations and have shown promising performances (Aharon, Elad, and Bruckstein, 2006; Yang et al., 2010; Liu, Tran, and Chin, 2016; Chen, Do, and Tran, 2010; Bosworth et al., 2015b). The proper choices of measurement subsets differ between applications and often require case-by-case treatment. Obviously, prior knowledge should help significantly in the selection process. For example, image datasets may have large variance overall but data remains relatively homogeneous within local regions. Hence, choosing to work with image patches often leads to satisfactory results in dictionary learning and deep learning (Aharon, Elad, and Bruckstein, 2006; Krizhevsky, Sutskever, and Hinton, 2012).

Without any prior information, a natural choice is to sample data uniformly at random with replacement, termed *bootstrap* (Efron, 1979). This simple sampling scheme has been shown to represent the entire system better than

specific predefined choices. It performs reasonably well when all measurements are equally good. In CS theory, many random matrices have been proven to be excellent sensing matrices. These operators act by shuffling and recombining entries of the original data samples, destroying any spatial or temporal structure and making the measurements even more democratic.

To incorporate information from multiple estimates, the *Bagging* (Breiman, 1996) (**B**ootstrap **Agg**regat**ing**) framework has been proposed by Leo Breiman. It is an efficient parallel ensemble method that improves the performance of unstable predictors. The algorithm consists of solving the same objective function multiple times independently from bootstrap samples and then averaging over multiple predictions to obtain the final solution.

Applying Bagging to find a sparse vector with a specific symmetric pattern was shown empirically to reduce estimation error when the sparsity level s is high (Breiman, 1996) in a forward subset selection problem, and for general sparse signals (Liu, Chin, and Tran, 2019). These experiments show the possibility of using Bagging to improve other sparse regression methods on general sparse signals. Although the well-known conventional Bagging method uses the bootstrap ratio 100%, some follow-up works have shown empirically that lower ratios improve Bagging in some classic classifiers: Nearest Neighbour Classifier (Hall and Samworth, 2005), CART Trees (Sabzevari, Martinez-Munoz, and Suarez, 2014), Linear SVM, LDA, and Logistic Linear Classifier (Zaman and Hirose, 2009). Based on this success, we hypothesize that reducing the bootstrap ratio will also improve performance of Bagging in sparse regression. Therefore, we set up the framework with a generic bootstrap ratio and study its

behavior with various bootstrap ratios.

In this thesis, we will demonstrate the generalized Bagging framework with bootstrap ratio L/m and number of estimates K as parameters. An important discovery is that in challenging cases when m is small, Bagging with a ratio L/m that is smaller than the conventional ratio 100% can lead to better performance.

Although Bagging can be applied to sparse regression problems, the solutions obtained using this method may not ultimately be very sparse. Individually solved predictors are not guaranteed to have the same support, and in the worst case, their average can be quite dense – its support size growing up to the number of estimates times the true sparsity level. To alleviate this problem, *Bolasso* (**Bo**otstrapping **Lasso**) has been proposed (Bach, 2008a). Bolasso first recovers the common support using the intersection of all bootstrapped estimators and then estimates the magnitudes by applying least squares on the support. However, this strategy is very aggressive. When the noise level is high, it commonly recovers the extremely sparse or even all-zero solution.

To resolve the support consistency issue in Bagging and avoid the overly aggressive two-step method Bolasso scheme, our second proposed method enforces the row sparsity constraint among all predictors using the $\ell_{1,2}$ norm. The final estimate is obtained by averaging over all estimators. We name this whole procedure JOBS (**J**oint-sparse **O**ptimization from **B**ootstrap **S**amples). The proposed method involves two key parameters: the bootstrap sample size L of random sampling with replacement from the original m measurements and the K number of those bootstrap vectors.

We will show that JOBS consistently and significantly outperforms the

baseline ℓ_1-minimization algorithm in the challenging case when the number of measurements m is limited. Our previous work (Liu, Chin, and Tran, 2019) has shown that Bagging improves the baseline ℓ_1 minimization when the bootstrap ratio L/m is smaller than the conventional full bootstrap sampling rate of 1. An interesting discovery is that the optimal bootstrap ratio JOBS is even lower than that of Bagging for similar optimal performance levels. The row sparsity prior among all estimators helps bring down the optimal bootstrap sampling ratio, and therefore less data is required for JOBS to achieve a similar performance as Bagging.

The main contributions of this thesis are as follows.

(i) We demonstrate that employing the powerful bootstrapping idea, inspired from machine learning, can improve the robustness of sparse recovery in noisy environments through a collaborative recovery scheme via two schemes: Bagging in sparse regression and JOBS. *(ii)* We explore the theoretical properties associated with finite L/m and K for the Bagging algorithm. *(iii)* We provide an in-depth analysis of the proposed JOBS strategy. Since the critical parameters in our method are the bootstrap sample size L and the number of bootstrap measurement vectors K, we derive analytically various error bounds with regards to these parameters. *(iv)* We confirm our optimal parameter settings and validate our theoretical framework via extensive simulations results. *(v)* We extend the theoretical analysis to the Bagging framework, using the same setting with various bootstrap sampling ratios and different number of estimates. *(vi)* We present a natural extension of the framework, employing a sub-sampling variation of the proposed scheme (named Sub-JOBS) as an alternative to bootstrapping, and

discuss the relationship between the sub-sampling variation to bootstrap. (*vii*) We demonstrate that the proposed JOBS recovery also benefits discriminative tasks such as face recognition over the baseline Sparse Representation-based Classification (SRC) framework, in which the conventional ℓ_1 minimization is employed (Wright et al., 2009). (*viii*) We show that a collaborative reconstruction scheme from random subsets is powerful for signal recovery from a compressed sensing hardware.

The outline of this thesis is as follows. Chapter 2 gives some background about this work, including compressed sensing, bootstrapping, bagging, bolasso, etc. Chapter 3 demonstrates the Bagging in Sparse Regression procedure, main theoretical results as well as some simulation results. Chapter 4 illustrates the proposed method JOBS, supported by theoretical performance bounds, simulation results as well as classifications results compared among the classic ℓ_1 minimization, Bagging method described in previous chapter as well as Bolasso. Chapter 5 shows the usage of our proposed method on real data from Compressed Sensing Optical Coherence Tomography (OCT) hardware. We will show that a collaborative scheme on randomly chosen subsets of original measurements can efficiently reconstruct three dimensional OCT images. Finally, Chapter 6 summarizes our proposed frameworks and Chapter 7 gives future directions.

Chapter 2

Background

2.1 Notations

In this section, we introduce some main notations used throughout the thesis. We first give definitions for ℓ_p vector norm and mixed $\ell_{p,q}$ norm for matrix. Then we list other main notations that we use throughout this thesis.

2.1.1 ℓ_p Vector Norm

A vector in n-dimensional vector space is denoted as $\boldsymbol{x} \in \mathbb{R}^n = (x_1, x_2, ..., x_n)^{\boldsymbol{T}}$, where subscripts $1, 2, ..., n$ denote indices and parentheses "()" denote stacking elements column-wise. Since all vectors are defined as arrays with finite number of rows, a generic form of a vector stacks its elements as multiple columns and then takes a transpose, which is represented by symbol $(\cdot)^{\boldsymbol{T}}$.

The ℓ_p norm $p > 0$ of a vector $\boldsymbol{x} \in \mathbb{R}^n = (x_1, x_2, ..., x_n)^{\boldsymbol{T}}$ is defined as:

$$\|\boldsymbol{x}\|_p = (\sum_{i=1}^{n} (x_i)^p)^{1/p}. \tag{2.1}$$

In the case when $p \geq 1$, the ℓ_p norm is a proper vector norm $\rho(\cdot)$ being positive

definite ($\rho(\boldsymbol{x}) \geq 0$, $\rho(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$), subadditive ($\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\rho(\boldsymbol{x}) + \rho(\boldsymbol{y}) \geq \rho(\boldsymbol{x} + \boldsymbol{y})$), and absolutely scalable ($\alpha > 0, \rho(\alpha\boldsymbol{x}) = |\alpha|\rho(\boldsymbol{x})$). When $0 < p < 1$, the ℓ_p norm can still be computed using equation (2.1), however it is not a proper vector norm because it is not subadditive, or in other words, non-convex.

The famous ℓ_1 norm that is commonly used to regress sparse vectors is the sum of the absolute value of each entry:

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|. \tag{2.2}$$

2.1.2 The ℓ_0 sparsity

The sparsity level is defined as counting the number of non-zero entries in a vector $\boldsymbol{x} \in \mathbb{R}^n$, which is:

$$\|\boldsymbol{x}\|_0 := \sum_{i=1}^{n} \mathbb{1}\{|x_i| > 0\}. \tag{2.3}$$

Note that, although we adopt the notation of ℓ_p norm for the sparsity measure (in which case $p = 0$), it is actually not a proper vector norm because it violates the absolutely scalable property. Because of this, the literature often refers to the ℓ_0 norm as a quasi-norm or pseudo-norm.

2.1.3 Mixed $\ell_{p,q}$ norm of a matrix

A matrix with n rows and K columns is denoted as $\boldsymbol{X} \in \mathbb{R}^{n \times K}$. It can be represented as $\boldsymbol{X} = (x_{ij}), i = 1, 2, .., n$ and $j = 1, 2, .., K$, where x_{ij} is the element on the (i, j)-th location. It can also represented as $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_K)$,

where $\boldsymbol{x}_j$ represents the j-th column of matrix $\boldsymbol{X}$. To introduce the mixed $\ell_{p,q}$ norm on a matrix, we also introduce composing a matrix by stacking multiple rows: $\boldsymbol{X} = (\boldsymbol{x}[1]^{\boldsymbol{T}}, \boldsymbol{x}[2]^{\boldsymbol{T}}, .., \boldsymbol{x}[n]^{\boldsymbol{T}})^{\boldsymbol{T}}$, where $\boldsymbol{x}[i]$ denotes the i-th row of matrix $\boldsymbol{X}$.

The mixed $\ell_{p,q}$ norm on matrix $\boldsymbol{X}$ is defined as:

$$\|\boldsymbol{X}\|_{p,q} = (\sum_{i=1}^{n} \|\boldsymbol{x}[i]^{\boldsymbol{T}}\|_q^p)^{1/p} = \|(\|\boldsymbol{x}[1]^{\boldsymbol{T}}\|_q, \|\boldsymbol{x}[2]^{\boldsymbol{T}}\|_q, ..., \|\boldsymbol{x}[n]^{\boldsymbol{T}}\|_q)^{\boldsymbol{T}}\|_p, \tag{2.4}$$

where $\boldsymbol{x}[i]$ denotes the $i-$th row of matrix $\boldsymbol{X}$. Intuitively, the mixed $\ell_{p,q}$ norm essentially takes ℓ_q norms on rows of $\boldsymbol{X}$ first; then stacks those as a vector and then computes its ℓ_p norm. Note when $p = q$, the $\ell_{p,p}$ norm of $\|\boldsymbol{X}\|$ is simply the ℓ_p vector norm of the vectorized $\boldsymbol{X}$. The row sparsity penalty that we employed $\ell_{1,2}$ norm in JOBS is essentially a special case of (2.4) taking $p = 1, q = 2$, which is in the following form:

$$\|\boldsymbol{X}\|_{1,2} = \sum(\|\boldsymbol{x}[1]^{\boldsymbol{T}}\|_2, \|\boldsymbol{x}[2]^{\boldsymbol{T}}\|_2, ..., \|\boldsymbol{x}[n]^{\boldsymbol{T}}\|_2). \tag{2.5}$$

2.1.4 Mixed $\ell_{p,q}$ norm over block partition of a vector

Similarly to the $\ell_{p,q}$ norm on matrix in (2.4), we introduce a more general form: the mixed $\ell_{p,q}$ norm over a block partition of a vector. The definition for $\ell_{p,q}$ norm over block partition $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_b\}$ for a vector $\|\boldsymbol{x}\|_{p,q|\mathcal{B}}$:

$$\|\boldsymbol{x}\|_{p,q|\mathcal{B}} = (\sum_{i=1}^{b} \|\boldsymbol{x}[\mathcal{B}_i]^{\boldsymbol{T}}\|_q^p)^{1/p} = \|(\|\boldsymbol{x}[\mathcal{B}_1]^{\boldsymbol{T}}\|_q, ..., \|\boldsymbol{x}[\mathcal{B}_b]^{\boldsymbol{T}}\|_q)\|_p. \tag{2.6}$$

It is not difficult to see that the $\ell_{p,q}$ norm of a matrix is a special case of $\ell_{p,q}$ norm over block of the vectorized version of that matrix. In fact, the mixed

$\ell_{1,2}$ norm on matrix $\boldsymbol{X}$ can also be expressed as a mixed $\ell_{1,2|\mathcal{B}}$ norm on the vectorized $\boldsymbol{X}$ given $\mathcal{B}$, where the block partition is row-wise. We will give the relationship in Section 4.9.1.

2.1.5 Other Main Notations List

Let matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ denote the original sensing matrix. Let vector $\boldsymbol{y} \in \mathbb{R}^m$ represent the measurement vector. Let $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_K$ be bootstrap samples, each containing L elements. For each bootstrapped sample $\mathcal{I}_j$, the corresponding bootstrapped sensing matrix $\boldsymbol{A}[\mathcal{I}_j]$ of size $L \times n$ and bootstrapped measurements vector $\boldsymbol{y}_{[\mathcal{I}_j]}$ of length L are generated and $\boldsymbol{x}_j \in \mathbb{R}^n$ is a feasible estimator for the j-th bootstrap sample. Concatenating K estimators $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_K$, we obtain the sparse-code matrix $\boldsymbol{X}$ of size $n \times K$. We summarize all relevant variables in Table 2.1.

2.2 Compressed Sensing

In Compressed Sensing, the problem of interest is to find the sparse vector $\boldsymbol{x}$ given the sensing matrix $\boldsymbol{A}$ as well as the measurement vector $\boldsymbol{y}$. Let the pseudo-norm ℓ_0 norm be the sparsity level which counts the number of non-zero entries; the mathematical formulation is as follows:

$$\mathbf{P_1} : \min \|\boldsymbol{x}\|_0 \ \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{2.7}$$

Although minimizing the number of non-zero entries in vector $\boldsymbol{x}$ as in (2.7) is our goal, directly minimizing the sparsity level is proven to be NP-hard (Natarajan, 1995). Instead, a convex regularizer is preferable. Among various choices of

Table 2.1: Notation of Main Variables

m	total number of measurements
n	signal dimension
s	sparsity level
L	size of each bootstrap sample
L/m	bootstrap sampling ratio
K	number of bootstrap samples / the number of estimates
$\boldsymbol{A}$	the original sensing matrix of size $m \times n$
$\boldsymbol{y}$	the original measurements vector of size $m \times 1$
$\mathcal{I}$	a multi-set (it allows duplicate elements) or a set
$\mathcal{I}_j$	the j-th Bootstrap sample, $j = 1, 2, ..., K$, length of $\mathcal{I}_j = L$
$(\cdot)[\mathcal{I}]$	takes rows supported on $\mathcal{I}$ and throws away elements in $\mathcal{I}^c$
$\boldsymbol{A}[\mathcal{I}_j]$	bootstrapped sampling matrix for bootstrap sample $\mathcal{I}_j$
$\boldsymbol{y}_{[\mathcal{I}_j]}$	measurement vector corresponds to bootstrap sample $\mathcal{I}_j$
$\boldsymbol{x}_j$	the j-th column of matrix $\boldsymbol{X}$;
	a feasible solution corresponds to $(\boldsymbol{A}[\mathcal{I}_j], \boldsymbol{y}_{[\mathcal{I}_j]})$
$\widehat{\boldsymbol{x}}_j$	the optimal solution corresponds to $(\boldsymbol{A}[\mathcal{I}_j], \boldsymbol{y}_{[\mathcal{I}_j]})$
$(\cdot)[i]$	the i-th row of a matrix/ vector.
$\boldsymbol{x}[i]$	the i-th row of matrix $\boldsymbol{X}$
$\|\boldsymbol{X}\|_{p,q}$	takes ℓ_q norms on rows of $\boldsymbol{X}$; stacks those as a vector
	and then computes ℓ_p norm. The precise form is in (2.4).
$\|\boldsymbol{X}\|_{1,2}$	row sparsity norm
$\|\boldsymbol{X}\|_{1,1}$	equivalents to the ℓ_1 norm on vectorized $\boldsymbol{X}$
$\boldsymbol{x}^{\boldsymbol{\ell_1}}$	the ℓ_1 minimization solution
$\boldsymbol{x}^{\boldsymbol{B}}$	the Bagging solution
$\boldsymbol{x}^{\boldsymbol{J}}$	the JOBS solution

sparsity-promoting regularizers, the ℓ_1 norm is the most commonly used. The noiseless case is referred to as *Basis pursuit*:

$$\mathbf{P_1} : \min \|\boldsymbol{x}\|_1 \ \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{2.8}$$

The noisy version is known as *Basis pursuit denoising* (Chen, Donoho, and Saunders, 2001), or *least absolute shrinkage and selection operator* (Lasso) (Tibshirani, 1996):

$$\mathbf{P_1^\epsilon} : \min \|\boldsymbol{x}\|_1 \ \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \leq \epsilon, \tag{2.9}$$

where a non-negative scalar ϵ represents the energy level of the measurement noise. The unconstrained form of (2.10) is: for some $\lambda > 0$,

$$\mathbf{P_1^\lambda} : \min \lambda\|\boldsymbol{x}\|_1 + \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2. \tag{2.10}$$

Many algorithms are developed to solve (2.9) such as in (Figueiredo, Nowak, and Wright, 2007; Beck and Teboulle, 2009; Osborne, Presnell, and Turlach, 2000; Combettes and Wajs, 2005; Boyd et al., 2011; Donoho, Maleki, and Montanari, 2009), etc.

2.3 Geometric Illustration of Proper Condition for ℓ_1 Minimization

Figure 2.1 illustrates level sets of the ℓ_0 norm in equation (2.3) and the ℓ_1 norm in equation (2.2) level sets for two dimensional vector $\boldsymbol{x} = (x_1, x_2)^{\boldsymbol{T}}$. From the figure, we can see that the ℓ_0 norm is non-smooth when near axes or the origin whereas the relaxation ℓ_1 norm is smooth, which is preferable for optimization

algorithms.

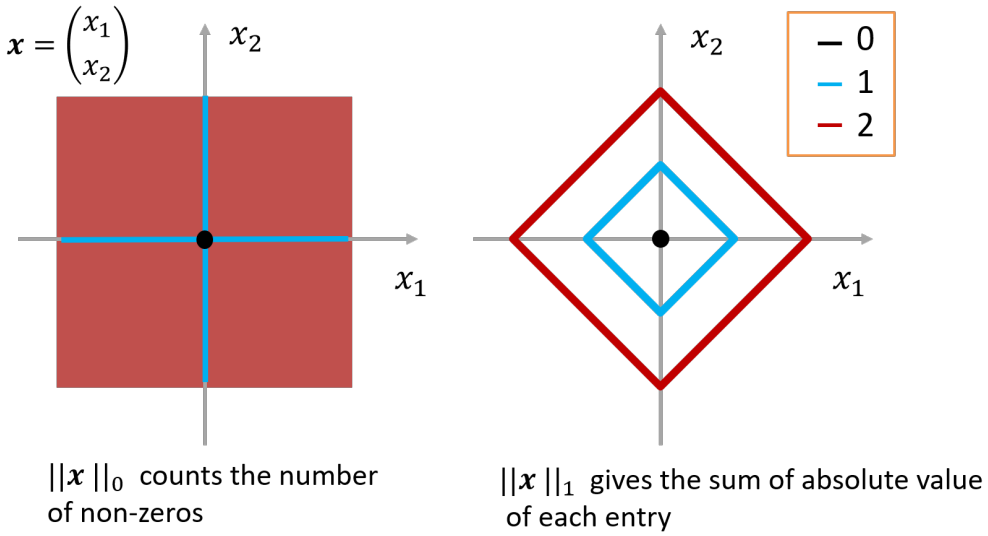


Figure 2.1: A example of ℓ_0 and ℓ_1 norm level sets for two dimensional vector.

Figure 2.2 gives a geometry demonstration of three different cases. All these three cases has the same true solutions that is $(x_1, x_2) = (0, 1)^{\boldsymbol{T}}$. Among three cases, the sensing matrices and the resulting measurements, that are green lines in these figures, are different. In the first case, ℓ_1 minimization recovers the true solution. In the second case, since the constraint is exactly parallel to level sets of ℓ_1 minimization norm, the ℓ_1 minimization is not unique. Although the true solution lies in the set of ℓ_1 minimization solutions, since there are multiple minima, it is treated as a failure recovery case. In the third case, the ℓ_1 minimization solution differs from the true solution and it is a failure case. This example shows that a proper condition of sensing matrix $\boldsymbol{A}$ is needed to guarantee correct recovery. This geometry property of $\boldsymbol{A}$ visualizes the null space property (Cohen, Dahmen, and DeVore, 2009) of sensing matrix $\boldsymbol{A}$ that characterise the correctness of sparse recovery and it will be elaborated in later

chapters (Chapter 3 and Chapter 4).

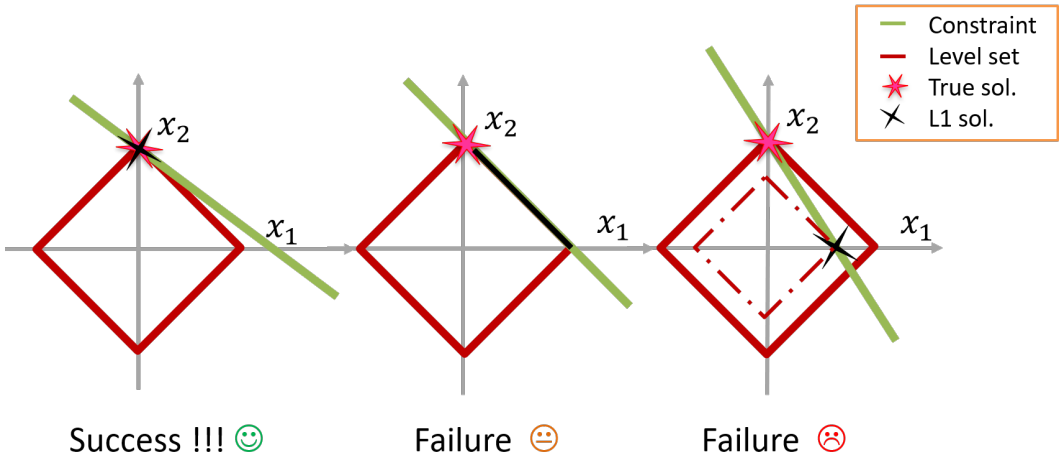


Figure 2.2: The geometry demonstration of ℓ_1 minimization. The pink star is the true solution. The green lines are constraints and red diamond shape line is the unit level set of the ℓ_1 norm. In the middle figure, all points lie on the black line are optimal ℓ_1 solutions.

2.4 Applications of Compressed Sensing/ Sparse Regression/ Sparse Coding

The field of compressed sensing, also known as sparse regression or sparse coding has a wide range of applications in the field of machine learning.

2.4.1 Signal Recovery of CS Hardware Systems

In the past decade, the information theory community introduced the concept of compressed sensing (CS) (Candès and Tao, 2005; Candes, Romberg, and Tao, 2006; Donoho, 2006; Baraniuk, 2007; Candès and Wakin, 2008), suggesting that the sparsity of natural signals can be utilized to reduce the number of samples required to capture signals of interest.

In many systems, sensing processes are time and energy consuming. Efficient compressed sensing design has been shown success in many imaging systems such as single pixel camera (Duarte et al., 2008), fast MRI imaging Lustig, Donoho, and Pauly, 2007, and Optical Coherence Tomography imaging (Lustig, Donoho, and Pauly, 2007; Guo et al., 2010; Liu and Kang, 2010). These work all have shown that much less measurements required by the Shannon/Nyquist theory.

2.4.2 Classification

Sparse regression is not only important in estimation problems, it also plays a crucial role in discriminative classification tasks, which are very important problem in machine learning.

A general framework of using sparse code features for classification is illustrated in Figure 2.3. Sparse Representation Classification proposed by Wright in Wright et al., 2009 uses features from training data directly as a dictionary, while other works in dictionary learning (Aharon, Elad, and Bruckstein, 2006; Mairal, Bach, and Ponce, 2012) aim at learning a better representative dictionary from training data features. These algorithms have shown a lot of success in various classification algorithms and achieve state-of-the art performance.

2.5 Generalized Sparsity

There are generalized definitions of the classic sparsity, which measures the sparsity level over some pre-defined patterns. We here introduce two common types of patterns: block sparsity and row sparsity.

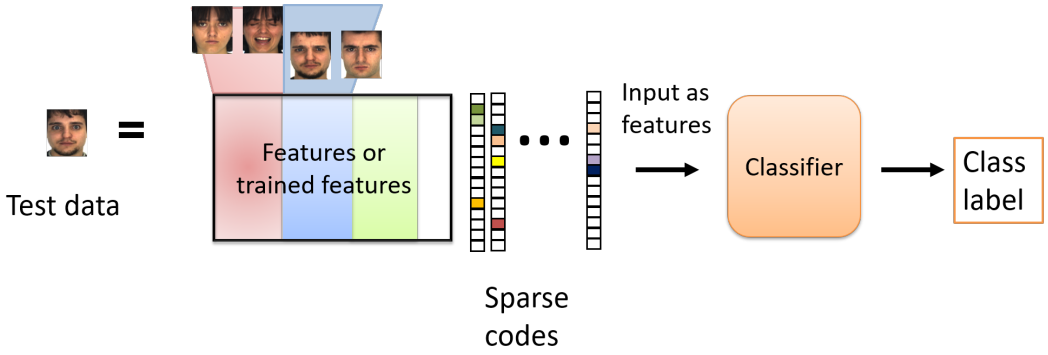


Figure 2.3: A general classification frameworks based on sparse codes/ representations.

2.5.1 Block Sparsity

The classic sparse vector has few non-zeros entries. A general notation of sparsity can be extended to blocks, which counts how many non-zero blocks there are in a vector. For a block to be a non-zero block, The classic sparsity and block sparsity of a vector $\boldsymbol{x}$ are depicted in Figure 2.4.

A more general sparsity can be defined over overlapping blocks, which is also known as group sparsity.

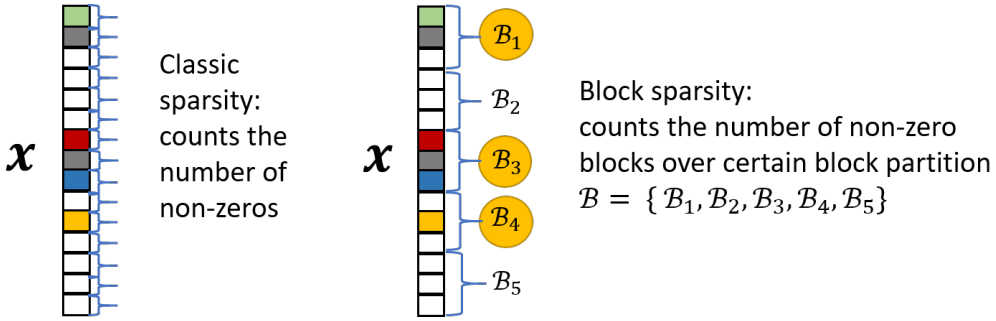


Figure 2.4: The illustration of classic sparsity and block sparsity. The classic sparsity is 6 and the block sparsity is 3 in this block partition (The blocks highlighted by yellow solid circles are non-zero blocks).

2.5.2 Row Sparsity

Row sparsity measures the sparsity of number of non-zero rows. It is a generalization of vector sparsity to matrix rows. Figure 2.5 depicts a 1 row-sparse matrix with only one row that is not a zero vector. The right panel of this figure shows that the row sparsity of a matrix can be equivalently defined as a block sparsity over its vectorized version. As we see in the figure, if we define blocks as row indices in its matrix: $\mathcal{B}_1 = 1, 4, \mathcal{B}_2 = 2, 5$ and $\mathcal{B}_3 = 3, 6$, row sparsity can be defined as a block sparsity pattern. We will explain the rigorous mathematical form in Section 4.9.1.

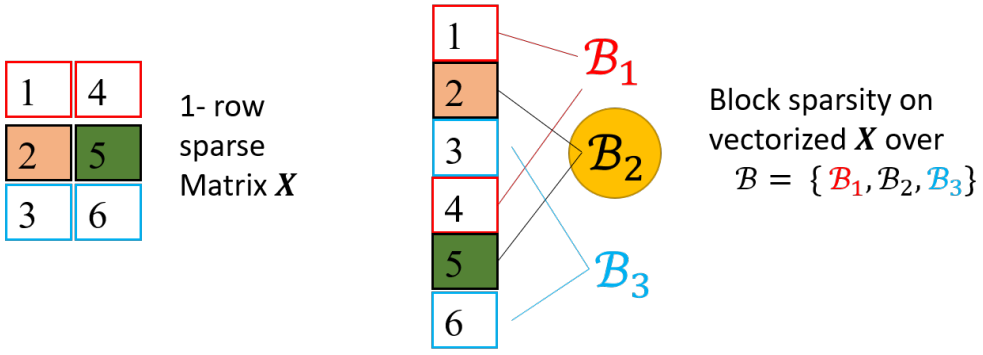


Figure 2.5: Illustration of row sparsity. **Left**: example of a row-sparse matrix. **Right**: row sparsity is a special case of block sparsity (The blocks highlighted by yellow solid circles are non-zero blocks).

2.6 Bootstrap

In statistics, the term *Bootstrapping* refers to a famous re-sampling scheme. It is a statistical method for estimating the underlining true sampling distribution empirically by sampling with replacement from the original pool of samples.

Bootstrapping methods has been widely used in deriving robust estimates and

confidence intervals of various statistics such as mean, median and conduction hypothesis testings (*Bootstrapping*).

2.7 Bootstrap Aggregating (Bagging)

Bagging procedure (Breiman, 1996), is designed to improve the stability and accuracy of machine learning algorithms in classification and regression problems. In the sparse recovery problem, Bagging method can incorporate the information from multiple estimates, Specifically, bagging solves objectives multiple times (say K times) *independently* from bootstrap samples and then averages over multiple predictions. Applying bagging in sparse regression was shown to reduce estimation error when the sparsity level s is high (Breiman, 1996). However, individually solved predictors aren't guaranteed to have the same support and in the worst case, their average can be quite dense: with its support size growing up to Ks.

2.8 Bootstrap Lasso (Bolasso)

To alleviate the problem of creating non-sparse solutions, Bolasso was proposed (Bach, 2008a). This is an algorithm that does support recovery via Bootstrap samples and then recovers the amplitude of the signal. Bolasso firstly recovers the support of the final estimate by detecting the common support among K individually solved predictors and then applies least squares on the common support. However, this strategy is very aggressive. When the noise level is high, it commonly recovers very sparse or even zero solution.

Chapter 3

Bagging in Sparse Regression

Classical sparse regression based on ℓ_1 minimization solves the least squares problem with all available measurements via sparsity-promoting regularization. In challenging practical applications with high levels of noise and missing or adversarial samples, solving the problem using all measurements simultaneously may fail. Bagging, a powerful ensemble method from machine learning, has shown the ability to improve the performance of unstable predictors in difficult practical settings. Although Bagging is most well-known for its application in classification problems, here we demonstrate that employing Bagging in sparse regression improves performance compared to the baseline method (ℓ_1 minimization). Although the original Bagging method uses a bootstrap sampling ratio of 1, such that the sizes of the bootstrap samples L are the same as the total number of data points m, we generalize the bootstrap sampling ratio in our framework to explore the optimal sampling ratios for various cases.

(Part of the contents of this chapter has been published in (Liu, Chin, and Tran, 2019).)

3.1 Introduction

Compressed Sensing (CS) and Sparse Regression studies solving the linear inverse problem in the form of least squares with an additional sparsity-promoting penalty term. Formally speaking, the measurements vector $\boldsymbol{y} \in \mathbb{R}^m$ is generated by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the sensing matrix, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of sparse coefficients with very few non-zero entries, and $\boldsymbol{z}$ is a noise vector with bounded energy. The problem of interest is finding the sparse vector $\boldsymbol{x}$ given $\boldsymbol{A}$ as well as $\boldsymbol{y}$. Among various choices of sparse regularizers, the ℓ_1 norm is the most commonly used. The noiseless case is referred to as *Basis Pursuit* (BP) whereas the noisy version is known as *basis pursuit denoising* (Chen, Donoho, and Saunders, 2001), or *least absolute shrinkage and selection operator* (LASSO) (Tibshirani, 1996) as in (2.9) and (2.10).

The performance of ℓ_1 minimization in recovering the true sparse solution has been thoroughly investigated in the CS literature (Candes, 2008; Candes, Romberg, and Tao, 2006; Donoho, 2006; Candess and Romberg, 2007). CS theory reveals that if the sensing matrix $\boldsymbol{A}$ has good properties, then BP recovers the ground truth and the LASSO solution is close enough to the true solution with high probability (Candes, 2008).

Classical sparse regression recovery based on ℓ_1 minimization solves the problem with all available measurements. In practice, it is often the case that not all measurements are available or required for recovery. Some measurements might be severely corrupted/missing or adversarial samples that break down the algorithm. These issues could lead to the failure of the sparse regression algorithm.

The Bagging procedure (Breiman, 1996) proposed by Leo Breiman is an efficient parallel ensemble method that improves the performance of unstable predictors. In Bagging, we first generate a bootstrap sample by randomly drawing m samples uniformly with replacement from all m data points. We repeat the process K times and generate K bootstrap samples. Then one bootstrapped estimator is computed for each bootstrap sample, and the final Bagged estimator is the average of all K estimators.

Applying Bagging to find a sparse vector with a specific symmetric pattern was shown empirically to reduce estimation error when the sparsity level s is high (Breiman, 1996) in a forward subset selection problem. This experiment shows the possibility of using Bagging to improve other sparse regression methods on general sparse signals. Although the well-known conventional Bagging method uses the bootstrap ratio 100%, some follow-up works have shown empirically that lower ratios improve Bagging in some classic classifiers: Nearest Neighbour Classifier (Hall and Samworth, 2005), CART Trees (Sabzevari, Martinez-Munoz, and Suarez, 2014), Linear SVM, LDA, and Logistic Linear Classifier (Zaman and Hirose, 2009). Based on this success, we hypothesize that reducing the bootstrap ratio will also improve performance of Bagging in sparse regression. Therefore, we set up the framework with a generic bootstrap ratio and study its behavior with various bootstrap ratios.

Note that, we use the notation L as the sizes of bootstrap samples, m as the number of all measurements, and K as the number of estimates. *(i)* We demonstrate the generalized Bagging framework with bootstrap ratio L/m and number of estimates K as parameters. *(ii)* We explore the theoretical

properties associated with finite L/m and K. *(iii)* We present simulation results with various parameters L/m and K and compare the performances of ℓ_1 minimization, conventional Bagging, and Bolasso (Bach, 2008a), another modern technique that incorporates Bagging into sparse recovery. An important discovery is that in challenging cases when m is small, Bagging with a ratio L/m that is smaller than the conventional ratio 100% can lead to better performance.

3.2 Proposed Method

3.2.1 Bagging in Sparse Regression

Our proposed method is sparse recovery using a generalized Bagging procedure. It is accomplished in three steps. First, we generate K bootstrap samples, each of size L, randomly sampled uniformly and independently with replacement from the original m data points. This results in K measurements and sensing matrices pairs: $\{\boldsymbol{y}_{[\mathcal{I}_1]}, \boldsymbol{A}[\mathcal{I}_1]\}, \{\boldsymbol{y}_{[\mathcal{I}_2]}, \boldsymbol{A}[\mathcal{I}_2]\}....., \{\boldsymbol{y}_{[\mathcal{I}_K]}, \boldsymbol{A}[\mathcal{I}_K]\}$. We use the notation $(\cdot)[\mathcal{I}]$ on matrices or vectors to denote retaining only the rows supported on $\mathcal{I}$ and throwing away all other rows in the complement $\mathcal{I}^c$. Second, we solve the sparse recovery problem independently on each of those pairs; mathematically, for all $j = 1, 2, .., K$, we find

$$\boldsymbol{x}_j^{\boldsymbol{B}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min}\, \lambda\|\boldsymbol{x}\|_1 + \|\boldsymbol{y}_{[\mathcal{I}_j]} - \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}\|_2^2. \tag{3.1}$$

The proposed approach is in the form of LASSO, and numerous optimization methods can be used to solve it, such as (Boyd et al., 2011; Berg and Friedlander, 2008; Wright, Nowak, and Figueiredo, 2009).

Finally, the Bagging solution is obtained by averaging all K estimators from

solving (3.1):

$$\text{Bagging:} \quad \boldsymbol{x^B} = \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{x}_j^{\boldsymbol{B}}. \tag{3.2}$$

Compared to the ℓ_1 minimization solution which is solved using all of the measurements, the bagged solution $\boldsymbol{x^B}$ is obtained by resampling without increasing the number of original measurements. We will show that in some cases, the bagged solution outperforms the base ℓ_1 minimization solution.

3.3 Preliminaries

We summarize the theoretical results of CS theory which we need to analyze our algorithm mathematically. We introduce the Null Space Property (NSP), as well as the Restricted Isometry Property (RIP). We also provide the tail bound of the sum of i.i.d. bounded random variables, which is needed to prove our theorems.

3.3.1 Null Space Property (NSP)

The NSP (Cohen, Dahmen, and DeVore, 2009) for standard sparse recovery characterizes the necessary and sufficient conditions for successful sparse recovery using ℓ_1 minimization.

Definition 1 (NSP, from (Cohen, Dahmen, and DeVore, 2009)) *Every $s-$sparse signal $\boldsymbol{x} \in \mathbb{R}^n$ is a unique solution to $\mathbf{P_1}: \ \min \|\boldsymbol{x}\|_1 \ \ s.t. \ \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ if and only if $\boldsymbol{A}$ satisfies NSP of order s: for any set $\boldsymbol{S} \subset \{1, 2, .., n\}$, with cardinality $s : card(\boldsymbol{S}) \leq s$,*

$$\|\boldsymbol{v}[\boldsymbol{S}]\ \|_1 < \|\boldsymbol{v}[\boldsymbol{S}^c]\ \|_1,$$

for all $\boldsymbol{v} \in \text{Null}(\boldsymbol{A})\backslash\{\boldsymbol{0}\}$*, where* $\boldsymbol{v}[\boldsymbol{S}]$ *denotes the vector equals to* $\boldsymbol{v}$ *on a index set* $\boldsymbol{S}$ *and zero elsewhere.*

3.3.2 Restricted Isometry Property (RIP)

Although NSP directly characterizes the ability of success for sparse recovery, checking the NSP condition is computationally intractable. It is also not suitable to use NSP for quantifying performance in noisy conditions since it is a binary (True or False) metric instead of a continuous range. The Restricted isometry property (RIP) (Candes, 2008) is introduced to overcome these difficulties.

Definition 2 (RIP, from (Candes, 2008)) *A matrix* $\boldsymbol{A}$ *with* ℓ_2*-normalized columns satisfies RIP of order* s *if there exists a constant* $\delta_s(\boldsymbol{A}) \in [0, 1)$ *such that for every* $s-$*sparse* $\boldsymbol{v} \in \mathbb{R}^n$*,*

$$(1 - \delta_s(\boldsymbol{A}))\|\boldsymbol{v}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{v}\|_2^2 \leq (1 + \delta_s(\boldsymbol{A}))\|\boldsymbol{v}\|_2^2. \tag{3.3}$$

3.3.3 Noisy Recovery Bounds based on RIP constants

It is known that satisfying the RIP conditions implies that the NSP conditions are also satisfied for sparse recovery (Candes, 2008). More specifically, if the RIP constant of order $2s$ is strictly less than $\sqrt{2} - 1$, then it implies that NSP is satisfied in the order s. We recall Theorem 1.2 in (Candes, 2008), where the noisy recovery performance for ℓ_1 minimization is bounded based on the RIP constant. This error bound is associated with the $s-$sparse approximation error and the noise level.

Theorem 3 (Noisy recovery for ℓ_1 **minimization (Candes, 2008))** *Let*

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 \leq \epsilon$, $\boldsymbol{x}_0$ *is* $s-$*sparse that minimizes* $\|\boldsymbol{x} - \boldsymbol{x}^\star\|$ *over all* $s-$*sparse signals. If* $\delta_{2s}(\boldsymbol{A}) \leq \delta < \sqrt{2}-1$, $\boldsymbol{x}^{\boldsymbol{\ell_1}}$ *be the solution of* ℓ_1 *minimization, then it obeys*

$$\|\boldsymbol{x}^{\boldsymbol{\ell_1}} - \boldsymbol{x}^\star\|_2 \leq \mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|_1 + \mathcal{C}_1(\delta)\epsilon,$$

where $\mathcal{C}_0(\cdot), \mathcal{C}_1(\cdot)$ *are some constants, which are determined by RIP constant* δ_{2s}. *The form of these two constants terms are* $\mathcal{C}_0(\delta) = \frac{2(1-(1-\sqrt{2})\delta)}{1-(1+\sqrt{2})\delta}$ *and* $\mathcal{C}_1(\delta) = \frac{4\sqrt{1+\delta}}{1-(1+\sqrt{2})\delta}$.

3.3.4 Tail Bound of the Sum of i.i.d. Bounded Random Variables

This exponential bound is similar in structure to Hoeffidings' inequality. Proving this bound requires working with the moment generating function of a random variable.

Lemma 4 (Tail bound of the sum of i.i.d. bounded Random variables) *Let* $Y_1, Y_2, ..., Y_n$ *be i.i.d. observations of bounded random variable* Y*:* $a \leq Y \leq b$ *and the expectation* $\mathbb{E}Y$ *exists, for any* $\epsilon > 0$, *then*

$$\mathbb{P}\{\sum_{i=1}^{n} Y_i \geq n\epsilon\} \leq \exp\{-\frac{2n(\epsilon - \mathbb{E}Y)^2}{(b-a)^2}\}. \tag{3.4}$$

3.4 Main Theoretical Results for Bagging in Sparse Regression

3.4.1 Noisy Recovery for Employing Bagging in Sparse Regression

We derive the performance bound for employing Bagging in sparse regression, in which the final estimate is the average over multiple estimates solved individually from bootstrap samples. We give the theoretical results for the case that true signal $\boldsymbol{x}^\star$ is exactly $s-$sparse and the general case with no assumption of the sparsity level of the ground truth signal. Note that, the theorems are based on deterministic sensing matrix, measurements, and noise: $\boldsymbol{A}, \boldsymbol{y}, \boldsymbol{z}$, in which all vector norms are equivalent.

Theorem 5 (Bagging: error bound for $\|\boldsymbol{x}^\star\|_0 = s$) *Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 < \infty$. If for all bootstrap matrices $\delta_{2s}(\boldsymbol{A}[\mathcal{I}_j]) < \sqrt{2}-1, j = 1, 2, ..., K$, then there exist a scaler $\delta_{L,K}$ such that $\delta_{2s}(\boldsymbol{A}[\mathcal{I}_j]) \leq \delta_{L,K} < \sqrt{2}-1$, and when the true solution is exactly $s-$sparse, for any $\tau > 0$, the Bagging solution $\boldsymbol{x}^{\boldsymbol{B}}$ satisfies*

$$\mathbb{P}\left\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2 \leq \mathcal{C}_1(\delta_{L,K})(\sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2 + \tau)\right\} \geq 1 - \exp\frac{-2K\tau^4}{L^2\|\boldsymbol{z}\|_\infty^4}, \tag{3.5}$$

where $\mathcal{C}_1(\cdot)$ is the same non-decreasing function of δ as in Theorem 3.

We also study the behavior of Bagging for a general signal $\boldsymbol{x}^\star, \|\boldsymbol{x}^\star\|_0 \geq s$, in which the performance involves the $s-$sparse approximation error. We use the vector $\boldsymbol{e}$ to denote this error, and $\boldsymbol{e} = \boldsymbol{x}^\star - \boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is the best s-sparse approximation of the ground truth signal over all $s-$sparse signals.

Theorem 6 (Bagging: error bound for the general sparse recovery) *Let* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 < \infty$. *If there exists a constant related to parameters* (L, K) *such that, If for all bootstrapped matrices* $\delta_{2s}(\boldsymbol{A}[\mathcal{I}_j]) < \sqrt{2} - 1, j = 1, 2, ..., K$, *then there exist a scaler* $\delta_{L,K}$ *such that* $\delta_{2s}(\boldsymbol{A}[\mathcal{I}_j]) \leq \delta_{L,K} < \sqrt{2} - 1$, *and for any* $\tau > 0$, *the Bagging solution* $\boldsymbol{x}^{\boldsymbol{B}}$ *satisfies*

$$\begin{aligned} &\mathbb{P}\left\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2 \leq (\mathcal{C}_0(\delta_{L,K})s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta_{L,K})(\sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2 + \tau)\right\} \\ &\geq 1 - \exp\frac{-2K{\mathcal{C}_1}^4(\delta)\tau^4}{(b')^2}, \end{aligned} \tag{3.6}$$

where $\mathcal{C}_0(\cdot), \mathcal{C}_1(\cdot)$ *are the same non-decreasing functions of* δ *as in Theorem 3, and* $b' = (\mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta)\sqrt{L}\|\boldsymbol{z}\|_\infty)^2$.

Theorem 6 gives the performance bound for Bagging in general signal recovery without the $s-$sparse assumption, and it reduces to Theorem 5 when the $s-$sparse approximation error is zero, i.e., $\|\boldsymbol{e}\|_1 = 0$. Both Theorem 5 and 6 above show that increasing the number of estimates K improves the result by increasing the lower bound of the certainty for the same performance level.

We give the proof sketch that demonstrate the key idea to prove both Theorem 5 and Theorem 6. Main tools used are Theorem 3 and Lemma 4. Some special treatments are required to deal with terms while proving Theorem 6. For more technical details, full proofs can be found in next Section.

Proof Sketch: Similar to the sufficient condition in Theorem 3, the sufficient condition to analyze Bagging is that all matrices resulting from Bagging have well-behaved RIP constants of order $2s$ bounded by a universal constant δ.

Let $\mathcal{I}$ denote a generic multi-set containing L elements and each element in

$\mathcal{I}$ are independent and identically, obeying a discrete uniform distribution from sample space $\{1, 2, .., m\}$. The squared error function $f(\boldsymbol{x}_{(\mathcal{I})}) = \|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2$, where $\boldsymbol{x}_{(\mathcal{I})}$ is the solution from ℓ_1 minimization on $\mathcal{I}$: $\boldsymbol{x}_{(\mathcal{I})} = \arg\min \lambda_{\mathcal{I}}\|\boldsymbol{x}\|_1 + \|\boldsymbol{y}_{[\mathcal{I}]} - \boldsymbol{A}[\mathcal{I}]\|_2^2$. Squared errors from K bootstrapped estimators $f(\boldsymbol{x}_j) = \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2, j = 1, 2, ..., K$ are realizations generated i.i.d. from the distribution of $f(\boldsymbol{x}_{(\mathcal{I})})$.

We proceed the proof using the Lemma 4. We choose the upper bound of the error to be a function of the expected value of noise power. For a exactly $s-$sparse true solution $\boldsymbol{x}^\star$, we pick $\epsilon = {\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2), \tau > 0$ and for a general signal $\boldsymbol{x}^\star$, we pick $\epsilon = (\mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta)\sqrt{\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2})^2 + {\mathcal{C}_1}^2(\delta)\tau^2, \tau > 0$. We can obtain the root of the expectation of squared error $\sqrt{\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2} = \sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2$. Then we need to compute the upper bound b and the lower bound a for the random variable $f(\boldsymbol{x}_{(\mathcal{I})})$. Since it is non-negative, we choose $a = 0$. The upper bound b is obtained from Theorem 3 and then the maximum value $\|\boldsymbol{z}\|_\infty$ is employed to further upper bound $\|\boldsymbol{z}_{\mathcal{I}_j}\|_2$. Through this process, we obtain the inequality: $\mathbb{P}\{\sum_j \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 - K\epsilon \leq 0\} \geq g(\mathbb{E}(f(\boldsymbol{x}), b, a)$, for some function g.

The Bagging solution is the average of all bootstrapped estimators. The key

inequality to establish is as follows:

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - \epsilon \leq 0\}$$

$$= \mathbb{P}\{K\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - \textstyle\sum_j f(\boldsymbol{x}_j) + \sum_j f(\boldsymbol{x}_j) - K\epsilon \leq 0\}$$

$$\geq \mathbb{P}\{K\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - \textstyle\sum_j f(\boldsymbol{x}_j) \leq 0, \sum_j f(\boldsymbol{x}_j) - K\epsilon \leq 0\}$$

$$= \mathbb{P}\{K\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - \textstyle\sum_j f(\boldsymbol{x}_j) \leq 0\}\mathbb{P}\{\sum_j f(\boldsymbol{x}_j) - K\epsilon \leq 0\}$$

$$= \mathbb{P}\{\textstyle\sum_j \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - K\epsilon \leq 0\}.$$

The first term is independent with the second term and it is true with probability 1 by Jensens' inequality. The bound for the second term is described in the previous paragraph. Here we successfully establish the relationship of error bound of the Bagging solution to the sum of squared errors of bootstrapped estimates.

3.4.2 Parameters Selection Guided by the Theoretical Analysis

Theorem 6 gives the performance bound for Bagging in sparse signal recovery without the $s-$sparse assumption, and it reduces to Theorem 5 when the $s-$sparse approximation error is zero $\|\boldsymbol{e}\|_1 = 0$. Theorem 6 can be used to analyze the cases with small m, where m is not sufficiently large enough compared to s.

Both Theorem 5 and 6 show that increasing the number of estimates K improves the result, by increasing the lower bound of certainty of the same performance. As for the sampling ratio L/m, because the RIP constant in general decreases with increasing L (proof with Gaussian assumption in (Baraniuk et al.,

2008)) and $\mathcal{C}_1(\delta)$ is a non-decreasing function of δ, a larger L in general results in a smaller $\mathcal{C}_1(\delta)$. The second factor associated with the noise power term, $\sqrt{L/m}$, suggests a smaller L.

Combining two factors indicates that the best L/m ratio is in between a small and a large number. In the experiment results, we will show that when m is small, varying L/m from $0-1$ creates peaks with the largest value at $L/m < 100\%$. The first factor is dominating in the stable case when there are enough measurements, in which a larger L leads to better performance.

3.5 Proofs of Main Theorems

3.5.1 Proof of Theorem 5: Performance Bound of Bagging for Exactly s-sparse Signals

Let $\boldsymbol{x}_1^{\boldsymbol{B}}, \boldsymbol{x}_2^{\boldsymbol{B}}, ..., \boldsymbol{x}_K^{\boldsymbol{B}}$ be the solutions of individually solved problems and the solution of the bagging scheme $\boldsymbol{x}^{\boldsymbol{B}}$ is obtained from their average: $\boldsymbol{x}^{\boldsymbol{B}} = \frac{1}{K}\sum_{j=1}^{K} \boldsymbol{x}_j^{\boldsymbol{B}}$. We consider the distance to the true solution $\boldsymbol{x}^\star$ from each estimate separately. Here, the desired upper bound is the square root of the expected power of each noise vector: $(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2} = \sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2$, where $\mathcal{I}$ is a multi-set of size L with each element randomly sampled from $\{1, 2, ..., m\}$. For any $\tau > 0$,

we have:

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2-\mathcal{C}_1(\delta)((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2}+\tau)\leq 0\}$$

$$=\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2-\mathcal{C}_1(\delta)(((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2}+\tau)^2)^{1/2}\leq 0\}$$

$$\geq\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2-\mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2+\tau^2)^{1/2}\leq 0\}$$

$$=\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2+\tau^2)\leq 0\}.$$

Consider using the average of errors for each estimate $\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2$, we can establish

$$\begin{aligned}
&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2-\mathcal{C}_1(\delta)((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2}+\tau)\leq 0\}\\
&=\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2\\
&+\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2+\tau^2)\leq 0\}\\
&\geq\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2\leq 0,\\
&\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2+\tau^2)\leq 0\}\\
&\text{(from the independence of two terms)}\\
&=\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-\frac{1}{K}\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2\leq 0\}\\
&\times\mathbb{P}\{\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}}-\boldsymbol{x}^{\star}\|_2^2-K{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2+\tau^2)\leq 0\}.
\end{aligned}$$

By Jensen's inequality, the bagging error is smaller than the averaged error of each individual estimator as in (4.22) and the first term holds with probability 1. Therefore, we have:

$$\begin{aligned}
&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2 - \mathcal{C}_1(\delta)((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2} + \tau) \leq 0\} \\
\geq &\mathbb{P}\{\sum_{j=1}^{K} \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - K{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2) \leq 0\} \\
= &1 - \mathbb{P}\{\sum_{j=1}^{K} \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 \geq K{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2)\}.
\end{aligned} \tag{3.7}$$

From this procedure, we can reduce the error bound for the bagging algorithm to bound the sum of individual errors.

Let the random variable of error for each bagged estimator be $\boldsymbol{x}_{(\mathcal{I})}$: $\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2$, where $\mathcal{I}$ denotes a bootstrap sample of size L and $\boldsymbol{x}_{(\mathcal{I})}$ is the bagged solution from ℓ_1 minimization on the bootstrap sample $\mathcal{I}$:
$\boldsymbol{x}_{(\mathcal{I})} = \arg\min \|\boldsymbol{x}\|_1 \;\; \text{s.t. } \|\boldsymbol{y}_{[\mathcal{I}]} - \boldsymbol{A}[\mathcal{I}]\|_2^2 \leq \epsilon_{(\mathcal{I})}$. The power of all errors for each bagged estimators $\|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2$ are realizations generated i.i.d. from the distribution of $\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2$. We proceed with the proof using Lemma 4 that establishes the tail bound of the sum of i.i.d. bounded random variables. It is a generalization of Hoeffding's inequality and the details of its proof can be found in Appendix 3.8.

In this case, we consider the lower bound a and the upper bound b of the error $\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2$. Clearly this term is non-negative, hence, we can set $a = 0$. The upper bound is obtained from the error bound of ℓ_1-minimization in Theorem 3. For all $\mathcal{I}$:

$$\mathbb{P}\{\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2 - {\mathcal{C}_1}^2(\delta)\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 \leq 0\} = 1. \tag{3.8}$$

According to the norm equivalence inequality, we have

$$\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 \leq (\sqrt{L}\|\boldsymbol{z}_{[\mathcal{I}]}\|_\infty)^2 \leq (\sqrt{L}\|\boldsymbol{z}\|_\infty)^2 = L\|\boldsymbol{z}\|_\infty^2. \tag{3.9}$$

From this, we can set $b = {\mathcal{C}_1}^2(\delta)L\|\boldsymbol{z}\|_\infty^2$.

We can now apply the sum of i.i.d. bounded random variable in Theorem 4 to analyze our problem. By (3.7), the parameter ζ in (3.4) turns out to be: $\zeta = {\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2)$. Hence,

$$\mathbb{P}\{\sum_{j=1}^{K}\|\boldsymbol{x}_j - \boldsymbol{x}^\star\|_2^2 - K\zeta \geq 0\} \leq \exp\{-\frac{2K(\zeta - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|)^2}{{\mathcal{C}_1}^4(\delta)L^2\|\boldsymbol{z}\|_\infty^4}\}. \tag{3.10}$$

To simplify the right hand side, let us consider the expected bagged error: $\mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 = \frac{1}{|m^L|}\sum_{\mathcal{I}}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2$. Our bound in (3.8) implies that

$$\mathbb{P}\{\frac{1}{|m^L|}\sum_{\mathcal{I}}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \leq \frac{1}{|m^L|}\sum_{\mathcal{I}}{\mathcal{C}_1}^2(\delta)\|\boldsymbol{z}_{\mathcal{I}}\|_2^2\} = 1,$$

which is equivalent to

$$\mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \leq \frac{1}{|m^L|}\sum_{\mathcal{I}}{\mathcal{C}_1}^2(\delta)\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 = \mathbb{E}\ {\mathcal{C}_1}^2(\delta)\|\boldsymbol{z}_{\mathcal{I}}\|_2^2 = {\mathcal{C}_1}^2(\delta)\mathbb{E}\|\boldsymbol{z}_{\mathcal{I}}\|_2^2. \tag{3.11}$$

From here, it is easy to see that

$$\begin{aligned}
&\zeta - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \\
=&{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2) - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \\
\geq&{\mathcal{C}_1}^2(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + \tau^2) - {\mathcal{C}_1}^2(\delta)\mathbb{E}\|\boldsymbol{z}_{\mathcal{I}}\|_2^2 = {\mathcal{C}_1}^2(\delta)\tau^2.
\end{aligned} \tag{3.12}$$

The right hand side of (3.10) is upper bounded by $\exp\{-\frac{2K\tau^4}{L^2\|\boldsymbol{z}\|_\infty^4}\}$. Substituting

this result into (3.7), we can obtain the result in our main bagging theorem.

3.5.2 Proof of Theorem 6: Bagging Performance Bound of Bagging for Approximately Sparse Signals

In this section, we are working with the case when the true solution $\boldsymbol{x}^\star$ is only approximately sparse. In other words, its sparsity level may exceed s and the $s-$sparse approximation error is no longer necessarily zero. Let ϵ_s denote the sparse approximation error $\epsilon_s = \mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1$. The square root of the expected power of each noise vector is $(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2} = \sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2$. We consider the following bound:

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2 - (\epsilon_s + \mathcal{C}_1(\delta)((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2} + \tau)) \leq 0\}$$

$$=\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 - (\epsilon_s + \mathcal{C}_1(\delta)((\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2} + \tau))^2 \leq 0\}$$

$$\geq\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 - \epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2 + {\mathcal{C}_1}^2(\delta)\tau^2) \leq 0\}.$$

Set $\zeta' = (\epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2 + {\mathcal{C}_1}^2(\delta)\tau^2$ and consider using the averages of the errors $\frac{1}{K}\sum_{j=1}^K \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2$ as an intermediate term. Repeating the same proving technique as in (3.7) yields

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 - \zeta'\} \geq \mathbb{P}\{\sum_{j=1}^K \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 - K\zeta' \leq 0\}$$

$$= 1 - \mathbb{P}\{\sum_{j=1}^K \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 \geq K\zeta'\}.$$

According to Lemma 4, we have:

$$\mathbb{P}\{\sum_{j=1}^K \|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2 \geq K\zeta'\} \leq \exp\{-\frac{2K(\zeta' - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2)^2}{(b'-a')^2}\}. \tag{3.13}$$

Here, $a' = 0$ and $b' = (\epsilon_s + \mathcal{C}_1(\delta)\sqrt{L}\|\boldsymbol{z}\|_\infty)^2$. The lower bound a' is set to zero since the error for any bagged estimator $\|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^\star\|_2^2$ is non-negative. The upper bound b' can be obtained using Theorem 3 and substituting in the upper bound of the noise power as derived in (3.9).

Next, consider the term $\zeta' - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 = (\mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta)\sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2)^2 + {\mathcal{C}_1}^2(\delta)\tau^2 - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2$. We can upper bound the expected value of the error of bagged estimator with same approach in (3.11). From Theorem 3, for all $\mathcal{I}$:

$$\mathbb{P}\{\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \leq (\epsilon_s + \mathcal{C}_1(\delta)\|\boldsymbol{z}_{[\mathcal{I}]}\|_2)^2\} = 1. \tag{3.14}$$

Since $\mathcal{I}$ takes value of all m^L choices with equal probability, the following result is implied from (3.14):

$$\mathbb{P}\{\mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^\star\|_2^2 \leq \mathbb{E}(\epsilon_s + \mathcal{C}_1(\delta)\|\boldsymbol{z}_{[\mathcal{I}]}\|_2)^2\} = 1. \tag{3.15}$$

Since $f(x) = x^2$ is a convex function, applying Jensen's inequality results in

$$(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2)^2 \leq \mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2.$$

Since the square root $x^{1/2}$ is a increasing function of x, taking square root preserves the sign of the inequality:

$$\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2 \leq (\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2}. \tag{3.16}$$

Then, from (3.15), we have:

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2 &\leq \mathbb{E}(\epsilon_s + \mathcal{C}_1(\delta)\|\boldsymbol{z}_{[\mathcal{I}]}\|_2)^2 \\
&= \epsilon_s^2 + {\mathcal{C}_1}^2(\delta)\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + 2\epsilon_s\mathcal{C}_1(\delta)\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2 \\
&\text{(by (3.16))} \\
&\leq \epsilon_s^2 + {\mathcal{C}_1}^2(\delta)\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2 + 2\epsilon_s\mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{\mathcal{I}}\|_2^2)^{1/2} \\
&= (\epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2.
\end{aligned}$$

Finally, we can bound the term $\zeta' - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2$ as follows:

$$\begin{aligned}
&\zeta' - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2 \\
=&(\epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2 + {\mathcal{C}_1}^2(\delta)\tau^2 - \mathbb{E}\|\boldsymbol{x}_{(\mathcal{I})} - \boldsymbol{x}^{\star}\|_2^2 \\
\geq&((\epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2 + {\mathcal{C}_1}^2(\delta)\tau^2 - (\epsilon_s + \mathcal{C}_1(\delta)(\mathbb{E}\|\boldsymbol{z}_{[\mathcal{I}]}\|_2^2)^{1/2})^2 \\
=&{\mathcal{C}_1}^2(\delta)\tau^2.
\end{aligned}$$

One can observe that the upper bound of this difference is ${\mathcal{C}_1}^2(\delta)\tau^2$, which is the same as in the case of the exact $s-$sparse signal in (3.12). The bound for (3.13) can be upper bounded by

$$\mathbb{P}\{\sum_{j=1}^{K}\|\boldsymbol{x}_j^{\boldsymbol{B}} - \boldsymbol{x}^{\star}\|_2^2 - K\zeta' \geq 0\} \leq \exp\{-\frac{2K{\mathcal{C}_1}^4(\delta)\tau^4}{(b')^2}\},$$

where $b' = (\mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta)\sqrt{L}\|\boldsymbol{z}\|_\infty)^2$.

3.6 Experimental Results

In this section, we perform sparse recovery on simulated data to study the performance of our algorithm. In our experiment, all entries of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are i.i.d. samples from the standard normal distribution $\mathcal{N}(0, 1)$. The signal dimension $n = 200$ and various numbers of measurements from 50 to 2000 are explored. For the ground truth signals, their sparsity levels are all $s = 50$, and the non-zero entries are sampled from the standard Gaussian with their locations being generated uniformly at random. For the noise processes $\boldsymbol{z}$, which entries are sampled i.i.d. from $\mathcal{N}(0, \sigma^2)$, with variance $\sigma^2 = 10^{-\mathrm{SNR}/10}\|\boldsymbol{A}\boldsymbol{x}\|_2^2$, where SNR represents the Signal to Noise Ratio. In our experiment, we add white Gaussian noise to make the SNR $= 0$ dB. Gaussian model is chosen since it is the most commonly used model for noise processes. Although its support is not bounded, in numerical experiments, realizations of Gaussian are always finite values. We use the ADMM (Boyd et al., 2011) implementation of Lasso to solve all sparse regression problems, in which the parameter $\lambda^{(K,L)}$ balances the least squares fit and the sparsity penalty.

We study how the bootstrap sampling ratio L/m as well as the number of estimates K affects the result. In our experiment, we take $K = 30, 50, 100$, while the bootstrap ratio L/m varies from 0.1 to 1. We report the Signal to Noise Ratio (SNR) as the error measure for recovery: $\mathrm{SNR}(\boldsymbol{x}, \boldsymbol{x}^\star) = -10 \log_{10} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2 / \|\boldsymbol{x}^\star\|_2^2$ averaged over 20 independent trials. For all algorithms, we evaluate $\lambda^{(K,L)}$ at different values from .01 to 200 and then select optimal values that gives the maximum averaged SNR over all trials.

Bagging and BoLasso with the various parameters K, L and ℓ_1 minimization

are studied. The results are plotted in Figure 3.1. The colored curves show the cases of Bagging with various number of estimates K. The intersections of colored curves and the purple solid vertical lines at $L/m = 100\%$ illustrates conventional Bagging with a full bootstrap rate. The grey circle highlights the best performance and the grey area highlights the optimal bootstrap ratio L/m. The performance of ℓ_1 minimization is depicted by the black dashed lines, while the best Bolasso performance is plotted using light green dashed lines. In those figures, for each condition with a choice of K, L, the information available to Bagging and Bolasso algorithms are identical, and ℓ_1 minimization always has access to all m measurements.

From Figure 3.1, we see that when m is small, Bagging can outperform ℓ_1 minimization. As m decreases, the margin increases. The important observation is that with a low number of measurements (m is between s to $2s$: $50 - 100$, s is the sparsity level), and a reduced bootstrap ratio L/m ($60\% - 90\%$), Bagging beats the conventional choice of full bootstrap ratio 100% for all different choices of K. Also with a reduced ratio and a small K our algorithm is already quite robust and outperforms ℓ_1 minimization by a large margin. When the number of measurements is moderate $m = 3s = 150$, Bagging still beats the baseline; however, the peaks take at full bootstrapping ratio and reduced bootstrap ratios does not gain more benefits. Increasing the level measurement makes the base algorithm more stable and the advantage of Bagging starts decaying.

3.7 Summary

We extend the conventional Bagging scheme in sparse recovery with the bootstrap sampling ratio L/m as adjustable parameters and derive error bounds for the algorithm associated with L/m and the number of estimates K. Bagging is particularly powerful when the number of measurements m is small. This condition is notoriously difficult, both in terms of improving sparse recovery results and obtaining tight bounds of theoretical properties. Despite these challenges, Bagging outperforms ℓ_1 minimization by a large margin and the reduced sampling rate has a larger margin over the conventional Bagging algorithm $L/m = 1$. When the number of measurements m is $s-2s$, where s is the sparsity level, the conventional Bagging achieves $270\% - 29\%$ and the generalized Bagging achieves $367\% - 32\%$ SNR improvement over the original ℓ_1 minimization with reduced sampling rate. Our Bagging scheme achieves acceptable performance even with very small L/m (around 0.6) and relative small K (around 30 in our experimental study). The error bounds for Bagging show that increasing K will improve the certainty of the bound, which is validated in the simulation. For a parallel system that allows a large number of processes to be run at the same time, a large K is preferred since it in general gives a better result.

3.8 Appendix: Proof of Lemma 4

To prove of this lemma, We would need the Markov's inequality for non-negative random variables here. Let X be a non-negative random variable and suppose

that $\mathbb{E}X$ exists. For any $t > 0$, we have:

$$\mathbb{P}\{X > t\} \leq \frac{\mathbb{E}X}{t}. \tag{3.17}$$

We also need the upper bound of the moment generating function (MGF) of the random variable Y. Suppose that $a \leq Y \leq b$, then for all $t \in \mathbb{R}$,

$$\mathbb{E}\exp\{tY\} \leq \exp\{t\mathbb{E}Y + \frac{t^2(b-a)^2}{8}\}. \tag{3.18}$$

Back to Lemma 4, for $t > 0$,

$$\mathbb{P}\{\sum_{i=1}^{n} Y_i \geq n\zeta\} = \mathbb{P}\{\exp\{\sum_{i=1}^{n} Y_i\} \geq \exp\{n\zeta\}\}$$

$$= \mathbb{P}\{\exp\{t\sum_{i=1}^{n} Y_i\} \geq \exp\{tn\zeta\}\}$$

using the Markov inequality in (3.17), we have:

$$\mathbb{P}\{\sum_{i=1}^{n} Y_i \geq n\zeta\} \leq \exp\{-tn\zeta\}\mathbb{E}\{\exp\{t\sum_{i=1}^{n} Y_i\}\}$$

$$= \exp\{-tn\zeta\}\mathbb{E}\{\Pi_{i=1}^{n}\exp\{tY_i\}\}$$

$$= \exp\{-tn\zeta\}\Pi_{i=1}^{n}\mathbb{E}\{\exp\{tY_i\}\}$$

by upper bound for MGF in (3.18)

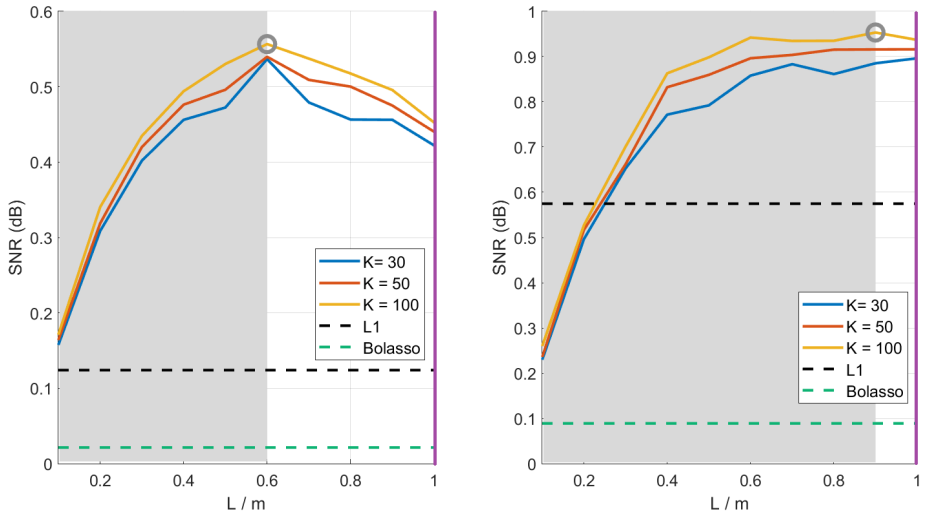
$$\mathbb{P}\{\sum_{i=1}^{n} Y_i \geq n\zeta\} \leq \exp\{-tn\zeta\}(\exp\{t\mathbb{E}Y + \frac{t^2(b-a)^2}{8}\})^n$$

$$= \exp\{-tn\zeta + tn\mathbb{E}Y + \frac{t^2(b-a)^2 n}{8}\}.$$

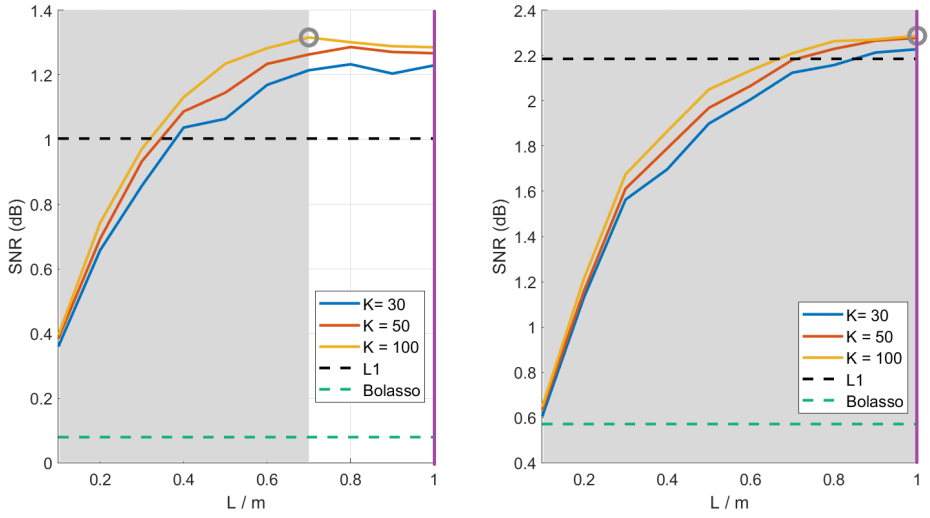
The right hand side is a convex function with respect to t. Taking the derivative with respect to t and set it zero, we obtain the optimal t, $t^\star = \frac{4\zeta - 4EY}{(b-a)^2}$. The right

hand side is minimized at value:

$$\exp\{-t^\star n\zeta + t^\star n\mathbb{E}Y + \frac{t^{\star 2}(b-a)^2 n}{8}\} = \exp\{\frac{-2n(\zeta - \mathbb{E}Y)^2}{(b-a)^2}\}.$$



(a) $m = 50$ **(b)** $m = 75$



(c) $m = 100$ **(d)** $m = 150$

Figure 3.1: Performance curves for Bagging with various sampling ratios L/m and number of estimates K, the best performance of Bolasso as well as ℓ_1 minimization. The Purple lines highlighted conventional Bagging with $L/m = 1$. In all cases, SNR $= 0$ dB and the number of measurements $m = 50, 75, 100, 150$ from left to right. The grey circle highlights the peak of Bagging, and the grey area highlights the bootstrap ratio at the peak point.

Chapter 4

JOBS: A Collaborative Regression Scheme

In this chapter, we elaborate our proposed method: a robust global sparse recovery strategy, named JOBS, which uses subsets of measurements to improve sparse regression in challenging circumstances. Here, K measurement vectors are generated from the original pool of m measurements via bootstrapping – each bootstrap sample containing L elements – and then a joint-sparse constraint is enforced to ensure support consistency among multiple predictors. The final estimate is obtained by averaging over K estimators. We will study the performance limits associated with different choices of bootstrap sampling ratio L/m and number of estimates K is analyzed theoretically and use simulations to validate our analysis.

(Part of the contents of this chapter has been published in (Liu, P, and Tran, 2019) and part of the contents of this chapter has been under review for Information Theory.)

4.1 Introduction

In compressed sensing (CS) and sparse recovery, solutions to the linear inverse problem in the form of least squares plus a sparsity-promoting penalty term have been extensively studied. Formally speaking, a the measurements vector $\boldsymbol{y} \in \mathbb{R}^m$ is generated by the model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z}$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is the sensing matrix, $\boldsymbol{x} \in \mathbb{R}^n$ contains the sparse codes with very few non-zero entries and $\boldsymbol{z}$ is noise vector with bounded energy. The problem of interest is to find the sparse vector $\boldsymbol{x}$ given the sensing matrix $\boldsymbol{A}$ as well as the measurement vector $\boldsymbol{y}$. However, directly minimizing the sparsity level, which is the number of non-zeros in $\boldsymbol{x}$, is proven to be NP-hard (Natarajan, 1995). Instead, a convex regularizer is preferable. Among various choices of sparsity-promoting regularizers, the ℓ_1 norm is the most commonly used. The noiseless case is referred to as *Basis pursuit* in (2.8). The noisy version is known as *Basis pursuit denoising* (Chen, Donoho, and Saunders, 2001), or *least absolute shrinkage and selection operator* (Lasso) (Tibshirani, 1996) as in (2.9) and (2.10).

The performance of ℓ_1 minimization in recovering the true sparse solution has been thoroughly investigated in the CS literature (Cohen, Dahmen, and DeVore, 2009; Candes, 2008; Candes, Romberg, and Tao, 2006; Donoho, 2006; Candess and Romberg, 2007). Pioneer works study the correctness and robustness of ℓ_1 minimization. They establish conditions for successful recovery based on the Null Space Property (NSP) (Cohen, Dahmen, and DeVore, 2009) and the Restricted Isometry Property (RIP) as well as quantify the recovery performance via the RIP constant (Candes, 2008; Candes, Romberg, and Tao, 2006). Additionally, under mild conditions on random sensing matrices, CS theory reveals that when

the true solution is sparse and with enough measurements, then (2.8) recovers the ground truth and solution to (2.9) is within a controllable neighborhood of the true solution with high probability (Candes, 2008). Unfortunately, in practice, measurements may not be available all at once. Moreover, certain parts of the data might be missing and/or severely corrupted. For example, in common streaming settings, measurements might be available sequentially or in small batches. Waiting for all measurements to be available wastes valuable processing time and buffering memory.

Alternatively, in sparse-representation-based classification, many schemes use local observations and have shown promising performances (Aharon, Elad, and Bruckstein, 2006; Yang et al., 2010; Liu, Tran, and Chin, 2016; Chen, Do, and Tran, 2010; Bosworth et al., 2015b). The proper choices of measurement subsets differ between applications and often require case-by-case treatment. Obviously, prior knowledge should help significantly in the selection process. For example, image datasets may have large variance overall but data remains relatively homogeneous within local regions. Hence, choosing to work with image patches often leads to satisfactory results in dictionary learning and deep learning (Aharon, Elad, and Bruckstein, 2006; Krizhevsky, Sutskever, and Hinton, 2012).

Without any prior information, a natural choice is to sample data uniformly at random with replacement, termed *bootstrap* (Efron, 1979). This simple sampling scheme has been shown to represent the entire system better than specific predefined choices. It performs reasonably well when all measurements are equally good. In CS theory, many random matrices have been proven to be

excellent sensing matrices. These operators act by shuffling and recombining entries of the original data samples, destroying any spatial or temporal structure and making the measurements even more democratic.

To incorporate information from multiple estimates, the *Bagging* (Breiman, 1996) (**B**ootstrap **Agg**regat**ing**) framework has been proposed. It solves the same objective function multiple times independently from bootstrap samples and then averages over multiple predictions to obtain the final solution. Applying the Bagging method in sparse regression has been shown to reduce estimation error when the sparsity level s is high for signals with a specific sparsity pattern (Breiman, 1996) and for general sparse signals (Liu, Chin, and Tran, 2019). Although Bagging is a general procedure for regression and classification tasks, in this work, we use *Bagging* to refer to employing Bagging procedure in sparse recovery. However, individually solved predictors are not guaranteed to have the same support, and in the worst case, their average can be quite dense – its support size growing up to the number of estimates times the true sparsity level. To alleviate this problem, *Bolasso* (**Bo**otstrapping **Lasso**) has been proposed (Bach, 2008a). Bolasso first recovers the common support using the intersection of all bootstrapped estimators and then estimates the magnitudes by applying least squares on the support. However, this strategy is very aggressive. When the noise level is high, it commonly recovers the extremely sparse or even all-zero solution.

in this work, to resolve the support consistency issue in Bagging and avoid the overly aggressive two-step method Bolasso scheme, we propose enforcing the row sparsity constraint among all predictors using the $\ell_{1,2}$ norm. The final

estimate is obtained by averaging over all estimators. We name this whole procedure JOBS (**J**oint-sparse **O**ptimization from **B**ootstrap **S**amples). The proposed method involves two key parameters: the bootstrap sample size L of random sampling with replacement from the original m measurements and the K number of those bootstrap vectors.

We will show that JOBS consistently and significantly outperform the baseline ℓ_1-minimization algorithm in the challenging case when the number of measurements m is limited. Our previous work (Liu, Chin, and Tran, 2019) has shown that Bagging improves the baseline ℓ_1 minimization when the bootstrap ratio L/m is smaller than the conventional full bootstrap sampling rate of 1. An interesting discovery is that the optimal bootstrap ratio JOBS is even lower than that of Bagging for similar optimal performance level. The row sparsity prior among all estimators helps bring down the optimal bootstrap sampling ratio and therefore less data is required for JOBS to achieve similar performance as Bagging.

The main contributions of this paper are as follows. *(i)* We demonstrate that employing the powerful bootstrapping idea, inspired from machine learning, can improve the robustness of sparse recovery in noisy environments through a collaborative recovery scheme. *(ii)* We provide an in-depth analysis of the proposed JOBS strategy. Since the critical parameters in our method are the bootstrap sample size L and the number of bootstrap measurement vectors K, we derive analytically various error bounds with regards to these parameters. *(iii)* We confirm our optimal parameter settings and validate our theoretical framework via extensive simulations results. *(iv)* We extend the theoretical

analysis to the Bagging framework, using the same setting with various bootstrap sampling ratios and different number of estimates. *(v)* We present a natural extension of the framework, employing a sub-sampling variation of the proposed scheme (named Sub-JOBS) as an alternative to bootstrapping, and discuss the relationship between the sub-sampling variation to bootstrap. (*vii*) Finally, we demonstrate that the proposed JOBS recovery also benefits discriminative tasks such as face recognition over the baseline Sparse Representation-based Classification (SRC) framework, in which the conventional ℓ_1 minimization is employed (Wright et al., 2009).

The outline of this chapter is as follows. Section 4.2 illustrates the JOBS procedure, demonstrate that it is a relaxation of ℓ_1 minimization, and provide relevant intuition for further analysis. Section 4.3 summarizes necessary theoretical background to analyze our algorithm. Section 4.4 then demonstrates all the major theoretical results of JOBS and Bagging with a generic L/m ratio and K – theoretical guarantee of JOBS solution and the worst case performance bounds for JOBS as well as Bagging. Section 4.5 presents the analysis and derivation of the results in the previous Section 4.4. Section 4.6 provides a detailed comparison between JOBS, Bagging, Bolasso, and ℓ_1 minimization on a synthetic dataset. Finally, Section 4.7 illustrates the application of JOBS on the classic classification task of face recognition using two real-world datasets.

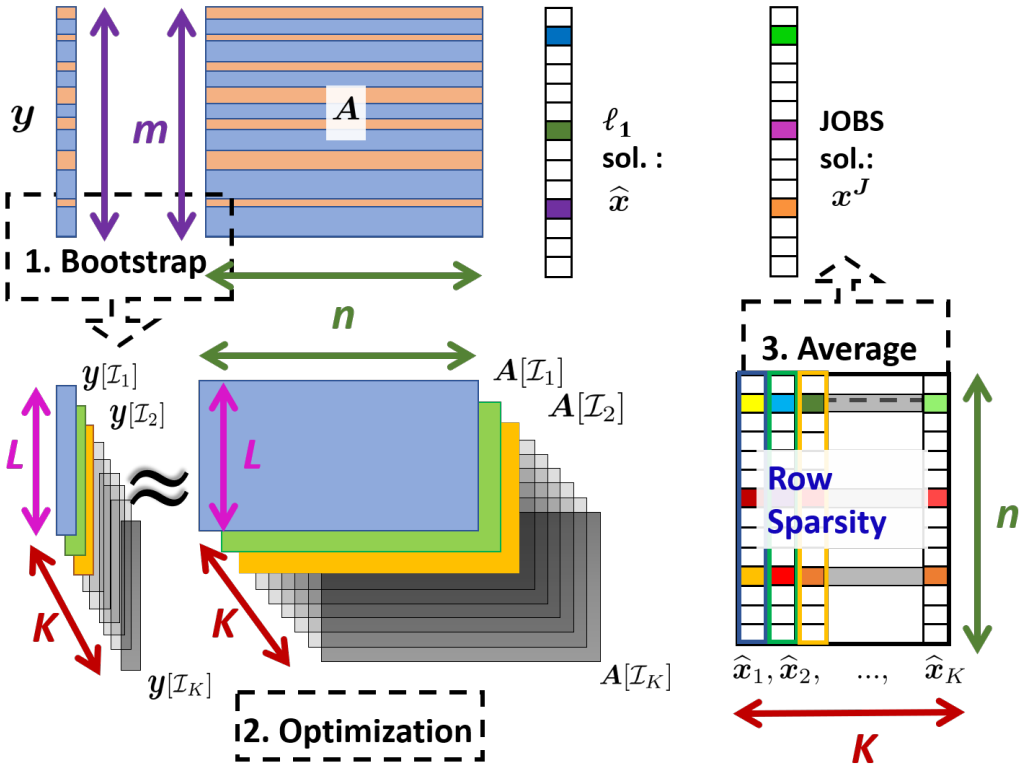


Figure 4.1: The Diagram of JOBS framework. The ℓ_1 minimization solution is obtained from solving optimization directly using the original sensing matrix $\boldsymbol{A}$ and the measurements vector $\boldsymbol{y}$. To obtain JOBS solution, K bootstrap samples of size L are generated from $\boldsymbol{A}$ and $\boldsymbol{y}$. A row-sparsity regularization is applied across all predictors. The final prediction is obtained by averaging.

4.2 Proposed Method: JOBS

4.2.1 JOBS

Our proposed method JOBS consists of three steps. First, we generate a bootstrap sample $\mathcal{I}$, in which each element is randomly sampled from the global measurement set $\{1, 2, .., m\}$. We repeat this process K times and create K bootstrap samples: $\{\mathcal{I}_1, \mathcal{I}_2, .., \mathcal{I}_K\}$. By taking corresponding rows from selected

bootstrap samples, the data now contain K pairs of sensing matrices measurements: $\{\boldsymbol{y}_{[\mathcal{I}_1]}, \boldsymbol{A}[\mathcal{I}_1]\}, \{\boldsymbol{y}_{[\mathcal{I}_2]}, \boldsymbol{A}[\mathcal{I}_2]\}....., \{\boldsymbol{y}_{[\mathcal{I}_K]}, \boldsymbol{A}[\mathcal{I}_K]\}$, where the operation $(\cdot)[\mathcal{I}]$ takes the rows of a matrix or vector supported on $\mathcal{I}$. Second, we solve the collaborative recovery on those sets. The optimization problem has both noiseless and noisy cases. The noiseless case problem is as follows. For all $j = 1, 2, ..., K$,

$$\mathbf{J_{12}} : \min \|\boldsymbol{X}\|_{1,2} \ \text{ s.t. } \ \boldsymbol{y}_{[\mathcal{I}_j]} = \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j, \tag{4.1}$$

and the noisy counterpart can be expressed as: for some non-negative number $\epsilon^J > 0$,

$$\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon^J}} : \widehat{\boldsymbol{X}} = \arg\min \|\boldsymbol{X}\|_{1,2} \ \text{ s.t. } \ \sum_{j=1}^{K} \|\boldsymbol{y}_{[\mathcal{I}_j]} - \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j\|_2^2 \leq \epsilon^J. \tag{4.2}$$

The proposed forms in $\mathbf{J_{12}}$, $\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon^J}}$ are in the form of block (group) sparse recovery (Berg and Friedlander, 2008) and there are numerous optimization methods for solving them such as (Boyd et al., 2011; Baron et al., 2009; Heckel and Bolcskei, 2012; Sun et al., 2009; Bach, 2008b; Berg and Friedlander, 2008; Wright, Nowak, and Figueiredo, 2009; Deng, Yin, and Zhang, 2011). in this work, we focus on the noisy form in (4.2). The noiseless form is presented only for the purpose of deeper understanding and analysis of theoretical properties.

Finally, the JOBS solution is obtained by averaging the columns of the solution $\widehat{\boldsymbol{X}}$ from (4.2):

$$\text{JOBS: } \boldsymbol{x}^{\boldsymbol{J}} = \frac{1}{K}\sum_{j=1}^{K} \widehat{\boldsymbol{x}}_j. \tag{4.3}$$

All supports (the locations of non-zero entries) of $\widehat{\boldsymbol{x}}_1, \widehat{\boldsymbol{x}}_2, ..., \widehat{\boldsymbol{x}}_K$ are the same because of the row sparsity constraint that we impose. Therefore, the sparsity

level of the JOBS solution $\boldsymbol{x^J}$ is guaranteed to be preserved during the averaging process unlike in the Bagging case. Figure 4.1 illustrates the entire proposed JOBS framework.

4.2.2 Implementation of JOBS

We present the pseudo-code for solving JOBS optimization problem via Alternating Direction Method of Multipliers(ADMM) updates. The key difference to Bagging and the baseline ℓ_1 minimization here is that we employ the soft-thresholding operation on each row in JOBS (described in line 6 of Algorithm 1), rather than the common entry-wise thresholding operation on each individual sparse-code element in Bagging.

Algorithm 1 ADMM implementation of JOBS

Require: Sensing matrix and measurements vector $(\boldsymbol{A}, \boldsymbol{y})$, bootstrap ratio and number of estimates $(L/m, K)$, sparse balancing ratio λ, learning rate ρ, maximum number of iterations MaxIter.
Initialization: $\widehat{\boldsymbol{X}}_0, \boldsymbol{W}_0, \boldsymbol{U}_0 \leftarrow \boldsymbol{O}$ (zero matrix of size $n \times K$)

1: generate K bootstrap samples of length L:
$\{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_K\}$, and its corresponding $\{\boldsymbol{A}[\mathcal{I}_j], \boldsymbol{y}_{[\mathcal{I}_j]}\}$
2: **for** $t = 1 :$ MaxIter **do**
3: $\widehat{\boldsymbol{X}}$ update: $\widehat{\boldsymbol{x}}_j \leftarrow$
$(\boldsymbol{A}[\mathcal{I}_j]^* \boldsymbol{A}[\mathcal{I}_j] + \rho \boldsymbol{I})^{-1} (\boldsymbol{A}[\mathcal{I}_j]^* \boldsymbol{y}_{[\mathcal{I}_j]} + \rho(\boldsymbol{w} - \boldsymbol{u}))$
4: $\widehat{\boldsymbol{X}} \leftarrow \alpha \widehat{\boldsymbol{X}} + (1 - \alpha) \boldsymbol{W}$
5: $\boldsymbol{W}$ update: applying shrinkage operations on each row. For $i = 1, 2, .., n$,
6: $\boldsymbol{w}[i] \leftarrow \text{Shrinkage}_{\lambda/\rho}(\widehat{\boldsymbol{x}}[i] - \boldsymbol{u}[i])$,
$\text{Shrinkage}_{\kappa}(\boldsymbol{x}) = \max(1 - \kappa / \|\boldsymbol{x}\|_2, 0)\boldsymbol{x}$
7: $\boldsymbol{U}$ update: $\boldsymbol{U} = \boldsymbol{U} + \boldsymbol{X} - \boldsymbol{W}$
8: **end for**
9: JOBS solution is the average columns of solution matrix $\widehat{\boldsymbol{X}}$: $\boldsymbol{x^J} = 1/K \sum \widehat{\boldsymbol{x}}_j$

4.2.3 Intuitive Explanation of Why JOBS Works

JOBS recovers the true sparse solution because it is essentially a relaxation of the original ℓ_1 minimization problem in a multiple vectors fashion. Therefore, it is not so surprising that JOBS relaxation can recover the true solution: exactly in the noiseless case and within some neighbourhood of the ground truth in noisy case.

We demonstrate that JOBS is a two-step relaxation procedure of ℓ_1 minimization. For a ℓ_1 minimization as in equation (2.8) with a unique solution $\boldsymbol{x}^\star$, the multiple measurement vectors (MMV) equivalence is: for $j = 1, 2, .., K$

$$\mathbf{P_1}(K) : \min \|\boldsymbol{X}\|_{1,1} \ \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_j, \tag{4.4}$$

where $\|\boldsymbol{X}\|_{1,1} = \sum_i \|\boldsymbol{x}[i]^{\boldsymbol{T}}\|_1$ as mentioned in Table 2.1. We show that this MMV form (4.4) is equivalent to the original problem (2.8). If the original problem $\mathbf{P_1}$ in (2.8) has a unique solution $\boldsymbol{x}^\star$, then the solution to the MMV problem $\mathbf{P_1}(K)$ in (4.4) yields a row sparse solution $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)$. This result can be derived via contradiction. The reverse direction is also true: if the MMV problem $\mathbf{P_1}(K)$ has a unique solution, it implies that the $\mathbf{P_1}$ must also have a unique solution. Details are stated in Lemma 18 in Section 4.9.2.

Since the $\ell_{1,1}$ norm of $\boldsymbol{X}$ essentially takes ℓ_1 norm of its vectorized version, it only enforces the sparsity for all elements in $\boldsymbol{X}$ without any structure such as the support consistency across its columns. To obtain the JOBS form, We first relax the $\ell_{1,1}$ norm in (4.4) to the $\ell_{1,2}$ norm. For all $j = 1, 2, .., K$

$$\mathbf{P_{12}}(K) : \ \min \|\boldsymbol{X}\|_{1,2} \ \text{ s.t. } \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_j. \tag{4.5}$$

From here, to obtain $\mathbf{J_{12}}$ in (4.1), we further drop all constraints that are not in $\mathcal{I}_j$ from (4.5) for estimator $\boldsymbol{x}_j, j = 1, 2, ..., K$. This two-step relaxation process is illustrated in Figure 4.2. Figure 4.3 gives an geometric illustration of this two-step relaxation using an example, where red surfaces are level sets of norms; blue and yellow hyper-planes are constraints; pink point is the true solution and black stars are optimization solutions. Detailed parameters are explained in Section 4.9.4.

The noisy version can be obtained similarly. We formulate the MMV version of the original ℓ_1 problem; relax the regularizer from $\ell_{1,1}$ norm to $\ell_{1,2}$ norm, and then further relax the objective function by dropping the constraints that are not on the selected subset $\mathcal{I}_j$ for the $j-$th estimate $\boldsymbol{x}_j$ to obtain the proposed form $\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon^J}}$.

Because JOBS procedure is a two-step relaxation of the ℓ_1 minimization, it gives some insight of why JOBS algorithm can correctly recover sparse solution, which is important for analyzing the algorithm. In Section 4.4, we will establish the correctness of JOBS algorithm rigorously.

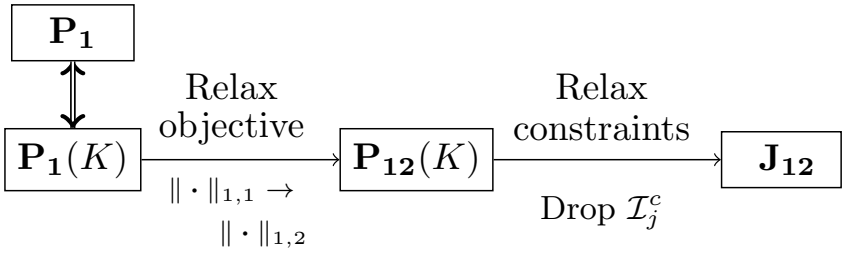
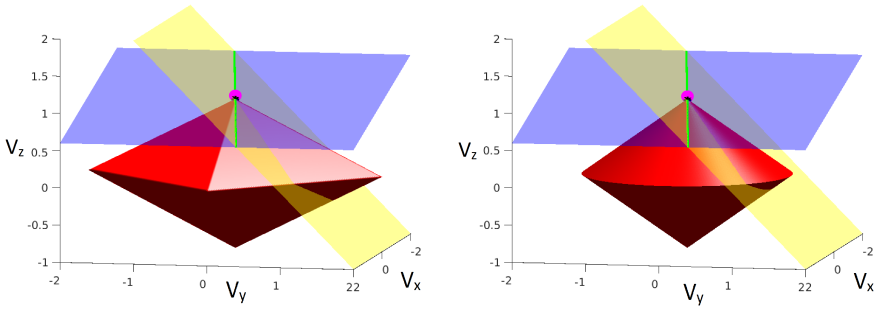
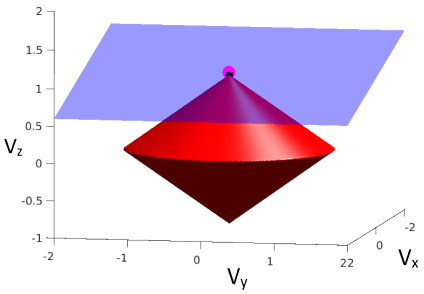


Figure 4.2: JOBS framework is a two-step relaxation of ℓ_1 minimization



(a) $\mathbf{P_1}(2)$: the $\ell_{1,1}$ norm minimization over all constraints

(b) $\mathbf{P_{12}}(2)$: the $\ell_{1,2}$ norm minimization over all constraints



(c) $\mathbf{J_{12}} : \mathcal{I}_1 = \{1\}, \mathcal{I}_2 = \{1\}$.
JOBS relaxation, drop one constraint.

Figure 4.3: JOBS is a two-step relaxation scheme of ℓ_1 minimization. The first relaxation: Relaxing from $\ell_{1,1}$ norm (Fig 4.3a) to $\ell_{1,2}$ norm (Fig. 4.3b). The second relaxation: further relaxing constraint in Fig 4.3b by dropping one constraint resulting in Fig. 4.3c. Red surfaces are level sets; Blue and yellow hyper planes are constraints; pink dot is the true solution and black dots are optimization solutions.

4.2.4 The sub-sampling Variation: Sub-JOBS

Bootstrapping (random sampling with replacement) creates duplicates within a bootstrap sample. Although it simplifies the analysis, in practice, duplicate information does not add value. One natural extension of the proposed framework is to use sub-sampling, which is sampling without replacement instead of with replacement. The sub-sampling variation of Bagging is known as Subagging

estimator (short for **Sub**sampling **Agg**regat**ing**) in the literature (Bühlmann and Yu, 2000; Buhlmann, 2003). We adopt a similar name for the sub-sampling variation of the proposed method: Sub-JOBS, where the prefix"Sub" represents sub-sampling. The only difference to the original scheme is that for each bootstrap sample $\mathcal{I}_j$, L distinct samples are generated by random sampling without replacement from m measurements rather than the conventional bootstrapping scheme. Note that, for any two different sub-sampling samples $\mathcal{I}_j, \mathcal{I}_t, j \neq t$, there may be shared samples.

in this work, all the theoretical results are for the bootstrapping version for simplicity of presentation. The numerical results and discussion for both the original bootstrapping scheme as well as the sub-sampling variation will be shown in Section 4.6.2. The connection between bootstrap and sub-sampling is also explained in details in Section 4.9.6.

4.3 Preliminaries

We summarize the theoretical results that are needed for understanding and analyzing our algorithm mathematically. We offer a quick review of several concepts including block sparsity, Null Space Property (NSP) (Cohen, Dahmen, and DeVore, 2009), Restricted Isometry Property (RIP) (Candes, 2008) for classical sparse signal recovery as well as Block Null Space Property (BNSP) (Gao, Peng, and Zhao, 2015), Block Restricted Isometry Property (BRIP) (Eldar and Mishali, 2009) for block sparse signal recovery.

4.3.1 Block Sparsity

Since row sparsity is a special case of block sparsity (or more precisely, the non-overlapping group sparsity) (Eldar and Mishali, 2009), we therefore can employ the tools from block sparsity to analyze our problem. Block sparsity is a generalization of the standard ℓ_1 sparsity. To start, we recall its definition.

Definition 7 (Block Sparsity, from (Eldar and Mishali, 2009)) $\boldsymbol{x} \in \mathbb{R}^n$ *is* $s-$*block sparse with respect to a partition* $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_b\}$ *of* $\{1, 2, ..., n\}$ *if for* $\boldsymbol{x} = (\boldsymbol{x}_{[\mathcal{B}_1]}, \boldsymbol{x}_{[\mathcal{B}_2]}, ..., \boldsymbol{x}_{[\mathcal{B}_b]})$*, the block sparsity level is* $\|\boldsymbol{x}\|_{0,2|\mathcal{B}} := \sum_{i=1}^{b} \mathbb{1}\{\|\boldsymbol{x}_{[\mathcal{B}_i]}\|_2 > 0\} \leq s$ *and the relaxation* $\ell_{1,2}$ *norm is* $\|\boldsymbol{x}\|_{1,2|\mathcal{B}} := \sum_{i=1}^{b} \|\boldsymbol{x}_{[\mathcal{B}_i]}\|_2$.

The block sparsity level $\|\boldsymbol{x}\|_{0,2|\mathcal{B}}$ counts the number of non-zero blocks of the given a block partition $\mathcal{B}$. The $\ell_{1,2}$ norm $\|\boldsymbol{x}\|_{1,2|\mathcal{B}} := \sum_{i=1}^{b} \|\boldsymbol{x}_{[\mathcal{B}_i]}\|_2$ is one of its convex relaxations. For the same sparse vector $\boldsymbol{x}$, its block sparsity level with respect to a non-overlapping block partition is in general smaller than its sparsity level.

The $\ell_{1,2}$ minimization is a special case of block sparse minimization, with each element in the block partition containing all indices of a row and the details are in Section 4.9.1. The results of block sparsity such as BNSP, BRIP can be useful tools to analyze our algorithm.

4.3.2 Block-Null Space Property (BNSP)

For standard sparse recovery, its null sparse property as described in Definition 1 and restricted isometry property as in Definition 2 are important properties for its performance bounded. Similarly, for block sparse signal recovery, there

are block Null Space Property (BNSP) and Block Restricted Isometry Property (BRIP) and crucial to study the theoretical performances.

We will stated them in the following sections. BNSP is obtained from a more general result of BNSP of $\ell_{p,2}$ block norm stated in (2.6) from (Gao, Peng, and Zhao, 2015) taking $p = 1$.

Definition 8 (BNSP, from (Gao, Peng, and Zhao, 2015)) *Every $s-$block sparse signal $\boldsymbol{x}$ with respect to block assignment $\mathcal{B}$, is a unique solution to* $\min \|\boldsymbol{x}\|_{1,2|\mathcal{B}}$ *s.t.* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ *if and only if matrix $\boldsymbol{A}$ satisfies block null space property over $\mathcal{B}$ of order s: for any set $\boldsymbol{S} \subset \{1, 2, ..., n\}$ with card$(\boldsymbol{S}) \leq s$,*

$$\|\boldsymbol{v}[\boldsymbol{S}]\ \|_{1,2|\mathcal{B}} < \|\boldsymbol{v}[\boldsymbol{S}^c]\ \|_{1,2|\mathcal{B}},$$

for all $\boldsymbol{v} \in \text{Null}(\boldsymbol{A})\backslash\{\boldsymbol{0}\}$, where $\boldsymbol{v}[\boldsymbol{S}]$ denotes the vector equal to $\boldsymbol{v}$ on a block index set $\boldsymbol{S}$ and zero elsewhere.

4.3.3 Block-Restricted Isometry Property (BRIP)

Although NSP directly characterizes the ability of success for sparse recovery, verifying the BNSP condition is computationally intractable and it is also not suitable for quantifying performance in noisy cases since it is a binary (True or False) metric instead of a continuous one. Restricted Isometry Properties: BRIP (Eldar and Mishali, 2009) are introduced for those purposes.

Definition 9 (BRIP, from (Eldar and Mishali, 2009)) *A matrix $\boldsymbol{A}$ with ℓ_2-normalized columns satisfies Block RIP with respect to block partition $\mathcal{B}$ of order s if there exists a constant $\delta_{s|\mathcal{B}}(\boldsymbol{A}) \in [0, 1)$ such that for every $s-$block*

sparse $\boldsymbol{v} \in \mathbb{R}^n$ *over* $\mathcal{B}$,

$$(1 - \delta_{s|\mathcal{B}}(\boldsymbol{A}))\|\boldsymbol{v}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{v}\|_2^2 \leq (1 + \delta_{s|\mathcal{B}}(\boldsymbol{A}))\|\boldsymbol{v}\|_2^2. \tag{4.6}$$

If we take the location of each entry as one block, the block sparsity RIP reduces to the standard RIP condition. Therefore, BRIP is a generalization of RIP.

4.3.4 Noisy Recovery Bounds based on RIP Constants

It is well-known that certain RIP conditions imply NSP conditions for both classical sparse recovery and block sparse recovery. More specifically, if the RIP constant in the order $2s$ is strictly less than $\sqrt{2} - 1$, then it implies that NSP is satisfied in the order of s. This applies to sparse recovery (Candes, 2008) and block sparse recovery (Eldar and Mishali, 2009).

Stated below are the error bound for conventional sparse recovery based on ℓ_1 minimization and the RIP constant as well as for block sparse recovery based on BRIP constant. According to Theorem 3, the sparse recovery bound is

$$\|\boldsymbol{x}^{\boldsymbol{\ell_1}} - \boldsymbol{x}^\star\|_2 \leq \mathcal{C}_0(\delta)s^{-1/2}\|\boldsymbol{e}\|_1 + \mathcal{C}_1(\delta)\epsilon,$$

where $\mathcal{C}_0(\cdot), \mathcal{C}_1(\cdot)$ are certain functions depending on the RIP constant $\delta_{2s}(\boldsymbol{A})$. They are in the form of non-decreasing functions of δ: $\mathcal{C}_0(\delta) = \frac{2(1-(1-\sqrt{2})\delta)}{1-(1+\sqrt{2})\delta}$ and $\mathcal{C}_1(\delta) = \frac{4\sqrt{1+\delta}}{1-(1+\sqrt{2})\delta}$.

Theorem 10 (Block sparse recovery error bound, from (Eldar and Mishali, 2009)) *Let* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 \leq \epsilon$; $\boldsymbol{x}_{0|\mathcal{B}}$ *is* $s-$*block sparse and minimizes* $\|\boldsymbol{x} - \boldsymbol{x}^\star\|_2$ *over all* $s-$*block sparse signals, and the vector* $\boldsymbol{e}_{\mathcal{B}}$ *represents*

the s-sparse approximation error vector $\boldsymbol{e}_{\mathcal{B}} = \boldsymbol{x}^{\star} - \boldsymbol{x}_{0|\mathcal{B}}$. *If* $\delta_{2s|\mathcal{B}} < \sqrt{2} - 1$, *then there exists a scaler* δ *such as* $\delta_{2s|\mathcal{B}}(\boldsymbol{A}) \leq \delta < \sqrt{2} - 1$, *and the solution of block sparse minimization* $\boldsymbol{x}^{\boldsymbol{\ell}_{1,2|\mathcal{B}}}$ *satisfies*

$$\|\boldsymbol{x}^{\boldsymbol{\ell}_{1,2|\mathcal{B}}} - \boldsymbol{x}^{\star}\|_2 \leq \mathcal{C}_0(\delta) s^{-1/2} \|\boldsymbol{e}_{\mathcal{B}}\|_{1,2|\mathcal{B}} + \mathcal{C}_1(\delta)\epsilon,$$

where $\mathcal{C}_0(\cdot), \mathcal{C}_1(\cdot)$ *are the same non-decreasing functions of* δ *as in Theorem 3.*

4.3.5 Sample Complexity for i.i.d. Gaussian or Bernoulli Random Matrices

With $\boldsymbol{A}$ being a random matrix in which entries are identically and independently distributed (i.i.d.), previous work in (Baraniuk et al., 2008) builds a relationship between the sample complexity for random matrices to a desired RIP constant as a direct implication from Johnson-Lindenstrauss lemma as stated below.

Theorem 11 (Sample Complexity, from (Baraniuk et al., 2008)) *Let entries of* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *from Gaussian distribution* $\mathcal{N}(0, 1/m)$ *or Bernoulli* $1/\sqrt{m}$ *Bern(*0.5*). Let* $\xi, \delta \in (0, 1)$, *and if* $m \geq \beta \delta^{-2}(s \ln(n/s) + \ln(\xi^{-1}))$ *for a universal constant* $\beta > 0$, *then* $\mathbb{P}(\delta_s(\boldsymbol{A}) \leq \delta) \geq 1 - \xi$.

By rearranging the terms in this theorem, the sample complexity can be derived: when m is sufficiently large, which is in the order of $\mathcal{O}(2s \ln(n/2s))$, there is a high probability that the RIP constant of order $2s$ is sufficiently small.

4.4 Theoretical Results

4.4.1 BNSP for JOBS

Similarly to previous CS analysis in (Candes, 2008), we first give the null space property to characterize the exact recovery condition of our algorithm. The BNSP for JOBS is stated as follows.

Lemma 12 (BNSP for JOBS) *A set of bootstrapped sensing matrices* $\{\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K]\}$ *satisfies BNSP of order* s *if* $\forall\ (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_K) \in \text{Null}(\boldsymbol{A}[\mathcal{I}_1]) \times \text{Null}(\boldsymbol{A}[\mathcal{I}_2]) ... \times \text{Null}(\boldsymbol{A}[\mathcal{I}_K]) \backslash \{(\boldsymbol{0}, \boldsymbol{0}, ..., \boldsymbol{0})\}$, *such that for all* $\boldsymbol{S}$: $\boldsymbol{S} \subset \{1, 2, ..., n\}$, $card(\boldsymbol{S}) \leq s$, $\|\boldsymbol{V}[\boldsymbol{S}]\|_{1,2} < \|\boldsymbol{V}[\boldsymbol{S}^c]\|_{1,2}$.

Theorem 13 (Correctness of JOBS) *The noiseless JOBS program* $\mathbf{J_{12}}$ *in (4.1) successfully recovers all the* $s-$*row sparse solution if and only if* $\{\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K]\}$ *satisfies BNSP of the order of* s *described in Lemma 12. The solution is of the form* $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)$, *where* $\boldsymbol{x}^\star$ *is the unique true sparse solution. Then, the JOBS solution* $\boldsymbol{x}^{\boldsymbol{J}}$, *which is the average over columns of* $\boldsymbol{X}^\star$, *is* $\boldsymbol{x}^\star$.

The BNSP of JOBS characterizes the existence and uniqueness of the solution, and Theorem 13 establishes the correctness of JOBS. Since the final estimate the average of the solution, the latter part of Theorem 13 implies that the JOBS solution is also optimal $\boldsymbol{x}^{\boldsymbol{J}} = \boldsymbol{x}^\star$.

The first part of Theorem 13 for the BNSP of JOBS can be obtained directly from Theorem 8, which is a special case in (Gao, Peng, and Zhao, 2015), whereas

the second part for the correctness of the JOBS solution can be derived by showing that $\boldsymbol{X}^\star$ is feasible and achieves the lower bound of the $\ell_{1,2}$ norm among all feasible solutions. The proof of this Theorem is shown in Section 4.5.1.

4.4.2 BRIP for JOBS

Let the JOBS block diagonal matrix $\boldsymbol{A}^{\boldsymbol{J}} = \text{block\_diag}(\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K])$, where block_diag denotes the operator that stacks matrices as a block diagonal matrices, and $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$ is the block partition of all indices of vectorized matrix $\boldsymbol{X} \in \mathbb{R}^{n\times K}$: $\text{vec}(\boldsymbol{X}) \in \mathbb{R}^{nK}$ that correspond to the row sparsity pattern. The vectorized JOBS formulation can be written as:

$$\begin{aligned} &\min_{\text{vec}(\boldsymbol{X})\in\mathbb{R}^{nK}} \|\text{vec}(\boldsymbol{X})\|_{1,2|\mathcal{B}} \ \text{ s.t.} \\ &\|\text{vec}(\boldsymbol{y}_{[\mathcal{I}_1]}, \boldsymbol{y}_{[\mathcal{I}_2]}, ..., \boldsymbol{y}_{[\mathcal{I}_K]}) \\ &- \text{block\_diag}(\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K])\text{vec}(\boldsymbol{X})\|_2^2 \leq \epsilon^J \end{aligned} \tag{4.7}$$

Let $\delta_{s|\mathcal{B}}$ denote the row sparse BRIP constant of order s over a given block partition $\mathcal{B}$ and δ_s denote the standard RIP constant of order s. We have the following proposition for JOBS.

Proposition 14 (BRIP for JOBS) *For all $s \leq n, s \in \mathbb{Z}^+$,*

$$\delta_{s|\mathcal{B}}(\boldsymbol{A}^{\boldsymbol{J}}) = \max_{j=1,2,...,K} \delta_s(\boldsymbol{A}[\mathcal{I}_j]). \tag{4.8}$$

It is not surprising at all that the BRIP of JOBS depends on the worst case among all K bootstrapped matrices since a smaller RIP constant indicates better recovery ability. If there are duplicated rows of $\boldsymbol{A}[\mathcal{I}_j]$ resulted from bootstrapping, they are removed before computing the RIP constant scaled by a

value related to the occurrence. The proof of this proposition is elaborated in Section 4.9.5.

4.4.3 Noisy Recovery for JOBS

Next, we analyze the error bound for JOBS using BNSP and BRIP in the noisy case. Note that our theorems are based on deterministic sensing matrix, measurements and noise vectors: $\boldsymbol{A}, \boldsymbol{y}, \boldsymbol{z}$ and the randomness in our framework is introduced by the bootstrap sampling process.

From previous analysis, we have established that if the BRIP constant of order $2s$ is less than $\sqrt{2}-1$, it implies that $\{\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K]\}$ satisfies BNSP of order s. Then, Theorem 13 establishes that the optimal solution to $\mathbf{J_{12}}$ in (4.1) is the $s-$row sparse signal $\boldsymbol{X}^\star$ with every column being $\boldsymbol{x}^\star$. Similar to the bound in Theorem 10, the reconstruction error is determined by the $s-$block sparse approximation error and the noise level. The Hoeffding's tail bound is used to obtain the worst case performance for JOBS. The following theorem states the performance bound for JOBS when the ground truth signal $\boldsymbol{x}^\star$ is exactly $s-$sparse.

Theorem 15 (JOBS: error bound for $\|\boldsymbol{x}^\star\|_0 = s$) *Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 < \infty$. If the BRIP constant of the JOBS matrix $\delta_{2s|\mathcal{B}}(\boldsymbol{A}^{\boldsymbol{J}}) < \sqrt{2}-1$, then there exists a constant related to parameters (L, K) such that, $\delta_{2s|\mathcal{B}}(\boldsymbol{A}^{\boldsymbol{J}}) \leq \delta_{L,K} < \sqrt{2}-1$ and when the true solution is exactly $s-$sparse, for any $\tau > 0$, JOBS solution $\boldsymbol{x}^{\boldsymbol{J}}$ satisfies*

$$\mathbb{P}\left\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 \leq \mathcal{C}_1(\delta_{L,K})(\sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2 + \tau)\right\} \geq 1 - \exp\frac{-2K\tau^4}{L\|\boldsymbol{z}\|_\infty^4}, \tag{4.9}$$

where $\mathcal{C}_1(\cdot)$ is the same non-decreasing functions of δ as in Theorem 3.

The relationship to the upper bound of RIP constant is discussed in Section 4.4.5. In the more general case, when the sparsity level of $\boldsymbol{x}^\star$ possibly exceeds s, there is no guarantee that the non $s-$sparse part will be preserved by the relaxations in the JOBS framework. Namely, let $\boldsymbol{X}^{\boldsymbol{J}\star}$ denote the true solution for the noiseless row sparse recovery program $\mathbf{J_{12}}$. If BNSP of order greater than s is not guaranteed to be satisfied, then we cannot guarantee that $\boldsymbol{X}^{\boldsymbol{J}\star} = \boldsymbol{X}^\star$. However, if $\boldsymbol{x}^\star$ is nearly $s-$sparse, then $\boldsymbol{X}^{\boldsymbol{J}\star}$ is not far away from $\boldsymbol{X}^\star$. Since $\widehat{\boldsymbol{X}}$, recovered from $\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon}^{\boldsymbol{J}}}$ in (4.2), is close to $\boldsymbol{X}^{\boldsymbol{J}\star}$ via the block sparse recovery bound, $\widehat{\boldsymbol{X}}$ must also be close enough to $\boldsymbol{X}^\star$. This result is stated in the following theorem.

Theorem 16 (JOBS: error bound for the general case) *Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{z}$, $\|\boldsymbol{z}\|_2 < \infty$. If the BRIP constant of the JOBS matrix $\delta_{2s|\mathcal{B}}(\boldsymbol{A}^{\boldsymbol{J}}) < \sqrt{2}-1$, then there exists a constant related to parameters (L, K) such that, $\delta_{2s|\mathcal{B}}(\boldsymbol{A}^{\boldsymbol{J}}) \leq \delta_{L,K} < \sqrt{2}-1$, and for any $\tau > 0$, JOBS solution $\boldsymbol{x}^{\boldsymbol{J}}$ satisfies*

$$\begin{aligned} &\mathbb{P}\left\{ \|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 \leq \|\boldsymbol{e}\|_2 + \mathcal{C}_1(\delta_{L,K})\left(\sqrt{\frac{L}{m}}\|\boldsymbol{A}\boldsymbol{e}+\boldsymbol{z}\|_2 + \tau\right)\right\} \\ &\geq 1 - \exp\frac{-2K\tau^4}{L(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}, \end{aligned} \tag{4.10}$$

where $\mathcal{C}_1(\cdot)$ is the same non-decreasing function of δ as in Theorem 3; $\boldsymbol{e}$ is the s-sparse approximation error: $\boldsymbol{e} = \boldsymbol{x}^\star - \boldsymbol{x}_0$ with $\boldsymbol{x}_0$ containing the largest s components of the true solution $\boldsymbol{x}^\star$; and $\|\boldsymbol{A}\|_{\infty,1} = \max_{i=1,2,\ldots,m}(\|\boldsymbol{a}[i]^{\boldsymbol{T}}\|_1)$ denotes the largest ℓ_1-norm of all rows of $\boldsymbol{A}$.

The error bound in Theorem 16 relates to $s-$sparse approximation error

as well as the noise level, which is similar to ℓ_1 minimization and block sparse recovery bounds. JOBS also introduces a relaxation error bounded by $\|\boldsymbol{e}\|_2$. The smaller the power of $\boldsymbol{e}$, the lower the upper bound will be. When $\boldsymbol{e} = \mathbf{0}$, $\boldsymbol{x}^\star$ is exactly $s-$sparse, then Theorem 16 reduces to Theorem 15.

We use Theorem 16 to explain the case when the number of measurements is low compared to the true sparsity level s. The trade-offs for a good choice of the bootstrap sample size L and the number of bootstrap samples K are discussed in Section 4.4.5.

4.4.4 Comparison to Noisy Recovery for Bagging in Sparse Recovery

We also derive the performance bound for employing the Bagging scheme in sparse recovery problems, in which the final estimate is the average over multiple estimates solved individually from bootstrap samples. We derive the theoretical results for the case that the true signal $\boldsymbol{x}^\star$ is exactly $s-$sparse and the general case that it is only approximately $s-$sparse. The results are in previous chapter in Theorem 5 and Theorem 6.

It is interesting to contrast the error bound for JOBS compared to Bagging. The RIP condition for Bagging is the same as the RIP condition for JOBS, under the assumption that all bootstrapped matrices $\boldsymbol{A}[\mathcal{I}_j]$s are well-behaved in the worst case analysis. When $\|\boldsymbol{x}^\star\|_0 = s$, the bound in Bagging is worse than JOBS since the certainty for algorithm is at least $1 - \exp \frac{-2K\tau^4}{L^2\|\boldsymbol{z}\|_\infty^4}$, compared to the certainty bound $1 - \exp \frac{-2K\tau^4}{L\|\boldsymbol{z}\|_\infty^4}$ in JOBS. With a L squared term instead of L in the denominator, the certainty bound is larger for JOBS given the same

choices of L and K.

As for the general signal recovery bound of Bagging in Theorem 6, the error bound for bagging does not contain the multiple vector measurements relaxation error as the one in JOBS. On the other hand, the uncertainty term in the exponential term involves more complicated terms. This bound is nontrivial comparing to the one for JOBS. Although the $s-$sparse assumption limits to exact $s-$sparse signals, for signals that are approximately $s-$sparse, or with low energy in the $s-$sparse approximation (i.e., $\|\boldsymbol{e}\|_1$ is small), the behavior would be close to that of the exact $s-$sparse case.

4.4.5 Parameters Selection from Theoretical Analysis

Our analysis of error bounds of JOBS actually guides us to the optimal choices of two important parameters: the bootstrap sample ratio L/m and the number of bootstrap samples K. We focus on analyzing how the error bound in (4.10) of Theorem 16 provides guidance to optimal parameter setting.

First, consider the sampling ratio L/m. The BRIP constant in general decreases with increasing L. The intuition behind this trend is that taking more measurements tends to gain a better ability to preserve the information of the signal after projection. The RIP constant for ℓ_1 minimization was proven to be smaller with high probability for a larger number of measurements (Baraniuk et al., 2008), i.e., a larger L indicates a smaller RIP constant. Although the result is based on the assumption that the sensing matrix is either random Gaussian or Bernoulli, in practice, for general matrices, more measurements leads to better recovery which indicates a smaller upper bound of RIP constant.

Increasing L decreases the upper bound of RIP constant for each $\boldsymbol{A}[\mathcal{I}_j]$. According to the BRIP constant for JOBS that we have proven in Proposition 14, the BRIP is the maximum RIP constant over K bootstrapped matrices. With increasing L, the upper bound of RIP constant for each $\boldsymbol{A}[\mathcal{I}_j]$ increases, therefore BRIP constant will also become smaller. Next, since $\mathcal{C}_1(\delta)$ is a non-decreasing function of δ and a larger L results in a smaller δ, one can see that $\mathcal{C}_1(\delta)$ is also smaller. On the other hand, the second factor associated with the noise power term, $\sqrt{L/m}$, suggests a smaller L to obtain a smaller upper bound for the noise energy.

Combining these two factors indicates that the best L/m ratio should be somewhere in between. In the experimental results, we show that when m is small, varying L/m from $0-1$ creates peaks with the largest value at $L/m \approx 0.4$. However, the first factor, which is the relationship between the BRIP constant and L, is dominating in the stable case (when m is large), so that larger L leads to better performances.

As for the number of estimates K, increasing K has a weak effect in increasing the BRIP constant. However, the maximum is taken over K RIP constants. Consequently, for any JOBS matrix $\boldsymbol{A}^{\boldsymbol{J}}$ generated from $K+1$ bootstrap samples, there exists one generated from K bootstrap samples with smaller or equal BRIP constant. Furthermore, the sample complexity result in (Baraniuk et al., 2008) shows that for the random matrices with the same sizes, the RIP constants are fairly concentrated. It is reasonable to deduce that increasing K does not increase the BRIP by a significant margin. In the sparse regression simulation, we find that increasing K in general does not reduce the performance.

The number of estimates K mainly affects the uncertainty in (4.10), which decays exponentially with K, so a large K is indeed preferable in this sense. The certainty can be written as $p(K) = 1 - \exp\{-\alpha K\}$, for some constant $\alpha > 0$. By taking the derivative $p'(K) = \alpha \exp\{-\alpha K\} > 0$, we know that the growth rate of $p(x)$ is non-negative and decreasing with K. This phenomenon is also verified in our simulation. Although increasing K will in general improve the results, the performance tends to be flattened out, and the improvement margin decreases.

4.5 Proofs of Main Theoretical Results in Section 4.4

4.5.1 Proof of Theorem 13: Correctness of JOBS

The first part of Theorem 13 can be directly shown from the BNSP for block sparse minimization problems as in (Eldar and Mishali, 2009). We only need to show the procedure to prove the latter part. If BNSP of order s is satisfied for $\{\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K]\}$, then each bootstrap matrix $\boldsymbol{A}[\mathcal{I}_j]$ satisfies the Null Space Property (NSP) of order s, which is proven in Section 4.9.3. Consequently, for all $j = 1, 2, ..., K$, $\boldsymbol{x}^\star$ also turns out to be the optimal solution to all estimators: $\boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}_j} \|\boldsymbol{x}_j\|_1 \;\text{ s.t. }\; \boldsymbol{y}_{[\mathcal{I}_j]} = \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j$.

For $\boldsymbol{X}$ to be a feasible solution, consider its $\ell_{1,2}$ norm, we have:

$$\|\boldsymbol{X}\|_{1,2} = \sum_{i=1}^{n}(\sum_{j=1}^{K}(x_{ij}^2))^{1/2} = \sqrt{K}\sum_{i=1}^{n}(\frac{1}{K}\sum_{j=1}^{K}(x_{ij}^2))^{1/2}.$$

By concavity of the square root, we have

$$
\begin{aligned}
&\|\boldsymbol{X}\|_{1,2} \\
&\geq \sqrt{K}\sum_{i=1}^{n}\frac{1}{K}\sum_{j=1}^{K}\sqrt{x_{ij}^2} = \sqrt{K}\frac{1}{K}\sum_{j=1}^{K}\sum_{i=1}^{n}|x_{ij}| \\
&\geq \sqrt{K}\frac{1}{K}\sum_{j=1}^{K}\min_{\substack{\boldsymbol{x}_j:x_{1j},\ldots,x_{nj} \\ \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j=\boldsymbol{y}[\mathcal{I}_j]}}\sum_{i=1}^{n}|x_{ij}| \\
&= \sqrt{K}\frac{1}{K}\sum_{j=1}^{K}\min_{\boldsymbol{x}_j:\boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j=\boldsymbol{y}[\mathcal{I}_j]}\|\boldsymbol{x}_j\|_1 \\
&= \sqrt{K}\|\boldsymbol{x}^\star\|_1.
\end{aligned}
$$

Since $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)$ is a feasible solution and $\|\boldsymbol{X}^\star\|_{1,2} = \|(\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)\|_{1,2}$ $= \sqrt{K}\|\boldsymbol{x}^\star\|_1$, it achieves the lower bound. By the uniqueness part of the theorem, we can concluded that $\boldsymbol{X}^\star$ is the unique solution. Since the JOBS solution takes the average over columns of multiple estimates, we can easily deduce that JOBS returns the correct answer.

4.5.2 Proof of Theorem 15: JOBS Performance Bound of for Exactly $s-$sparse Signals

If the true solution is exactly $s-$sparse, the sparse approximation error is zero. Then the noise level of performance only relates to measurements noise. For ℓ_1 minimization, $\boldsymbol{z}$ is the noise vector and we use matrix $\boldsymbol{Z} = (\boldsymbol{z}_{[\mathcal{I}_1]}, \boldsymbol{z}_{[\mathcal{I}_2]}, ..., \boldsymbol{z}_{[\mathcal{I}_K]})$ to denote the noise matrix in JOBS. We bound the distance of $\|\boldsymbol{Z}\|_{2,2}$ to its expected value using Hoeffding's inequalities stated in (Hoeffding, 1963).

Theorem 17 (Hoeffding's Inequalities) *Let $X_1, ..., X_n$ be independent bounded random variables such that X_i falls in the interval $[a_i, b_i]$ with probability one.*

Denote their sum by $S_n = \sum_{i=1}^{n} X_i$. *Then for any* $\zeta > 0$, *we have:*

$$\mathbb{P}\Big\{S_n - \mathbb{E}\boldsymbol{S}_n \geq \zeta\Big\} \leq \exp \frac{-2\zeta^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \quad \textit{and} \tag{4.11}$$

$$\mathbb{P}\Big\{S_n - \mathbb{E}\boldsymbol{S}_n \leq -\zeta\Big\} \leq \exp \frac{-2\zeta^2}{\sum_{i=1}^{n}(b_i - a_i)^2}. \tag{4.12}$$

Here, the entire noise vector is $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y} = (z[1], z[2], ..., z[m])^T$, $\|\boldsymbol{z}\|_\infty = \max_{i=1,2,...,m} |z[i]| < \infty$. We consider the matrix $\boldsymbol{Z} \circ \boldsymbol{Z} = (\xi_{ji})$, where $\circ$ is the entry-wise product. The quantity that we are interested in $\|\boldsymbol{Z}\|_{2,2}$ is the sum of all entries in $\boldsymbol{Z} \circ \boldsymbol{Z}$. Each element in this matrix $\boldsymbol{Z} \circ \boldsymbol{Z}$ is drawn i.i.d from the squares of entries in $\boldsymbol{z}$: $\{z[1], z[2], ..., z[m]\}$ with equal probability. Let Ξ be the underlining random variable and Ξ obeys a discrete uniform distribution:

$$\mathbb{P}(\Xi = z^2[i]) = \frac{1}{m}, i = 1, 2, ..., m. \tag{4.13}$$

The lower and upper bound of Ξ is then

$$0 \leq \min_i z^2[i] \leq \Xi \leq \|\boldsymbol{z}\|_\infty^2. \tag{4.14}$$

We use zero as lower bound for Ξ instead of the minimum value to simplify the terms. The expected power of $\boldsymbol{Z}$ is

$$\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 = \frac{KL}{m}\|\boldsymbol{z}\|_2^2. \tag{4.15}$$

Applying Hoeffding's inequality for any $\tau > 0$ leads to

$$\mathbb{P}\{\|\boldsymbol{Z}\|_{2,2}^2 - \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 - \tau \leq 0\} \geq 1 - \exp \frac{-2\tau^2}{KL\|\boldsymbol{z}\|_\infty^4}. \tag{4.16}$$

Next, let $\widehat{\boldsymbol{X}}$ be the solution of $\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon}^J}$. Theorem 10 yields

$$\mathbb{P}\{\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_{2,2}^2 - \mathcal{C}_1^2(\delta)\|\boldsymbol{Z}\|_{2,2}^2 \leq 0\} = 1. \tag{4.17}$$

Let Δ denote the difference between the solution to the truth solution scaled by the $\mathcal{C}_1$ constant. Hence, $\Delta = \frac{1}{\mathcal{C}_1(\delta)}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_{2,2}$ and (4.17) becomes

$$\mathbb{P}\{\Delta - \|\boldsymbol{Z}\|_{2,2} \leq 0\} = 1. \tag{4.18}$$

Since $\boldsymbol{Z}$ depends on the choice of $\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_K$, we derive the typical performance by studying the distance of the solution to the expected noise level of JOBS.

$$\begin{aligned}
&\mathbb{P}\{\Delta^2 - \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&= \mathbb{P}\{\Delta^2 - \|\boldsymbol{Z}\|_{2,2}^2 + \|\boldsymbol{Z}\|_{2,2}^2 - \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&\geq \mathbb{P}\{\Delta^2 - \|\boldsymbol{Z}\|_{2,2}^2 \leq 0, \|\boldsymbol{Z}\|_{2,2}^2 - \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&\quad \text{(The first and the second parts are independent)} \\
&= \mathbb{P}\{\Delta^2 - \|\boldsymbol{Z}\|_{2,2}^2 \leq 0\}\mathbb{P}\{\|\boldsymbol{Z}\|_{2,2}^2 - \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&\quad \text{(using (4.18) and (4.16))} \\
&\geq 1 - \exp\frac{-2\tau^4}{KL\|\boldsymbol{z}\|_\infty^4}.
\end{aligned}$$

In summary, this procedure results in

$$\mathbb{P}\{\Delta^2 \leq \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 + \tau^2\} \geq 1 - \exp\frac{-2\tau^4}{KL\|\boldsymbol{z}\|_\infty^4}. \tag{4.19}$$

We can bound the squared error as follows:

$$\begin{aligned}
&\mathbb{P}\{\Delta \leq (\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau\} \\
&= \mathbb{P}\{\Delta^2 \leq \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 + \tau^2 + 2\tau(\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2}\} \qquad (4.20) \\
&\geq \mathbb{P}\{\Delta^2 \leq \mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2 + \tau^2\}.
\end{aligned}$$

Combining (4.19) and (4.20), we arrive at

$$\mathbb{P}\{\Delta \leq (\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau\} \geq 1 - \exp\frac{-2\tau^4}{KL\|\boldsymbol{z}\|_\infty^4}. \qquad (4.21)$$

Since $f(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}^\star\|_2^2$ is convex, we can apply Jensens' inequality to establish:

$$\|\frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{x}}_j - \boldsymbol{x}^\star\|_2^2 \leq \frac{1}{K}\sum_{j=1}^{K}\|\widehat{\boldsymbol{x}}_j - \boldsymbol{x}^\star\|_2^2. \qquad (4.22)$$

The JOBS estimate is averaged column-wise over all estimates: $\boldsymbol{x}^{\boldsymbol{J}} = \frac{1}{K}\sum_{j=1}^{K}\widehat{\boldsymbol{x}}_j$. Therefore, equation (4.22) is essentially

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2^2 - \frac{1}{K}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_{2,2}^2 \leq 0\} = 1. \qquad (4.23)$$

Now, we consider the typical performance of the JOBS solution and recall that Δ denotes the difference between the solution to the truth solution scaled

by the $\mathcal{C}_1$: $\Delta = \frac{1}{\mathcal{C}_1(\delta)}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_{2,2}$. We can then bound the probability of error.

$$\begin{aligned}
&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 - \frac{\mathcal{C}_1(\delta)}{\sqrt{K}}((\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau) \leq 0\} \\
=&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 - \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_2 \\
&+ \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_2 - \frac{\mathcal{C}_1(\delta)}{\sqrt{K}}((\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau) \leq 0\} \\
\geq&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 - \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_2 \leq 0, \Delta \leq (\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau\} \\
=&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 - \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^\star\|_2 \leq 0\}\mathbb{P}\{\Delta \leq (\mathbb{E}\|\boldsymbol{Z}\|_{2,2}^2)^{1/2} + \tau\} \\
&\text{(by (4.23) and (4.21))} \\
\geq&1 - \exp\frac{-2\tau^4}{KL\|\boldsymbol{z}\|_\infty^4}.
\end{aligned}$$

Substituting the expected noise level derived in (4.15) yields

$$\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^\star\|_2 \leq \mathcal{C}_1(\delta)(\sqrt{\frac{L}{m}}\|\boldsymbol{z}\|_2 + \frac{\tau}{\sqrt{K}})\} \geq 1 - \exp\frac{-2\tau^4}{KL\|\boldsymbol{z}\|_\infty^4}.$$

By replacing $\tau/\sqrt{K}$ with τ, the quantity on the right hand side of the equation then becomes $1 - \exp\frac{-2K\tau^4}{L\|\boldsymbol{z}\|_\infty^4}$ and we have proved the theorem.

4.5.3 Proof of Theorem 16: JOBS Performance Bound of JOBS for General Sparse Signals

Now we consider the case that the BNSP is only satisfied for order s whereas there is no $s-$sparse assumption on the true solution. Therefore, the JOBS algorithm can only guarantee the correctness of the $s-$row-sparse part and our

best hope is to be able to recover the best $s-$sparse approximation of the true solution. Let $\boldsymbol{x_0}$ be the best $s-$sparse approximation of the true solution $\boldsymbol{x^\star}$ and $\boldsymbol{e}$ denote the difference of the sparse approximation: $\boldsymbol{e} = \boldsymbol{x^\star} - \boldsymbol{x}_0$. We rewrite the measurements to include the $s-$sparse approximation error as part of noise: for $j = 1, 2, ..., K$,

$$\begin{aligned} \boldsymbol{y}_{[\mathcal{I}_j]} &= \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x^\star} + \boldsymbol{z}_{[\mathcal{I}_j]} \\ &= \boldsymbol{A}[\mathcal{I}_j](\boldsymbol{x}_0 + (\boldsymbol{x^\star} - \boldsymbol{x}_0)) + \boldsymbol{z}_{[\mathcal{I}_j]} \\ &= \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_0 + \widetilde{\boldsymbol{z}}_j, \end{aligned} \tag{4.24}$$

where $\widetilde{\boldsymbol{z}}_j = \boldsymbol{A}[\mathcal{I}_j](\boldsymbol{x^\star} - \boldsymbol{x}_0) + \boldsymbol{z}_{[\mathcal{I}_j]} = \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{e} + \boldsymbol{z}_{[\mathcal{I}_j]}$.

To bound the distance of solution of $\mathbf{J}_{\mathbf{12}}^{\boldsymbol{\epsilon}\boldsymbol{J}}$: $\widehat{\boldsymbol{X}}$ to the true solution $\boldsymbol{X^\star}$, we evaluate its distance to the exactly s row-sparse matrix $\boldsymbol{X_0} = (\boldsymbol{x_0}, \boldsymbol{x_0}, ..., \boldsymbol{x_0})$ as an intermediate step. Since $\boldsymbol{e} = \boldsymbol{x^\star} - \boldsymbol{x_0}$, we have: $\boldsymbol{X^\star} - \boldsymbol{X_0} = (\boldsymbol{e}, \boldsymbol{e}, ..., \boldsymbol{e})$ and $\|\boldsymbol{X_0} - \boldsymbol{X^\star}\|_{2,2} = \sqrt{K}\|\boldsymbol{e}\|_2$. Then, the distance of $\widehat{\boldsymbol{X}}$ to the true solution $\boldsymbol{X^\star}$ can be decomposed into two components:

$$\begin{aligned} \|\widehat{\boldsymbol{X}} - \boldsymbol{X^\star}\|_{2,2} &= \|\widehat{\boldsymbol{X}} - \boldsymbol{X_0} + \boldsymbol{X_0} - \boldsymbol{X^\star}\|_{2,2} \\ &\leq \|\widehat{\boldsymbol{X}} - \boldsymbol{X_0}\|_{2,2} + \|\boldsymbol{X_0} - \boldsymbol{X^\star}\|_{2,2} \\ &= \|\widehat{\boldsymbol{X}} - \boldsymbol{X_0}\|_{2,2} + \sqrt{K}\|\boldsymbol{e}\|_2. \end{aligned} \tag{4.25}$$

To bound the first component in (4.25): $\|\widehat{\boldsymbol{X}} - \boldsymbol{X_0}\|_{2,2}$, the procedure is similar to the prove the exactly $s-$sparse case. We use the recovery guarantee from the row sparse recovery result in Theorem 10, which gives an upper bound

of this term associated with the power of the noise matrix $\widetilde{\boldsymbol{Z}} = (\widetilde{\boldsymbol{z}}_1, \widetilde{\boldsymbol{z}}_2, ..., \widetilde{\boldsymbol{z}}_K)$:

$$\begin{aligned} \|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 &= \sum_{j=1}^{K} \|\widetilde{\boldsymbol{z}}_j\|_2^2 = \sum_{j=1}^{K} \|\boldsymbol{A}[\mathcal{I}_j]\boldsymbol{e} + \boldsymbol{z}_{[\mathcal{I}_j]}\|_2^2 \\ &= \sum_{j=1}^{K} \sum_{i \in \mathcal{I}_j} (\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle + z[i])^2. \end{aligned} \tag{4.26}$$

Next, let $\widetilde{\Xi} = (\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle + z[i])^2$ with $\boldsymbol{a}[i], \boldsymbol{z}[i]$ generated uniformly from all rows of $\boldsymbol{A}$ and $\boldsymbol{z}$. Since $\widetilde{\Xi}$ is non-negative, $\Xi \geq 0$, the lower bound is 0. Its upper bound can be derived using the Hölders inequality:

$$\begin{aligned} \widetilde{\Xi} = (\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle + z[i])^2 &\leq (\|\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle\|_1 + \|\boldsymbol{z}\|_\infty)^2 \\ &\leq (\|\boldsymbol{a}[i]^T\|_1 \|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^2 \\ &\leq (\max_i \|\boldsymbol{a}[i]^T\|_1 \|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^2 \\ &= (\|\boldsymbol{A}\|_{\infty,1} \|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^2, \end{aligned} \tag{4.27}$$

where $\|\boldsymbol{A}\|_{\infty,1} = \max_{i=1,2,..,m} \|\boldsymbol{a}[i]^T\|_1$. Since $\boldsymbol{A}$ is deterministic with all bounded entries, the quantity $\|\boldsymbol{A}\|_{\infty,1}$ is bounded.

Also, from (4.26), the expectation of $\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2$ is

$$\begin{aligned} \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 &= \sum_{j=1}^{K} \sum_{i \in \mathcal{I}_j} \mathbb{E}(\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle)^2 + 2\mathbb{E}z[i]\langle \boldsymbol{a}[i], \boldsymbol{e} \rangle + \mathbb{E}z[i]^2 \\ &= \frac{KL}{m} \|\boldsymbol{A}\boldsymbol{e} + \boldsymbol{z}\|_2^2. \end{aligned} \tag{4.28}$$

Obtaining the the lower and upper bound of $\widetilde{\Xi}$, we can then apply Hoeffding's inequality to get the tail bound of $\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2$. It can be written as follows: for any

$\tau > 0$,

$$\mathbb{P}\{\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \tau \leq 0\} \geq 1 - \exp \frac{-2\tau^2}{KL(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}. \quad (4.29)$$

Similarly, as in the proof of Theorem 15, here we consider the distance from the recovered solution $\widehat{\boldsymbol{X}}$ to the exactly $s-$row-sparse solution $\boldsymbol{X_0}$. Let $\widetilde{\Delta}$ be $\widetilde{\Delta} = \frac{1}{\mathcal{C}_1(\delta)}\|\widehat{\boldsymbol{X}} - \boldsymbol{X_0}\|_{2,2}$ and, according to Theorem 10, we have

$$\mathbb{P}\{\|\widetilde{\Delta} - \|\widetilde{\boldsymbol{Z}}\|_{2,2} \leq 0\} = 1. \quad (4.30)$$

Combing (4.29) and (4.30) allows us to conclude that

$$\begin{aligned}
&\mathbb{P}\{\widetilde{\Delta}^2 - \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&= \mathbb{P}\{\widetilde{\Delta}^2 - \|\boldsymbol{Z}\|_{2,2}^2 + \|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&\geq \mathbb{P}\{\widetilde{\Delta}^2 - \|\boldsymbol{Z}\|_{2,2}^2 \leq 0, \|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&= \mathbb{P}\{\widetilde{\Delta}^2 - \|\boldsymbol{Z}\|_{2,2}^2 \leq 0\}\mathbb{P}\{\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 - \tau^2 \leq 0\} \\
&\geq 1 - \exp \frac{-2\tau^4}{KL(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}.
\end{aligned}$$

We can then bound the expected square root of noise power:

$$\begin{aligned}
&\mathbb{P}\{\widetilde{\Delta} \leq (\mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2)^{1/2} + \tau\} \quad \text{(by (4.20))} \\
&\geq \mathbb{P}\{\widetilde{\Delta}^2 \leq \mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2 + \tau^2\} \\
&\geq 1 - \exp \frac{-2\tau^4}{KL(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}.
\end{aligned} \quad (4.31)$$

The final JOBS estimates $\boldsymbol{x^J}$ is $\boldsymbol{x^J} = \frac{1}{K}\sum_{j=1}^K \widehat{\boldsymbol{x}}_j$ and as a direct result of (4.23),

we have

$$\begin{aligned}
\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^{\star}\|_2 &\leq \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}^{\star}\|_{2,2} \\
&\text{(by (4.25))} \\
&\leq \frac{1}{\sqrt{K}}\|\widehat{\boldsymbol{X}} - \boldsymbol{X}_{\boldsymbol{0}}\|_{2,2} + \|\boldsymbol{e}\|_2 = \frac{\mathcal{C}_1(\delta)\widetilde{\Delta}}{\sqrt{K}} + \|\boldsymbol{e}\|_2.
\end{aligned} \tag{4.32}$$

Combing the results from (4.31) and (4.32) yields

$$\begin{aligned}
&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^{\star}\|_2 \leq \frac{\mathcal{C}_1(\delta)((\mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2)^{1/2} + \tau)}{\sqrt{K}} + \|\boldsymbol{e}\|_2\} \\
\geq &\mathbb{P}\{\frac{\mathcal{C}_1(\delta)\widetilde{\Delta}}{\sqrt{K}} + \|\boldsymbol{e}\|_2 \leq \frac{\mathcal{C}_1(\delta)((\mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2)^{1/2} + \tau)}{\sqrt{K}} + \|\boldsymbol{e}\|_2\} \\
= &\mathbb{P}\{\widetilde{\Delta} \leq (\mathbb{E}\|\widetilde{\boldsymbol{Z}}\|_{2,2}^2)^{1/2} + \tau\} \\
\geq &1 - \exp\frac{-2k\tau^4}{(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}.
\end{aligned} \tag{4.33}$$

Finally, by substituting in the expected noise level derived in (4.28), we arrive at

$$\begin{aligned}
&\mathbb{P}\{\|\boldsymbol{x}^{\boldsymbol{J}} - \boldsymbol{x}^{\star}\|_2 \leq \mathcal{C}_1(\delta)(\sqrt{\frac{L}{m}}\|\boldsymbol{A}\boldsymbol{e} + \boldsymbol{z}\|_2 + \frac{\tau}{\sqrt{K}}) + \|\boldsymbol{e}\|_2\} \\
&\geq 1 - \exp\frac{-2\tau^4}{KL(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}.
\end{aligned} \tag{4.34}$$

Replacing τ with $\tau/\sqrt{K}$, the quantity on the right hand side of the equation then becomes $1 - \exp\frac{-2K\tau^4}{L(\|\boldsymbol{A}\|_{\infty,1}\|\boldsymbol{e}\|_\infty + \|\boldsymbol{z}\|_\infty)^4}$ and we have proved the theorem.

4.6 Experimental Results on Sparse Regression

In this section, we perform sparse recovery on a generic synthetic dataset to study the performance of the proposed algorithm. In our experiment, all entries of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are i.i.d. samples from the standard normal distribution $\mathcal{N}(0, 1)$. The signal dimension $n = 200$, and various numbers of measurements from 50 to 200 are explored. We will focus on the results with small number of measurements from 50 to 150, where JOBS has an advantage over the conventional ℓ_1-minimization. The ground truth signals $\boldsymbol{x}^\star$ have their sparsity levels set to $s = 50$. The location of each non-zeros entry is selected uniformly at random whereas its magnitude is sampled from the standard Gaussian distribution. For the noise processes $\boldsymbol{z}$, all entries are sampled i.i.d. from $\mathcal{N}(0, \sigma^2)$, with variance $\sigma^2 = 10^{-\mathrm{SNR}/10}\|\boldsymbol{A}\boldsymbol{x}\|_2^2$, where SNR represents the Signal-to-Noise Ratio. In our experiment, we study three different noise levels: when SNR $= 0, 1$ and 2 dB.

We employ the Alternating Direction Method of Multipliers (ADMM) implementation of block (group) Lasso (Boyd et al., 2011) to solve the unconstrained version of the noisy form of JOBS as in (4.2). The parameter $\lambda_{L,K} > 0$ balances the least squares fit and the joint sparsity penalty based on the choice of bootstrap sampling size and number of bootstrap samples (L, K):

$$\min_{\boldsymbol{X}} \ \lambda_{L,K}\|\boldsymbol{X}\|_{1,2} + \frac{1}{2}\sum_{j=1}^{K} \|\boldsymbol{y}_{[\mathcal{I}_j]} - \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j\|_2^2. \tag{4.35}$$

The same solver is used to solve Bagging, Bolasso, and ℓ_1-minimization with $K = 1$ for a fair comparison. The implementation details of ADMM for solving JOBS optimization is in Section 4.2.2.

We explore how two key parameters – the number of estimates K and the

bootstrapping ratio L/m – affect sparse regression results. In our experiment, we vary $K = 30, 50, 100$ while setting the bootstrap ratio L/m from 0.1 to 1 with an increment of 0.1. We report the average recovered Signal to Noise Ratio (SNR) as the error measure to evaluate the recovery performance: $\text{SNR}(\widehat{\boldsymbol{x}}, \boldsymbol{x}^\star) = -10 \log_{10} \|\widehat{\boldsymbol{x}} - \boldsymbol{x}^\star\|_2^2 / \|\boldsymbol{x}^\star\|_2^2$ (dB) averaged over 20 independent trials. For all algorithms, we vary the balancing parameter $\lambda_{L,K}$ at different values from .01 to 200 and then select the optimal value that gives the maximum averaged SNR over all trials at each (L, K).

4.6.1 Performance of JOBS, Bagging, Bolasso and ℓ_1 minimization with Small Number of Measurements

We study the performance of JOBS, Bagging and Bolasso, as well as ℓ_1 minimization, using the same parameters (L, K). We plot the performance of JOBS and Bagging with various bootstrap sampling ratios L/m and the number of estimates K in Figure 4.4 and Figure 4.5 for four different total numbers of measurements m ranging from 50 to 150. The solid curves show ours results with various number of estimates K. The grey circle highlights the best performance whereas the grey area highlights the optimal bootstrap ratio L/m. The smaller the grey area, the smaller the optimal bootstrap ratio is. In these figures, for each condition with a particular choice of (L, K), the information available to JOBS, Bagging and Bolasso algorithms is identical and ℓ_1-minimization always has access to all m measurements. The performance of ℓ_1 minimization is depicted by the black dashed lines. Since the performance of Bolasso is much lower than the other algorithms, only the best Bolasso performances among all choices of $(L/m, K)$ are shown using the green dashed lines.

From Figure 4.4, we can observe that when the number of measurements m is limited, JOBS outperforms ℓ_1 minimization significantly. As m increases in Figure 4.5, the margin decreases. When the number of measurements is low (the sparsity level $s = 50$ and m is only $50 - 150$, which is between $1 \times s$ to $3 \times s$), and with very small bootstrap sampling ratio L/m (L/m is only $0.3 - 0.5$), JOBS and Bagging are quite robust and outperform all other algorithms using the same parameters (L, K). In addition, although JOBS and Bagging are similar in terms of the best performance limit, Bagging requires higher L/m ratios (typically ≥ 0.6) to achieve peak performance. The grey area in Figure 4.4 and Figure 4.5 highlighting the optimal bootstrap ratio L/m is further left for JOBS than for Bagging. In short, applying the correct prior on multiple estimates shows its advantage most prominently when the total amount of data is limited. However, when the level of measurements is high enough, bootstrapping loses its advantages and ℓ_1 becomes the preferred strategy. We will see in the following subsection that for the alternative sub-sampling scheme, JOBS at least reaches ℓ_1 minimization solution when the number of measurements is high. Finally, increasing the number of bootstrap vectors (estimates) K seems to improve recovery results in general.

4.6.2 Results for the sub-sampling Variation: Sub-JOBS

In a similar manner as the previous section, we study the performance of the sub-sampling variation of the original framework: Sub-JOBS where the prefix "Sub-" denotes sub-sampling. We study how varying the sub-sampling ratio L/m from 0.1 to 1 as well as the number of estimates K from 30 to 100 affects the result. The same variation is also adopted in Bagging and Bolasso for comparison.

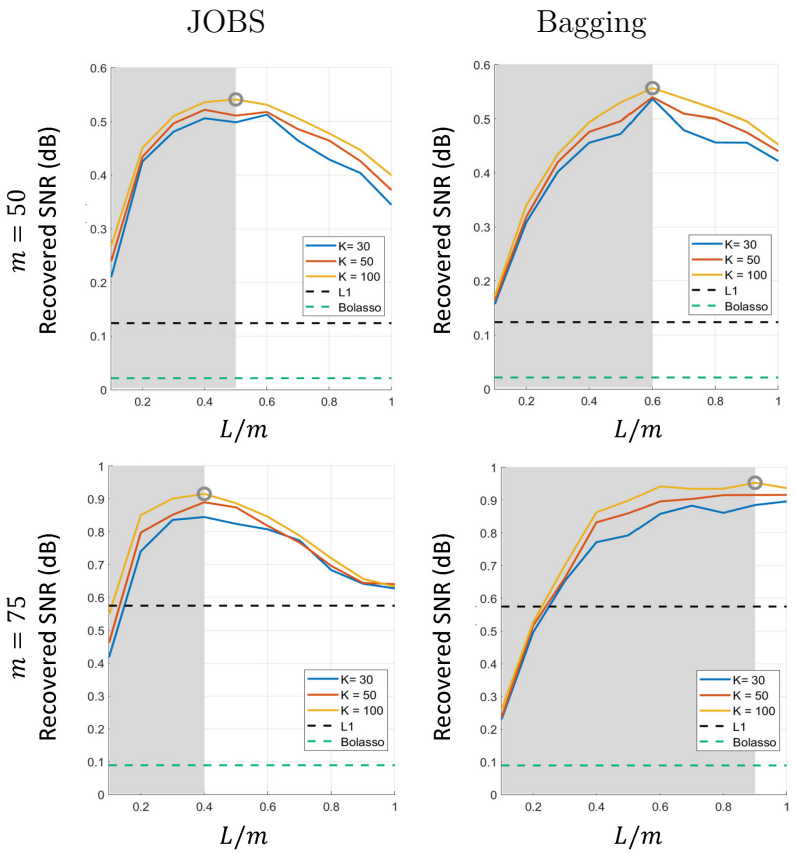


Figure 4.4: Recovery SNR (dB) performance curves for JOBS and Bagging (with various L,K) versus the peak Bolasso performance among various L,K and ℓ_1 minimization. The number of measurements are $m = 50, 75$ from top to bottom. Noise level is set to SNR = 0 dB. The grey circles highlight peaks while the grey area highlights the optimal bootstrap ratio. The optimal JOBS bootstrap ratio is smaller than that of Bagging. The y-axis of plots in the same row has been calibrated to have the same range.

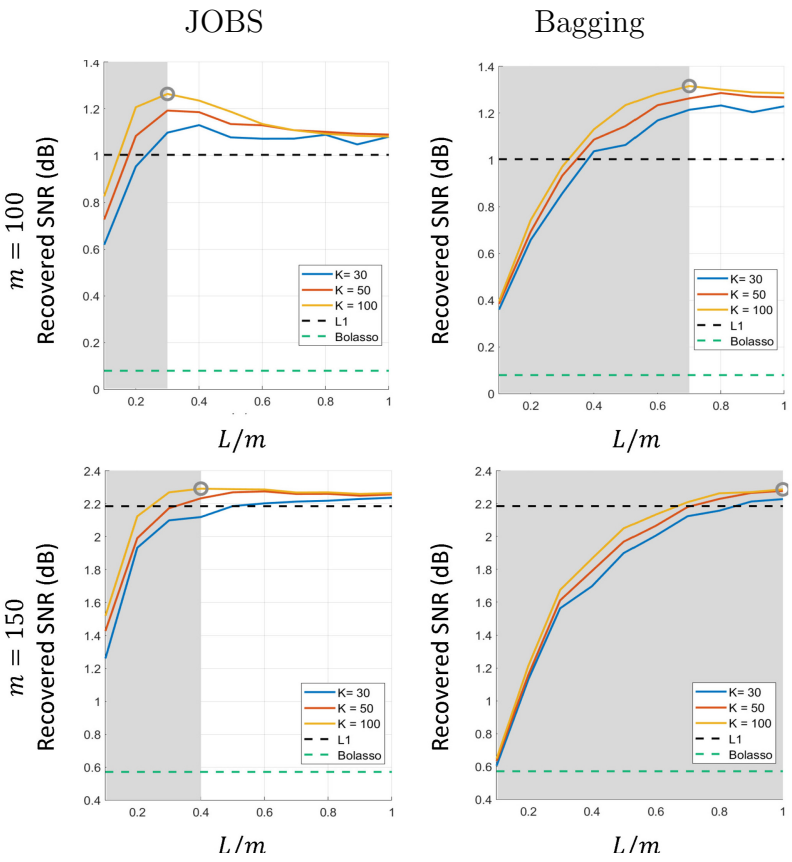


Figure 4.5: Recovery SNR (dB) performance curves for JOBS and Bagging (with various L,K) versus the peak Bolasso performance among various L,K and ℓ_1 minimization. The number of measurements are $m = 100, 150$ from top to bottom. Noise level is set to SNR = 0 dB. The grey circles highlight peaks while the grey area highlights the optimal bootstrap ratio. The optimal JOBS bootstrap ratio is smaller than that of Bagging. The y-axis of plots in the same row has been calibrated to have the same range.

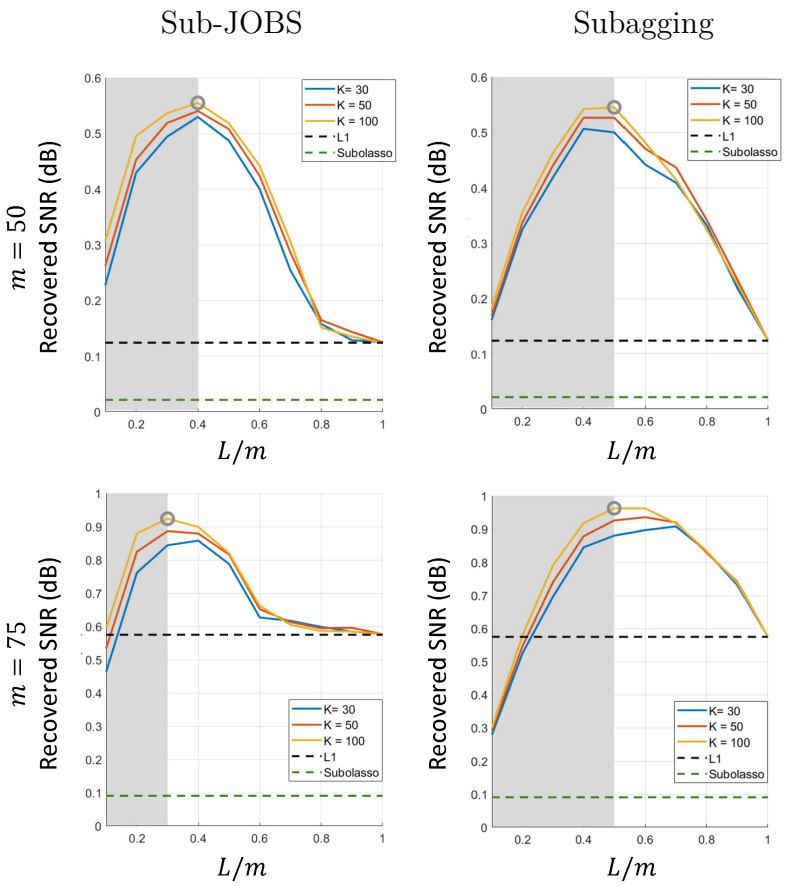


Figure 4.6: Recovered SNR (dB) performance curves for the sub-sampling schemes: Sub-JOBS, Subagging (with various L,K) versus Subolasso and ℓ_1 minimization. The number of measurements are $m = 50, 75$ from top to bottom. The noise level is set to SNR = 0 dB. Grey circles highlight the peaks and the grey area highlights the optimal sub-sampling ratio. The optimal sampling ratio of Sub-JOBS is smaller than that of Subagging. The y-axis of plots in the same row has the same range.

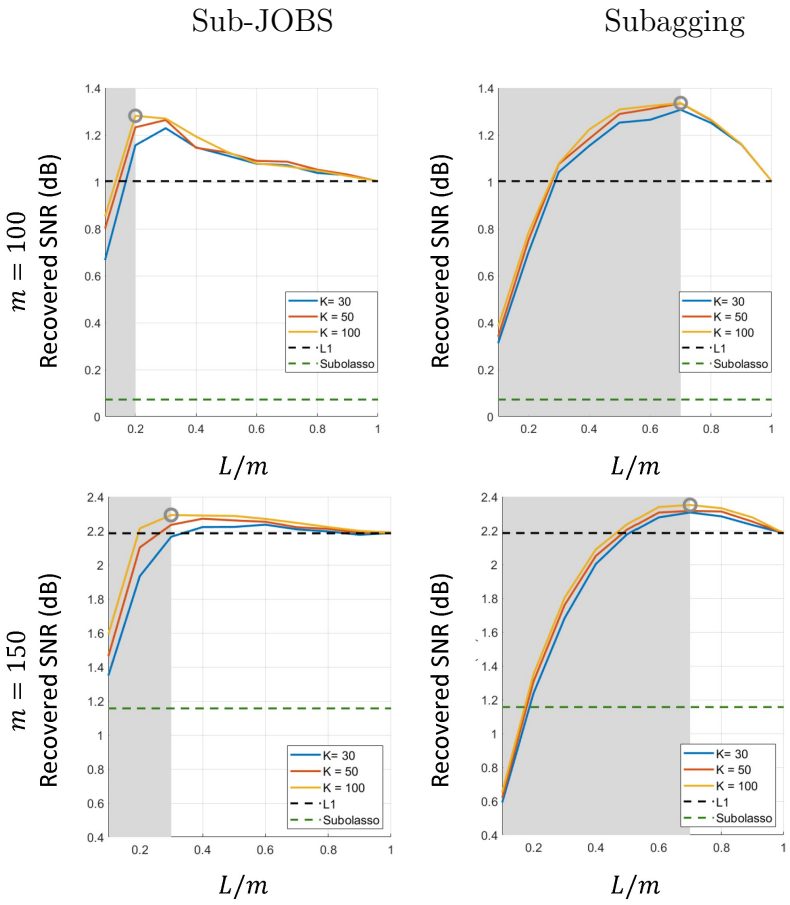


Figure 4.7: Recovered SNR (dB) performance curves for the sub-sampling schemes: Sub-JOBS, Subagging (with various L,K) versus Subolasso and ℓ_1 minimization. The number of measurements are $m = 100, 150$ from top to bottom. The noise level is set to SNR = 0 dB. Grey circles highlight the peaks and the grey area highlights the optimal sub-sampling ratio. The optimal sampling ratio of Sub-JOBS is smaller than that of Subagging. The y-axis of plots in the same row has the same range.

Table 4.1: The averaged sparsity ratios (± one standard deviation) of recovered optimal solutions of JOBS, Bagging, Bolasso (Top rows: original scheme; Bottom rows: sub-sampling variations) and ℓ_1 minimization. The numerical threshold for non-zero is 10^{-2}. SNR = 0 dB.

	Bootstrapping-based methods			
m	JOBS	Bagging	Bolasso	ℓ_1 min.
50	89% (±3%)	91% (±2%)	0.025% (±0.1%)	5.7% (±2%)
75	78% (±4%)	82% (±5%)	0.20% (±0.4%)	8.7% (±3%)
100	71% (±4%)	91% (±2%)	0.25% (±0.3%)	15% (±3%)
150	47% (±6%)	87% (±5%)	3.6% (±1%)	21% (±4%)
	Sub-sampling variations			
m	Sub-JOBS	Subagging	Subolasso	ℓ_1 min.
50	89% (±3%)	91% (±3%)	0.025% (±0.1%)	5.7% (±2%)
75	72% (±4%)	87% (±3%)	0.13% (0.3%)	8.7% (±3%)
100	57% (±3%)	74% (±7%)	0.60% (±0.5%)	15% (±3%)
150	55% (±6%)	79% (±8%)	3.8% (±2%)	21% (±4%)

All experimental settings are identical to those in the previous section except that the bootstrapping resampling scheme is replaced by sub-sampling for each subset $\mathcal{I}_j$.

Figure 4.6 and Figure 4.7 depict the performances of three different algorithms with the same parameters K, L. Similarly to the case in Figure 4.4 and Figure 4.5, one immediately observes that both JOBS and Bagging outperform ℓ_1 minimization and the sub-sampling version of Bolasso: Subolasso. Furthermore, JOBS achieves the best performance with smaller L than Bagging. Since sub-sampling potentially contains more information than bootstrapping, it also reduces the length of the subsets L necessary to achieve the best performance. For JOBS, the best sub-sampling ratio L/m at which the peak value is achieved reduces to $0.2 - 0.4$ for small m (ranging from $50 - 150$), whereas for Bagging,

the optimal sub-sampling ratio becomes $0.5 - 0.7$.

With the same L, K, the sub-sampling variation in general gives better performance than bootstrapping since there are no duplicated measurements. This is also not surprising since a sub-sampling ratio from $0 - 1$ corresponds to a bootstrap ratio from $0 - \infty$. The relationship between bootstrapping and sub-sampling ratios is well known and can be found in Figure 4.11 in Section 4.9.6. For the same number of estimates K, the performance of JOBS under a certain bootstraping ratio L/m is very similar to the performance of Sub-JOBS with corresponding sub-sampling ratio taking at the ratio where it results in the expected unique number of samples from bootstrapping.

We observe two key properties of the Sub-JOBS. *(i)* While m is small, the optimal sub-sampling ratios L/m in Figure 4.6, Figure 4.7 are smaller than the optimal bootstrap ratios in Figure 4.4 and Figure 4.5 for both JOBS and Bagging, since the grey and white boundaries are further left in sub-sampling variations. *(ii)* When more measurements m become available, JOBS and Bagging begin to lose their advantages over ℓ_1 minimization with the bootstrap scheme, whereas for the sub-sampling variation, JOBS and Bagging both approach ℓ_1-minimization performance with reasonably small L/m and K.

4.6.3 JOBS Solutions are Consistently Sparser than Bagging Solutions at Similar Performance Level

In JOBS, we have more precise control over the sparsity level. Individually solved predictors in Bagging are not guaranteed to have the same support. In the worst case, the average predictor from Bagging can be quite dense. We

also observe in our experimental results that JOBS generally produces sparser solutions than Bagging.

Here we analyze the sparsity of the reconstructed signals through the sparsity ratio. For a reconstructed vector $\widehat{\boldsymbol{x}}$ with a set threshold $\tau > 0$, the sparsity ratio is defined as the number of elements with whose magnitude higher the threshold over the total number elements in the vector:

$$\mathrm{sr}(\widehat{\boldsymbol{x}}, \tau) = \mathbb{1}\{|\widehat{\boldsymbol{x}}[i]| \geq \tau\}/\mathrm{len}(\widehat{\boldsymbol{x}}). \tag{4.36}$$

We calculate the average sparsity ratio over reconstructed samples. For our sparse regression task with the SNR = 0 dB, the threshold is set at 0.01. We take the peak performance solution and calculate their averaged sparsity ratios.

The sparsity ratios for JOBS, Bagging, Bolasso and their sub-sampling variations as well as ℓ_1 minimization are illustrated in Table 4.1. The averaged sparsity ratio for the optimal Bagging solution fluctuates between 82% to 91% as the number of measurements m changes from 50 to 150. The averaged sparsity ratio of the JOBS solution ranges from 89% to 47% as m varies from 50 to 150. The ground truth sparsity ratio is set at 25%. When the number of measurements increases, the sparsity ratio of the JOBS solution tends to approach that of the ground truth. With the same number of measurements m, an optimal JOBS solution is sparser than an optimal Bagging solution. As expected, Bolasso solutions are much too sparse due to the strict constraint of being in the intersection of supports for all estimators in order to be in the final estimated support.

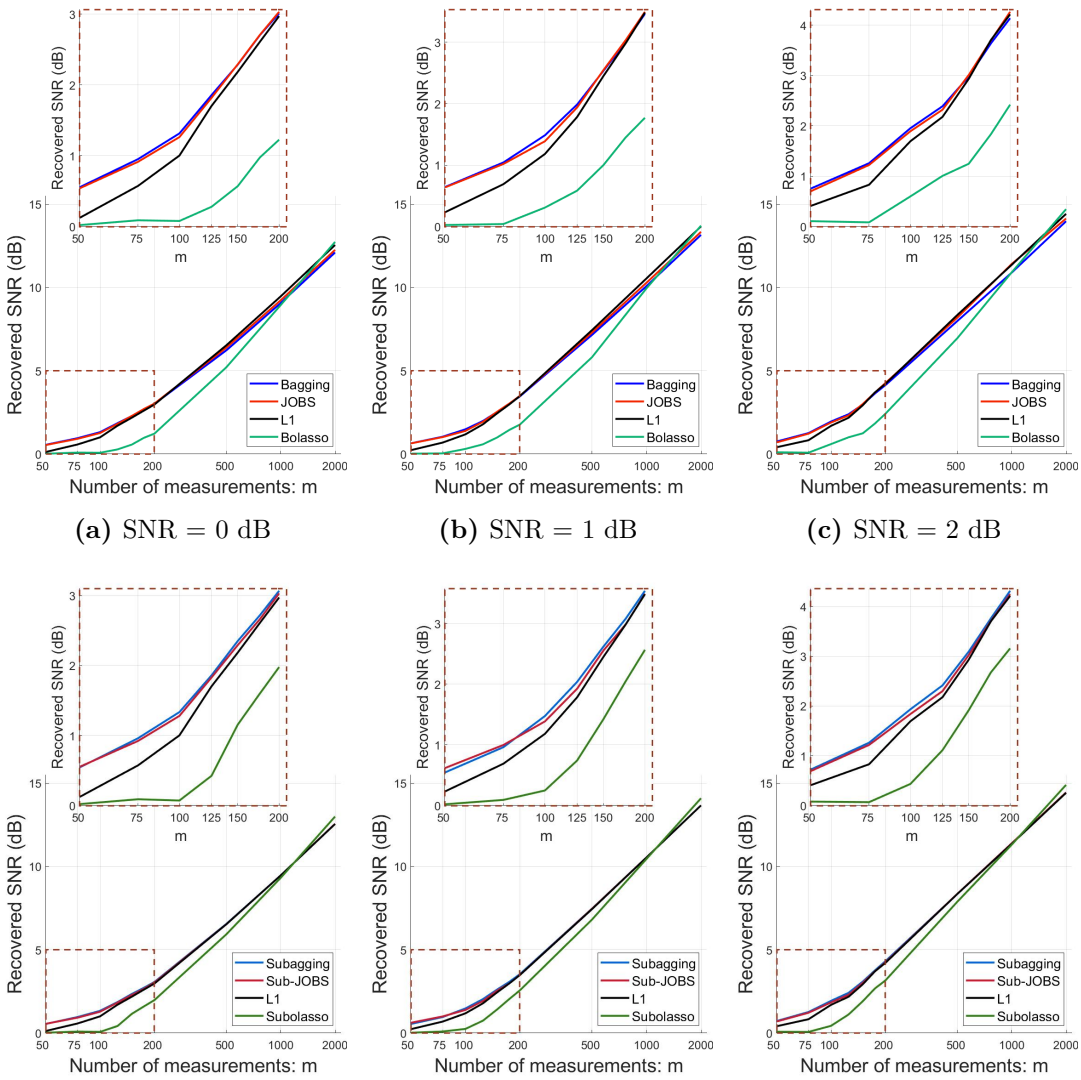
Another observation from Table 4.1 is: when the number of measurements

m increases, the sparsity levels of optimal JOBS solution and Bagging solution decreases whereas the optimal ℓ_1 minimization solution and Bolasso solution increases.

4.6.4 JOBS Optimal Sampling Ratio is Consistently Smaller than that of Bagging

From our experiments, we notice that both JOBS and Bagging algorithms outperform the classical ℓ_1 minimization algorithm in the challenging case when the total number of measurements m is low. The peak performance of JOBS and Bagging are comparable. Table 4.2 shows the optimal ratios for JOBS algorithm and for Bagging with the number of measurements m from 50-200 and various SNR ratios SNR $= 0, 1, 2$ dB and the optimal value of the estimates K is the maximum value. The optimal bootstrap sampling ratio for JOBS is smaller than that for Bagging, both for the original bootstrap version and the sub-sampling variation.

For the sub-sampling variations: Sub-JOBS and Subagging, their optimal sub-sampling ratios are smaller than those of the bootstrap versions under the same values of m and K. This observation matches our expectation of sub-sampling due to the fact that bootstrapping creates duplicated measurements, and thus it requires a larger sample size to achieve the same behavior as its sub-sampling variation.



(a) SNR = 0 dB **(b)** SNR = 1 dB **(c)** SNR = 2 dB

(d) SNR = 0 dB **(e)** SNR = 1 dB **(f)** SNR = 2 dB

Figure 4.8: Overall recovery performances with various number of measurements for for JOBS, Bagging, Bolasso in (a)-(c) and their sub-sampling schemes: Sub-JOBS, Subagging, Subolasso and ℓ_1 minimization in (d)-(f), both compared with ℓ_1 minimization with a full range of number of measurements from 50 to 2000 and various SNR values at $0, 1, 2$dB. The x-axis is plotted in log scale. In the challenging case of limited m measurements and high noise level, the margin between Sub-JOBS and ℓ_1 minimization is larger (zoomed-in figures on the top row). Peak performances of sub-sampling variations are similar and slightly better than the original bootstrap versions for JOBS, Bagging and Bolasso.

Table 4.2: The Empirical Optimal Sampling Ratios L/m with Limited Measurements m. Various noise levels with SNR $= 0, 1, 2$dB.

SNR (dB)	m	JOBS	Bagging	Sub-JOBS	Subagging
0	50	0.5	0.6	0.4	0.5
0	75	0.4	0.9	0.3	0.5
0	100	0.3	0.7	0.2	0.7
0	150	0.4	1	0.3	0.7
1	50	0.6	0.8	0.4	0.5
1	75	0.4	1	0.3	0.5
1	100	0.3	1	0.3	0.7
1	150	0.5	1	0.4	0.8
2	50	0.5	0.8	0.5	0.6
2	75	0.4	0.7	0.4	0.5
2	100	0.4	1	0.3	0.7
2	150	0.5	1	0.5	0.8

4.6.5 Lower Computational Complexity of JOBS than Bagging due to Smaller Optimal Sampling Ratios

The computational complexity of the JOBS algorithm is obviously higher than the conventional ℓ_1 minimization algorithm. The extra computations allow us to achieve an improvement in recovery performance as illustrated in the experimental results. Regarding the algorithm itself, although the mixed $\ell_{1,2}$ norm at first glance seems more complicated to optimize, in terms of actual implementation, the computation complexity for JOBS and Bagging turn out to be very similar for a given sampling parameter (L, K). With the ADMM implementation, Bagging uses iterative soft thresholding on each element of the solution matrix whereas JOBS requires iterative row-wise soft thresholding to achieve row sparsity (refer to Section 4.2.2 for details). The theoretical

complexity level for both algorithms is $\mathcal{O}(n^2(L+n)K) + T\mathcal{O}(n^2K)$, where T is number of iterations. For ℓ_1 minimization, the theoretical complexity level is $\mathcal{O}(n^2(m+n)) + T\mathcal{O}(n^2)$. The theoretical complexity of JOBS is less than K times the complexity of ℓ_1 minimization. The complexity terms that relate to the number of iterations T counts the complexity of all operations needed at every iteration, whereas the complexity term that does not relate to the number of iterations correponds to line 3 in Algorithm 1 which remains the same across every iteration. While calculating the computational complexity for all algorithms, we assume that the inverse operation uses the Gaussian Elimination implementation with $\mathcal{O}(n^3)$ complexity.

With a large enough K, JOBS achieves optimal performance with a much smaller vector size L compared to Bagging. Therefore, to obtain the peak performance, the complexity for JOBS turns out to be lower than that of Bagging. Bagging can benefit greatly from a parallel implementation. Similarly, a distributed implementation reduces the running time for JOBS.

4.6.6 Peak Performances over a Large Range of Measurements

We take a closer look at the peak performances of various recovery schemes with a wide range of measurement numbers and various SNR setting at $0, 1, 2$ dB. For the three bootstrapping algorithms, the optimal choices of parameters K and L as found and presented in the previous section (indicated by the grey circles in Figure 4.4, Figure 4.5 Figure 4.6) and Figure 4.7 are selected throughout the entire experiment. We then explore the recovery power over a wider range of available measurements, including the oversampling cases. The

number of measurements m ranging from 50 to 2000. Figure 4.8 depicts the peak performances of a wide range of number of measurements across all SNR settings, using three bootstrapped methods and their sub-sampling variations as well as the ℓ_1 minimization.

As aforementioned, when the number of measurements m is low, JOBS and Bagging outperform ℓ_1 minimization. The larger the noise level, the larger the margin becomes. As m decreases, the margin also increases. However, when the level of m is high enough, the two bootstrapping-based strategies start losing their advantages over ℓ_1 minimization. The performance limits of JOBS and Bagging are comparable (within 3% difference). Figure 4.4 and Figure 4.5 show that, in general, JOBS achieves comparable performance to Bagging with significantly smaller L and K values. JOBS and Bagging performance tend to get close to ℓ_1 minimization as m increases. The remaining bootstrap-based scheme of Bolasso only performs similarly to other algorithms for very large m. The sub-sampling variations of the Sub-JOBS, Subagging, and Subolasso also behave similarly.

The optimal bootstrap sampling ratio in the original version and the optimal sub-sampling ratios for sub-sampling variations approximately matches to the relationship in Figure 4.11. The optimal sub-sampling ratios are slightly smaller than the original bootstrap sampling ratios in Figure 4.8 for the same SNR.

4.7 Experimental Results on Classification

In this section, we extend the JOBS framework to classification problems. We solve the sparse regression problem using ℓ_1 minimization, JOBS, Bagging,

Bolasso and their sub-sampling variations, all within a Sparse Representation-based Classification (SRC) framework proposed by Wright in (Wright et al., 2009). The SRC framework predicts a class label of test data based on the representation error from sparse regression results given the training data as the sparsifying dictionary. In this section, we study how sparse regression solutions from JOBS, Bagging and Bolasso affect classification results. In other words, we would like to confirm that improvements in regression directly leads to improvements in classification.

4.7.1 The SRC Algorithm

Many efforts have been made in developing classification algorithms based on sparse representation. SRC is one of the earliest, the simplest, and also the most well-known of these methods. The basic idea is as follows. Ideally, the test data can be best represented by the linear combination of past observed data samples from the same class. The SRC procedure first uses a dimension reduction technique to generate feature vectors of all the data. Then, features of different classes from the training set are concatenated to form the sparsifying dictionary. For a given test data point, we first solve the sparse representation with respect to the dictionary from training data. The test data should be better represented by the training data from its own class, with most coefficients corresponding to other classes set to zero. Hence, the predicted classification label is assigned based on the class that gives the minimum class-wise representation error (Wright et al., 2009).

Major extensions of the original SRC frameworks mainly concentrate on

earning a better representative dictionary than the direct training data features (Aharon, Elad, and Bruckstein, 2006; Mairal, Bach, and Ponce, 2012). in this work, we only aim to demonstrate that improving the sparse regression stage in SRC also leads to better classification results. The classification experiments have been performed on two common face recognition datasets: the extended Yale B dataset (Georghiades, Belhumeur, and Kriegman, 2001) and the cropped AR dataset (Martinez and Kak, 2001).

4.7.2 The Extended Yale B Dataset

The Extended Yale B database consists of 2414 frontal-face images of 38 individuals (Georghiades, Belhumeur, and Kriegman, 2001). For each subject, there are $59 - 64$ images. Those different images are taken under different laboratory-controlled lighting conditions. All images have been registered, normalized, and resized to 192×168 pixels. Among all the data, 90% of data are randomly selected for training and the rest are used for testing.

4.7.3 The Cropped AR Dataset

Apart from Yale B dataset, we also use a more challenging dataset called the cropped AR dataset (Martinez and Kak, 2001). This dataset consists of 2600 images from 100 individuals with 50 male subjects and 50 female subjects. For each subject, 26 images with different illumination conditions, facial expressions and common occlusions such as from glasses and scarfs worn by individuals. Some examples of these face pictures are shown in Figure 4.9. All images have been registered and cropped to dimension 165 pixels $\times$120 pixels. Among all the data, we again randomly select 90% of data for training and the rest are

used for testing. Finally, for each task, the number of test samples among all classes are the same. Therefore, the performance of test samples from each class contributes the same weight to the final cumulative accuracy.

(a) **(b)** **(c)**

Figure 4.9: Examples of face pictures in the cropped AR data set. Left: a whole face picture of a person. Middle: a face picture of a person with sun glasses. Right: a face picture of a person with scarf.

4.7.4 Face Recognition Experiment Results

We extract low dimensional features before performing classification. Here we use random projection for dimension reduction. A random Gaussian matrix of size $m \times d_{\text{Img}}$ is used, where m is the number of rows of the projection matrix and d_{Img} is the dimension of the images. For the AR dataset, d_{Img} is $165 \times 120 = 19800$ and for the Yale B dataset, the dimension is $192 \times 168 = 32256$. These generate the so-called random features as described in SRC (Wright et al., 2009). The random projection setting is also consistent with the choice of random sensing matrix $\boldsymbol{A}$ in our regression experiment. However, in this face recognition experiment, we further reduce the number of features (measurements) in the AR dataset to $m = 50$ and the images in the Yale B dataset $m = 30$ since the Yale B dataset is less challenging and thus desirable performance can be achieved with fewer

features. In this classification experiment, there is no extra added noise on training or testing data features. Some face images contain glasses and scarf in the cropped AR dataset, therefore there are inherent sparse noise in this dataset.

We compute the sparse representation using ℓ_1 minimization, JOBS, Bagging and Bolasso in the same manner as the regression experiment. From these sparse representations, we then compute the class-wise residues as in the SRC framework to arrive at the final prediction.

The classification results are shown in Table 4.3 and Table 4.4. On the cropped AR dataset, classification based on sparse representations generated by JOBS shows a consistent improvement of 3% in classification accuracy over the baseline ℓ_1 minimization. Similarly to the regression case, the optimal bootstrapping ratio for JOBS is quite low (only around 0.5). Bagging solutions do not result in any improvement over the baseline ℓ_1 algorithm. Both achieve the same accuracy of 0.855. The best accuracy for Bolasso is slightly lower: 0.790 whereas the optimal accuracy for JOBS is 0.880, obtained with optimal bootstrap sampling ratio and number of estimates of $(L/m, K) = (0.5, 30)$.

On the easier dataset Yale B, the accuracy for JOBS is 0.939, also a slight improvement over the accuracy of 0.925 for ℓ_1 minimization. Bagging yields 0.921 while Bolasso achieves 0.901. The Yale-B dataset is relatively simple compared to the AR dataset. Hence, any improvements on a high baseline accuracy is considerably more difficult.

Similarly to the sparse regression task, we also compare the sparsity levels of optimal solutions for different algorithms. We calculate the sparsity ratio as previously in equation (4.36), which expresses in percentage the ratio of the

Table 4.3: Classification Accuracies with various methods on Yale-B data set. The number of random features is 30 and the split ratio of training and testing set is 0.91.

	Baseline	Bootstrapping-based methods		
Methods	ℓ_1 min.	JOBS	Bagging	Bolasso
Accuracy	0.925	**0.939**	0.921	0.901
$(L/m, K)^\star$	(1,1)	(0.6,10)	(0.9,10)	(0.9,10)
		Sub-sampling variations		
		Sub-JOBS	Subagging	Subolasso
Accuracy		0.930	0.930	0.930
$(L/m, K)^\star$		(0.4,50)	(0.9, 10)	(0.7, 30)

Table 4.4: Classification Accuracies with various methods on AR data set. The number of random features is 50 and the split ratio of training and testing set is 0.92.

	Baseline	Bootstrapping-based methods		
Methods	ℓ_1 min.	JOBS	Bagging	Bolasso
Accuracy	0.855	**0.880**	0.855	0.790
$(L/m, K)^\star$	(1,1)	(0.5,30)	(1,50)	(0.9,30)
		Sub-sampling variations		
		Sub-JOBS	Subagging	Subolasso
Accuracy		0.875	0.870	0.785
$(L/m, K)^\star$		(0.6,10)	(0.9, 30)	(0.9, 10)

number zero entries versus the the signal dimension, then averaging over all sparse representations among all test data. We set a threshold of amplitudes to be 1×10^{-6} for being non-zero. We compute the sparsity ratio for each reconstructed signal, and then calculate the mean and standard deviation for each algorithm. The result is in Table 4.5 and Table 4.6. The Bagging solution is the most dense among all algorithms whereas Bolasso generates the most sparse solution. The sparsity ratio of JOBS and ℓ_1 minimization are in between these two extremes. Noticeably, in harder AR dataset experiment with occlusion

inherent in the data introduced by glasses and scarfs, JOBS solution is slightly more sparse than ℓ_1 solution. Our solutions in the classification experiment match the solutions in the sparse regression experiment in terms of sparsity levels and validates our intuition on sparsity levels and performances of various algorithms.

Table 4.5: Comparison of Sparsity Ratios ($\pm$ one standard deviation) of different algorithms expressed in percentages. For all algorithms, the numerical threshold for being non-zero is 10^{-6}. Bagging generates the most dense solutions. JOBS and ℓ_1 minimization generates solutions with moderate sparsity levels while Bolasso generates the most sparse solutions. Yale B ($m = 30$).

Baseline	Bootstrapping-based methods		
ℓ_1 min.	JOBS	Bagging	Bolasso
0.93% ($\pm$0.3%)	2.4% ($\pm$0.4%)	5.94% ($\pm$1%)	0.48% ($\pm$0.3%)
	Sub-sampling variations		
	Sub-JOBS	Subagging	Subolasso
	2.5% ($\pm$0.4%)	3.8% ($\pm$0.8%)	0.39% ($\pm$0.2%)

Table 4.6: Comparison of Sparsity Ratios ($\pm$ one standard deviation) of different algorithms expressed in percentages. For all algorithms, the numerical threshold for being non-zero is 10^{-6}. Bagging generates the most dense solutions. JOBS and ℓ_1 minimization generates solutions with moderate sparsity levels while Bolasso generates the most sparse solutions. AR ($m = 50$).

Baseline	Bootstrapping-based methods		
ℓ_1 min.	JOBS	Bagging	Bolasso
3.6% ($\pm$0.5%)	2.7%($\pm$0.5%)	27% ($\pm$3%)	0.56% ($\pm$0.3%)
	Sub-sampling variations		
	Sub-JOBS	Subagging	Subolasso
	2.4% ($\pm$0.4%)	13% ($\pm$2%)	0.63% ($\pm$0.2%)

4.8 Summary

We propose a collaborative signal recovery framework named JOBS, motivated from powerful bootstrapping ideas in machine learning. JOBS improves the robustness of sparse recovery in challenging scenarios of noisy environments and/or limited measurements. We carefully analyze theoretical properties of JOBS such as BNSP and BRIP. We further derive error bounds for JOBS as well as for a closely related scheme called Bagging and analyze their theoretic recovery behaviors with respect to two key parameters: the bootstrap sampling ratio L/m and the number of estimates K. We also study a common sub-sampling variation of the framework and study its connection to the original bootstrap scheme experimentally. Finally, experiments on sparse regression as well as classification (face recognition task) are conducted for validation. Simulation results show that the proposed algorithm consistently outperforms Bagging and Bolasso among most parameter settings (L, K).

We summarize below several important properites that we discovered for JOBS. (*i*) JOBS is particularly powerful when the number of measurements m is limited, outperforming ℓ_1 minimization by a large margin. (*ii*) JOBS achieves desirable performances with relatively low bootstrap ratio L/m (peak performance occurs at $0.3 - 0.5$ whereas the sub-sampling variation requires only $0.2 - 0.4$). It also demands a relatively small K (around 30 in our experimental study). (*iii*) The optimal sampling ratio for JOBS is lower than that of Bagging while achieving similar results. This results in a lower computation complexity for JOBS. (*iv*) JOBS solutions are generally more sparse than Bagging's – a desirable property in sparse recovery.

Future work may include developing online implementation of JOBS for streaming applications, extending the JOBS framework to other forms of collaborative sparse regression, and exploration of the framework to improve dictionary learning.

4.9 Appendix

4.9.1 The Row Sparsity Norm is a Special Case of Block (group) Sparsity

Consider the matrix that contains all bootstrapped estimators: $\boldsymbol{X} \in \mathbb{R}^{n \times K} = [x_{ij}], i = 1, 2, ..., n, j = 1, 2, ..., K$, the row sparsity norm: the $\ell_{1,2}$ norm of $\boldsymbol{X}$ is equivalent to the a block assignment on the vectorized $\boldsymbol{X}$: vec($\boldsymbol{X}$). Each block assignment $\mathcal{B}_i$ in assignment $\{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$ takes the all elements in $i-$th row of vec($\boldsymbol{X}$): $\{x_{i1}, x_{i2}, ..., x_{iK}\}$. The vectorized JOBS formulation is in (4.7).

Although blocks in block sparsity conventionally refer to indices sets that are adjacent, block sparsity result can be easily extended to the cases with non-adjacent indices blocks. In the vectorized JOBS problem, the indices in row sparsity block is not adjacent due to the vectorization convention stacks columns rather than rows. Rearranging the columns in the JOBS matrix $\boldsymbol{A^J}$ can make the row sparsity block adjacent without changing the nature of the problem.

4.9.2 Proof of the Reverse Direction for Noiseless Recovery

Lemma 18 *If the MMV problem* $\mathbf{P_1}(K)$ *, $K > 1$, in (4.4) has a unique solution, it will be of form* $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)$*. Then, there is a unique solution to* $\mathbf{P_1}$*:*

$\boldsymbol{x}^\star$.

Let us prove the other direction. If $\mathbf{P_1}(K)$ has a unique solution, the solution must be in the form of $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star, ..., \boldsymbol{x}^\star)$, and it implies that $\mathbf{P_1}$ has a unique solution $\boldsymbol{x}^\star$.

If $\mathbf{P_1}(K)$ has a unique solution, then it is equivalent to say that $\boldsymbol{A}$ satisfied BNSP of order s. For all $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_K) \neq \boldsymbol{O}, \boldsymbol{v}_j \in \text{Null}(\boldsymbol{A})$, we have $\forall\ \boldsymbol{S}, |\boldsymbol{S}| \leq s, \|\boldsymbol{V}[\boldsymbol{S}]\|_{1,2} < \|\boldsymbol{V}[\boldsymbol{S}^c]\|_{1,2}$. This implies that $\forall\ \boldsymbol{V} = (\boldsymbol{v}, \boldsymbol{0}, \boldsymbol{0}, ..., \boldsymbol{0}), \boldsymbol{v} \in \text{Null}(\boldsymbol{A})\backslash\{\boldsymbol{0}\}$, BNSP is satisfied. Since in this case, except the first column, all others are zero and therefore do not contribute any to the group norm. Mathematically, for all $\boldsymbol{S}$, $\|\boldsymbol{V}[\boldsymbol{S}]\|_{1,2} = \|\boldsymbol{v}[\boldsymbol{S}]\|_1$. We, therefore, will have the BNSP of order s, implying the NSP for $\boldsymbol{A}$ of order s.

4.9.3 Implications of Block Null Space Property of JOBS Matrix

Using a similar analysis as in previous subsection 4.9.2, we conclude that a block diagonal matrix satisfies BNSP of order s implies that each submatrix satisfies NSP of order s. The block diagonal JOBS matrix $\boldsymbol{A^J} = \text{block\_diag}(\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K])$ satisfies BNSP of order s. Then, for all $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_K) \neq \boldsymbol{O}, \boldsymbol{v}_j \in \text{Null}(\boldsymbol{A}[\mathcal{I}_j]), j = 1, 2, .., K$, we have $\forall\ \boldsymbol{S}, |\boldsymbol{S}| \leq s, \|\boldsymbol{V}[\boldsymbol{S}]\|_{1,2} < \|\boldsymbol{V}[\boldsymbol{S}^c]\|_{1,2}$. This implies that $\forall\ \boldsymbol{V} = (\boldsymbol{0}, ..., \boldsymbol{v}_j, ..., \boldsymbol{0}), \boldsymbol{v}_j \in \text{Null}(\boldsymbol{A}[\mathcal{I}_j])\backslash\{\boldsymbol{0}\}$, BNSP is satisfied, which essentially states that NSP is statisfied for $\boldsymbol{A}[\mathcal{I}_j]$.

4.9.4 A Toy Example Shows the Correctness of JOBS

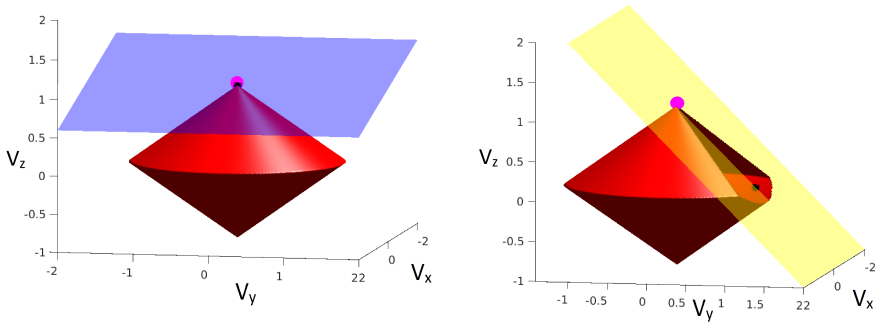
We give a toy example to illustrate our scheme geometrically and demonstrate the feasibility of our algorithm. Let the dimension of signal $n = 2$ and number of measurements $m = 2$. In this example, true solution $\boldsymbol{x}^\star = \begin{pmatrix} 0 & 1 \end{pmatrix}^{\boldsymbol{T}}$ The sensing matrix $\boldsymbol{A} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$ and the measurement vector is $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\star = \begin{pmatrix} 1 & 1 \end{pmatrix}^{\boldsymbol{T}}$.

Since $\boldsymbol{A}$ is full rank and $m = n$, the ℓ_1 solution is $\boldsymbol{x}^\star$. For JOBS algorithm, we let the sizes of subsets $L = 1$ and the number of subsets is $K = 2$. The true MMV solution $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star)$. Our solution candidate $\boldsymbol{X}$ lies in the domain of 2 by 2 matrices with 4 free parameters. For visualization purpose, we reduce the degree of freedom by 1 by adding an extra constraint: let elements in the last row be the same. Now, $\boldsymbol{X} \in \mathcal{X} = \left\{ \begin{pmatrix} v_x & v_y \\ v_z & v_z \end{pmatrix} : v_x, v_y, v_z \in \mathbb{R} \right\}$. This treatment is reasonable since $\boldsymbol{X}^\star = (\boldsymbol{x}^\star, \boldsymbol{x}^\star) \in \mathcal{X}$. We plot level sets of $\ell_{1,1}$ and $\ell_{1,2}$ norms with respect to v_x, v_y and v_z.

For subset $\mathcal{I} = \{1\}$ corresponds to the first constraint introduced by the dot product of the first row of $\boldsymbol{A}$ and $\boldsymbol{x}$, which is $2v_x + v_z = 1$, and $\mathcal{I} = \{2\}$ denotes to the second one, which is $v_y + v_z = 1$. Fig. 4.3a shows $\ell_{1,1}$ norm minimization and Fig. 4.3b depicts relaxation version. Fig. 4.10a is a successful case whereas Fig. 4.10b displays a failure. This illustrates that JOBS is a two-step relaxation of the ℓ_1 minimization and the success demonstrates the feasibility of JOBS under proper conditions.

4.9.5 Proof of Proposition 14

To prove this proposition, we give an alternative form of RIP and BRIP which are stated in the following two propositions. Alternative form of RIP as a function



(a) $\mathbf{J_{12}} : \mathcal{I}_1 = \{1\}, \mathcal{I}_2 = \{1\}$ **(b)** $\mathbf{J_{12}} : \mathcal{I}_1 = \{2\}, \mathcal{I}_2 = \{2\}$

Figure 4.10: Two cases of JOBS. **Left**: a successful recovery case. **Right**: a failure recovery case. $\mathbf{J_{12}}$. The blue and yellow planes are the first and second constraints, respectively. The green line is their intersection. The pink point is the true solution and black points are reconstructed solutions.

of matrix induced norm is given as follows.

Proposition 19 (Alternative form of RIP) *Matrix* $\boldsymbol{A}$ *has* ℓ_2*-normalized columns, and* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{S} \subset \{1, 2, ..., n\}$ *with size smaller or equal to* s *and* $\boldsymbol{A_S}$ *takes columns of* $\boldsymbol{A}$ *with indices in* $\boldsymbol{S}$*. The RIP constant of order* s *of* $\boldsymbol{A}$*,* $\delta_s(\boldsymbol{A})$ *is:*

$$\delta_s(\boldsymbol{A}) = \max_{\boldsymbol{S} \subseteq \{1,2,...,n\}, |\boldsymbol{S}| \leq s} \|\boldsymbol{A}_{\boldsymbol{S}}^{\boldsymbol{T}} \boldsymbol{A}_{\boldsymbol{S}} - \boldsymbol{I}\|_{2 \to 2}, \tag{4.37}$$

where $\boldsymbol{I}$ *is an identity matrix of size* $s \times s$ *and* $\|\cdot\|_{2\to 2}$ *is the induced* $2-$*norm defined as: for any matrix* $\boldsymbol{A}$*,* $\|\boldsymbol{A}\|_{2\to 2} = \sup_{\boldsymbol{x} \neq \boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}$.

This proposition can be directly derived from the definition of RIP constant. Similarly, we can derive the alternative form of BRIP constant as a function of matrix induced norm.

Proposition 20 (Alternative form of BRIP) *Let matrix* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *have* ℓ_2*-normalized columns and let* $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_n\}$ *be the group sparsity pattern*

that defines the row sparsity pattern, with $\mathcal{B}_i$ contains all indices corresponding to all elements of the $i-$th row. For $\mathcal{S} \subseteq \{1, 2, ..., n\}$, denote $\mathcal{B}(\mathcal{S}) = \{\mathcal{B}_i, i \in \mathcal{S}\}$ taking several blocks (groups) in $\mathcal{S}$. The Block-RIP constant of order s of $\boldsymbol{A}$: $\delta_{s|\mathcal{B}}(\boldsymbol{A})$ is

$$\delta_{s|\mathcal{B}}(\boldsymbol{A}) = \max_{\mathcal{S}\subseteq\{1,2,...,n\},|\mathcal{S}|\leq s} \|\boldsymbol{A}^{\boldsymbol{T}}_{\mathcal{B}(\mathcal{S})}\boldsymbol{A}_{\mathcal{B}(\mathcal{S})} - \boldsymbol{I}\|_{2\to 2}. \tag{4.38}$$

Without loss of generality, let us assume that all columns of $\boldsymbol{A}$ in the original ℓ_1 minimization have unit ℓ_2 norms. Therefore, $\boldsymbol{A}$ does not have any zero column. Before we calculate the RIP constant of the bootstrapped sensing matrices, we need to perform two operations: remove the duplicate rows from bootstrapped sensing matrices and then normalize the columns.

First, we remove the duplicated rows using the weighted scheme. In the noisy recovery problem, for a multi-set $\mathcal{I}$ that may contain duplicate, the set $\mathcal{U}$ denotes the set of all unique elements. In the constraint optimization, we can express the sum using occurrence times in $\mathcal{I}$ for each element using c_i. $\|\boldsymbol{A}[\mathcal{I}]\boldsymbol{x} - \boldsymbol{y}[\mathcal{I}]\|_2^2 = \sum_{i\in\mathcal{I}} \|\boldsymbol{a}[i]\boldsymbol{x} - \boldsymbol{y}[i]\|_2^2 = \sum_{i\in\mathcal{U}} \|\sqrt{c_i}\boldsymbol{a}[i]\boldsymbol{x} - \boldsymbol{y}[i]\|_2^2$. Therefore, the original program is equivalent to reducing the duplicated rows in the bootstrap sample using $\sqrt{c_i}$ as weights. Because sampling with replacement is uniform, therefore the expected values of occurrence times for each sample are the same. To denote this operation, we have $\boldsymbol{R} \in \mathbb{R}^{u\in L}$, $\boldsymbol{R} = \text{diag}(\sqrt{c_1}, \sqrt{c_2}, ..., \sqrt{c_u})\boldsymbol{I}[\mathcal{U}]$, each row of $\boldsymbol{I}[\mathcal{U}]$ correponds to the unique vector of a row and this operation deletes the duplicated rows.

Second, we normalize the columns of these matrices using the following normalization procedure. For $\boldsymbol{M} \in \mathbb{R}^{u\times n}$, since the original matrix $\boldsymbol{A}$ does not have any zero column, $\boldsymbol{Q}(\boldsymbol{M}) \in \mathbb{R}^{n\times n}$ is a normalization matrix of $\boldsymbol{M}$ such that

$\boldsymbol{MQ}(\boldsymbol{M})$ has ℓ_2-normalized columns. Clearly, the normalization matrix $\boldsymbol{Q}$ of $\boldsymbol{M}$ is obtained by:

$$\boldsymbol{Q}(\boldsymbol{M}) = \text{diag}(\|\boldsymbol{m}_1\|_2^{-1}, \|\boldsymbol{m}_2\|_2^{-1}, ..., \|\boldsymbol{m}_n\|_2^{-1}), \tag{4.39}$$

where $\boldsymbol{m}_j$ denotes $j-$th column of $\boldsymbol{M}$.

Similary, we can construct $\boldsymbol{Q}_j$s using (4.39) to normalize the columns. Let the original JOBS matrix in the vectorized problem be $\boldsymbol{A^J} = \text{block\_diag}(\boldsymbol{A}[\mathcal{I}_1], \boldsymbol{A}[\mathcal{I}_2], ..., \boldsymbol{A}[\mathcal{I}_K])$. We first normalize each block and then obtain the normalized bootstrapped sensing matrix as: $\widetilde{\boldsymbol{A}[\mathcal{I}_j]} = \boldsymbol{R}_j \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{Q}_j$. The original JOBS matrix can be transferred into the normalized version $\widetilde{\boldsymbol{A}^J} = \text{block\_diag}(\widetilde{\boldsymbol{A}[\mathcal{I}_1]}, \widetilde{\boldsymbol{A}[\mathcal{I}_2]}, ..., \widetilde{\boldsymbol{A}[\mathcal{I}_K]})$.

Now, we consider the BRIP constant for $\boldsymbol{A^J}$. In this derivation, column selection of a matrix is written as a right multiplication of the matrix $\boldsymbol{I_S}(\cdot)$.

$$\begin{aligned}
\delta_{s|\mathcal{B}}(\boldsymbol{A^J}) &= \delta s|\mathcal{B}(\widetilde{\boldsymbol{A^J}}) \\
&= \max_{\mathcal{S}\subseteq\{1,2,..,n\},|\mathcal{S}|\leq s} \|(\widetilde{\boldsymbol{A^J}}\boldsymbol{I}_{\mathcal{B}(\mathcal{S})})^{\boldsymbol{T}}\widetilde{\boldsymbol{A^J}}\boldsymbol{I}_{\mathcal{B}(\mathcal{S})} - \boldsymbol{I}\|_{2\to 2} \\
&= \max_{\substack{\mathcal{S}\subseteq\{1,2,..,n\},\\ |\mathcal{S}|\leq s}} \max_j \|(\widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}})^{\boldsymbol{T}}\ \widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}} - \boldsymbol{I}\|_{2\to 2} \\
&= \max_{\substack{\mathcal{S}\subseteq\{1,2,..,n\},\\ |\mathcal{S}|\leq s}} \|\text{block_diag}((\widetilde{\boldsymbol{A}[\mathcal{I}_1]}\boldsymbol{I}_{\mathcal{S}})^{\boldsymbol{T}}\widetilde{\boldsymbol{A}[\mathcal{I}_1]}\boldsymbol{I}_{\mathcal{S}} - \boldsymbol{I}, \\
&\qquad ..., (\widetilde{\boldsymbol{A}[\mathcal{I}_K]}\boldsymbol{I}_{\mathcal{S}})^{\boldsymbol{T}}\widetilde{\boldsymbol{A}[\mathcal{I}_K]}\boldsymbol{I}_{\mathcal{S}} - \boldsymbol{I})\|_{2\to 2}.
\end{aligned}$$

The induced $2-$norm of a matrix equals to the max singular value of $\|\boldsymbol{D}\|_{2\to 2} = \sigma_{\max}(\boldsymbol{D})$ and if $\boldsymbol{D}$ is a block diagonal matrix $\boldsymbol{D} = \text{diag}(\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_K)$, then

$\sigma_{\max}(\boldsymbol{D}) = \max_{j=1,2,\ldots,K} \sigma_{\max}(\boldsymbol{D}_j)$. Applying this property leads to

$$\begin{aligned}
\delta_{s|\mathcal{B}}(\boldsymbol{A^J}) &= \max_{\substack{\mathcal{S}\subseteq\{1,2,..,n\},\\ |\mathcal{S}|\leq s}} \max_j \|(\widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}})^{\boldsymbol{T}}\ \widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}} - \boldsymbol{I}\|_{2\to 2} \\
&= \max_j \max_{\substack{\mathcal{S}\subseteq\{1,2,..,n\},\\ |\mathcal{S}|\leq s}} \|(\widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}})^{\boldsymbol{T}}\ \widetilde{\boldsymbol{A}[\mathcal{I}_j]}\boldsymbol{I}_{\mathcal{S}} - \boldsymbol{I}\|_{2\to 2} \\
&= \max_j \delta_s(\boldsymbol{A}[\mathcal{I}_j]).
\end{aligned}$$

4.9.6 Distribution of the Unique Number of Elements for Bootstrapping

The bootstrap is essentially sampling with replacement, which is likely to create duplicate information. The performance of sampling with replacement and sampling without replacement (sub-sampling) can be linked by studying the quantity of the number of unique elements. In this section, we give the analytic form of the number of unique samples when there are finite number of measurements m and bootstrap sample L, as well as the form for asymptotic case as $m \to \infty$. The finite case is studied in a well-known statistics problem – the Birthday Problem (*The Birthday Problem*). We also show empirically that the finite m case is close in the asymptotic sense.

4.9.6.1 Unique Number of Bootstrap Samples with Finite Sample m

We generate L samples from m samples uniformly at random with replacement ($L \leq m$). Let U denote the number of distinct samples among L samples. Clearly we have the number of distinct samples is between $[1, L]$ and the probability

mass function is given by (*The Birthday Problem*), same as the famous Birthday problem in statistics:

$$\mathbb{P}(U = u) = \binom{m}{u} \sum_{j=0}^{u} (-1)^j \binom{u}{j} (\frac{u-j}{m})^L, u = 1, 2, ..., L. \tag{4.40}$$

In our problem, we are interested in finding the lower bound of U with certainty $1 - \alpha$

$$\mathbb{P}(U \geq d) = \sum_{U=d}^{L} \binom{m}{u} \sum_{j=0}^{u} (-1)^j \binom{u}{j} (\frac{u-j}{m})^L \geq 1 - \alpha. \tag{4.41}$$

Therefore for

$$1 \geq \alpha \geq \sum_{u=0}^{d-1} \binom{m}{u} \sum_{j=0}^{u} (-1)^j \binom{u}{j} (\frac{u-j}{m})^L, \tag{4.42}$$

equation (4.41) is satisfied.

4.9.6.2 Asymptotic Unique Ratios of Bootstrap Samples

The theoretically unique percentage for asymptotic case when the number of total number of measurements goes to infinity $m \to \infty$ has been studied in the literature (Weiss, 1958; Mendelson et al., 2016). In the limit case, the limiting distribution of the number of unique elements U is normal. The asymptotic mean for the unique number of elements over total number of measurements m is $\mathbb{E}\frac{U}{m} = 1 - \exp\{-r\}$, where r is the bootstrap sampling rate. The asymptotic variance of the unique ratio is then $\text{Var}\frac{U}{m} = \frac{1}{m}(\exp\{-r\} - (1 + r)\exp\{-2r\})$, which converges to zero when m is large.

4.9.6.3 Finite Number of Measurements m Cases are Empricially close to the Asymptotic Case

We generate 10000 trials of random sampling with replacement and then calculate the empirical unique percentage by counting the ratio of the number of unique elements over the total number of measurements m. The theoretical mean is consistently lower than the mean for a finite m. From the plot, the average unique elements in finite m cases $m = 50, 75, 100, 150$ are not so different from the theoretical value of the infinite sample size.

The empirical mean and the asymptotic value are plotted in Figure 4.11a, indicating that the numeric unique percentage is not that far from the asymptotic value even when the number of estimates is finite and small. Figure 4.11b illustrates the region between the mean plus and minus one standard of deviation. As the asymptotic case, the theoretical standard deviation converges to zero. We plotted the cases $m = 150$ and $m = 50$ compared to the asymptotic case. For both, the variance is tight and gets smaller when m becomes larger. For the same m, the variance of the unique number of elements become larger when the bootstrap ratio L/m is large.

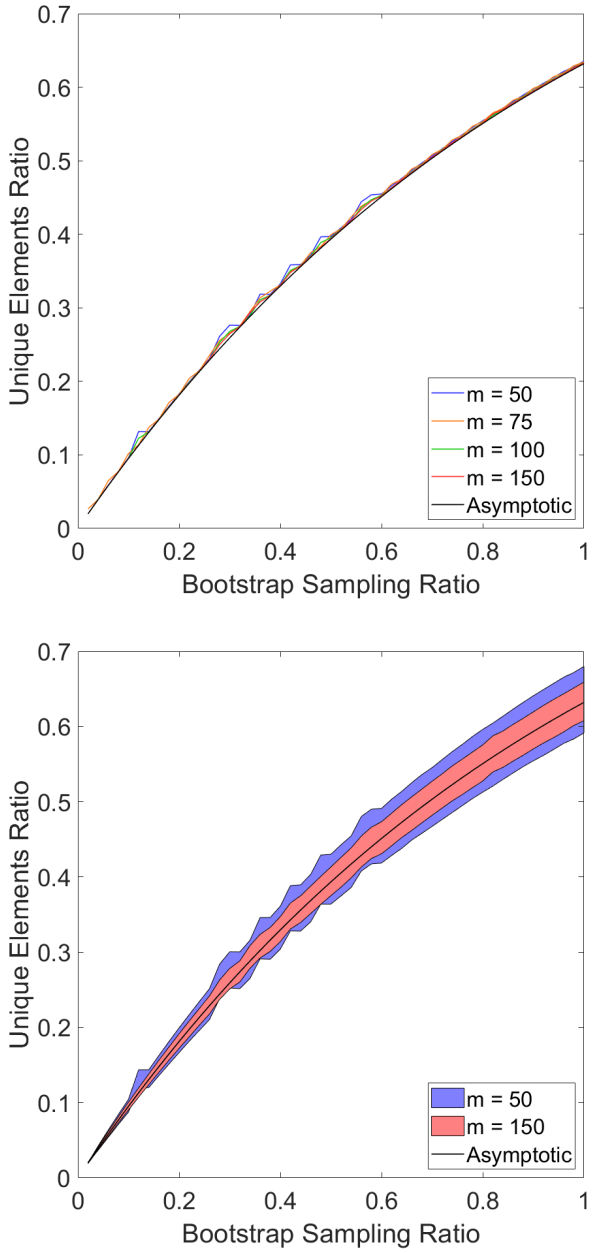


Figure 4.11: Unique element ratios with various bootstrapping ratios. **Top**: The mean of unique element ratios under various bootstrapping ratios with various total number of measurements: $m = 50, 75, 100, 150$ and theoretical asymptotic value when $m \to \infty$. **Bottom**: The area between of empirical mean plus and minus one empirical standard deviation. The blue and the red area corresponds to $m = 50$ and 150 respectively. The black line is the asymptotic mean and the asymptotic variance converges to zero.

Chapter 5

Collaborative Scheme in CS 3-D OCT Recovery

In this chapter, we demonstrate an application that employs collaborative regression from sub-sampled measurements from a compressed sensing imaging hardware. The algorithm has shown significant improvement in signal recovery over conventional ℓ_1 minimization. The algorithm is also efficient in recovering signals of large dimensions.

(Part of the contents of this chapter has been under review for Optics Express.)

5.1 Introduction

Throughout the past decade, optical coherence tomography (OCT) has proven to be a versatile tool in medical diagnostics allowing for example, straightforward assessment of the progress of macular degeneration, multiple sclerosis, and glaucoma (Fercher et al., 2003; Tomlins and Wang, 2005). Using the spectrally-dependent interference of two light waves, OCT interrogates depth information of a sample, including many different tissues, and can be scanned to collect a

three-dimensional (3D) data cube of an object's structure. The massive amount of data collected in a volumetric OCT acquisition poses challenges for data throughput, storage, and manipulation, typically requiring data compression prior to any data processing (Zhang and Kang, 2010; Grulkowski et al., 2009). Additionally, in many applications the signal capture should be performed quickly to avoid motion artifacts that can distort the image (Yun et al., 2004; Zhang et al., 2009).

In the past decade the information theory community introduced the concept of compressed sensing (CS) (Candès and Tao, 2005; Candes, Romberg, and Tao, 2006; Donoho, 2006; Baraniuk, 2007; Candès and Wakin, 2008), suggesting that the sparsity of natural signals can be leveraged to reduce the number of samples required to capture signals of interest. This has been adopted by the medical field and applied to MRI imaging, photo-acoustic imaging, and OCT (Lustig, Donoho, and Pauly, 2007; Guo et al., 2010; Liu and Kang, 2010). Particularly, extensive work on under sampling OCT data followed by implementing CS algorithms has demonstrated successful reconstruction with less than 20% of measurements required by the Shannon/Nyquist theory (Liu and Kang, 2010; Xu, Huang, and Kang, 2014). By taking advantage of the compressibility of volumetric OCT data (Wu et al., 2012), the data cube can be under sampled without the loss of image quality (Young et al., 2011). However many of these methods still require the entire data cube to be recorded at the Nyquist rate followed by digital under sampling after acquisition (Liu and Kang, 2010; Xu, Huang, and Kang, 2014; Wu et al., 2012; Young et al., 2011). Although compressed reconstruction after data acquisition allows for real time visualization, such digital domain sub-sampling

fails to address the physical domain bottleneck at the signal acquisition stage due to serial sampling limits of an ADC.

We have developed an architecture to implement CS at high speed in the optical domain using optical signal processing in a technique we termed Continuous High-Rate Photonically Enabled Compressed Sensing (CHiRP-CS) and have leveraged this architecture for high-speed flow microscopy, ultra wide-band radio frequency (RF) sensing, and preliminary work on OCT (Bosworth and Foster, 2013; Bosworth et al., 2015b; Bosworth et al., 2015a; Stroud et al., 2016).

For OCT we directly record optically computed inner products between the interference signals and know binary patterns such that each ADC sample contains information spanning the entire A-scan depth profile. This allows for real-time optical domain data compression of the OCT signal prior to detection by a high-speed ADC. Consequently, we can remove the limits imposed by Nyquist sampling on A-scan rates and thus decouple imaging rate from ADC sampling rate. Similar recent work, investigated the compressive acquisition and reconstruction of A-scans at 1.51-MHz rates using a 66% compression ratio (Mididoddi et al., 2017). Here, we demonstrate A-scan acquisition rates from 14.4-MHz to 144-MHz using compression ratios from 26% to 2.6%, respectively. Such low compression ratios are achieved by implementing compressive sampling in the axial A-scan dimension followed by joint reconstruction of the entire volumetric C-scan data utilizing the multi-dimensional sparsity of the full 3D signals of interest.

5.2 Challenges

We image a test sample composed of an empty PDMS microfluidic channel mounted on a glass slide and acquire a 1-mm by 1.5-mm transverse scan with a 10-um resolution resulting in a $100 \times 150 \times 384$ OCT data cube. To visualize the reconstruction result, we determine the 3D power spectrum of the recovered spectral modulation signal by calculating the magnitude for each discrete spectral modulation frequency from $\widehat{\boldsymbol{X}}$ by summing the squared coefficients of positive and negative frequencies at each corresponding location. For our 384 reconstructed spectral pixels, this procedure results in 192 axial pixels in the depth image.

The 3D C-scan of the reference signal captured using time-stretch OCT is shown in Figure 5.1a, in which the axial dimension spans from bottom to top. If we just apply ℓ_1 minimization, the typical reconstructed solution is displayed in Fig 5.1b. This solution detects the locations of the layers fairly accurately however we lose a lot of energy towards the top of the C-scan corresponding to the region of high frequency spectral modulation. Also, the energy for the middle slice with curved channel is aliased to these higher frequency bands.

Although using ℓ_1 minimization will not give a recovered image of acceptable quality due to missing major depth information, it is acceptable for use as a start point since it detects the other two major frequency bands correctly. We therefore proposed a weighted ℓ_1 minimization that promotes a row-sparsity pattern across each frequency band, and we will use the ℓ_1 minimization solution as a initialization point for our proposed framework.

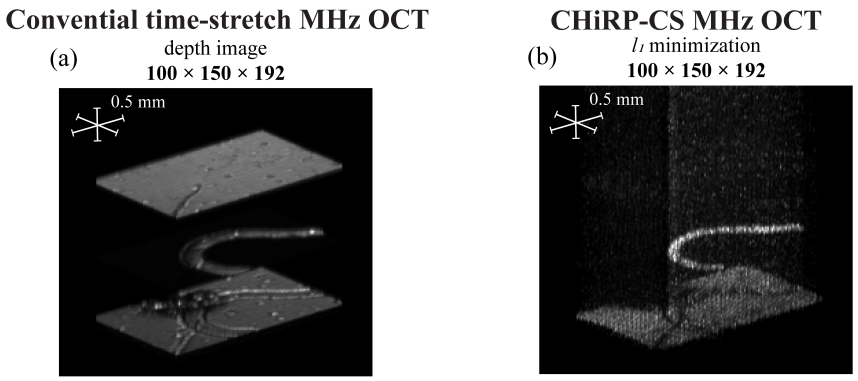


Figure 5.1: C-scan recovery from 80 compressed measurements, or an 18-MHz A-scan rate. a) Ground truth reference. b) The output of the ℓ_1 minimization recovery of measurements from our CHiRP-CS hardware system. The entire top board that corresponds high frequency is missing.

5.3 Proposed Method

5.3.1 Problem Formulation

To scan a 3D object $\boldsymbol{S}$ of size $N_1 \times N_2 \times N_3$, the compressed measurements were taken along the third, A-scan dimension. We generate $N_1 \times N_2$ number of measurements for a C-scan, with each measurement of length M. Let $\boldsymbol{y}^{(i,j)} \in \mathbb{R}^M, i = 1, 2, ..., N_1, j = 1, 2, ..., N_2$ denote the measurements vector collected from the signal at a spatial location (i, j) : $\boldsymbol{s}^{(i,j)} \in \mathbb{R}^M$, and $\boldsymbol{A}^{(i,j)} \in \mathbb{R}^{M \times N_3}$ is the sensing matrix associated with that measurement vector. For the reconstructed cube in the frequency domain, these dimensions are: $N_1 = 100, N_2 = 150$ and $N_3 = 384$. The recovered image is of size $100 \times 150 \times 192$. The depth dimension is half of the one in the frequency domain since along the depth direction, the final image is calculated on the energy from both positive and negative frequencies. In the recovery algorithm, the number of measurements per line M ranges from

10 to 100, corresponding to compression ratios M/N_3 from 2.6% to 26%. The mathematical model of the system in matrix form is simply:

$$\boldsymbol{y}^{(i,j)} = \boldsymbol{A}^{(i,j)}\boldsymbol{s}^{(i,j)} + \boldsymbol{z}^{(i,j)}, \tag{5.1}$$

where $\boldsymbol{z}^{(i,j)}$ represents the noise vector added to the (i,j)-th noiseless measurement. The noise vector is generated from multiple sources such as the linearization approximation error of the system, noise from the data collection process, as well as pre-processing error, etc.

With OCT signals, we know the spectral modulation frequencies correspond to different depths of the object. Additionally, in many OCT applications, the object is sparse in the number of reflected layers, resulting a small number of cosine tones along the compressed A-scan dimension. Therefore, if we represent the signal in frequency domain, we would expect the coefficients to be sparse. Accordingly, the inverse discrete Fourier transform matrix is used as the sparsifying basis in the recovery algorithm.

Similar to many noise processes, the noise of this system is predominately of high frequency. Consequently, the high frequency parts of the signal become less distinguishable from the noise. While applying sparse recovery with the classic ℓ_1 min norm, we observe that as the sparsity level λ is increasing, the power of the high frequency part vanishes much faster than the low frequency part. As a result, there is a bias toward the low frequency interference depths. To resolve this issue, we employ a weighted ℓ_1 minimization to this problem. The general weighted ℓ_1 minimization method was proposed in (Khajehnejad et al., 2009).

5.3.2 Collaborative Weighted Sparse Recovery

Our proposed method is composed of two steps. First, we generate a initialization from the classic ℓ_1 minimization problem. The weight vector $\boldsymbol{w}$ is then computed based on the power spectrum of the initialization solution.

$$\widehat{\boldsymbol{X}}_{\boldsymbol{0}} = \underset{\boldsymbol{X}_0 \in \mathbb{C}^{N_3 \times N_1 N_2}}{\arg\min} \frac{1}{2} \sum_{k=1}^{N_1 N_2} \|\boldsymbol{y}^{(k)} - \boldsymbol{A}^{(k)} \boldsymbol{D} \boldsymbol{x}^{(k)}\|_2^2 + \lambda_0 \sum_{k=1}^{N_1 N_2} \|\boldsymbol{x}_0^{(k)}\|_1, \tag{5.2}$$

where $\|\boldsymbol{x}\|_1$ calculates the sum of absolute values of all entries of $\boldsymbol{x}$: $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{N_3} |x_i|$ and $\lambda_0 > 0$ is the sparsity balancing ratio. The non-negative regularization parameter λ_0 balances the ratio of sparsity of the solution and the fitness of the solution with respect to the measurements. The larger value of sparsity level leads to a more sparse solution.

Second, we pass the ℓ_1 minimization solution to the weighted ℓ_1 minimization algorithm. For non-negative weights $\boldsymbol{w} = (w_1, w_2, ..., w_{N_3})^T, w_i \geq 0$, the weighted ℓ_1 norm of vector $\boldsymbol{x}$ given the weight vector $\boldsymbol{w}$ is defined as: $\|\boldsymbol{x}\|_{\boldsymbol{w},1} = \sum_{i=1}^{N_3} w_i |x_i|$. For variable matrix $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N_1 N_2)})$, we find:

$$\widehat{\boldsymbol{X}} = \underset{\boldsymbol{X} \in \mathbb{C}^{N_3 \times N_1 N_2}}{\arg\min} \frac{1}{2} \sum_{k=1}^{N_1 N_2} \|\boldsymbol{y}^{(k)} - \boldsymbol{A}^{(k)} \boldsymbol{D} \boldsymbol{x}^{(k)}\|_2^2 + \lambda \sum_{k=1}^{N_1 N_2} \|\boldsymbol{x}^{(k)}\|_{\boldsymbol{w},1}, \tag{5.3}$$

where the sparisfying transform $\boldsymbol{\Phi}$ is the inverse discrete Fourier basis. The weight $\boldsymbol{w}$ in (5.3) is a function associated with the power spectrum of $\widehat{\boldsymbol{X}}_{\boldsymbol{0}}$ from solving (5.2). Smaller weights encourage higher amplitudes by reducing the contribution in the penalty function. We estimated the support based on the amplitudes of the ℓ_1 minimization solution. To suppress noise on non-support locations and hence enhance the image quality, the weight vector $\boldsymbol{w}$ is set to be small for locations in the support and large for non-support locations. Essentially,

our choice of $\boldsymbol{w}$ enforces joint sparsity along the depth dimension. Within the support, the recovered locations with smaller amplitudes have smaller weights in order to boost the energy of the reconstructed signal in high frequency bands. More details on determining the weight vector are explained in the Section 5.3.3.

The above two optimization problems (5.2)(5.3) can be solved efficiently by several methods such as proximal gradient descent methods (Combettes and Wajs, 2005; Beck and Teboulle, 2009), gradient projection (Figueiredo, Nowak, and Wright, 2007), alternating minimization (Boyd et al., 2011), approximate message passing (Donoho, Maleki, and Montanari, 2009), etc. In our implementation, we use gradient projection for sparse reconstruction (GPSR) (Figueiredo, Nowak, and Wright, 2007) to solve both ℓ_1 and weighted ℓ_1 minimization programs.

The full image reconstruction process is illustrated in Figure 5.2, where the inputs are the compressed measurements $\boldsymbol{y}$ and pseudo-random binary pattern $\boldsymbol{A}$, and the output is the reconstructed image $\widehat{\boldsymbol{X}}$. The iterative ℓ_1 minimization algorithm using GPSR is fed the measurements and the known binary patterns to reconstruct an initial image. This is sent into a weighted ℓ_1 minimization algorithm that produces the final image. The details regarding how to set the weight vector in this method is elaborated in next section.

5.3.3 Details on Designing Weighting Vectors

In this section, we explain the details of designing a proper weighting scheme. We first plot the 1D power spectrum for each depth for the ℓ_1 minimization solution. In Figure 5.3a, clearly for non-support part, the energy is reasonably large. The 1D power spectrum statistics along the third dimension are used to

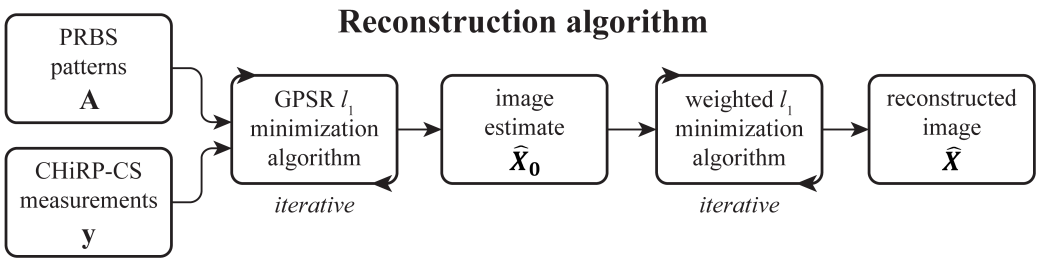
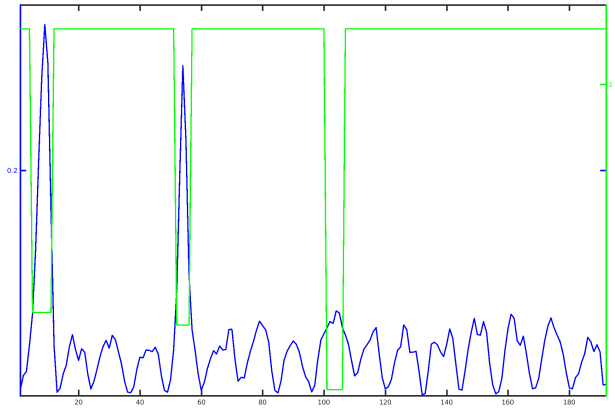


Figure 5.2: The flowchart of the proposed two-step weighted algorithm. The collaborative row-sparsity is enforced inexplicitly through the weighted ℓ_1 minimization step.

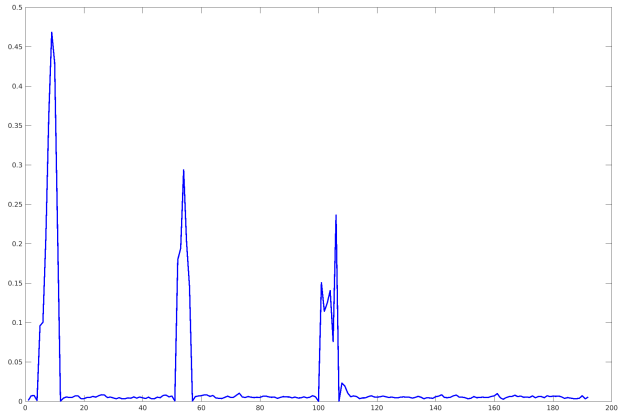
determine the weights for the weighted ℓ_1 minimization algorithm to resolve these issues. First, we search for peaks from the ℓ_1 minimization solution to determine the support and non-support part of the signal. We use three parameters to determine the peaks: the threshold ratio τ between 0 to 1 on amplitudes to be in the support, the minimum width $\delta \in \boldsymbol{Z}^+$ to be a strong peak, and minimum radius from the peak boundaries to the local maximum (peak centers) $\rho \in \boldsymbol{Z}^+$.

We first eliminate the support locations based on the first two parameters. Then for the detected peaks, we find the local maximum within each local region, treat it as the peak center and extend its boarder to 2ρ if the border to center distance is less than ρ, encouraging a smoother transition between support and non-support part in the solution. In our experiment, we pick $\tau = 15\%, \delta = 3, \rho = 2$. We find peaks within top 15% magnitudes with width at least 3 pixels and radius at least 2 pixels from the peak center.

Next we compute the weight vector $\boldsymbol{w}$ according to the result from peak detection. Within each peak region, the square root of the averaged power are used as weights. Since the ℓ_1 minimization recovery solution has a higher energy at a lower frequency peak and a lower energy at a high frequency peak, this



(a) Blue line: one dimensional (1-D) Spectrum of ℓ_1 minimization; Green linke: Weights assigned from detected peak indicator function



(b) The one dimension power spectrum of reconstructed signal

Figure 5.3: The effect of using reweighted sparse recovery on 1D power spectrum. **Top**: A typical power spectrum from ℓ_1 minimization in blue and the weight in green is calculated by the algorithm. **Bottom**: 1D spectrum of reweighted sparse recovery solution by using the weights in green in (a).

method assigns a lower weight for a high frequency peak compared to a lower one. For the non-support part, we set their weights to be a constant higher than all the weights for the support. Here, we set them to be two times the square root of the max value of the 1D power spectrum vector. All parameters in the peak finding and weight functions are determined empirically. The higher the weight, the more the magnitude of corresponding location tends toward zero because the weight contributes more to the penalty. Our designed weight vector significantly decreases the reconstruction noise and boosts the energy in high frequency bands. The weighting function calculated from our proposed method is shown in Figure 5.3a. By applying this weight, we use reweighted ℓ_1 minimization as given in equation (5.3). The resulting 1D power spectrum is shown in 5.3b. As we expected, the third profile peak occurs and the power on non-support are suppressed significantly.

5.4 Experimental Results

5.4.1 The Comparison between Randomly Sub-sampled Measurements and Temporally-continuous Measurements

We compare the result of using subsampled measurements $\boldsymbol{A}$ and $\boldsymbol{y}$ with using temporally-continuous measurements. Figure 5.4 gives the results with 50 measurements per A-scan direction with two different sampling pattern: one is sequentially-continuous measurements and the other one is random subsets. Random sub-sampling gives better results than a sequentially-continuous measurements of the same size. In the top layer, the reconstruction result from

sequentially-continuous measurements has more aliasing than the result from random subsets.

This is because random sub-sampled measurements are more incoherent than sequentially continuous measurements, which means that the random subsets provide more useful information with the same number of measurements.

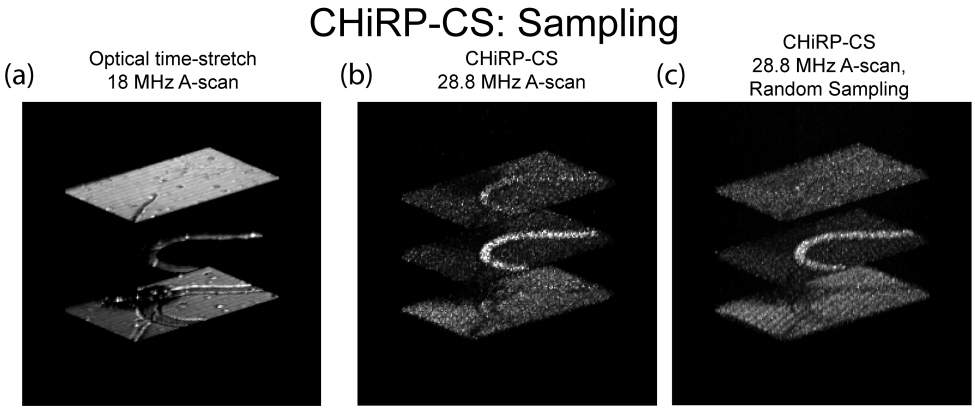


Figure 5.4: Comparison of reconstruction with continuous measurements versus random measurements. (a) The reference. (b) The reconstruction result using 50 continuous measurements. PSNR = 24.3 dB. (c) The reconstruction results using a random set of 50 measurements at each location. PSNR = 25.7 dB.

5.4.2 Performance with Various Number of Measurements (Sampling rate)

The impact of compression rate on image quality is illustrated in Figure 5.5. In Figure 5.5a, the increase in PSNR from 30 to 50 measurements corresponds to properly reconstructing the third layer. The 100 measurement reconstruction shows some noise reduction, but the increase in PSNR is minimal. Although reconstructions using 10 and 30 samples fail to reconstruct all of the layers, there

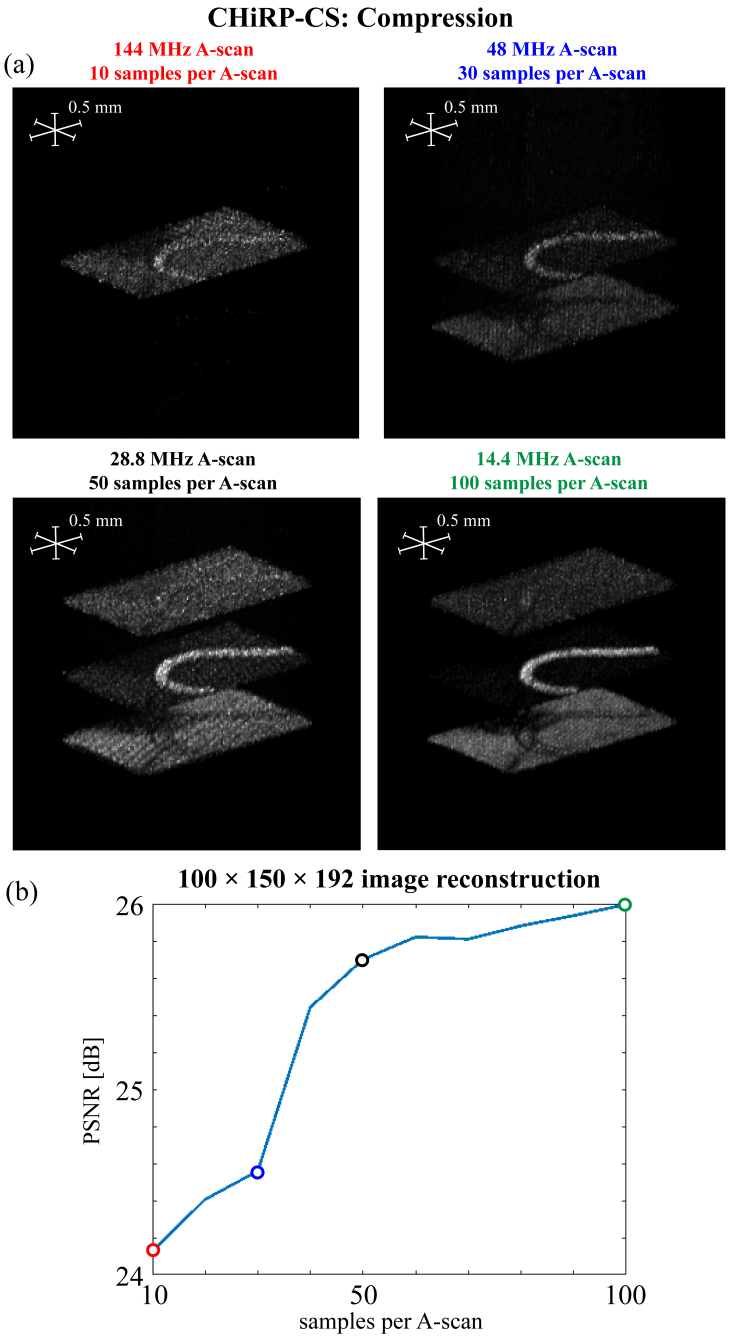


Figure 5.5: (a) Example C-scan reconstructions of an 100 x 150 x 192 depth image with 10, 30, 50, and 100 measurements, or 144-MHz, 48-MHz, 28.8-MHz, and 1.44-MHz A-scan rates, respectively. (b) The PSNR of the CS reconstruction vs the number of compressed measurements used for reconstruction shows an increase in PSNR around 50 measurements where the third layer becomes clearly visible.

is still a clear object reconstructed. This ability is due to the sparsity enforced by the compressed reconstruction, meaning that the sparser the image, the fewer samples that are needed to be used to form an image. As shown in Figure 5.5b, when the number of measurements increases, the algorithm is able to recover more details of the signal and the reconstruction image quality is improved visually, as quantified by Peak Signal-to-Noise Ratio (PSNR). We calculate PSNR by comparing the reference Nyquist-sampled time-stretch measurement of the section of microfluidic channel to our CS reconstructions.

5.5 Summary

We demonstrate a compressed sensing OCT system that both addresses the need for high-speed data acquisition and offers physical-domain data compression. Using this approach we show real-time data compression of the OCT signals allowing A-scan acquisition from deeply sub-Nyquist sets of measurements.

In this system, different frequencies corresponds to various depth information. Using conventional ℓ_1 minimization, the high frequencies components are not recovered correctly and resulting one major depth profile is completely missing. Additionally, we develop a reconstruction approach that leverages the joint sparsity in both axial and transverse dimensions to efficiently and accurately reconstruct the full C-scan data cube. The resulting reweighted scheme is able to recover all depth profiles. Experimentally, we show successful C-scan reconstruction of all layers in a microfluidic channel on a glass slide with only 13% of the measurements required by Nyquist sampling, or a 28.8-MHz A-scan rate. Our results demonstrate that the conventional limit on axial pixel rate

imposed by the ADC sampling rate in conventional Nyquist-sampling OCT systems can be surpassed using sub-Nyquist CS sampling resulting in faster collection times, less data, and lower cost detectors and ADCs.

5.6 Appendix: Hardware Systems

5.6.1 CHiRP-CS Sampling System

The CHiRP-CS optical system is shown in the lower half of Figure 5.6. The approach begins with ultrafast laser pulses from a MLL source at a center wavelength of 1550 nm and native 90-MHz repetition rate. Pulses from this laser are first temporally multiplexed twice up to a repetition rate of 360 MHz and then sent through a 853 ps/nm dispersion optical fiber module to spread the 22 nm of optical bandwidth over 20 ns. This resulting spectrum-to-time mapping allows the spectrum of the laser pulses to be modulated in time with an EOM operating at 11.52 Gb/s using a pre-programmed pseudo-random binary sequence (PRBS). With the 2.77-ns repetition period of the laser pulse train and the over 20-ns chirped pulse duration, neighboring pulses overlap greatly, resulting in up to three pulses modulated simultaneously. However this modulation is imparted on different portions of their spectra resulting in unique spectral patterns on each pulse. Modulating the 360-MHz pulse source at 11.52 Gb/s results in sequential pulses spectrally shifting the PRBS pattern by 32 bits. In order to reach our final 1.44-GHz repetition rate, the patterned source is temporally multiplexed twice more, with large delays of 166 ns and 306 ns. The dispersed pulses are re-compressed in 50 km of SMF resulting in a sequence of temporally distinct short pulses with unique spectral patterns at a 1.44-GHz repetition

rate. This approach results in minimum spectral features of 12.53 GHz, over almost 3 THz of optical bandwidth, determined by the dispersion and EOM modulation rate. Notably, given the 11.52-Gb/s modulation rate the 12.53-GHz minimum spectral features are near the modulation limit from interbit spectral distortion. Subsequently passing the spectrally-coded pulses through the OCT interferometer piece-wise multiplies the binary spectral pattern with the OCT interference signal of interest. Finally, detecting the temporally compressed pulse after this process achieves optical integration. Thus this process optically computes the vector inner product between the binary spectral patterns and the OCT interference signal such that only a single ADC sample of each pulse is required to capture each compressed measurement.

5.6.2 OCT Interferometer System

The OCT interferometer is constructed with a mirror as a reference arm and a two-dimensional laterally scanning sample arm with a single 7.5-mm focal length aspheric lens. The input CHiRP-CS source is split by a 80/20 coupler, where 80 percent enters the sample arm and 20 percent reaches the reference mirror. The returned pulses are then recombined in a 50/50 coupler and detected in a balanced configuration by a 1.6-GHz amplified photo-detector. As illustrated in Figure 5.6, a single sample of the compressed pulse amplitude yields the inner product between the spectral interference signals and the unique binary spectral pattern imposed on each pulse by the CHiRP-CS system. Each inner product contains information spanning the entire spectrum and can be described mathematically as

$$y = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + z, \tag{5.4}$$

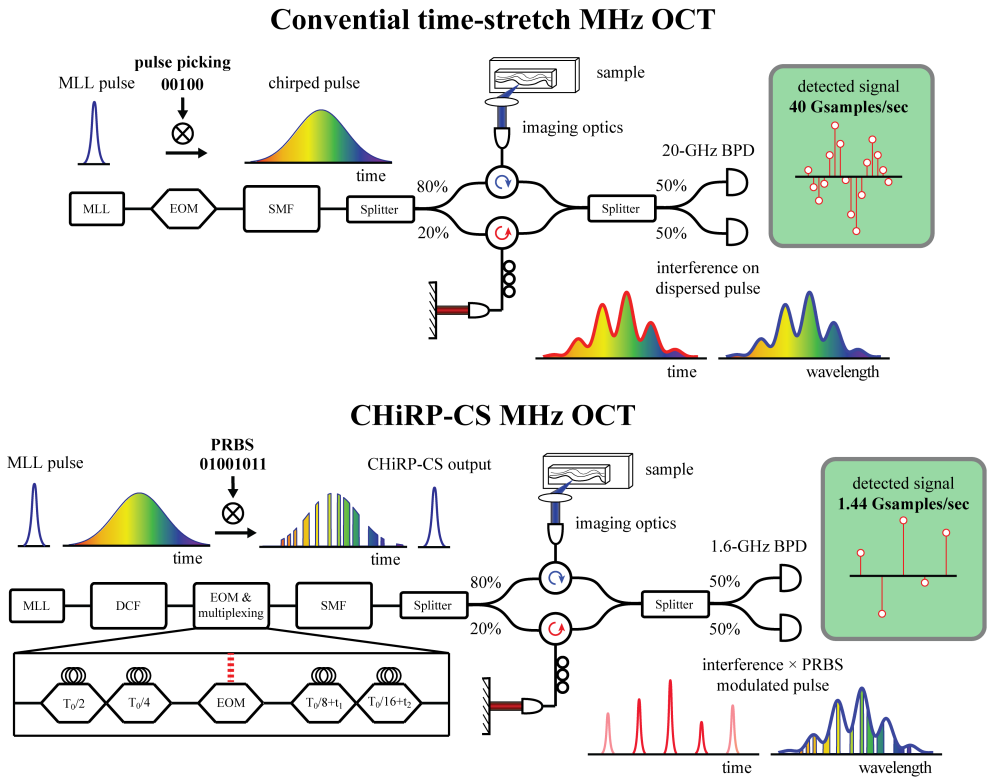


Figure 5.6: Experimental setup for conventional time-stretch MHz OCT is shown on top. A 90-MHz MLL is pulse picked down to a 18-MHz repetition rate and dispersed to over 8 nanoseconds using SMF. This is sent into the OCT interferometer and the returned pulses are detected with a 20-GHz balanced photo-detector and digitized at 40 Gsamples/s. Our CHiRP-CS MHz OCT system is shown at the bottom. Pulses from a 90-MHz MLL are dispersed in DCF, spectral encoded with a PRBS using an EOM, then temporally compressed in SMF. The pulses are temporally multiplexed four times, before and after the modulation, for a final 1.44-GHz repetition rate. The pulses are sent into the OCT interferometer and detected with a 1.6-GHz balanced photo-detector and digitized at 1.44 Gsamples/s. MLL - mode-locked laser, SMF - single mode fiber, DCF - dispersion compensating fiber, EOM - electro-optic modulator, PRBS - psudeo-random binary sequence, BPD - balanced photo-detector.

where $\boldsymbol{a}$ is one row in the known pseudo-random binary pattern $\boldsymbol{A}$, and $\boldsymbol{s}$ is the interference signal under test a sample location, and z denotes the noise. We can then reconstruct the depth profile using a sequence of such compressed measurements (vector $\boldsymbol{y}$) and compressed sensing algorithms necessitating far fewer measurements than a comparable Nyquist sampling system such as a time-stretch system shown in the top of Figure 5.6. Specifically, here we reconstruct the OCT spectra onto 384 spectral pixels using 10 to 100 samples yielding compression ratios from 2.6% to 26%. Practically, this compression allows us to use a much lower speed detector and ADC than the comparable time-stretch system, while achieving the same imaging speed or, alternately higher imaging speed using comparable electronics.

In order to investigate the fidelity of our CS approach, we first acquire Nyquist-sampled time-stretch OCT measurement for comparison (Xu et al., 2014; Goda et al., 2012). As shown on top in Figure 5.6, this time-stretch system is implemented as follows. The 90-MHz MLL is sent into an EOM to be pulse picked down to 18 MHz. Pulse picking is necessary to avoid pulse overlap after the pulses then propagate through 853 ps/nm dispersion module to achieve sufficient spectrum-to-time mapping for comparable axial dimension to our CS system. This signal detected by a 20-GHz linear balanced photo-detector, then digitized by a 20-GHz bandwidth 40-GS/s oscilloscope. This time-stretch measurement is used as the ground truth in comparison to our compressed sensing approach to evaluate reconstructed image quality.

Chapter 6

Conclusion and Discussions

In this thesis, we proposed alternative methods for solving sparse regression compared to the conventional Lasso approach.

The first framework that we propose is Bagging in Sparse Regression, which employs bagging procedure in the bootstrap samples. In this work, we extend the conventional Bagging scheme in sparse recovery with an adjustable bootstrap sampling ratio L/m and derive error bounds for the algorithm associated with L/m and the number of estimates K in Theorem 5 and Theorem 6. Bagging is particularly powerful when the number of measurements m is small.

The performance limits associated with different choices of bootstrap sampling ratio L/m and number of estimates K are analyzed theoretically. Simulation results show that a lower L/m ratio $(0.6 - 0.9)$ leads to better performance than the conventional choice $(L/m = 1)$, especially in challenging cases with low levels of measurements. With the reduced sampling rate, SNR improves over the original Bagging method by up to 24% and over the base algorithm ℓ_1 minimization by up to 367%. With a properly chosen sampling ratio, a reasonably small number of estimates $(K = 30)$ gives a satisfying result, although increasing

K is discovered to always improve or at least maintain performance. Moreover, the reduced sampling rate shows a performance improvement measured by the recovered SNR, and it is over the conventional Bagging algorithm by up to 24%.

Although Bagging in sparse regression is very robust and outperform ℓ_1 minimization by significant margins in noisy cases, it is not guaranteed that the Bagging solution is sparse during the averaging process. The second framework that we propose a collaborative signal recovery framework named JOBS, which is a collaborative scheme also using bootstrap samples. The usage of row sparsity among different estimators enforces support consistency and hence the final solutions' sparsity level is preserved.

Similar to Bagging, JOBS improves the robustness of sparse recovery in challenging scenarios of noisy environments and/or limited measurements. We carefully analyze theoretical properties of JOBS such as BNSP and BRIP and then we further derive error bounds for JOBS described in Theorem 15 and Theorem 16. We analyze theoretic recovery behaviors with respect to two key parameters: the bootstrap sampling ratio L/m and the number of estimates K. We also extend the bootstrap scheme to a common sub-sampling variation of the framework and study its connection to the original bootstrap scheme. Finally, experiments on sparse regression as well as classification (face recognition task) are conducted for validation. Simulation results show that the proposed algorithm consistently outperforms Bagging and Bolasso among most parameter settings (L, K).

JOBS yields state-of-the-art recovery performances, outperforming ℓ_1 minimization and other existing bootstrap-based techniques in the challenging case

of limited measurements. With a proper choice of the bootstrap sampling ratio in the range of $(0.3 - 0.5)$ and a reasonably large number of estimates ($K \geq 30$), the recovery SNR improvement over the baseline ℓ_1-minimization algorithm can reach up to 336%. JOBS also improves the baseline performance in face recognition on the cropped AR dataset by 3% with the optimal bootstrapping sampling ratio set at 0.5. In both regression and classification tasks, JOBS solutions are validated to be more sparse than Bagging solutions.

The following points are several important properties that we discovered for JOBS. (*i*) JOBS is particularly powerful when the number of measurements m is limited, outperforming ℓ_1 minimization by a large margin. (*ii*) JOBS achieves desirable performances with relatively low bootstrap ratio L/m (peak performance occurs at $0.3 - 0.5$ whereas the sub-sampling variation requires only $0.2 - 0.4$). It also demands a relatively small K (around 30 in our experimental study). (*iii*) The optimal sampling ratio for JOBS is lower than that of Bagging while achieving similar results. This results in a lower computation complexity for JOBS. (*iv*) JOBS solutions are generally more sparse than Bagging's – a desirable property in sparse recovery. (*v*) Extending the framework in classification tasks, only JOBS framework also improves classification accuracy; the improvement over ℓ_1 minimization is 3% for cropped AR dataset and 1.5% for Yale B dataset with approximately $90\% - 10\%$ training-testing split. (*vi*) JOBS has an advantage over Bagging because the expected amount of data required for obtaining peak performance is less; we have precise control of the sparsity level, and heuristically we produce sparser solutions compared to Bagging, which matches our motivation of developing JOBS framework.

In the OCT recovery work, the problem has two main challenges: (i) The speeding construction on sensing system, with limited measurements at each location, the reconstruction is quite challenging. (ii) Both noise and some part of signal are high frequency and therefore the high frequency signals are hard to recover. We have shown the power for the collaborative approach on subsets of measurements. The reconstruction result is much better than the conventional sparse recovery. The algorithm also outperforms the same collaborative approach but choosing continuous measurements.

Chapter 7

Future works

In this thesis, we have proposed a general collaborative framework and there are a lot of real-world applications that have a similar flavor. Although the major contributions of this thesis lie in the theoretical understanding of the framework, there are various potential future works that are related to this topic. In this section, we will discuss a few future directions along this line of research.

7.1 Extension to Dictionary Learning

There are many dictionary learning frameworks that utilize sparse representation to perform classification such as (Aharon, Elad, and Bruckstein, 2006), (Mairal, Bach, and Ponce, 2012), etc. These dictionary learning methods have shown a better performance than the SRC algorithm that we mentioned in the experiment. The reason is that in dictionary learning frameworks, not only is the sparse solution optimized, but also the dictionary matrix $\boldsymbol{A}$ is optimized, which improves the representation of data.

A natural extension will be incorporating this novel sparse recovery framework in dictionary learning and improves supervised learning. We expect the improvement in the regression will improve differentiation between different classes on difficult examples and therefore improve classification frameworks based on dictionary learning and sparse representation.

7.2 Non-ideal Sensing Matrix

In our simulations in both regression and classification frameworks, an i.i.d. random matrix is used as sensing matrix $\boldsymbol{A}$. In practice, there are cases that the sensing matrix are non-i.i.d. or highly correlated. These conditions occur commonly and are lack of tight bounds in theoretical studies. In numerical studies, we have shown that the JOBS method improves hard cases at the ranges where the conventional method tends to fail. It is also possible that this technique will be able to improve the performance in hard cases such as a non-i.i.d random matrix or highly correlated cases.

7.3 Spatially-Correlated Subsets

For some image recognition problems, not random subsets, instead, spatially-correlated subsets are observed. For example, partial face recognition is a problem that often arises in practical settings and applications. The main challenge is that the lack of information will cause the failure of feature extraction and images with different sizes and alignments may need to be treated differently. Our proposed framework solved these two issues simultaneously. We propose a sparse representation-based algorithm for this problem. Our method firstly trains

a dictionary and the classifier parameters in a supervised dictionary learning framework and then aligns the partially observed test image and seeks for the sparse representation with respect to the training data alternatively to obtain its label. We also analyze the performance limit of sparse representation-based classification algorithms on partial observations. We did experiment on AR face data-set. We training on holistic face of training and the testing is performed on testing data, which are image patches of various sizes of the original test data. Details are presented in our work (Liu, Tran, and Chin, 2016).

In our experiment, each test is evaluated on one image patch. The collaborative scheme in JOBS has shown to improve the result if multiple subsets are involved. For classification on spatially-correlated measurements (image patches for iamge classification), a collaborative scheme over multiple patches of one test may potentially improve the classification result.

7.4 Variations in Optimization and Obtaining Final Estimator

There are variations of regularizer that can be imposed for row-sparsity. In this work, the proposed method uses $\ell_{1,2}$ norm. Other choices of regularizer can be used such as infinity norm, non-convex norm, greedy approach such as simultaneous orthogonal matching pursuit can possibly be used. The final estimator in this paper is obtained by taking average over multiple bootstrapped estimators. In statistics, median is also a common choice for robust estimator.

7.5 Streaming Implementations

Classical optimization methods for representation learning that process all data at once (batch method) to minimize certain loss functions often requires large amount of memory and are computationally costly. In large-scale problems or in the streaming setting where new data are coming in constantly, these algorithms are not very efficient because they are limited by the increased storage and complexity requirements. Online Learning methods, which process one data point at a time, and mini-batch algorithms which update predictions over a small set of training data are desirable in training systems to dynamically adapt to new data, solving a summation approximation of the loss functions on individual samples or batches of samples. JOBS has shown to be effective in with lower L, which will be an advantage if the algorithm is streaming. The streaming variation of the proposed method could possibly be in the form of the following equation, in which the superscript t denotes the $t-th$ iteration:

$$\boldsymbol{X}^{(t+1)} = \underset{X^{(t)}}{\arg\min} \|\boldsymbol{X}^{(t)}\|_{1,2} \;\; \text{s.t.} \;\; \sum_{j=1}^{K} \|\boldsymbol{y}_{[\mathcal{I}_j]} - \boldsymbol{A}[\mathcal{I}_j]\boldsymbol{x}_j\|_2 \leq \epsilon^J, \{\mathcal{I}_1\mathcal{I}_2, .., \mathcal{I}_K\}^{(t)}. \tag{7.1}$$

Potentially a small L and reasonable K can be used for the streaming algorithm.

In the proposed work JOBS, we also have shown that JOBS procedure is a relaxation of the ℓ_1 minimization for all choices of $\mathcal{I}$ and therefore (7.1) serves as the stochastic approximation using stochastic mini-batch method with relaxation. Hopefully the streaming form of JOBS can improve the convergence of mini-batch method for ℓ_1 minimization.

7.6 Extension of JOBS Framework to Other Penalty Functions

JOBS is a framework as a relaxation of the standard sparsity penalty ℓ_1 minimization. Similar framework can be extended to other penalty functions such as: *i)* Other block or group sparsity defined by particular group partition *ii)* non-convex sparse promoting penalty function such as $\ell_p, 0 < p < 1$. *iii)* Classification loss functions.

7.7 Regression via Deep Learning Framework

In the era of big data, Deep learning methods today have become one of the most powerful methods for classification tasks. Key to the success is the ability to learn rich feature hierarchies (Girshick et al., 2014) , with low-level features like edges and colors learned at lower layers, which are combined together in the higher layers to detect complex shapes and patterns in a fully-differentiable end-to-end framework. Although most advantages of deep learning are initially discovered in classification, doing regression via deep learning recently has achieved some success. Reconstruction algorithms such as ReconNet (Kulkarni et al., 2016), DeepInverse (Mousavi and Baraniuk, 2017) and (Lohit, Kulkarni, and Turaga, 2016) have shown promising performances. We would like to try to employ deep learning to solve regression problem of our proposed method JOBS and hopefully the performance can be further improved.

7.8 Multi-layer Sparse Coding Neural Networks

Convolution Neural Network (CNN) recently has shown its success in various application field such as vision, speech and natural language processing. Recently there have been works that use multi-layer sparse coding (Sulam et al., 2018; Sun, Nasrabadi, and Tran, 2019) to achieve a hierarchical sparse features extraction structure, which has shown to reach CNN performances on multiple tasks.

Our work improves on sparse regression, which is a single layer sparse coding layer in a multi-layer framework. As such, an interesting extension is to see whether a similar technique can improve the performances of a supervised learning framework based on multi-layer sparse coding networks.

References

Cohen, A., W. Dahmen, and R. DeVore (2009). "Compressed sensing and best k-term approximation". In: *Journal of the American mathematical society* 22.1, pp. 211–231.

Candes, E. J (2008). "The restricted isometry property and its implications for compressed sensing". In: *Comptes Rendus Mathematique* 346.9, pp. 589–592.

Candes, E. J, J. Romberg, and T. Tao (2006). "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *IEEE Trans. on Info. theory* 52.2, pp. 489–509.

Donoho, D. L (2006). "Compressed sensing". In: *IEEE Trans. on Info. theory* 52.4, pp. 1289–1306.

Candess, E. and J. Romberg (2007). "Sparsity and incoherence in compressive sampling". In: *Inverse prob.* 23.3, p. 969.

Aharon, M., M. Elad, and A. Bruckstein (2006). "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". In: *IEEE Trans. on Sig. Proc.* 54.11, pp. 4311–4322.

Yang, J., J. Wright, T. S Huang, and Y. Ma (2010). "Image super-resolution via sparse representation". In: *IEEE trans. on Image Proc.* 19.11, pp. 2861–2873.

Liu, L., T. D Tran, and S. P Chin (2016). "Partial face recognition: A sparse representation-based approach". In: *the 41st Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 2389–2393.

Chen, Y., T. T Do, and T. D Tran (2010). "Robust face recognition using locally adaptive sparse representation". In: *the 17th IEEE Int. Conf. on Image Proc. (ICIP)*. IEEE, pp. 1657–1660.

Bosworth, B. T, J. R Stroud, D. N Tran, T. D Tran, S. Chin, and M. A Foster (2015b). "Ultrawideband compressed sensing of arbitrary multi-tone sparse radio frequencies using spectrally encoded ultrafast laser pulses". In: *Optics Letters* 40.13, pp. 3045–3048.

Krizhevsky, A., I. Sutskever, and G. E Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems (NeurIPs)*, pp. 1097–1105.

Efron, B. (1979). "Bootstrap methods: another look at the jackknife". In: *The Annals of Stat.* 7.1, pp. 1–26.

Breiman, L. (1996). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.

Liu, L., S. P Chin, and T. D Tran (2019). "Reducing Sampling Ratios and Increasing Number of Estimates Improve Bagging in Sparse Regression". In: *2019 53rd Annual Conf. on Information Sciences and Systems (CISS)*. IEEE.

Hall, P. and R. J Samworth (2005). "Properties of bagged nearest neighbour classifiers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.3, pp. 363–379.

Sabzevari, M., G. Martinez-Munoz, and A. Suarez (2014). "Improving the robustness of bagging with reduced sampling size". In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.

Zaman, F. and H. Hirose (2009). "Effect of subsampling rate on subbagging and related ensembles of stable classifiers". In: *International Conference on Pattern Recognition and Machine Intelligence*. Springer, pp. 44–49.

Bach, F. R (2008a). "Bolasso: model consistent lasso estimation through the bootstrap". In: *Proceedings of the 25th Int. Conf. on Machine learning (ICML)*. ACM, pp. 33–40.

Wright, J., A. Y Yang, A. Ganesh, S. S Sastry, and Y. Ma (2009). "Robust face recognition via sparse representation". In: *IEEE Trans. on Pattern Anal. Mach. Intelligence* 31.2, pp. 210–227.

Natarajan, B. K. (1995). "Sparse approximate solutions to linear systems". In: *SIAM J. on computing* 24.2, pp. 227–234.

Chen, S., D. L Donoho, and M. Saunders (2001). "Atomic decomposition by basis pursuit". In: *SIAM review* 43.1, pp. 129–159.

Tibshirani, R. (1996). "Regression shrinkage and selection via the Lasso". In: *J. of the Royal Stat. Society. Series B*, pp. 267–288.

Figueiredo, M. AT, R. D Nowak, and S. J Wright (2007). "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems". In: *Selected Topics in Sig. Proc., IEEE J. of* 1.4, pp. 586–597.

Beck, A. and M. Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. on imaging sciences* 2.1, pp. 183–202.

Osborne, M. R, B. Presnell, and B. A Turlach (2000). "A new approach to variable selection in least squares problems". In: *IMA J. of numerical anal.* 20.3, pp. 389–403.

Combettes, Patrick L and Valerie R Wajs (2005). "Signal recovery by proximal forward-backward splitting". In: *Multiscale Modeling and Simulation* 4.4, pp. 1168–1200.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011). "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine Learning* 3.1, pp. 1–122.

Donoho, D., A. Maleki, and A. Montanari (2009). "Message-passing algorithms for compressed sensing". In: *Proceedings of the National Academy of Sciences* 106.45, pp. 18914–18919.

Candès, E. J. and T. Tao (2005). "Decoding by linear programming". In: *IEEE Trans. Inf. Theory*, pp. 4203–4215.

Baraniuk, R. G (2007). "Compressive sensing". In: *IEEE Signal Process. Mag.* 24.4.

Candès, E. J and M. B Wakin (2008). "An introduction to compressive sampling [a sensing/sampling paradigm that goes against the common knowledge in data acquisition]". In: *IEEE Signal Process. Mag.* 25.2, pp. 21–30.

Duarte, M. F, M. A Davenport, S. Takhar, J. N Laska, T. Sun, K. F Kelly, and R. G Baraniuk (2008). "Single-pixel imaging via compressive sampling". In: *IEEE signal processing magazine* 25.2, pp. 83–91.

Lustig, M., D. Donoho, and J. M Pauly (2007). "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic Resonance in Medicine* 58.6, pp. 1182–1195.

Guo, Z., C. Li, L. Song, and L. V Wang (2010). "Compressed sensing in photoacoustic tomography in vivo". In: *Journal of biomedical optics* 15.2, p. 021311.

Liu, X. and J. U Kang (2010). "Compressive SD-OCT: the application of compressed sensing in spectral domain optical coherence tomography". In: *Optics express* 18.21, pp. 22010–22019.

Mairal, J., F. Bach, and J. Ponce (2012). "Task-driven dictionary learning". In: *IEEE Trans. on Pattern Anal. Mach. Intelligence* 34.4, pp. 791–804.

Bootstrapping. https://en.wikipedia.org/wiki/Bootstrapping.

Berg, E. and M. P. Friedlander (2008). "Probing the Pareto frontier for basis pursuit solutions". In: *SIAM J. on Scientific Computing* 31.2, pp. 890–912. DOI: 10.1137/080714488. URL: http://link.aip.org/link/?SCE/31/890.

Wright, S. J, R. D Nowak, and M. AT Figueiredo (2009). "Sparse reconstruction by separable approximation". In: *IEEE Trans. on Sig. Proc.* 57.7, pp. 2479–2493.

Baraniuk, R., M. Davenport, R. DeVore, and M. Wakin (2008). "A simple proof of the restricted isometry property for random matrices". In: *Constructive Approx.* 28.3, pp. 253–263.

Liu, L., Chin S. P, and T. D Tran (2019). "JOBS: Joint-Sparse Optimization from Bootstrap Samples". In: *IEEE International Symposium on Information Theory, ISIT 2019*. IEEE, pp. 2689–2693.

Baron, D., M. F Duarte, M. B Wakin, S. Sarvotham, and R. G Baraniuk (2009). "Distributed compressive sensing". In: *arXiv preprint arXiv:0901.3403*.

Heckel, R. and H. Bolcskei (2012). "Joint sparsity with different measurement matrices". In: *Proc. of 50th Annual Allerton Conf. on Communication, Control, and Computing, (ALLERTON)*. IEEE, pp. 698–702.

Sun, L., J. Liu, J. Chen, and J. Ye (2009). "Efficient recovery of jointly sparse vectors". In: *Advances in neural information processing systems (NeurIPs)*, pp. 1812–1820.

Bach, F. R (2008b). "Consistency of the group lasso and multiple kernel learning". In: *The J. of Machine Learning Research* 9, pp. 1179–1225.

Deng, W., W. Yin, and Y. Zhang (2011). "Group sparse optimization by alternating direction method". In: *Rice CAAM Report TR11-06*. International Society for Optics and Photonics, 88580R.

Bühlmann, P. L and B. Yu (2000). "Explaining bagging". In: *Research report/Seminar für Statistik, Eidgenössische Technische Hochschule Zürich*. Vol. 92. Seminar für Statistik, Eidgenössische Technische Hochschule (ETH).

Buhlmann, P. L (2003). "Bagging, Subagging and Bragging for Improving some Prediction Algorithms". In: *Research Report*. Seminar Fur Statistik, ETH Zurich, Switzerland.

Gao, Y., J. Peng, and Y. Zhao (2015). "On the Null Space Property of ℓ_q-Minimization for in Compressed Sensing". In: *J. of Function Spaces* 2015.

Eldar, Y. C and M. Mishali (2009). "Robust recovery of signals from a structured union of subspaces". In: *IEEE Trans. on Info. Theory* 55.11, pp. 5302–5316.

Hoeffding, W. (1963). "Probability inequalities for sums of bounded random variables". In: *J. of the American statistical association* 58.301, pp. 13–30.

Georghiades, A. S, P. N Belhumeur, and D. J Kriegman (2001). "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose". In: *IEEE Trans. on Pattern Anal. Mach. Intelligence* 23.6, pp. 643–660.

Martinez, A. M and A. C Kak (2001). "PCA versus LDA". In: *IEEE Trans. on Pattern Anal. Mach. Intelligence* 23.2, pp. 228–233.

The Birthday Problem. http://www.math.uah.edu/stat/urn/Birthday.html.

Weiss, I. (1958). "Limiting distributions in some occupancy problems". In: *The Annals of Mathematical Statistics* 29.3, pp. 878–884.

Mendelson, A. F, M. A Zuluaga, B. F Hutton, and S. Ourselin (2016). "What is the distribution of the number of unique original items in a bootstrap sample?" In: *arXiv*.

Fercher, Adolf F, Wolfgang Drexler, Christoph K Hitzenberger, and T. Lasser (2003). "Optical coherence tomography-principles and applications". In: *Reports on progress in physics* 66.2, p. 239.

Tomlins, P. H and R. Wang (2005). "Theory, developments and applications of optical coherence tomography". In: *Journal of Physics D: Applied Physics* 38.15, p. 2519.

Zhang, K. and J. U Kang (2010). "Real-time 4D signal processing and visualization using graphics processing unit on a regular nonlinear-k Fourier-domain OCT system". In: *Optics express* 18.11, pp. 11772–11784.

Grulkowski, I., M. Gora, M. Szkulmowski, I. Gorczynska, D. Szlag, S. Marcos, A. Kowalczyk, and M. Wojtkowski (2009). "Anterior segment imaging with Spectral OCT system using a high-speed CMOS camera". In: *Optics express* 17.6, pp. 4842–4858.

Yun, S. H, G. J Tearney, J. F de Boer, and B. E Bouma (2004). "Pulsed-source and swept-source spectral-domain optical coherence tomography with reduced motion artifacts". In: *Optics Express* 12.23, pp. 5614–5624.

Zhang, K., W. Wang, J. Han, and J. U Kang (2009). "A surface topology and motion compensation system for microsurgery guidance and intervention based on common-path optical coherence tomography". In: *IEEE Transactions on Biomedical Engineering* 56.9, pp. 2318–2321.

Xu, D., Y. Huang, and J. U Kang (2014). "Volumetric (3D) compressive sensing spectral domain optical coherence tomography". In: *Biomedical optics express* 5.11, pp. 3921–3934.

Wu, A. B, E. Lebed, M. V Sarunic, and M. F Beg (2012). "Quantitative evaluation of transform domains for compressive sampling-based recovery of sparsely sampled volumetric OCT images". In: *IEEE Transactions on Biomedical Engineering* 60.2, pp. 470–478.

Young, M., E. Lebed, Y. Jian, P. J Mackenzie, M. F Beg, and M. V Sarunic (2011). "Real-time high-speed volumetric imaging using compressive sampling

optical coherence tomography". In: *Biomedical optics express* 2.9, pp. 2690–2697.

Bosworth B, T and M. A Foster (2013). "High-speed ultrawideband photonically enabled compressed sensing of sparse radio frequency signals". In: *Optics letters* 38.22, pp. 4892–4895.

Bosworth, B. T, J. R Stroud, D. N Tran, T. D Tran, S. Chin, and M. A Foster (2015a). "High-speed flow microscopy using compressed sensing with ultrafast laser pulses". In: *Optics express* 23.8, pp. 10521–10532.

Stroud, J. R, B. T Bosworth, D. N Tran, T. D Tran, S. Chin, and M. A Foster (2016). "72 MHz A-scan optical coherence tomography using continuous high-rate photonically-enabled compressed sensing (CHiRP-CS)". In: *CLEO: Science and Innovations*. Optical Society of America, SM2I–1.

Mididoddi, C. K, F. Bai, G. Wang, J. Liu, S. Gibson, and C. Wang (2017). "High-throughput photonic time-stretch optical coherence tomography with data compression". In: *IEEE Photonics Journal* 9.4, pp. 1–15.

Khajehnejad, M. A, W. Xu, A S Avestimehr, and B. Hassibi (2009). "Weighted ℓ_1 minimization for sparse recovery with prior information". In: *2009 IEEE international symposium on information theory*. IEEE, pp. 483–487.

Xu, J., C. Zhang, J. Xu, K Wong, and K Tsia (2014). "Megahertz all-optical swept-source optical coherence tomography based on broadband amplified optical time-stretch". In: *Optics letters* 39.3, pp. 622–625.

Goda, K., A. Fard, O. Malik, G. Fu, A. Quach, and B. Jalali (2012). "High-throughput optical coherence tomography at 800 nm". In: *Optics express* 20.18, pp. 19612–19617.

Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587.

Kulkarni, K., S. Lohit, P. Turaga, R. Kerviche, and A. Ashok (2016). "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449–458.

Mousavi, A. and R. G Baraniuk (2017). "Learning to invert: Signal recovery via deep convolutional networks". In: *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, pp. 2272–2276.

Lohit, S., K. Kulkarni, and P. Turaga (2016). "Direct inference on compressive measurements using convolutional neural networks". In: *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, pp. 1913–1917.

Sulam, J., V. Papyan, Y. Romano, and M. Elad (2018). "Multilayer convolutional sparse modeling: Pursuit and dictionary learning". In: *IEEE Trans. on Sig. Proc.* 66.15, pp. 4090–4104.

Sun, X., N. M Nasrabadi, and T. D Tran (2019). "Supervised Deep Sparse Coding Networks for Image Classification". In: *IEEE trans. on Image Proc.* 29, pp. 405–418.

Biography

Luoluo Liu received the B.S. degree in electrical engineering from Southwest Jiaotong University, China in 2013 and M.S. degrees in Electrical and Computer Engineering, applied mathematics and statistics, in 2015, 2019 respectively, both from the Johns Hopkins University, Baltimore, MD. Currently, she is pursuing the Ph.D. degree from The Johns Hopkins University in the Department of Electrical and Computer Engineering. Her research interests are compressed sensing and sparse representations, machine learning, and large-scale optimization problems, including a wide range of applications such classification, object detection, solving inverse problems as well as theory. She worked at Siemens Healthineers at Princeton, New Jersey in the summer of 2018 as a deep learning research intern. She owns a patent on quality assessment of Magnetic Resonance images using deep learning approach. Luoluo received National Scholarship from Ministry of Education of the P.R. China in 2012 and university outstanding undergraduate student in 2013.