Deterministic Compressed Sensing

Sina Jafarpour

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: A. Robert Calderbank

November 2011

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Deterministic Compressed Sensing	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Princeton University,Department of Computer Science,Princeton,NJ,08540**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The central goal of compressed sensing is to capture attributes of a signal using very few measurements. The initial publications by Donoho and by Candes and Tao have been followed by applications to image compression, data streaming, medical signal processing, digital communication and many others. The emphasis has been on random sensing but the limitations of this framework include performance guarantees storage requirements, and computational cost. This thesis will describe two deterministic alternatives. The rst alternative is based on expander graphs. We rst show how expander graphs are appropriate for compressed sensing in terms of providing explicit and e cient sensing matrices as well as simple and e cient recovery algorithms. We show that by reformulating signal reconstruction as a zero-sum game we can e ciently recover any sparse vector. We provide a saddle-point reformulation of the expander-based sparse approximation problem, and propose an e cient expander-based sparse approximation algorithm, called the GAME algorithm. We show that the restricted isometry property of expander matrices in the ‘1-norm ensures that the GAME algorithm always recovers a sparse approximation to the optimal solution with an ‘1=‘1 data-domain approximation guarantee. We also demonstrate resilience to Poisson noise. The Poisson noise model is appropriate for a variety of applications, including low-light imaging and digital streaming where the signal-independent and/or bounded noise models used in the compressed sensing literature are no longer applicable. We develop a novel sensing paradigm based on expander graphs and propose a MAP algorithm for recovering sparse or compressible signals from Poisson observations. We support our results with experimental demonstrations of reconstructing average packet arrival rates and instantaneous packet counts at a router in a communication network, where the arrivals of packets in each ow follow a Poisson process. The second alternative is based on error correcting codes. We show that deterministic sensing matrices based on second order Reed Muller codes optimize average case performance. We also describe a very simple algorithm, one-step thresholding that succeeds in average case model selection and sparse approximation, where more sophisticated algorithms, developed in the context of random sensing, fail completely. Finally, we provide an algorithmic framework for structured sparse recovery, where some extra prior knowledge about the sparse vector is also available. Our algorithm called Nesterov Iterative Hard-Thresholding (NIHT) uses the gradient information in the convex data error objective to navigate over the non-convex set of structured sparse signals. Experiments show however that NIHT can empirically outperform ‘1-minimization and other state-of-the-art convex optimization-based algorithms in sparse recovery.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**209**	

Abstract

The central goal of compressed sensing is to capture attributes of a signal using very few measurements. The initial publications by Donoho and by Candes and Tao have been followed by applications to image compression, data streaming, medical signal processing, digital communication and many others. The emphasis has been on random sensing but the limitations of this framework include performance guarantees, storage requirements, and computational cost. This thesis will describe two deterministic alternatives.

The first alternative is based on expander graphs. We first show how expander graphs are appropriate for compressed sensing in terms of providing explicit and efficient sensing matrices as well as simple and efficient recovery algorithms. We show that by reformulating signal reconstruction as a zero-sum game we can efficiently recover any sparse vector. We provide a saddle-point reformulation of the expander-based sparse approximation problem, and propose an efficient expander-based sparse approximation algorithm, called the GAME algorithm. We show that the restricted isometry property of expander matrices in the ℓ_1-norm ensures that the GAME algorithm always recovers a sparse approximation to the optimal solution with an ℓ_1/ℓ_1 data-domain approximation guarantee.

We also demonstrate resilience to Poisson noise. The Poisson noise model is appropriate for a variety of applications, including low-light imaging and digital streaming, where the signal-independent and/or bounded noise models used in the compressed sensing literature are no longer applicable. We develop a novel sensing paradigm based on expander graphs and propose a MAP algorithm for recovering sparse or compressible signals from Poisson observations. We support our results with experimental demonstrations of reconstructing average packet arrival rates and instantaneous packet counts at a router in a communication network, where the arrivals of packets in each flow follow a Poisson process.

The second alternative is based on error correcting codes. We show that deterministic sensing matrices based on second order Reed Muller codes optimize average case performance. We also describe a very simple algorithm, one-step thresholding, that succeeds in average case model selection and sparse approximation, where more sophisticated algorithms, developed in the context of random sensing, fail completely.

Finally, we provide an algorithmic framework for structured sparse recovery, where some extra prior knowledge about the sparse vector is also available. Our algorithm, called Nesterov Iterative Hard-Thresholding (NIHT) uses the gradient information in the convex data error objective to navigate over the non-convex set of structured sparse signals. Experiments show however that NIHT can empirically outperform ℓ_1-minimization and other state-of-the-art convex optimization-based algorithms in sparse recovery.

Acknowledgements

My foremost gratitude goes to my adviser, Robert Calderbank, for making the four years of my Ph.D. studies extremely enjoyable and productive. Robert Calderbank is not only a fantastic adviser, but he is also a brilliant mathematician, engineer, inventor, a great friend, and certainly the most influential person in my life, after my parents.

Robert is a great mathematician. In my first year, his unique way of using Algebra for solving complicated Communication and Quantum Computing problems seemed just like doing magic to me. I am extremely grateful to him for teaching me how to do this magic, and for teaching me how to think positively, constructively, and joyfully while doing this magic.

Robert is a wonderful engineer. I will never forget his advice "in a successful research team, everyone should know everything". His mastery in a vast range of areas, from pure mathematics to engineering, has always greatly motivated me to learn about as many inter-disciplinary topics as possible. I'm thankful to him because of his enormous supports that made it possible for me to work with experts in the fields of computer science, mathematics, and electrical engineering.

Robert is a also great inventor. Using mathematics for inventing new practical technologies is something that interests him most. He taught me how to look for a practical problem that interests me enough, and how to tackle the problem with my own toolbox, and from my own perspective. I am always impressed by his extraordinary ability of applying his expertise to practical problems outside his field.

I am extremely fortunate to have such a wonderful adviser. Robert cares about his students a lot. He considers their success in their academic lives as his own success. He always gives complete freedom to his students to choose the problem that interests them most, and then uses his extraordinary ability to find new applications for those problems. Meeting with Robert has always provided me with infinite positive energy. I remember times that I did not have any new result to discuss with him, but I still wanted to meet with him, to get this positive energy, and to learn more from him. I'm especially thankful to Robert for being constantly available for discussions, even at 5 o'clock in the mornings, and even when he was traveling to Asia or Europe. I further thank him for providing me with everything I needed for my research, from buying a laptop, to giving me the access to powerful computational resources. I am also grateful to Robert for his support of my meeting and working with other researchers around the world. I feel so indebted to Robert, that I cannot explain my true gratitude to him by words.

Secondly, I would like to thank my co-adviser, Rob Schapire, for everything that he taught me during our research meetings, in his three machine learning classes, and while I was a TA for his Artificial Intelligence course. Rob is an excellent teacher, researcher, and a true friend. You can always rely on his helps and supports while

facing with any academic or real-life challenge. Rob is a fantastic teacher. It was his excellent machine learning courses that made me consider machine learning as my main research interest. In his classes, Rob always encouraged us to ask any question that came into our mind. I was always impressed by his ability of taking these questions on the fly, and teaching us deep concepts by the means of them. He is also a great researcher. In our meetings, he always listened to me very calm and relaxed while I was explaining my raw ideas. He never mentioned something he was not sure enough about, and was always open to new ideas. One thing that I particularly enjoyed about collaborating with him was that he never imposed his own ideas, and always mentioned them only as suggestions, so that I was able to carefully analyze and criticize them.

Many thanks to the Van Gogh Project team members: Andrei Brasoveanu, Eugene Brevdo, Robert Calderbank, Bruno Cornelis, Ingrid Daubechies, Ann Dooms, Shannon Hughes, Gungor Polatkan, Maximiliaan Martens and Josephine Wolff, for providing a friendly environment, that made the times working on the Van Gogh Project among the most productive and elated times I spent in Princeton. I specially enjoyed the teamwork among the members. By working on this project, I gradually learned how to interact with people with different backgrounds, from art historians to electrical engineers. I specially thank Ingrid Daubechies for her wonderful supervision of the project. Ingrid is a great mathematician. You can find about her great mathematical achievements easily by using a search engine. But one thing about her that impressed me even more was that she always tried to be her own person and do what she likes most. She was always ready to directly talk with the museums, and convince them to trust us and give us their real-data, so that we could try our ideas on those real datasets. I was always amazed by her ability of devising complicated mathematical models for the paintings and at the same time explaining these complicated models to the art historians. I also highly thank Gungor Polatkan for many fruitful discussions that we had even after midnights during the different parts of the project, for his constructive comments in my practice talks, and for showing me a nice Hookah Place near Princeton.

During my summer internships I benefited a lot from the support of my mentors Chris Burges in Microsoft Research, and Patrick Haffner and Howard Karloff in AT&T Labs. I thank Chris for teaching me what "real" massive datasets look like, and how a machine learning expert is supposed to manipulate them efficiently. I thank Patrick and Howard for their invaluable supports, for the fruitful discussions that we had over the last three years, and also for showing me how Theoretical Computer Science, if used properly, has the capability of providing "brilliant" and "correct" practical solutions to real world applications.

I thank Volkan Cevher for offering me a summer internship in EPFL, and for many fruitful discussions that we had over skype and during his visit at Princeton, leading to several publications in the last year of my Ph.D. I also thank Maryam Fazel, MohammadTaghi Hajiaghayi, Henry Landau, David Nahamoo and Irina Rish, Guillermo

Sapiro, Roelof van Zwol, and Rebecca Willett for arranging my short visits to University of Washington, University of Maryland, AT&T Labs, Bell Labs, IBM Research, University of Minnesota, Yahoo! Labs, and Duke University.

I want to take on this opportunity to thank my thesis committee members. I am thankful to Bernard Chazelle, and Amit Signer for serving on my oral generals committee and guiding me in the research. In addition, many thanks to Robert Calderbank, Rob Schapire, and Ingrid Daubechies for reading this thesis carefully, and to David Blei for providing constructive comments.

During the four years of my Ph.D. studies, I have collaborated with many excellent researchers and scientists. I thank all my collaborators for many important things that I learned from them. Thanks to Waheed Bajwa, Srinivas Bangalore, MohammadHossein Bateni, Eugene Brevdo, Andrei Brasoveanu, Chris Burges, Volkan Cevher, Robert Calderbank, Ingrid Daubechies, Marco Duarte, Patrick Haffner, MohammadTaghi Hajiaghayi, Zachary Harmany, Babak Hassibi, Stephen Howard, Howard Karloff, Jeremy Kent, Victoria Kostina, Roummel Marcia, Maximiliaan Martens, Taniya Mishra, Maria Nastasescu, Dan Pei, Ali Pezeshki, Gungor Polatkan, Maxim Raginsky, Alan Ritter, Rob Schapire, Carlos Scheidegger, Rebecca Willett, Josephine Wolff, and Weiyu Xu.

I am very grateful to the administrative staffs at Princeton University, specially Beth Jarvie, Melissa Lawson, and Valerie Marino, who made many things very easy. Also thanks to Kathy Peterson for organizing my visits to Duke University, and to Frank and Dolly Pokrass for their excellent hospitality during my Duke visits.

I would also like to thank all my current and former colleagues at Princeton University, Microsoft Research, and AT&T Labs. In particular, thanks to Vaneet Aggarwal, Lorne Applebaum, Amir Bennatan, Waheed Bajwa, MohammadHossein Bateni, Aditya Bhaskara, Eugene Brevdo, Melissa Carroll, Allison Chaney, Yuejie Chi, Hamid Chinaei, Eden Chlamtac, Mihai Cucuringu, Marco Duarte, Rong Ge, Sean Gerrish, Sreechakra Goparaju, Moritz Hardt, Andrew Harms, Jeff Huang, Shannon Hughes, Pallika Kanani, U Kang, Berk Kapicioglu, Shi Li, Adriana Lopez-Alt, Mohammad Mahmoody, Rajsekar Manokaran, Indraneel Mukherjee, Ali Pezeshki, Gungor Polatkan, Yury Polyanskiy, Yury Pritykin, Alan Ritter, Sushant Sachdeva, Aarti Singh, Karthik Sridharan, David Steurer, Umar Syed, Rodrigo Toso, Anuradha Venugopalan, Aravindan Vijayaraghavan, Yihong Wu, Yiyue Wu, and many others with whom I had many helpful discussions.

I take this opportunity to specially thank my awesome roommates Bahman Hekmatshoar, Mohammad Mahmoody, and Amir Momen Roknabadi. I thank them for amazing, memorable, and joyful moments that we shared, and also for understanding my "no fixed sleep time" habit. I also wish to thank Ali Abdi, Farnaz Absalan, Nasser Alidoust, Nima Alidoust, Sasan Amini, Zahra Aminzare, Hossein Azari, Javad Azimi, Behtash Babadi, MohammadHossein Bateni, Siavosh Behbahani, Mahdi Cheraghchi, Hamid Chinaei, Effat Emamian, Omid Etesami, Mohammad

Farajzadeh, Hamed Firooz, Mahya Ghandehari, Hani Goodarzi, MohammadTaghi Hajiaghayi, Amir Hajian, Hamed Hatami, Masoud Jafari, Darioush Jalali, Majid Janzamin, Maziyar Kalantarzadeh, Maryam Kamali, MohammadReza Kamali, Amin Karbasi, Roya Karimian, Amin Khajehnejad, Sara Kherad, Soheil Mohajer, Mohammad Moharrami, Zuheir Montazeri, Atefeh Mostavi, Marzieh Nabi, Arefeh Nasri, Hanieh Nowzari, Shayan Oveis-Gharan, Vahed Qazvinian, Sanaz Rezaei, Farnaz Ronaghi, Keivan Sadri, Babak Saleh, Alireza Salehi, Sina Salek, Hanie Sedghi, Alireza Shabani, Solamz Shariat, Mohsen Sharifani, Emad Soroush, Mahmood Taherzad, Ali Tajer, Amir Talebi, Noori Torabi, Mostafa Vafadoost, Alireza Zaheri, and many others that I have probably forgotten to mention their names.

I thankfully acknowledge the fellowships and grants that supported my research at Princeton University. I was supported by the Princeton University Fellowship, NSF grants CCF-0915299, and DMS-0914892, SToMP UIUC 2006-06918-10, DARPA 20074005-32905-G, FA9550-08-1-0341, and ONR-N00014-08-1 1110.

Above all, I express my heartfelt gratitude to my beloved parents, my lovely grandparents, and my dear sister Saba. Without their love, support, and encouragement none of my achievements have been possible. I dedicate this thesis to my lovely grandmother, Mahin Sadri.

"Creativity is the ability to introduce order into the randomness of nature"
Eric Hoffer

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Preliminaries

Chapter 1

Introduction

An emerging challenge for information and inference systems is to acquire and analyze the ever-increasing high-dimensional data produced by the vast natural and manmade phenomena. Sampling, streaming, and recoding of even the most primitive data, e.g., in medical imaging and network monitoring, now produce a data deluge that severely stresses the available analog-to-digital converter, digital communication bandwidth and storage resources; hence, the traditional paradigm of capturing an entire data set only to compress it for the subsequent transmission or storage is becoming no longer feasible.

Surprisingly, while the ambient data dimension is large in many problems, the relevant information therein typically resides in a much lower dimensional space. This observation has led to several new theoretical and algorithmic developments under different communities, including theoretical computer science, machine learning, applied mathematics, and digital signal processing. One such development is called *compressed sensing* (CS),[1] which exploits sparse representations [95, 54, 20].

The central goal of compressed sensing is to capture attributes of a signal using very few measurements. In most work to date, this broader objective is exemplified by the important special case in which the measurement data constitute a vector $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$, where $\boldsymbol{\Phi}$ is an $M \times N$ matrix called the *sensing matrix*, $\boldsymbol{\alpha}^*$ is a vector in $\mathbb{R}^N$, which can be well-approximated by a k-sparse vector, where a k-sparse vector is a vector which has at most k non-zero entries, and $\boldsymbol{e_M}$ is additive measurement noise.

When $\boldsymbol{\Phi}$ satisfies the so-called restricted isometry property (RIP), it preserves the geometric information of the set of sparse signals [95, 56]. Based on this observation, we can tractably, stably and provably approximate sparse signals from $M \gtrsim 2k \log\left(\frac{N}{k}\right)$ measurements using convex optimization [54, 55] or greedy algorithms [237, 199].

The role of random measurement in compressive sensing (see [54] and [95]) can be

[1] Not to be confused with the canonical acronym of Computer Science.

viewed as analogous to the role of random coding in Shannon theory. Both provide worst-case performance guarantees in the context of an adversarial signal/error model. Although it is known that certain probabilistic processes generate $M \times N$ measurement matrices that satisfy the RIP with high probability, there is no practical algorithm for verifying whether a given measurement matrix has this property. Storing the entries of a random sensing matrix, and performing matrix-vector multiplications may also require significant resources.

These drawbacks lead us to consider constructions with *deterministic* alternatives, which do not suffer from the same drawbacks. This thesis will describe two deterministic alternatives. The frameworks presented here provide

- easily checkable conditions on special types of deterministic sensing matrices guaranteeing successful sparse approximation and model-selection guarantees;
- storage efficiency, as the entries of these matrices can be computed on the fly, and
- recovery algorithms with lower complexities exploiting the *structure* of the sensing matrices.

The first framework is based on expander graphs. We show that by reformulating signal reconstruction as a zero-sum game we can efficiently recover any sparse vector. We also demonstrate resilience to Poisson noise.

The second alternative is based on algebraic error correcting codes. We show that deterministic sensing matrices based on second order Reed Muller codes optimize average case performance.

Chapter 2

Notation

2.1 Vector Properties

Throughout this thesis, nonnegative reals (respectively, integers) will be denoted by $\mathbb{R}_+$ (respectively, $\mathbb{Z}_+$). By $[-1,1]$, we mean the interval between -1 and 1, whereas $\{-1,1\}$ is the discrete set with the elements -1 and 1. For every integer N, we denote $[N] \doteq \{1,\cdots,N\}$.

We denote vectors by bold small letters $\boldsymbol{v}$, and we denote matrices by bold capital letters $\boldsymbol{\Phi}$. Given a vector $\boldsymbol{u} \in \mathbb{R}^N$ and a set $\mathcal{S} \subseteq [N]$, we will denote by $\boldsymbol{u}_{\mathcal{S}}$ the vector obtained by setting to zero all coordinates of $\boldsymbol{u}$ that are in $\mathcal{S}^c$, the complement of $\mathcal{S}$. Similarly if $\boldsymbol{\Phi}$ is an $M \times N$ matrix, then $\boldsymbol{\Phi}_{\mathcal{S}}$ denotes the $M \times |\mathcal{S}|$ submatrix of $\boldsymbol{\Phi}$ which is obtained by restricting the columns of $\boldsymbol{\Phi}$ to the subset $\mathcal{S}$. Also $\boldsymbol{v_{i\to j}}$ denotes the vector $\boldsymbol{v}$ restricted to entries $i, i+1, \cdots, j$, that is $\boldsymbol{v_{i\to j}} \doteq (v_i, v_{i+1}, \cdots, v_j)$.

A vector $\boldsymbol{v} \in \mathbb{R}^N$ is k-sparse if it has at most k non-zero entries. The support of the k-sparse vector $\boldsymbol{v}$, denoted as $\mathrm{Supp}(\boldsymbol{v})$, indicates the positions of the non-zero elements. The ℓ_0 pseudo-norm of $\boldsymbol{v}$, also called the Hamming weight of $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\|_0$, and indicates the number of non-zero entries of $\boldsymbol{v}$. In other words, $\boldsymbol{v}$ is k-sparse, if and only if $\|\boldsymbol{v}\|_0 \leq k$.

For each positive integer p, the ℓ_p norm of a vector $\boldsymbol{v}$ is defined as

$$\|\boldsymbol{v}\|_p \doteq \left(\sum_{i=1}^{N} |v_i|^p\right)^{\frac{1}{p}}.$$

Also, throughout this thesis we, for every vector $\boldsymbol{v}$ define

$$\|\boldsymbol{v}\|_{\min} \doteq \min_{i:v_i \neq 0} |v_i|,$$

as the magnitude of the smallest nonzero entry of $\boldsymbol{v}$. Holder inequality is an important inequality widely used for bounding the inner-products between arbitrary vectors.

Theorem 2.1 (Holder inequality). *Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be arbitrary vectors in $\mathbb{R}^N$, and let p and q be positive integers such that $\frac{1}{p}+\frac{1}{q}=1$. Then*

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle \leq \|\boldsymbol{u}\|_p \|\boldsymbol{v}\|_q,$$

and the equality holds if and only if there exist real non-negative numbers c_1 and c_2, not both of them zero, such that for every index $i \in \{1, \cdots, N\}$ $c_1|u_i|^p = c_2|v_i|^q$.

Two key operators which are widely used in this thesis are the *Hard Thresholding* and *Soft Thresholding* operators:

Definition 2.2 (Hard Thresholding). *Let $\mathrm{H}_k() : \mathbb{R}^N \to \mathbb{R}^N$ be a function that sets to zero all but the k largest coordinates in absolute value. More precisely, for each $\boldsymbol{v} \in \mathbb{R}^N$, let π be a permutation of $\{1, \cdots, N\}$ such that $|v_{\pi(1)}| \geq |v_{\pi(2)}| \geq \cdots \geq |v_{\pi(N)}|$. Then the vector $\mathrm{H}_k(\boldsymbol{v})$ is a k-sparse vector $\boldsymbol{\alpha}$ where $\alpha_{\pi(i)} = v_{\pi(i)}$ for $i \leq k$ and $\alpha_{\pi(i)} = 0$ for $i \geq k+1$.*

The *hard thresholding* operator $\mathrm{H}_k()$ gives the best k-sparse approximation of any vector $\boldsymbol{\beta} \in \mathbb{R}^N$, that is for every norm p

$$\mathrm{H}_k(\boldsymbol{v}) = \arg \min_{\boldsymbol{v'}:k-\text{sparse}} \|\boldsymbol{v} - \boldsymbol{v'}\|_p. \tag{2.1.1}$$

This best k-sparse approximation can be computed efficiently in time $O(N \log N)$ by first sorting the elements of $\boldsymbol{v}$, and then selecting the k largest elements. Also for every norm p, we define

$$\sigma_k(\boldsymbol{v}) \doteq \|\boldsymbol{v} - \mathrm{H}_k(\boldsymbol{v})\|_1. \tag{2.1.2}$$

In other words, $\sigma_k(\boldsymbol{v})$ is the the best k-term approximation error to $\boldsymbol{v}$ in the ℓ_1 norm.

Definition 2.3 (Soft Thresholding). *For $\theta \in \mathbb{R}^+$, we define the soft thresholding function $\mathrm{S}(\alpha, \theta)$ as*

$$\mathrm{S}(\alpha, \theta) = \begin{cases} \theta & \text{if } \alpha > \theta \\ -\theta & \text{if } \alpha < -\theta \\ \alpha & \text{otherwise.} \end{cases} \tag{2.1.3}$$

For a subset $\mathcal{S} \subseteq [N]$ we will denote by $\boldsymbol{I}_{\mathcal{S}}$ the vector with components $1_{\{i \in \mathcal{S}\}}$, $1 \leq i \leq N$. Given a vector $\boldsymbol{u}$, we will denote by $\boldsymbol{u}^+$ the vector obtained by setting to zero all negative components of $\boldsymbol{u}$: for all $1 \leq i \leq N$, $u_i^+ = \max\{0, u_i\}$. Given two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^N$, we will write $\boldsymbol{u} \succeq \boldsymbol{v}$ if $u_i \geq v_i$ for all $1 \leq i \leq N$. If $\boldsymbol{u} \succeq \varepsilon \boldsymbol{I}_{[N]}$ for some $\varepsilon \in \mathbb{R}$, we will simply write $\boldsymbol{u} \succeq \alpha$. We will write $\succ$ instead of $\succeq$ if the inequalities are strict for all i.

2.2 Matrix Properties

Let $\mathbf{\Phi}$ be an $M \times N$ matrix. We denote the i^{th} column of $\mathbf{\Phi}$ by $\boldsymbol{\varphi_i}$, and denote the entry at the j^{th} row of the i^{th} column of $\mathbf{\Phi}$ by $\varphi_{i,j}$. The null-space of $\mathbf{\Phi}$, denoted by $\mathcal{N}_{\mathbf{\Phi}}$, is the set of all N dimensional vectors $\boldsymbol{v}$ with $\mathbf{\Phi}\boldsymbol{v} = \mathbf{0}$. We also use $\mathbf{\Phi}^\top$ to denote the conjugate transpose of $\mathbf{\Phi}$, and use $\mathbf{\Phi}^\dagger$ to denote the MoorePenrose pseudoinverse of $\mathbf{\Phi}$, that is $\mathbf{\Phi}^\dagger \doteq (\mathbf{\Phi}^\top \mathbf{\Phi})^{-1}\mathbf{\Phi}^\top$.

An $N \times N$ matrix $\boldsymbol{U}$ is unitary if and only if $\boldsymbol{U}^\top = \boldsymbol{U}^{-1}$. It is well known that any $M \times N$ matrix $\mathbf{\Phi}$ can be decomposed as $\mathbf{\Phi} = \boldsymbol{U}\mathbf{\Sigma}\boldsymbol{V}$, where $\boldsymbol{U}$ is an $M \times M$ unitary matrix, $\boldsymbol{V}$ is an $N \times N$ unitary matrix, and $\mathbf{\Sigma}$ is an $M \times N$ diagonal matrix [149]. The elements of the diagonal of $\mathbf{\Sigma}$ are the *singular values* of $\mathbf{\Phi}$.

For each positive integer p, the ℓ_p norm of a matrix $\mathbf{\Phi}$ is defined as

$$\|\mathbf{\Phi}\|_p \doteq \max_{\boldsymbol{v} \in \mathbb{R}^N} \frac{\|\mathbf{\Phi}\boldsymbol{v}\|_p}{\|\boldsymbol{v}\|_p}. \tag{2.2.1}$$

In particular $\|\mathbf{\Phi}\|_2 = \sigma_{\max}(\mathbf{\Phi})$, where $\sigma_{\max}(\mathbf{\Phi})$ is the maximum singular value of $\mathbf{\Phi}$, and

$$\|\mathbf{\Phi}\|_\infty \doteq \max_{i:\{1,\cdots,M\}} \max_{j:\{1,\cdots,N\}} |\varphi_{i,j}|.$$

Similarly, for every integer k, the *restricted* ℓ_p *norm* of a matrix $\mathbf{\Phi}$ is defined as

$$\|\mathbf{\Phi}\|_{k,p} \doteq \max_{\boldsymbol{v}:k-\text{sparse}} \frac{\|\mathbf{\Phi}\boldsymbol{v}\|_p}{\|\boldsymbol{v}\|_p}. \tag{2.2.2}$$

Theorem 2.4. *Let $\mathbf{\Phi}$ be an $M \times N$ matrix, and let $\boldsymbol{v}$ be a vector in $\mathbb{R}^N$. Then $\|\mathbf{\Phi}\boldsymbol{v}\|_\infty \leq \|\mathbf{\Phi}\|_\infty \|\boldsymbol{v}\|_1$.*

Proof. For every index $j \in M$, from the triangle inequality we get

$$\left|(\mathbf{\Phi}\boldsymbol{v})_j\right| = \left|\sum_{i=1}^{N} v_i \varphi_{i,j}\right| \leq \|\mathbf{\Phi}\|_\infty \sum_{i=1}^{N} |v_i| = \|\mathbf{\Phi}\|_\infty \|\boldsymbol{v}\|_1.$$

Therefore $\|\mathbf{\Phi}\boldsymbol{v}\|_\infty = \max_{j \in M} |(\mathbf{\Phi}\boldsymbol{v})_j| \leq \|\mathbf{\Phi}\|_\infty \|\boldsymbol{v}\|_1$. □

An $M \times N$ matrix $\mathbf{\Phi}$ with normalized columns is called a dictionary.

Definition 2.5 (Tight Frame)**.** *A dictionary is a tight-frame with redundancy $\frac{N}{M}$ if for every vector $\boldsymbol{v} \in \mathbb{R}^N$, $\|\mathbf{\Phi}\boldsymbol{v}\|^2 = \frac{N}{M}\|\boldsymbol{v}\|^2$. If $\mathbf{\Phi}\mathbf{\Phi}^\dagger = \frac{N}{M}\mathrm{I}_{M\times M}$, then $\mathbf{\Phi}$ is a tight-frame with redundancy $\frac{N}{M}$.*

The following Proposition states that tight-frames have the lowest spectral norms among all dictionaries of the same size.

Proposition 2.6. *Let $\mathbf{\Phi}$ be an $M \times N$ dictionary. Then $\|\mathbf{\Phi}\|^2 \geq \frac{N}{M}$, and equality holds if and only if $\mathbf{\Phi}$ is a tight frame with redundancy $\frac{N}{M}$.*

2.3 Function Properties

Definition 2.7 (Convex set)**.** *A set $\mathcal{S}$ is convex if for every pair of points $\mathbf{P}$ and $\mathbf{Q}$ in $\mathcal{S}$, and every $\alpha \in [0,1]$, the point $\boldsymbol{R} = \alpha\mathbf{P} + (1-\alpha)\mathbf{Q}$ is also in $\mathcal{S}$.*

Definition 2.8 (Convex function)**.** *A function $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is convex if $\mathcal{S}$ is a convex set and moreover, for every pair of points $\mathbf{P}$ and $\mathbf{Q}$ in $\mathcal{S}$, and every $\alpha \in [0,1]$, we have*

$$\mathcal{R}(\alpha\mathbf{P} + (1-\alpha)\mathbf{Q}) \leq \alpha\mathcal{R}(\mathbf{P}) + (1-\alpha)\mathcal{R}(\mathbf{Q}).$$

Theorem 2.9 (Convex function)**.** *A differentiable function $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is convex if $\mathcal{S}$ is a convex set and moreover, for every pair of points $\mathbf{P}$ and $\mathbf{Q}$ in $\mathcal{S}$*

$$\mathcal{R}(\mathbf{P}) \geq \mathcal{R}(\mathbf{Q}) - \langle(\mathbf{P}-\mathbf{Q}), \nabla\mathcal{R}(\mathbf{Q})\rangle.$$

Definition 2.10. *A differentiable function $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ is strongly convex with parameter σ, if for every pair of points $\mathbf{P}$ and $\mathbf{Q}$ in $\mathcal{S}$,*

$$\mathcal{R}(\mathbf{P}) - \mathcal{R}(\mathbf{Q}) - \langle(\mathbf{P}-\mathbf{Q}), \nabla\mathcal{R}(\mathbf{Q})\rangle \geq \frac{\sigma}{2}\|\mathbf{P}-\mathbf{Q}\|_2^2.$$

Definition 2.11 (Big–O' notation)**.** *$f(n) = O(g(n))$ (alternatively, $f(n) \precsim g(n)$) if $\exists\ c_o > 0, n_o : \forall\ n \geq n_o, f(n) \leq c_o g(n)$, $f(n) = \Omega(g(n))$ (alternatively, $f(n) \succsim g(n)$) if $g(n) = O(f(n))$, and $f(n) = \Theta(g(n))$ (alternatively, $f(n) \asymp g(n)$) if $g(n) \precsim f(n) \precsim g(n)$.*

2.4 Concentration Inequalities

In this section, we provide the main concentration inequalities which are used throughout the thesis.

Theorem 2.12 (Gaussian tail bound)**.** *Let $X \sim \mathcal{N}(0, \sigma^2)$ be a zero-mean Gaussian random variable with variance σ^2. Then for all $0 \leq \epsilon$, we have*

$$\Pr\left[|X| \geq \epsilon\sigma\right] \leq 2\exp\left\{-\frac{\epsilon^2}{2}\right\}.$$

Theorem 2.13 ($\boldsymbol{\ell_\infty}$-Norm of the Projection of a Complex Gaussian Vector)**.** *Let $\boldsymbol{\Phi}$ be a real- or complex-valued $M \times N$ matrix having unit ℓ_2-norm columns and let $\boldsymbol{v}$ be an $N \times 1$ vector having entries independently distributed as $\mathcal{N}(0, \sigma^2)$. Then for any $\epsilon > 0$, we have*

$$\Pr\left(\|\boldsymbol{\Phi}^\top\boldsymbol{v}\|_\infty \geq \sigma\epsilon\right) < \frac{4N}{\sqrt{2\pi}} \cdot \frac{\exp(-\epsilon^2/2)}{\epsilon}. \tag{2.4.1}$$

Proof. Assume without loss of generality that $\sigma = 1$, since the general case follows from a simple rescaling argument. Let $\boldsymbol{\varphi_1}, \ldots, \boldsymbol{\varphi_N} \in \mathbb{C}^M$ be the N columns of $\boldsymbol{\Phi}$ and define

$$z_i \doteq \langle \boldsymbol{\varphi_i}, \boldsymbol{v} \rangle, \ i = 1, \ldots, N. \tag{2.4.2}$$

Note that the z_i's are identically (but not independently) distributed as $z_i \sim \mathcal{N}(0,1)$, which follows from the fact that $v_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and the columns of $\boldsymbol{\Phi}$ have unit ℓ_2-norms. The rest of the proof follows from the facts that

$$\begin{aligned}\Pr\left(\|\boldsymbol{\Phi}^\top \boldsymbol{v}\|_\infty \geq \epsilon\right) &\overset{(a)}{\leq} N \cdot \Pr\left(|\mathrm{Re}(z_1)|^2 + |\mathrm{Im}(z_1)|^2 \geq \epsilon^2\right) \\ &\overset{(b)}{\leq} 2N \cdot \Pr\left(|\mathrm{Re}(z_1)| \geq \frac{\epsilon}{\sqrt{2}}\right) \overset{(c)}{<} \frac{4N}{\sqrt{2\pi}} \cdot \frac{\exp(-\epsilon^2/2)}{\epsilon}.\end{aligned}$$

Here, (a) follows by taking a union bound over the event $\bigcup_i \{|z_i| \geq \epsilon\}$, (b) follows from taking a union bound over the event $\{|\mathrm{Re}(z_1)| \geq \epsilon/\sqrt{2}\} \cup \{|\mathrm{Im}(z_1)| \geq \epsilon/\sqrt{2}\}$ and noting that the real and imaginary parts of z_i's are identically distributed as $\mathcal{N}(0, \frac{1}{2})$, and (c) mainly follows by upper bounding the *complementary cumulative distribution function* [167]. □

Theorem 2.14 ($\boldsymbol{\chi^2}$-concentration [161])**.** *Let $X \sim \chi^2_m$ be a chi-squared random variable with m degrees of freedom, with mean $m\sigma^2$, and with standard deviation $\sqrt{2m}\sigma^2$. Then for all $0 \leq \epsilon \leq \frac{1}{2}$, we have*

$$\Pr\left[X - m\sigma^2 \geq \epsilon m \sigma^2\right] \leq \exp\left\{-\frac{3}{16} m\epsilon^2\right\}.$$

Theorem 2.15 (Azuma's Inequality [12])**.** *Suppose $\langle Z_0, Z_1, \cdots, Z_k \rangle$ is a bounded-difference martingale sequence, that is for each i, $\mathbb{E}[Z_i] = Z_{i-1}$, and $|Z_i - Z_{i-1}| \leq c_i$. Then for all $\epsilon > 0$,*

$$\Pr\left[|Z_k - Z_0| \geq \epsilon\right] \leq 2\exp\left\{\frac{-\epsilon^2}{2\sum_{i=1}^k c_i^2}\right\}.$$

In this thesis, we use the Azuma's Inequality for complex martingale random variables.

Theorem 2.16 (Complex Azuma's Inequality)**.** *Let $\langle Z_0, Z_1, \cdots, Z_k \rangle$ be a set of complex random variables such that, for each i, $\mathbb{E}[Z_i] = Z_{i-1}$, and $|Z_i - Z_{i-1}| \leq c_i$. Then for all $\epsilon > 0$,*

$$\Pr\left[|Z_k - Z_0| \geq \epsilon\right] \leq 4\exp\left\{\frac{-\epsilon^2}{8\sum_{i=1}^k c_i^2}\right\}.$$

Proof. For each random variable Z_i let $X_i \doteq \mathrm{Re}\,(Z_i)$ and $Y_i \doteq \mathrm{Im}\,(Z_i)$, so that $Z_i = X_i + iY_i$. Then $\mathbb{E}\,[X_i] = X_{i-1}$ and $\mathbb{E}\,[Y_i] = Y_{i-1}$. Moreover, by triangle inequality $|X_i - X_{i-1}| \leq |Z_i - Z_{i-1}| \leq c_i$, and $|Y_i - Y_{i-1}| \leq |Z_i - Z_{i-1}| \leq c_i$. Hence, $\langle X_0, \cdots, X_m \rangle$, and $\langle Y_0, \cdots, Y_m \rangle$ form martingale sequences. Now from the triangle inequality we have

$$\Pr\left[|Z_k - Z_0| \geq \epsilon\right] \leq \Pr\left[|X_m - X_0| \geq \frac{\epsilon}{2}\right]$$
$$+ \Pr\left[|Y_m - Y_0| \geq \frac{\epsilon}{2}\right] \leq 4\exp\left\{\frac{-\epsilon^2}{8\sum_{i=1}^{k} c_i^2}\right\}.$$

□

2.5 Group Theory

In this thesis, we will analyze deterministic sensing matrices for which the columns form a group $\mathcal{G}$ under pointwise multiplication. The multiplicative identity is the column $\mathbf{1}$ with every entry equal to 1. The following property is fundamental.

Lemma 2.17. *If a group $\mathcal{G}$ has at least one identity $\boldsymbol{f}$ different from the identity element, then the group $\mathcal{G}$ satisfies $\sum_{\boldsymbol{g}\in\mathcal{G}} \boldsymbol{g} = 0$*

Proof.

$$\boldsymbol{f}\left(\sum_{\boldsymbol{g}\in\mathcal{G}} g\right) = \sum_{\boldsymbol{g}\in\mathcal{G}} (\boldsymbol{f}\boldsymbol{g}) = \sum_{\boldsymbol{g}\in\mathcal{G}} \boldsymbol{g}.$$

Therefore we have

$$(\mathbf{1} - \boldsymbol{f})\left(\sum_{\boldsymbol{g}\in\mathcal{G}} \boldsymbol{g}\right) = 0,$$

and since $\boldsymbol{f} \neq \mathbf{1}$ we must have $\sum_{\boldsymbol{g}\in\mathcal{G}} \boldsymbol{g} = 0$. □

Chapter 3

An Overview of Compressed Sensing

3.1 What is Compressed Sensing?

The central goal of compressed sensing is to capture attributes of a signal using very few measurements [95, 48, 22]. In most work to date, this broader objective is exemplified by the important special case in which the measurement data constitute a vector $f = \mathbf{\Phi}\boldsymbol{\alpha}^*$, where $\mathbf{\Phi}$ is an $M \times N$ matrix called the *sensing matrix* , and $\boldsymbol{\alpha}^*$ is a k-sparse vector in $\mathbb{R}^N$ (with $k \ll N$) [60, 86].

There are three main objectives in compressive sensing:

- (O1): Efficient reconstruction of any k-sparse vector $\boldsymbol{\alpha}^*$ from the measurement vector $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^*$ efficiently.
- (O2): Minimizing the number of required measurements for reconstruction ($M \approx k \ll N$).
- (O3): Robustness against data-domain and measurement-domain noise.

Based on the above objectives, compressed sensing can be viewed as a process consisting of two complementary tasks:

1. (T1): Designing an *appropriate* $M \times N$ sensing matrix $\mathbf{\Phi}$,
2. (T2): Designing an *efficient* reconstruction algorithm.

Objective (O1) requires that the reconstruction algorithm recovers $\boldsymbol{\alpha}^*$ from $\boldsymbol{f}$ without knowing its support *a priori*. A necessary condition for this requirement is that no two k-sparse vectors are mapped to the same low-dimensional vector. Otherwise, there is no way to distinguish the two vectors from the low-dimensional measurement vector. This condition imposes a constraint on the number of required measurements.

Proposition 3.1. *Let $\mathbf{\Phi}$ be an $M \times N$ matrix that does not map any pair of k-sparse vectors into the same low-dimensional measurement vector. Then the rank of any $M \times 2k$ submatrix of $\mathbf{\Phi}$ is $2k$, and therefore $M \geq 2k$.*

Proof. Suppose that there exists an $M \times 2k$ submatrix of $\mathbf{\Phi}$ with rank less than $2k$. Let $\mathcal{B}$ denote the indices of the columns of this submatrix. Then $\mathbf{\Phi}_{\mathcal{B}}$ has non-empty null-space, and therefore there the exists a vector $\boldsymbol{v}$ in the null-space of $\mathbf{\Phi}$ which is $2k$-sparse with $\text{Supp}(\boldsymbol{v}) = \mathcal{B}$. Write $\boldsymbol{v} = \boldsymbol{\alpha} - \boldsymbol{\beta}$, where $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ are k-sparse vectors with disjoint supports. Now we have

$$\mathbf{\Phi}(\boldsymbol{\alpha}) - \mathbf{\Phi}(\boldsymbol{\beta}) = \mathbf{\Phi}(\boldsymbol{\alpha} - \boldsymbol{\beta}) = \mathbf{\Phi}\overline{\boldsymbol{v}} = 0.$$

In other words, there are two distinct k-sparse vectors $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ with $\mathbf{\Phi}\boldsymbol{\alpha} = \mathbf{\Phi}\boldsymbol{\beta}$. This means that no reconstruction algorithm can distinguish them by just looking at the measurement vector. Now suppose $M < 2k$. Then the rank of any $M \times 2k$ submatrix of $\mathbf{\Phi}$ is at most M which will be strictly smaller than $2k$. □

Remark 3.2. *If every $M \times 2k$ submatrix of $\mathbf{\Phi}$ has rank $2k$, then compressed sensing is information theoretically possible using the sensing matrix $\mathbf{\Phi}$. This means that for every k-sparse vector $\boldsymbol{\alpha}^*$, given $\mathbf{\Phi}\boldsymbol{\alpha}^*$, one can recover $\boldsymbol{\alpha}^*$ successfully by performing an exhaustive search over all $\binom{N}{k}$, k-dimensional subspaces of $\mathbb{R}^N$.*

Thus far, we have seen that there is a trade-off between the first two objectives of the compressed sensing, and in order to get Objective (O1) we need to have at least $2k$ measurements. The last objective in compressed sensing is about robustness against noise. Sparse approximation is a measure of stability of different compressed sensing methods, and was originally established by Kashin [166] with later improvements by Gluskin [128, 129].

Definition 3.3 (Sparse Approximation)**.** *Let p and q be positive integers. Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix , and let $\mathcal{A}_{\mathbf{\Phi}}$ be a reconstruction algorithm associated with $\mathbf{\Phi}$. Then $\mathcal{A}_{\mathbf{\Phi}}$ provides ℓ_p/ℓ_q sparse approximation guarantee if and only if there exists absolute constants C_1, C_2, such that for every $\boldsymbol{\alpha}^* \in \mathbb{R}^N$, and $\boldsymbol{e_M} \in \mathbb{R}^M$, given $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$, $\mathcal{A}_{\mathbf{\Phi}}$ can successfully recover a k-sparse vector $\hat{\boldsymbol{\alpha}}$ with*

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_p \leq \frac{C_1}{k^{1-\frac{1}{p}}}\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_q + C_2\|\boldsymbol{e_M}\|_p.$$

In the rest of this section we first focus on the noiseless compressed sensing problem. We will see that noiseless compressed sensing can be efficiently done by designing appropriate matrices based on Reed-Solomon codes [5], and then using specific algorithms for recovering sparse vectors [253]. However, robustness against the noise is a lacking in that approach.

An alternative approach is to start with a *generic* sparse recovery algorithm, and find sufficient conditions a sensing matrix should satisfy in order to guarantee the fidelity of

the recovery algorithm. Linear programming with matrices satisfying the Restricted Isometry Property is an example of this approach [54, 55]. In Section 3.3 we will see that this approach provides robustness against noise. However, the limitations of this approach are large storage and computational requirements. Robustness against the noise also imposes extra lower-bounds on the number of required measurements which are discussed in detail in Section 3.5.

To overcome these difficulties, in this thesis we introduce alternative deterministic matrices which are carefully designed to tackle the robust compressed sensing problem. Our sensing matrices are equipped with custom reconstruction algorithms. These algorithms exploit the structure of the sensing matrix, and provide efficient storage, compression, and recovery, as well as robustness against noise.

3.2 Noiseless Compressed Sensing

In Section 3.1 we introduced three major objectives of compressed sensing. If we ignore the Objective (O3), then we are left with the *noiseless compressed sensing* problem. The main tasks in the noiseless compressive sensing are

1. (NT1): Designing an $M \times N$ sensing matrix $\boldsymbol{\Phi}$ (with $M \approx k \ll N$), such that the rank of any $M \times 2k$ submatrix of $\boldsymbol{\Phi}$ is $2k$,

2. (NT2): Designing an *efficient* algorithm for solving the combinatorial ℓ_0 minimization problem

$$\begin{aligned} &\text{minimize} \|\boldsymbol{\alpha}'\|_0 \\ &\text{subject to } \boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'. \end{aligned} \tag{3.2.1}$$

Solving the combinatorial optimization problem of Equation (3.2.1) is in general NP-hard [197]. However, here we will see an example of a sensing matrix and a reconstruction algorithm that can efficiently recover any k-sparse vector using only $M = 2k$ measurements. We first show why the two tasks about are sufficient to guarantee the achievement of the Tasks (T1), and (T2) in compressed sensing.

Proposition 3.4. *Suppose that $\boldsymbol{\Phi}$ is a sensing matrix satisfying the Task (NT1), and let $\mathcal{A}_{\boldsymbol{\Phi}}$ be an algorithm which efficiently solves the optimization problem of Task (NT2). Let $\boldsymbol{\alpha}^*$ be an arbitrary k-sparse vector. Then given $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$, the reconstruction algorithm $\mathcal{A}$ efficiently recovers $\boldsymbol{\alpha}^*$ uniquely.*

Proof. (NT2) guarantees that the Algorithm ($\mathcal{A}_{\boldsymbol{\Phi}}$) always finds a k-sparse vector, $\hat{\boldsymbol{\alpha}}$, such that $\boldsymbol{f} = \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$. On the other hand, (NT1) guarantees that if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are two k-sparse vectors with $\boldsymbol{\Phi}\boldsymbol{\alpha} = \boldsymbol{\Phi}\boldsymbol{\beta}$, then $\boldsymbol{\alpha} = \boldsymbol{\beta}$. Therefore, since both $\boldsymbol{\alpha}^*$ and $\hat{\boldsymbol{\alpha}}$ are k-sparse and $\boldsymbol{\Phi}\boldsymbol{\alpha}^* = \boldsymbol{f} = \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$, we must have $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$. □

The ℓ_0-minimization problem of Equation 3.2.1 can be viewed as a channel coding problem using linear codes defined over the field of real numbers. To see this, let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix , with null-space $\mathcal{N}_{\mathbf{\Phi}}$. Let $\boldsymbol{\alpha}^+$ be a solution of $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}'$. Then *any* other solution of $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}'$, is given by $\boldsymbol{\alpha}^+ - \mathcal{N}_{\mathbf{\Phi}} = \{\boldsymbol{\alpha}^+ - \boldsymbol{v} | \boldsymbol{v} \in \mathcal{N}_{\mathbf{\Phi}}\}$. Thus the ℓ_0-minimization problem of Equation (3.2.1) is equivalent to the problem of finding $\boldsymbol{v} N \mathcal{N}_{\mathbf{\Phi}}$ which minimizes $\|\boldsymbol{\alpha}_1 - \boldsymbol{v}\|_0$.

If one thinks of $\mathcal{N}_{\mathbf{\Phi}}$ as a linear code defined over the field of real numbers, and of $\boldsymbol{f}$ as the received word, the ℓ_0-minimization problem is equivalent to finding the error vector of minimum (Hamming) weight over all the codewords $\boldsymbol{v} \in \mathcal{N}_{\mathbf{\Phi}}$. Problems of this nature have been widely studied in the language of coding theory [183]; however, these codes are typically defined over finite fields, whereas here all arithmetic is done over the field of real numbers.

Inspired by this simple but fundamental connection between the noiseless compressed sensing and the theory of error-correcting codes several coding theory based constructions are proposed for solving the noiseless compressed sensing problem [5, 217, 152, 56]. In particular, based on the theory of algebraic coding/decoding, Akcakaya and Tarokh [5] construct *Vandermonde* sensing matrices that generalize Reed-Solomon codes using Vandermonde matrices. Let $M = 2k$, and consider the $M \times N$ sensing matrix

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ z_1 & z_2 & \cdots & z_N \\ z_1^2 & z_2^2 & \cdots & z_N^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{M-1} & z_2^{M-1} & \cdots & z_N^{M-1} \end{bmatrix},$$

where $z_1, \cdots, z_N$ are N distinct, non-zero real numbers.

Observe that since $M = 2k$, any $M \times 2k$ sub-matrix of $\mathbf{\Phi}$ is a $2k \times 2k$ Vandermonde matrix, and therefore has rank $2k$. In the language of algebraic coding theory, the null-space of $\mathbf{\Phi}$ is a maximum distance separable linear code of length N, dimension $N - M$ and minimum distance $M + 1$, and can be viewed as a generalization of the Reed-Solomon code over the field of the real numbers.

The Vandermonde reconstruction algorithm (the roots of which go back to 1795! – see [87, 253]) , uses the same idea as the algebraic algorithm for decoding Reed-Solomon codes [183, 231]. It uses the input data to construct an error-locator polynomial; the roots of this polynomial identify the signals appearing in the sparse superposition. The whole reconstruction process can be done efficiently using only $O(k^2)$ operations.

The Vandermonde construction provides optimality in the number of required measurements ($M = 2k$), and efficiency in sparse reconstruction ($O(k^2)$ operations). Nevertheless, because the correspondence between the coefficients of a polynomial and its roots is not well conditioned, it is very difficult to make the algorithm robust against the noise. This difficulty becomes more clear in Section 3.5 in which we will see that in the robust compressed sensing framework at least $\Omega\left(k \log \frac{N}{k}\right)$ measurements are

necessary.

3.3 Robust Compressed Sensing

3.3.1 ℓ_1-minimization and Restricted Isometry Property

The coding theory approach to compressed sensing exploits the similarities between the ℓ_0-minimization of Equation (3.2.1) and the decoding step in channel coding. The first step in this approach is to *design* a proper sensing matrix and a recovery algorithm specific to the designed sensing matrix. However, as we saw earlier, there are fundamental challenges against making this approach robust.

The sparse approximation problem has also been extensively investigated in the statistics and machine learning communities [142]. In statistics and machine learning, we are provided with M training examples. Each training example consists of N distinct features, and the goal is to find a *sparse* combination of the features that best represents the labels of all training examples.

More precisely, let $\mathbf{\Phi}$ be the $M \times N$ matrix whose rows indicate the M training examples, and at each row, the columns represent the values of N different features for that training example. Also let $\boldsymbol{f}$ be an M-dimensional vector in $\mathbb{R}^N$ that corresponds to the (real-valued) labels for the M training examples. The goal is to find a sparse vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ such that $\mathbf{\Phi}\boldsymbol{\alpha}^*$ closely approximates $\boldsymbol{f}$.

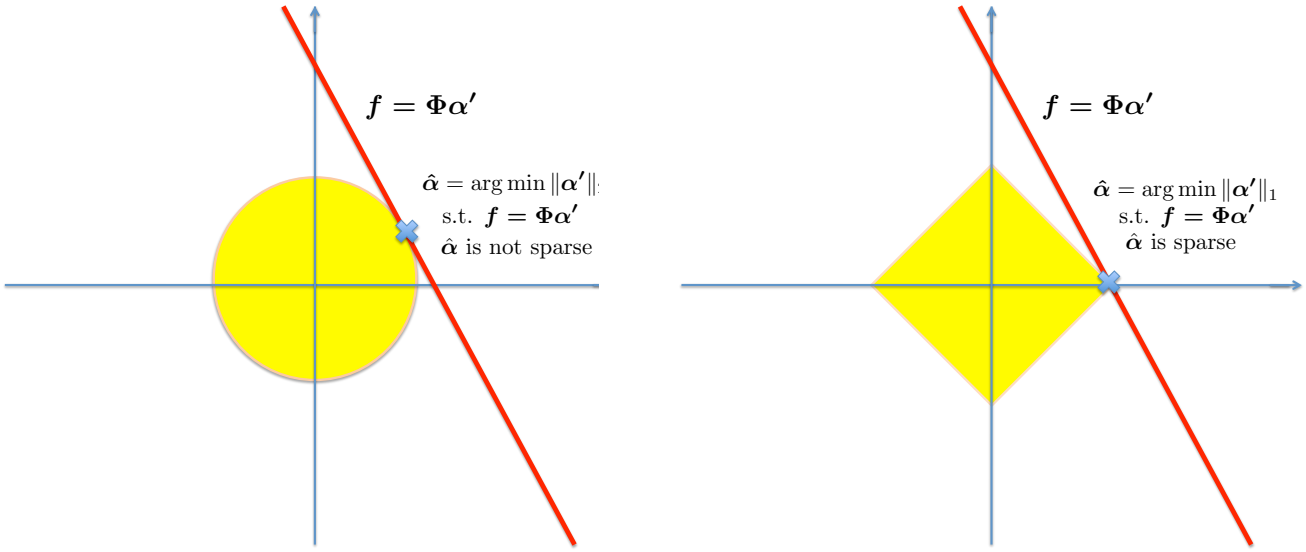
For simplicity, first consider the noiseless case in which there exists a k-sparse vector $\boldsymbol{\alpha}^*$ with $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^*$. In this case, the sparse feature selection problem reduces to the ℓ_0 minimization problem of Equation (3.2.1):

$$\begin{aligned} &\text{minimize } \|\boldsymbol{\alpha}'\|_0 \\ &\text{subject to } \boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}'. \end{aligned}$$

As the ℓ_0 pseudo-norm is non-convex, solving this optimization problem in general is NP-Hard [197]. The ℓ_1 minimization is an alternative and tractable approach, which suggests that instead of solving the non-convex ℓ_0 minimization problem, we alternatively solve the convex ℓ_1-minimization problem (also known as the Basis Pursuit (BP) problem [73]):

$$\begin{aligned} &\text{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\text{subject to } \boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}'. \end{aligned} \tag{3.3.1}$$

The ℓ_1 norm is a convex norm that has the most similarity to the non-convex pseudo-norm ℓ_0 [118]. The following 2-dimensional example provides insight into why ℓ_1-



(a) Visualization of the ℓ_2 minimization (3.3.2).

(b) Visualization of the ℓ_1 minimization (3.3.1).

Figure 3.1: An illustrative example indicating the advantage of ℓ_1 minimization over ℓ_2 minimization in finding a sparse point in the line $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$. The ℓ_2-minimization (3.3.2) finds a non-sparse point which is the intersection of the ℓ_2 ball, and the line $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$, whereas the ℓ_1-minimization (3.3.1) finds a sparse point which is the intersection of the ℓ_1 diamond, and the line $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$.

minimization is better able to select a sparse solution of $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$, than is ℓ_2-minimization. Here our feasible set $\mathcal{S} = \{\boldsymbol{\alpha}' : \boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'\}$ is a line in the plane, and the analytical solution to the ℓ_2-minimization problem

$$\begin{aligned} &\text{minimize } \|\boldsymbol{\alpha}'\|_2 \\ &\text{subject to } \boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}', \end{aligned} \tag{3.3.2}$$

(which is $\hat{\boldsymbol{\alpha}} = \boldsymbol{\Phi}^\dagger \boldsymbol{f}$) is a point in this line that has the closest Euclidean distance to the origin. This point can be found by blowing up a circle (the ℓ_2 ball) until it contacts the line $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$. However, as indicated in Figure 3.1(a) this closest point will live away from the coordinate axes, and hence will not be sparse. In contrast, the ℓ_1 ball in Figure 3.1(b) has points aligned with the coordinate axes. Therefore, when the ℓ_1 ball is blown up, it will first contact the line $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'$ at a point near the coordinate axes, that is it finds a sparse solution.

Thus far we have assumed that the vector $\boldsymbol{f}$ can be exactly represented by a sparse linear combination of the columns of $\boldsymbol{\Phi}$. However, in reality and in many statistics and machine learning applications $\boldsymbol{f}$ can only be well approximated by a sparse linear

combination of the columns of $\mathbf{\Phi}$. Over time, the ℓ_1-minimization approach has been generalized to address this more general problem.

Examples of these generalizations are, *Least Absolute Shrinkage* and *Selection Operator (LASSO)* [233, 106], which solves the ℓ_1-regularized regression problem

$$\text{minimize}_{\boldsymbol{\alpha}'} \|\boldsymbol{f} - \mathbf{\Phi}\boldsymbol{\alpha}'\|_2^2 + \lambda\|\boldsymbol{\alpha}'\|_1, \tag{3.3.3}$$

for a proper regularization parameter λ, the Basis Pursuit Denoising program [244, 243], which solves the second order cone program

$$\begin{aligned} &\text{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\text{subject to } \|\boldsymbol{f} - \mathbf{\Phi}\boldsymbol{\alpha}'\|_2 \leq \varepsilon_1, \end{aligned} \tag{3.3.4}$$

for some appropriately chosen ε_1, and the Dantzig Selector program [58], which solves the linear optimization

$$\begin{aligned} &\text{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\text{subject to } \|\mathbf{\Phi}^\top(\boldsymbol{f} - \mathbf{\Phi}\boldsymbol{\alpha}')\|_\infty \leq \varepsilon_2, \end{aligned} \tag{3.3.5}$$

for another suitably chosen parameter ε_2.

After the formalization of ℓ_1-minimization, several algorithms were proposed to efficiently solve the above optimization programs. Examples of such algorithms are the interior point methods [51, 52], Lasso modification to LARS [106, 171], homotopy methods [99], weighted least squares [163], and gradient-based methods [111, 257, 27, 242].

Experience with many applications has confirmed that ℓ_1-minimization algorithms and their extensions [59, 168] can robustly find sparse features that closely approximate the target vector $\boldsymbol{f}$. Therefore, ℓ_1 minimization appears to be a suitable algorithm for Task (T2) of compressed sensing. A natural question that now comes to mind is "what are proper sensing matrices for which the ℓ_1 minimization is *guaranteed* to recover a sparse vector, and among these sensing matrices, for what matrices is *uniqueness* in sparse recovery guaranteed?"

To answer this question, recall that Proposition 3.1 implies that if $\mathbf{\Phi}$ is a sensing matrix for which ℓ_1-minimization is able to recover any k-sparse vector $\boldsymbol{\alpha}^*$ from $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^*$, then no $2k$-sparse vector can be in the null-space of $\mathbf{\Phi}$. The following property is a stricter requirement for sensing matrices, and is introduced by Candès and Tao [56]:

Definition 3.5 (Restricted Isometry Property)**.** *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix. Then for every integer k,and $0 < \epsilon < 1$, $\mathbf{\Phi}$ satisfies the (k, ϵ) Restricted Isometry Property (abb. $\mathbf{\Phi}$ is (k, ϵ)-RIP), if for every k-sparse vector $\boldsymbol{\alpha}$ we have*

$$(1 - \epsilon)\|\boldsymbol{\alpha}\|_2 \leq \|\mathbf{\Phi}\boldsymbol{\alpha}\|_2 \leq (1 + \epsilon)\|\mathbf{\Phi}\boldsymbol{\alpha}\|_2.$$

Note that any sensing matrix which is $(2k, \epsilon)$-RIP for some positive ϵ, also satisfies the requirement of Proposition 3.1. To see this, suppose $\boldsymbol{\alpha}$ is a $2k$-sparse vector in the null-space of $\boldsymbol{\Phi}$. Then from the RIP with have

$$(1 - \epsilon)\|\boldsymbol{\alpha}\|_2 \leq \|\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2 = 0,$$

and therefore we must have $\|\boldsymbol{\alpha}\|_2 = 0$. In other words, the $(2k, \epsilon)$-RIP guarantees that no two k-sparse vectors can be mapped to the same low-dimension vector. That is, compressed sensing is information theoretically possible using any $(2k, \epsilon)$-RIP sensing matrix .

The following celebrated results of Candès, Romberg and Tao [54, 55], and Donoho et al. [95, 97] also guarantee that if a sensing matrix $\boldsymbol{\Phi}$ is $(2k, \epsilon)$-RIP for sufficiently small ϵ, then ℓ_1-minimization can exactly recover any k-sparse vector, in the noiseless compressed sensing framework, and moreover, the Basis Pursuit Denoising algorithm can stably approximate any k-sparse vector in the presence of data-domain and measurement-domain noise. This means that compressed sensing is also *computationally* possible using RIP sensing matrices and the generic ℓ_1-minimization algorithm.

Theorem 3.6 (Noiseless Compressed Sensing [57, 95])**.** *Let $\boldsymbol{\Phi}$ be a sensing matrix satisfying $(2k, 0.41)$-RIP. For every k-sparse vector $\boldsymbol{\alpha}^*$, let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$, and let $\hat{\boldsymbol{\alpha}}$ be the solution of the ℓ_1-minimization problem*

$$\begin{aligned} &\textit{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\textit{subject to } \boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}'. \end{aligned}$$

Then $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$.

Theorem 3.7 (Stable Compressed Sensing [55, 97])**.** *Let ϵ be a positive number smaller than 0.3, and let $\boldsymbol{\Phi}$ be a sensing matrix satisfying $(2k, \epsilon)$-RIP. Let $\boldsymbol{\alpha}^*$ be any arbitrary vector in $\mathbb{R}^N$, and let $\mathrm{H}_k(\boldsymbol{\alpha}^*)$ denote the best k-term approximation of $\boldsymbol{\alpha}^*$ defined by Equation (2.1.1). Finally let $\boldsymbol{e_M}$ be an arbitrary noise vector in $\mathbb{R}^M$, and let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$. Then the solution $\hat{\boldsymbol{\alpha}}$ of the Basis Pursuit Denoising problem*

$$\begin{aligned} &\textit{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\textit{subject to } \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2 \leq \|\boldsymbol{e_M}\|_2, \end{aligned}$$

satisfies the following ℓ_2/ℓ_1 sparse approximation guarantee:

$$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_2 \leq c_1 \frac{\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1}{\sqrt{k}} + c_2\|\boldsymbol{e_M}\|_2, \tag{3.3.6}$$

with $c_1 \doteq 2\frac{1+(\sqrt{2}-1)\epsilon}{1-(\sqrt{2}+1)\epsilon}$, and $c_2 \doteq \frac{2}{\sqrt{1+\epsilon}}c_1$.

Table 3.1: Summary of the ℓ_1-minimization problems used in RIP-based compressed sensing. In the deterministic noise model no assumption is made regarding the noise, whereas in the stochastic noise model the noise vector is assumed to be white Gaussian.

Optimization	Objective	Noise Model
Basis Pursuit (BP) [73]	minimize $\|\boldsymbol{\alpha}'\|_1$ s.t. $\boldsymbol{\Phi}\boldsymbol{\alpha}' = \boldsymbol{f}$	No noise
Basis Pursuit Denoising (BPDN) [244]	minimize $\|\boldsymbol{\alpha}'\|_1$ s.t. $\|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2 \leq \varepsilon$	Deterministic Noise
LASSO [106]	minimize $\|\boldsymbol{\Phi}\boldsymbol{\alpha}' - \boldsymbol{f}\|_2^2 + \lambda\|\boldsymbol{\alpha}'\|_1$	Stochastic Noise
Dantzig Selector (DS) [58]	minimize $\|\boldsymbol{\alpha}'\|_1$ s.t. $\|\boldsymbol{\Phi}^\top(\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}')\|_\infty \leq \varepsilon$	Stochastic Noise

Theorems 3.6 and 3.7 are fundamental as they provide *sufficient* conditions a sensing matrix should satisfy to provably guarantee that tractable ℓ_1-minimization can uniquely recover any sparse vector in the noiseless sensing regime [57, 49, 95] and can robustly find a sparse approximation to any vector in the presence of noise [54, 55, 97]. Table 3.1 summarizes the main ℓ_1-minimization problems that are widely used in compressed sensing applications.

Remark 3.8. *There is a fundamental difference between the coding theory approach and the statistics approach. In the coding theory approach, we first design a sensing matrix (e.g., the Reed-Solomon matrix) and then come up with a recovery algorithm specific to that particular sensing matrix (e.g., the algebraic decoding). In contrast, in the statistics approach we start with the generic ℓ_1-minimization algorithm, and then find specific properties (e.g. RIP) a sensing matrix should satisfy, so that the fidelity of the ℓ_1-minimization is guaranteed. Examples of RIP sensing matrices are introduced in Section 3.4.*

3.3.2 Greedy and Iterative Algorithms

So far we have seen that if a sensing matrix satisfies the $(2k, \epsilon)$-RIP with sufficiently small ϵ, then ℓ_1-minimization methods can stably recover any sparse vector. However, the best known running time for ℓ_1-minimization algorithms is $O(N^{1.5}M^2)$ [205] which is infeasible for many practical applications. These include important examples such as medical imaging or data streaming, where the number of pixels in the images or the traffic table sizes (N) are in the range 10^7 to 10^9. In these applications, more efficient and scalable algorithms are needed.

Greedy algorithms [237] provide an alternative to the ℓ_1-minimization approach. They aim to directly solve the original ℓ_0-minimization problem. Like the ℓ_1-minimization techniques, greedy algorithms were also developed before the formulation of RIP [85]. The greedy algorithms were initially developed as heuristic algorithms for approximately solving the non-convex optimization problem

$$\begin{aligned} &\text{minimize } \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2^2 \\ &\text{subject to } \boldsymbol{\alpha}' \in \Sigma_k, \end{aligned} \tag{3.3.7}$$

where $\boldsymbol{f}$ is an arbitrary vector in $\mathbb{R}^M$, $\boldsymbol{\Phi}$ is an $M \times N$ sensing matrix, and

$$\Sigma_k \doteq \{\boldsymbol{\alpha}' : \|\boldsymbol{\alpha}'\|_0 \leq k\},$$

is the union of all $\binom{N}{k}$ k-dimensional subspaces in $\mathbb{R}^N$. Since Σ_k is non-convex, the optimization problem of Equation (3.3.7) is not convex. Moreover, it is possible to show that (3.3.7) is in general NP-hard , that is, if there is no restriction on the sensing matrix $\boldsymbol{\Phi}$ [197].

Here we start with the Iterative Hard Thresholding (IHT) Algorithm [35, 114] which is the simplest algorithm for solving the optimization problem of Equation (3.3.7). IHT can be viewed as a special-case of the more generic *Gradient Projection* algorithm [105, 130] which is widely used in machine learning [37] and optimization [179].

Let $\mathcal{L} : \mathbb{R}^N \to \mathbb{R}$ be a differentiable loss function, and let Ω be a subset of $\mathbb{R}^N$ such that for every $\boldsymbol{v} \in \mathbb{R}^N$ it is possible to efficiently find the solution of the optimization problem

$$\min_{\boldsymbol{u} \in \Omega} \|\boldsymbol{v} - \boldsymbol{u}\|_2^2.$$

The Gradient projection algorithm is a simple and generic but powerful algorithm for solving the optimization problem

$$\begin{aligned} &\text{minimize } \mathcal{L}(\boldsymbol{\alpha}') \\ &\text{subject to } \boldsymbol{\alpha}' \in \Omega. \end{aligned} \tag{3.3.8}$$

It starts from an arbitrary point $\boldsymbol{\alpha}^0 \in \Omega$, and iteratively takes a step of length η along the gradient. Therefore, at each iteration we have one gradient update

$$\boldsymbol{\beta}^{\boldsymbol{t}} \doteq \boldsymbol{\alpha}^{t-1} - \eta \nabla \mathcal{L}(\boldsymbol{\alpha}^{t-1})$$

to reduce the value of the loss function, and then we need one projection step, i.e. projecting back the result to the set Ω by efficiently calculating the vector $\boldsymbol{\alpha}^t \doteq \min_{\boldsymbol{\alpha}' \in \Omega} \|\boldsymbol{\alpha}^t - \boldsymbol{\beta}^{\boldsymbol{t}}\|^2$, in order to ensure that $\boldsymbol{\alpha}^t$ is also in the feasible set Ω.

Here our loss function is the square loss $\mathcal{L}(\boldsymbol{\alpha}') = \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2^2$, with $\nabla\mathcal{L}(\boldsymbol{\alpha}') = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\alpha}' - \boldsymbol{f})$. Also the feasible set Ω is the set of all k-sparse vectors (Σ_k). As

Algorithm 1 Iterative Hard Thresholding Algorithm

Inputs: M-dimensional vector $\boldsymbol{f}$, $M \times N$ matrix $\boldsymbol{\Phi}$, number of iterations T, the sparsity level k, and update rate η.

Output: N-dimensional vector $\hat{\boldsymbol{\alpha}}$

Initialize $\boldsymbol{\alpha}^0 \doteq \mathbf{0}_N$.
for $t = 1, \ldots, T$ **do**
  Let $\boldsymbol{r^t} \doteq \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\alpha}^{t-1} - \boldsymbol{f})$. (Gradient step calculation)
  Let $\boldsymbol{\beta^t} \doteq \boldsymbol{\alpha}^{t-1} - \eta \boldsymbol{r^t}$. (Gradient update)
  Set $\boldsymbol{\alpha}^t \doteq \mathrm{H}_k(\boldsymbol{\beta^t})$. (Projection back to Σ_k)
end for
Output $\hat{\boldsymbol{\alpha}} \doteq \boldsymbol{\alpha}^T$.

a result, it follows from Equation (2.1.1) that the projection step is just a hard thresholding operation

$$\boldsymbol{\alpha}^t \doteq \min_{\boldsymbol{\alpha}' \in \Omega} \|\boldsymbol{\alpha}^t - \boldsymbol{\beta^t}\|^2 = \min_{\boldsymbol{\alpha}' \in \Sigma_k} \|\boldsymbol{\alpha}' - \boldsymbol{\beta^t}\|^2 = \mathrm{H}_k(\boldsymbol{\beta^t}),$$

and can be computed efficiently in time $O(N \log N)$ by sorting the elements of $\boldsymbol{\beta^t}$.

Algorithm 1 summarizes the Iterative Hard Thresholding (IHT) algorithm for solving the optimization problem of Equation (3.3.7). As mentioned earlier, this algorithm has been invented independently by several researchers in different communities as a heuristic algorithm for solving Equation (3.3.7) [85, 36, 147]. It turns out that the Restricted Isometry Property, which is a sufficient condition for the fidelity of the ℓ_1-minimization algorithms was also sufficient for proving the quick convergence of the IHT algorithm to the optimal solution of Equation (3.3.7) [36, 119].

The analysis that we discuss here is due to Garg and Khandekar [119], and provides a near linear-time algorithm that is guaranteed to find the solution of the program of Equation (3.3.7) as long as the sensing matrix is $(2k, 1/3)$-RIP.

In their analysis, they first show that the loss function $\mathcal{L}(\boldsymbol{\alpha}^t)$ always decreases by a constant factor at the end of every iteration. To prove this they show that the gradient descent step reduces the error significantly enough, while the RIP of $\boldsymbol{\Phi}$ implies that the sparsification step does not increase the error by too much. The following theorem summarizes the ℓ_2/ℓ_1 sparse approximation guarantee of the IHT Algorithm.

Theorem 3.9. *Let $\boldsymbol{\alpha^*}$ be an arbitrary vector in $\mathbb{R}^N$, and let $\boldsymbol{e_M}$ be the arbitrary noise vector in $\mathbb{R}^M$. Define*

$$\mathrm{SNR} \doteq \frac{\|\mathrm{H}_k(\boldsymbol{\alpha^*})\|_2}{\|\boldsymbol{e_M}\|_2 + \|\boldsymbol{\alpha^*} - \mathrm{H}_k(\boldsymbol{\alpha^*})\|_2}.$$

Let $\boldsymbol{\Phi}$ be an $M \times N$ matrix satisfying $(2k, \epsilon)$-RIP with $\epsilon \leq \frac{1}{3}$. Finally let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha^*} + \boldsymbol{e_M}$. Then there exists a constant $c_{\mathrm{IHT}} > 0$ that only depends on ϵ, such that

Algorithm 1 with $\eta = \frac{1}{1+\epsilon}$*, computes a* k*-sparse vector* $\hat{\boldsymbol{\alpha}}$ *satisfying*

$$\|\hat{\boldsymbol{\alpha}} - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_2 \leq c_{\mathrm{IHT}} \left(\sqrt{1+\epsilon} \left(\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_2 + \frac{\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1}{\sqrt{k}} \right) + \|\boldsymbol{e_M}\|_2 \right)$$

in at most $O(\log \mathrm{SNR})$ *iterations. Moreover, each iteration requires only* $O(\mathcal{V})$ *operations, where* $\mathcal{V}$ *bounds the cost of a matrix-vector multiplication by* $\boldsymbol{\Phi}$ *or* $\boldsymbol{\Phi}^\dagger$*.*

The Iterative Hard Thresholding Algorithm is a first order algorithm that solves the ℓ_0-minimization problem of Equation (3.3.7). By first order we mean at each iteration the algorithm only requires two matrix vector multiplications $\boldsymbol{\Phi}\boldsymbol{\alpha}^{t-1}$ and $\boldsymbol{\Phi}^\top(\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}^{t-1})$. Therefore, the algorithm can be implemented easily and efficiently. Nevertheless, the IHT algorithm usually performs significantly worse than ℓ_1 minimization (see Figure 3.2 or [185] for more discussions).

In order to overcome the sub-optimality of the IHT algorithm compared to convex, more complicated greedy algorithms were proposed over the years. The Compressive Sampling Matching Pursuit (CoSaMP) algorithm combines the idea of greedy gradient-projection with the idea of using convex optimization methods for sparse approximation, with the aim of achieving a high-performance, computationally efficient algorithm [199]. CoSaMP is an iterative algorithm that relies on two stages of sparse approximation: a first stage selects an enlarged candidate support set in a similar fashion to the IHT algorithm, while a second stage projects down this initial approximation to the desired sparsity level.

Similar to the IHT algorithm, at the begining of every iteration the gradient vector $\boldsymbol{r^t} = \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\alpha}^{t-1} - \boldsymbol{f})$ is calculated. In IHT then this gradient is directly added to the previous candidate $\boldsymbol{\alpha}^{t-1}$ in order to obtain the new candidate $\boldsymbol{\alpha}^t$. However, in contrast to IHT algorithm, CoSaMP is not first order. Here first the support of the significant entries of the gradient vector $\boldsymbol{r^t}$ is first added to the support of the previous candidate $\boldsymbol{\alpha}^{t-1}$, with the goal of obtaining a richer set Ω^t of the columns of the sensing matrix that best represents the vector $\boldsymbol{f}$. The new candidate is then a $|\Omega^t|$-sparse vector, supported on Ω^t whose non-zero entries that are obtained by solving the optimization problem

$$\text{minimize}_{\boldsymbol{\beta'_{\Omega^t}}} \|\boldsymbol{\Phi}_{\Omega^t}\boldsymbol{\beta'_{\Omega^t}} - \boldsymbol{f}\|_2^2,$$

(i.e., by projecting $\boldsymbol{f}$ into the span of $\boldsymbol{\Phi}_{\Omega^t}$.) Similar to IHT, the new candidate is finally further sparsified to ensure that it belongs to the feasible set Σ_k. The algorithm is formally detailed as Algorithm 2.

The following theorem is proved by Needell and Tropp [199], and provides an ℓ_2/ℓ_1 sparse approximation guarantee for CoSaMP.

Theorem 3.10. *Suppose that* $\boldsymbol{\Phi}$ *is an* $M \times N$ *sensing matrix which is* $(4k, 0.1)$*-RIP. Let* $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$ *be a vector of samples of an arbitrary signal, contaminated with*

Algorithm 2 CoSaMP Algorithm

Inputs: M-dimensional vector $\boldsymbol{f}$, $M \times N$ matrix $\boldsymbol{\Phi}$, number of iterations T, and the sparsity level k.

Output: N-dimensional vector $\hat{\boldsymbol{\alpha}}$

Initialize $\boldsymbol{\alpha}^0 \doteq \mathbf{0}_M$.
for $t = 1, \cdots, T$ **do**
  Let $\boldsymbol{r^t} \doteq \boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{\alpha}^{t-1} - \boldsymbol{f})$. (Gradient step calculation)
  Let $\hat{\Omega} \doteq \mathrm{Supp}(\mathrm{H}_{2k}(\boldsymbol{r^t}))$. (Identify large components)
  Let $\Omega^t \doteq \hat{\Omega} \cup \Omega^{t-1}$. (Enlarging the candidate set)
  Let $\boldsymbol{\beta^t}_{\Omega^t} = \boldsymbol{\Phi}^\dagger_{\Omega^t}\boldsymbol{f}$, and $\boldsymbol{\beta^t}_{\overline{\Omega^t}} = \mathbf{0}$. (Signal estimation by least squares)
  Let $\boldsymbol{\alpha}^t \doteq \mathrm{H}_k(\beta^t)$. (Projection back to Σ_k.)
end for
Output $\hat{\boldsymbol{\alpha}} \doteq \boldsymbol{\alpha}^T$.

arbitrary noise. Define

$$\mathrm{SNR} \doteq \frac{\|\mathrm{H}_k(\boldsymbol{\alpha^*})\|_2}{\|\boldsymbol{e_M}\|_2}. \tag{3.3.9}$$

Then the algorithm CoSaMP produces a k-sparse approximation $\hat{\boldsymbol{\alpha}}$ that satisfies

$$\|\boldsymbol{\alpha^*} - \hat{\boldsymbol{\alpha}}\|_2 \leq 20\left(\|\boldsymbol{\alpha^*} - \mathrm{H}_k(\boldsymbol{\alpha^*})\|_2 + \frac{\|\boldsymbol{\alpha^*} - \mathrm{H}_k(\boldsymbol{\alpha^*})\|_1}{\sqrt{k}} + \|\boldsymbol{e_M}\|_2\right),$$

in at most $O(\log \mathrm{SNR})$ iterations. Moreover, each iteration requires only $O(\mathcal{V})$ operations, where $\mathcal{V}$ bounds the cost of a matrix-vector multiplication by $\boldsymbol{\Phi}$ or $\boldsymbol{\Phi}^\dagger$.

Examples of other greedy algorithms include the classical Matching Pursuit (MP) [186], Orthogonal Matching Pursuit (OMP) [241, 126], stagewise OMP (StOMP) [96], regularized OMP (ROMP) [200], subspace pursuit [82], Iterative Soft Thresholding (IST) [107], and SAMP [89]. A comparison of a few key greedy algorithms for RIP-based Compressed sensing is provided in Table 3.2.

Greedy algorithms are favorable for compressed sensing due to their computational efficiency and also their simplicity of implementation. However, a major problem with most greedy algorithms is that the sparsity level k must be known to the user a priori. To solve this difficulty Donoho and Maleki have suggested using *tuned* greedy algorithms [185]. A tuned greedy recovery algorithm is a recovery algorithm that uses a *hard-coded* sparsity level k, which is determined as a function of the data dimension N, and the number of measurements M. The user does not need to know this hard-coded number. If the actual sparsity in $\boldsymbol{\alpha^*}$ is better than the assumed value, the algorithm still works, but if the sparsity is actually worse, the algorithm wont work even if tuned to assume that worse sparsity level.

The Tuned Two-Stage Thresholding (Tuned TST) [185] is then a generalization of the CoSaMP algorithm that does not require the sparsity level to be specified by the

Table 3.2: A comparison of a few selected greedy algorithms for RIP-based Compressed sensing and the Basis Pursuit Denoising algorithm. The algorithms are robust against noise. Here M and N denote the number of rows and columns of matrix $\boldsymbol{\Phi}$, $\mathcal{V}$ denotes the time taken in performing two matrix operations $\boldsymbol{\Phi}\boldsymbol{v}$ and $\boldsymbol{\Phi}^\top\boldsymbol{u}$. Also SNR is defined By Equation (3.3.9), and all bounds ignore the $O()$ constants.

Algorithm	Approach	Recovery Time	Recovery Condition	Recovery Guarantee
BPDN [55]	Convex Optimization	$N^{1.5}M^2$	$\boldsymbol{\Phi}$: $(2k,\epsilon)$-RIP with $\epsilon \leq \sqrt{2}-1$	ℓ_2/ℓ_1
IHT [119]	Greedy	$\mathcal{V}\log \text{SNR}$	$\boldsymbol{\Phi}$: $(2k,\epsilon)$-RIP with $\epsilon \leq \frac{1}{3}$	ℓ_2/ℓ_1
IHT [36]	Greedy	$\mathcal{V}\log \text{SNR}$	$\boldsymbol{\Phi}$: $(3k,\epsilon)$-RIP with $\epsilon \leq \frac{1}{\sqrt{32}}$	ℓ_2/ℓ_1
Subspace Pursuit [82]	Greedy	$\mathcal{V}\log \text{SNR}$	$\boldsymbol{\Phi}$: $(3k,\epsilon)$-RIP with $\epsilon \leq 0.06$	ℓ_2/ℓ_1
SAMP [89]	Greedy	$\mathcal{V}\log \text{SNR}$	$\boldsymbol{\Phi}$: $(3k,\epsilon)$-RIP with $\epsilon \leq 0.06$	ℓ_2/ℓ_1
CoSaMP [199]	Greedy	$\mathcal{V}\log \text{SNR}$	$\boldsymbol{\Phi}$: $(4k,\epsilon)$-RIP with $\epsilon \leq 0.1$	ℓ_2/ℓ_1

user. It has been reported that the Tuned TST algorithm empirically outperforms the original CoSaMP algorithm [185]. Therefore, in the rest of this thesis, unless specified explicitly, we use the tuned TST algorithm as the baseline greedy algorithm.

Theorems 3.9 and 3.10 provided theoretical ℓ_2/ℓ_1 sparse approximation bounds on the performances of IHT and CoSaMP algorithms, which are similar to the ℓ_2/ℓ_1 guarantee of ℓ_1-minimization methods (Theorem 3.7). However, a good asymptotic theoretical bound is not very useful if the runtime constants are very big. From a practical perspective, it is very important to quantify the exact reconstruction accuracy of the proposed greedy algorithms, and in particular to determine how well each greedy algorithm performs compared to the ℓ_1-minimization approach.

To see how each greedy algorithm compares to the ℓ_1-minimization empirically, the following Monte Carlo simulations is suggested by Donoho and Tanner [98] (see also [185]). *Fix the signal dimension $N = 800$, and sweep across k and M values. For each (k, M)-pair, repeat the following 100-times: (i) generate a k-sparse vector $\boldsymbol{\alpha}^*$ with random support, random sign, and unit norm, (ii) generate compressive measurements (no noise) using a RIP sampling matrix, and (iii) recover a k-sparse approximation $\hat{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha}^*$ using each greedy algorithm. Finally report the number of recoveries that obtain reconstruction error, $\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_2$ less than 10^{-2}.*

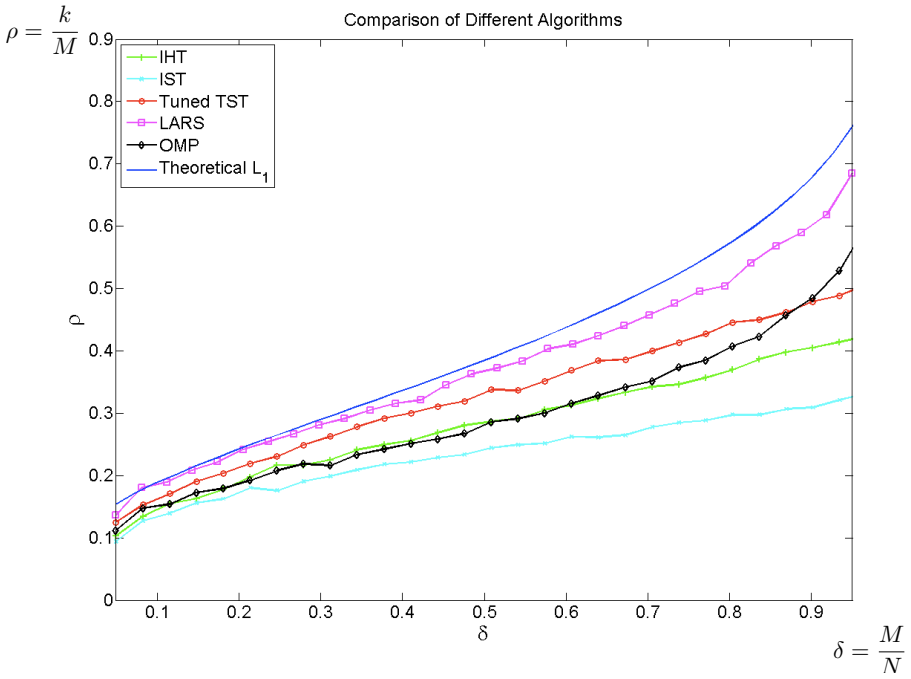


Figure 3.2: Phase Transitions of several baseline sparse approximation algorithms as provided in [185]. The upper curve indicates the theoretical phase transition of the ℓ_1-minimization (which is characterized by Donoho and Tanner [98]), and the lower curves show the empirical phase transitions of different algorithms.

Figure 3.2 compares the empirical performance of different sparse reconstruction algorithms. The curve corresponding to each algorithm shows the Phase Transition of that algorithm. Bellow the phase transition curve, the algorithm works well and above that curve it fails; the transition zone is narrow, and gets better defined at large problem sizes N (see [185] for further discussion).

As shown in Figure 3.2 the ℓ_1-minimization methods (e.g. LARS) always outperform the greedy algorithms in terms of the maximum sparsity level k that can be recovered using the algorithm. On the other hand, the greedy algorithms are typically significantly faster. In other words, there is always a trade-off between the performance and efficiency of the two approaches. If the sparsity value (k) is not too large, it is more beneficial to use fast greedy algorithms. On the other hand, if the sparsity value is higher than a threshold, then it is more performant to use the ℓ_1-minimization methods to increase the chance of successful recovery.

3.3.3 Bayesian Compressive Sensing

Bayesian compressed sensing provides a third approach for solving the robust compressed sensing problem [160, 4, 194]. In Bayesian compressed sensing it is often assumed *a priori* that the unknown sparse vector $\boldsymbol{\alpha}^*$ is sampled from a distribution that favors sparse vectors, and the noise vector $\boldsymbol{e_M}$ is sampled from some stochastic distribution (e.g. the multivariate Gaussian distribution). The main goal is then to estimate the hidden parameters of the underlying distributions through a *maximum a posteriori (MAP)* optimization in order to completely identify the posterior density function of $\boldsymbol{\alpha}^*$ [34].

More precisely, given the measurement vector $\boldsymbol{f} \in \mathbb{R}^M$, the goal is find an estimate $\hat{\boldsymbol{\alpha}}$ that maximizes the *posterior* probability $\log \Pr[\hat{\boldsymbol{\alpha}}|\boldsymbol{f}]$. It follows from Bayes rule that

$$\begin{aligned} \hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}'} \Pr[\boldsymbol{\alpha}'|\boldsymbol{f}] &= \arg\max_{\boldsymbol{\alpha}'} \left(\frac{\Pr[\boldsymbol{f}|\boldsymbol{\alpha}'] \Pr[\boldsymbol{\alpha}']}{\Pr[\boldsymbol{f}]} \right) \\ &= \arg\max_{\boldsymbol{\alpha}'} \left(\Pr[\boldsymbol{f}|\boldsymbol{\alpha}'] \Pr[\boldsymbol{\alpha}'] \right), \end{aligned} \tag{3.3.10}$$

where the last equality follows from the fact that $\boldsymbol{f}$ is already observed, and therefore $\Pr[\boldsymbol{f}]$ is a constant independent of $\boldsymbol{\alpha}'$.

The conditional distribution $\Pr[\boldsymbol{f}|\boldsymbol{\alpha}']$ models the noise process. The simplest noise model assumes that the measurement noise is white Gaussian of mean $\mathbf{0}_M$ and variance $\sigma_M^2 \mathrm{I}_{M\times M}$, therefore we have

$$\Pr[\boldsymbol{f}|\boldsymbol{\alpha}'] = \left(\frac{1}{\sqrt{2\pi\sigma_M}} \right)^N \exp\left\{ -\frac{\|\boldsymbol{\Phi}\boldsymbol{\alpha}' - f\|_2^2}{2\sigma_M^2} \right\}. \tag{3.3.11}$$

The prior distribution $\Pr[\boldsymbol{\alpha}']$ models the prior knowledge about the vector $\boldsymbol{\alpha}'$. In Bayesian compressed sensing, prior distributions that give more weight to sparse vectors are of more interest. A widely used sparseness prior is the Laplace density function with zero mean and variance $\sigma_d^2 \boldsymbol{I}_{N\times N}$ [32]:

$$\Pr[\boldsymbol{\alpha}'|] = \left(\frac{1}{\sqrt{2}\sigma_d} \right)^N \exp\left\{ -\sqrt{2}\frac{\|\boldsymbol{\alpha}'\|_1}{\sigma_d} \right\},$$

By choosing the Laplace distribution as the prior, and the white Gaussian distribution for modeling the noise we have

$$\hat{\boldsymbol{\alpha}} = \arg\max_{\boldsymbol{\alpha}'} \left(\Pr[\boldsymbol{f}|\boldsymbol{\alpha}'] \Pr[\boldsymbol{\alpha}'] \right) \tag{3.3.12}$$

$$\arg\max_{\boldsymbol{\alpha}'} \log \left(\Pr[\boldsymbol{f}|\boldsymbol{\alpha}'] \Pr[\boldsymbol{\alpha}'] \right)$$

$$= \arg\min_{\boldsymbol{\alpha}'} \frac{\|\boldsymbol{\Phi}\boldsymbol{\alpha}' - f\|_2^2}{2\sigma_M^2} + \frac{\sqrt{2}}{\sigma_d}\|\boldsymbol{\alpha}'\|_1.$$

The optimization problem of Equation (3.3.12) is a LASSO optimization problem (Equation 3.3.3) with parameter $\lambda = \frac{2\sqrt{2}\sigma_M^2}{\sigma_d}$. This provides another angle on why ℓ_1-minimization methods are attuned to sparse approximation. As Equation (3.3.12) indicates, the solution of the LASSO optimization with a suitable regularizing parameter λ is indeed the solution of a MAP optimization problem with Laplace prior and white Gaussian noise.

Since solving the map optimization problem $\text{maximize}_{\boldsymbol{\alpha}'} \log \Pr[\boldsymbol{\alpha}'|\boldsymbol{f}]$ with a Laplace prior is equivalent to solving a LASSO optimization, it also inherits the $O(M^2N^{1.5})$ computational complexity of solving the LASSO optimization [205]. To overcome this computational difficulty, Donoho, Maleki, and Montanari proposed the Approximate Message-Passing (AMP) algorithm for approximately solving the LASSO problem [91, 94]. Empirical observations suggest that the solution of the AMP Algorithm quickly converges to the optimal solution of LASSO [92, 93]. However, the convergence of the AMP has only been proved for sensing matrices obtained from Gaussian distributions [91, 25] which suffer from computation and storage limitations. (see Section 3.4.1 for further discussions). Proving the convergence of the AMP algorithm for efficient sensing matrices is an interesting and important open problem.

Another approach to overcome this computational difficulty is to use other sparsity-promoting prior distributions, so that the map optimization problem can be solved efficiently using the standard Bayesian optimization methods including the Markov Chain Monte Carlo method [127], and the Variational Inference method [248].

Ji, Xue, and Carin [160] have addressed this issue by introducing the Bayesian Compressive Sensing (BCS) algorithm, which uses the relevance vector machine (RVM) for sparse approximation [234]. Rather than imposing a Laplace prior on $\boldsymbol{\alpha}'$, in the RVM a hierarchical prior has been invoked [121]. The hierarchical prior has similar sparsity-promoting properties to the Laplace prior but allows convenient conjugate-prior properties which are useful for conveniently implementing a Markov Chain Monte Carlo (MCMC) or a variational Bayesian optimization algorithm [198].

Similarly Carmi et. al. [62] proposed the *Approximate Bayesian Compressive Sensing (ABCS)* algorithm which uses the semi-Gaussian prior distribution. A distinguishing feature of the semi-Gaussian distribution is greater concentration in the vicinity of the origin, which promotes sparsity more aggressively than ℓ_1-minimization.

Even though simulation results indicate that the BCS and ABCS algorithms have good performance and in some cases even outperform the ℓ_1-minimization method, they both have computational complexity $O(NM^2)$ which is still inefficient for many compressed sensing applications with $N \approx 10^9$, and $M \approx 10^6$.

One approach which can further reduce the computational complexity of the sparse recovery phase in Bayesian compressed sensing is to use the efficient Belief Propagation Algorithms [182]. Belief Propagation is a fast message-passing algorithm that has been extensively used for efficiently decoding the *Low-Density Parity Check (LDPC)*

Table 3.3: Summary of the main algorithms in the Bayesian compressed sensing framework.

Algorithm	Signal Prior	Sensing Matrix	Recovery Time	Convergence Guarantee
LASSO [233]	Laplace	RIP	$O(N^{1.5}M^2)$	Yes
BCS [160]	Hierarchical prior	RIP	$O(NM^2)$	Yes
ABCS [62]	semi-Gaussian	RIP	$O(NM^2)$	Yes
CS-BP [23]	Mixture of two Gaussians	LDPC	$O(N \log^2 N)$	No
SuPrEM [4]	Gaussian-scale mixtures	LDPC	$O(N)$	No

codes [227]. The fundamental connection between compressed sensing and the theory of error-correcting codes suggests the idea of adopting BP to solve the compressed sensing problem.

Recently, there have been several papers on using Belief Propagation algorithms for sparse recovery. In [222, 23], the authors introduced the belief propagation approach to compressive sensing, and applied it to the recovery of random signals, modeled by a two-state mixture of Gaussians (with more weight on the narrower Gaussian to promote the sparsity). Their proposed *Compressive Sensing Belief Propagation (CS-BP)* algorithm has $O(N \log^2 N)$ computational complexity and is significantly faster than BCS and ABSC. In a more recent paper, Akcakaya, Park, and Tarokh [4] used belief propagation on signals modeled as Gaussian-scale mixtures. Their proposed *Sum Product with Expectation Maximization (SuPrEM)* algorithm has $O(N)$ running time, and is shown to have an excellent empirical performance.

Nevertheless, the major problem with the belief propagation approach is that neither CS-BP nor SuPrEM is guaranteed to converge. In contrast to the other sparse reconstruction algorithms, much less is known about the theoretical performance of the CS-BP and SuPrEM algorithms. Analyzing the convergence rates of the Belief Propagation algorithms is an interesting and important open problem in Bayesian compressed sensing. Table 3.3 summarizes a comparison of different Bayesian compressive sensing algorithms.

3.4 Construction of RIP Sensing Matrices

3.4.1 Random RIP Constructions

Thus far we have seen that if a matrix is $(2k, \epsilon)$-RIP for sufficiently small ϵ, then the ℓ_1-minimization, greedy, and Bayesian algorithms can stably approximate any sparse vector $\boldsymbol{\alpha}^*$ from the low-dimensional vector $\boldsymbol{f}$. Therefore, the problem of stable compressed sensing is now reduced to the problem of finding RIP sensing matrices. Here we provide examples of sensing matrices satisfying the RIP.

Definition 3.11 (Gaussian sensing matrix)**.** *A Gaussian sensing matrix , is an $M \times N$ matrix whose entries are sampled independently and identically from a $\mathcal{N}\left(0, \frac{1}{M}\right)$ distribution.*

Definition 3.12 (Rademacher sensing matrix)**.** *A Rademacher sensing matrix is an $M \times N$ matrix such that each entry of it is assigned to be $\pm\frac{1}{\sqrt{M}}$, each with probability $\frac{1}{2}$.*

The following theorem by Baraniuk et al. [21] shows that Gaussian and Rademacher processes generate $M \times N$ matrices that satisfy the RIP with high probability:

Theorem 3.13. *Suppose that M, N, and $0 < \epsilon < 1$ are given. Let Φ be an $M \times N$ Rademacher (or Gaussian) sensing matrix . Then there exist absolute constants $c_1, c_2 > 0$, such that Φ is (k, ϵ)-RIP for any $k \leq \frac{c_1 \epsilon^2 M}{\log(N/M)}$ with probability at least $1 - 2\exp\{-c_2 \epsilon^2 M\}$.*

Theorem 3.13 is of particular interest as it concludes that stable compressed sensing is possible using *random Gaussian or Rademacher sensing matrices* combined with ℓ_1-minimization. Moreover, only $O\left(k \log N/k\right)$ measurements are required in order to be able to successfully recover any k-sparse vector. As we will see in Section 3.5 at least $\Omega\left(k \log N/k\right)$ measurements are always required to have stable compressed sensing, and therefore Gaussian and Rademacher matrices are optimal with respect to the number of required measurements.

However, there is no efficient algorithm to verify whether a given random Gaussian or Rademacher matrix satisfies the RIP or not. Moreover, since Gaussian and Rademacher matrices do not have any structure, memory is an issue since $\Omega(MN)$ bits are required to store the whole matrix. Moreover, due to the lack of structure of these matrices, any matrix-vector multiplication requires $\Omega(MN)$ operations which makes the encoding less efficient.

To overcome the difficulties of using Gaussian or Rademacher sensing matrices, alternative RIP matrices have been introduced.

Definition 3.14 (Subsampled unitary matrices)**.** *Let U be any $N \times N$ unitary matrix. Choose a subset Ω of cardinality $|\Omega| = M$ uniformly at random from the set*

$\{1, \cdots, N\}$. *Let* $\mathbf{\Phi}$ *be the* $M \times N$ *matrix obtained by sampling* M *rows of* U *corresponding to the indices in* Ω *and renormalizing the resulting columns so that they have unit* ℓ_2*-norms. Then* $\mathbf{\Phi}$ *is an* $M \times N$ *subsampled unitary matrix.*

The following theorem indicates that as long as M is sufficiently large, any subsampled unitary matrix is RIP with overwhelming probability.

Theorem 3.15 ([218])**.** *For each integer* N, k*, and for any* $t > 1$ *and any* $\epsilon \in (0, 1)$*, let*

$$M \geq c_3 N \|\mathbf{\Phi}\|_\infty^2 kt \log^4 N,$$

then the subsampled matrix $\mathbf{\Phi}$ *is* (k, ϵ)*-RIP with probability exceeding* $1 - 10 \exp\left\{-c_4 \epsilon^2 t\right\}$*, where* c_3 *and* c_4 *are absolute constants that do not depend on* $M, N, or k$*.*

The following corollary follows from Theorem 3.15 by taking $t = O\left(\frac{\log N}{\epsilon^2}\right)$:

Corollary 3.16. *Let* U *be any* $N \times N$ *unitary matrix whose entries have magnitude* $\frac{1}{\sqrt{N}}$*. For each integer* k *and any* $\epsilon \in (0, 1)$*, let*

$$M \geq \frac{c_3' k \log^5 N}{\epsilon^2},$$

where c_3' *is an absolute constant. Then the subsampled matrix* $\mathbf{\Phi}$ *is* (k, ϵ)*-RIP with probability exceeding* $1 - \frac{1}{N}$*.*

Partial Hadamard matrices are one example of subsampled unitary matrices satisfying the conditions of Corollary 3.16.

Definition 3.17 (Hadamard Matrix)**.** *The Hadamard transform* H_n *is a* $2^n \times 2^n$ *matrix that can be defined recursively in the following way: We define the* 1×1 *Hadamard matrix* H_0 *by the identity* $H_0 = 1$*, and then define* H_n *for* $n > 0$ *by:*

$$H_n \doteq \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}. \tag{3.4.1}$$

Without loss of generality suppose $N = 2^n$. The Hadamard matrix H_n can be viewed as a symmetric unitary matrix with $\|H_n\|_\infty = \frac{1}{\sqrt{N}}$. Therefore, it follows from Corollary 3.16 that as long as $M \geq \frac{c_3' k \log^5 N}{\epsilon^2}$, an $M \times N$ subsampled Hadamard matrix is (k, ϵ)-RIP.

Note that in contrast to random Gaussian and Rademacher matrices, only $O(M \log N)$ random bits are required to store a partial Hadamard matrix. Moreover, the matrix-vector multiplication $H_n \boldsymbol{v}$ can be performed efficiently in time $O(N \log N)$ using the fast Walsh-Hadamard transform [112].

Therefore, partial Hadamard matrices are superior to random Gaussian or Rademacher matrices in terms of the required storage and computational time of calculating

matrix-vector multiplications. The main drawback of partial Hadamard matrices is sub-optimality in the number of required measurements for satisfying the RIP. With Gaussian and Rademacher matrices only $O\left(\frac{k \log N/k}{\epsilon^2}\right)$ measurements are required to guarantee the RIP with overwhelming probability, whereas $O\left(\frac{k \log^5 N}{\epsilon^2}\right)$ measurements are required for partial Hadamard matrices to have the same RIP with the same probability.

3.4.2 Deterministic RIP Constructions

Thus far, we have seen Gaussian, Rademacher, and the partial Hadamard matrices as examples of *random* sensing matrices satisfying the RIP. As mentioned earlier, Gaussian and Rademacher matrices suffer from storage and computational issues, and partial Hadamard matrices suffer from sub-optimality in the number of measurements. In addition, there is no efficient algorithm to verify whether a random matrix is RIP or not. Therefore, it is desirable to construct *deterministic* matrices satisfying the RIP.

Most explicit constructions of RIP matrices are based on bounding the mutual coherence between the columns of the sensing matrix .

Definition 3.18 (Mutual coherence)**.** *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix with normalized columns. The mutual coherence between the columns of $\mathbf{\Phi}$ is then defined as*

$$\mu \doteq \max_{i \neq j} |\langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle|.$$

The following lemma connects the RIP property of any sensing matrix $\mathbf{\Phi}$ with normalized columns to the mutual coherence of $\mathbf{\Phi}$.

Lemma 3.19. *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix with normalized columns and with mutual coherence μ. Then $\mathbf{\Phi}$ is (k, ϵ)-RIP with $\epsilon = (k-1)\mu$.*

Proof. Let $\boldsymbol{\alpha}$ be any k-sparse vector. We have

$$\begin{aligned} |\|\mathbf{\Phi}\boldsymbol{\alpha}\|_2^2 - \|\boldsymbol{\alpha}\|_2^2| &= |\sum_{j \neq i} \alpha_i \alpha_j \langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle| \leq \sum_{j \neq i} |\alpha_i||\alpha_j||\langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle| \qquad (3.4.2) \\ &\leq \mu \sum_{j \neq i} |\alpha_i||\alpha_j| = \mu \left(\left(\sum_{i=1}^{N} |\alpha_i| \right)^2 - \|\boldsymbol{\alpha}\|_2^2 \right) \leq \mu(k-1)\|\boldsymbol{\alpha}\|_2^2, \end{aligned}$$

The last inequality follows from the Cauchy-Schwarz inequality and the fact that $\boldsymbol{\alpha}$ is k-sparse:

$$\left(\sum_{i=1}^{N} |\alpha_i| \right)^2 = \|\boldsymbol{\alpha}\|_1^2 \leq k\|\boldsymbol{\alpha}\|_2^2.$$

□

Examples of sensing matrices with small mutual coherence have been constructed by Calderbank et. al. [140], Applebaum et. al. [11], Bajwa et. al. [15], Kashin [166], Alon et. al. [7], DeVore [88], Iwen [154], and Nelson and Temlyakov [201]. All these constructions have mutual coherence $\mu \leq \frac{\log N}{\sqrt{M}\log M}$, and therefore satisfy RIP for $k = O\left(\epsilon \frac{\sqrt{M}\log M}{\log N}\right)$.

On the other hand, the Welch bound [251], demonstrates that the mutual coherence of a matrix with normalized columns cannot be too small. More precisely, there is a universal lower bound

$$\mu \geq \sqrt{\frac{\log N}{M \log {}^{M}/_{\log N}}} \geq \frac{1}{\sqrt{M}},$$

as long as $M \leq \frac{N}{2}$. Therefore, by estimating RIP parameters in terms of the coherence parameter we cannot construct $M \times N$ (k, ϵ)-RIP matrices with $k \geq \sqrt{M}$, and $\epsilon < 1$.

Most explicit constructions of RIP matrices are based on the mutual coherence and suffer from the $k = O(\sqrt{N})$ barrier, but a recent result by Bourgain et. al [39] uses the methods of additive combinatorics to do slightly better.

Theorem 3.20 ([39]). *There is an effective constant $\varepsilon_0 > 0$ and an explicit number M_0 such that for any positive integers $M \geq M_0$, and $M \leq N \leq M^{1+\varepsilon_0}$, there is an explicit $M \times N$ matrix with is (k, ϵ)-RIP, with $k = M^{0.5+\varepsilon_0}$, and $\epsilon = M^{-\varepsilon_0}$.*

Table 3.4 compares various properties of different RIP matrices. The result of Bourgain et. al. breaks the bottleneck $M = \Omega(k^2)$ of the low-coherence matrices. However, it is still significantly sub-optimal compared to the $M = O(k \log {}^{N}/_{k})$ measurements of random sensing matrices. The problem of finding deterministic RIP matrices with close to optimal ($M \approx k \log \frac{N}{k}$) number of measurement is an important open problem in the theory of compressed sensing.

A negative result by Chandar, proves that if a sensing matrix has only $0, 1$ entries, or if it is too sparse, then that matrix cannot satisfy the RIP [70]. This negative result and several unsuccessful attempts in designing optimal deterministic or structured RIP matrices suggests that maybe the RIP condition is too restrictive for compressed sensing. This is indeed the main subject of this thesis, in which we show that it is possible to design deterministic matrices that do not satisfy the RIP, but still provide almost every feature obtainable from random RIP matrices, as well as extra advantages which are not obtainable (or are in some cases even impossible) via the random RIP matrices. Before discussing these matrices, we first investigate the information theoretic limitations of the compressed sensing.

Table 3.4: Properties of different sensing matrices satisfying the (k, ϵ) Restricted Isometry Property. All bounds ignore the $O(.)$ constants.

Matrix	Number of measurements	Memory (random bits)	Matrix-vector multiplication	Random vs. Deterministic
Gaussian (Rademacher) [21]	$k \log\left(\frac{N}{k}\right)$	MN	MN	Random
Partial Hadamard (Fourier)[218]	$k \log^5 N$	$M \log N$	$N \log N$	Random
Incoherent Tight-Frames [11, 15]	k^2	-	$N \log N$	Deterministic
Randomness Extractors [39]	$k^{\frac{2}{2+\varepsilon_0}}$	-	NM	Deterministic

3.5 Compressed Sensing Lower Bounds

In Section 3.3, we saw that if a sensing matrix is RIP, then it is possible to obtain ℓ_2/ℓ_1 sparse approximation guarantees using ℓ_1-minimization and greedy algorithms. We have also seen examples of RIP matrices with $M = O\left(k \log\left(\frac{N}{k}\right)\right)$ measurements. Now, a natural question that comes to mind is "*What kinds of improvements are possible over this existing RIP-based approach?*" To answer this question, we focus on the following specific questions.

- (Q1): Are $M = \Omega\left(k \log\left(\frac{N}{k}\right)\right)$ measurements necessary for stable compressed sensing?
- (Q2): Is ℓ_2/ℓ_1 the tightest sparse approximation guarantee in stable compressed sensing? Is it possible to derive other ℓ_p/ℓ_q bounds? How do they compare to the ℓ_2/ℓ_1 bound?
- (Q3): Is RIP necessary for stable compressed sensing? Is it possible to find RIP-less sensing matrices with similar (or even better) performances?

Here we answer the first two questions. Answering the third question is the subject of the rest of this thesis. To answer the first two questions, we first define the best k-term approximation which is a fundamental problem in approximation theory [75], and is highly related to the sparse approximation problem (Definition 3.3) for stable compressed sensing.

Definition 3.21 (Best $\boldsymbol{k}$-term Approximation)**.** *Let p and q be positive integers. Let $\boldsymbol{\Phi}$ be an $M \times N$ sensing matrix , and let $\mathcal{A}_{\boldsymbol{\Phi}}$ be a reconstruction algorithm associated*

with $\boldsymbol{\Phi}$. Then $\mathcal{A}_{\boldsymbol{\Phi}}$ provides ℓ_p/ℓ_q best k-term approximation guarantee if and only if there exists an absolute constant C, such that for every vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$, given $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$, $\mathcal{A}_{\boldsymbol{\Phi}}$ can successfully recover a k-sparse vector $\hat{\boldsymbol{\alpha}}$ with

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_p \leq \frac{C}{k^{1-\frac{1}{p}}} \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_q.$$

The ℓ_p/ℓ_q best k-term approximation problem is a special case of the ℓ_p/ℓ_q sparse approximation problem defined in Definition 3.3. However, in contrast to the general sparse approximation problem, there is no measurement noise in the best k-term approximation setting. The best k-term approximation analyses provide powerful tools for proving lower-bounds on the number of required measurements for stable compressed sensing. These analyses originated from work in functional analysis and approximation theory by Kashin [166], and were later improved and generalized by Gluskin [128, 129], Cohen et. al. [75], and Ba et. al [13].

The following theorem is due to Cohen et. al [75] and implies that ℓ_1/ℓ_1 best k-term approximation is achievable from any algorithm that provides ℓ_2/ℓ_1 guarantees.

Theorem 3.22 ([75])**.** *Let $\boldsymbol{\Phi}$ be an $M \times N$ sensing matrix and let $\mathcal{A}_{\boldsymbol{\Phi}}$ be a reconstruction algorithm such that $(\boldsymbol{\Phi}, \mathcal{A}_{\boldsymbol{\Phi}})$-provides an ℓ_2/ℓ_1 sparse approximation guarantee. Then $(\boldsymbol{\Phi}, \mathcal{A}_{\boldsymbol{\Phi}})$ also provides an ℓ_1/ℓ_1 guarantee.*

Proof. Let $\boldsymbol{\alpha}^*$ be an arbitrary vector in $\mathbb{R}^N$, let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$. Also let $\hat{\boldsymbol{\alpha}} = \mathcal{A}_{\boldsymbol{\Phi}}(\boldsymbol{f})$, and $\boldsymbol{\Delta} = \boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}$. The ℓ_2/ℓ_1 guarantee of $(\boldsymbol{\Phi}, \mathcal{A}_{\boldsymbol{\Phi}})$ implies that $\hat{\boldsymbol{\alpha}}$ is k-sparse, and that

$$\|\boldsymbol{\Delta}\|_2 \leq \frac{C}{\sqrt{k}} \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1. \tag{3.5.1}$$

Let $\mathcal{S} = \mathrm{Supp}(\mathrm{H}_k(\boldsymbol{\alpha}^*)) \cap \mathrm{Supp}(\hat{\boldsymbol{\alpha}})$. Since both $\hat{\boldsymbol{\alpha}}$ and $\mathrm{H}_k(\boldsymbol{\alpha}^*)$ are k-sparse, we have $|\mathcal{S}| \leq 2k$.

Therefore, it follows from Holder's inequality that

$$\|\boldsymbol{\Delta}_{\boldsymbol{\mathcal{S}}}\|_1 \leq \sqrt{2k} \|\boldsymbol{\Delta}_{\boldsymbol{\mathcal{S}}}\|_2 \leq \sqrt{2k} \|\boldsymbol{\Delta}\|_2. \tag{3.5.2}$$

Combining (3.5.1) and (3.5.2), yields

$$\|\boldsymbol{\Delta}_{\boldsymbol{\mathcal{S}}}\|_1 \leq \frac{\sqrt{2k}C_1}{\sqrt{k}} \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 = \sqrt{2}C \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1. \tag{3.5.3}$$

On the other hand, since $\mathcal{S}$ includes the top k coordinates of $\boldsymbol{\alpha}^*$

$$\|\boldsymbol{\Delta}_{\overline{\boldsymbol{\mathcal{S}}}}\|_1 = \|\boldsymbol{\alpha}^*_{\overline{\boldsymbol{\mathcal{S}}}}\|_1 \leq \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1.$$

Therefore

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1 = \|\boldsymbol{\Delta}\|_1 = \|\boldsymbol{\Delta}_{\boldsymbol{\mathcal{S}}}\|_1 + \|\boldsymbol{\Delta}_{\overline{\boldsymbol{\mathcal{S}}}}\|_1 \leq \left(1 + \sqrt{2}C_1\right) \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1.$$

□

The following theorem of Ba et. al. [13] provides lower bounds on the number of required measurements for obtaining ℓ_1/ℓ_1 and ℓ_2/ℓ_1 guarantees.

Theorem 3.23 ([13]). *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix and let $\mathcal{A}_{\mathbf{\Phi}}$ be a reconstruction algorithm such that $(\mathbf{\Phi}, \mathcal{A}_{\mathbf{\Phi}})$-provides an ℓ_1/ℓ_1 sparse approximation guarantee, then $M = \Omega\left(k \log\left(\frac{N}{k}\right)\right)$.*

By combining Theorem 3.22 and Theorem 3.23, we obtain similar lower bounds on the number of required measurements for ℓ_2/ℓ_1 guarantees.

Corollary 3.24. *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix and let $\mathcal{A}_{\mathbf{\Phi}}$ be a reconstruction algorithm such that $(\mathbf{\Phi}, \mathcal{A}_{\mathbf{\Phi}})$-provides an ℓ_2/ℓ_1 guarantee, then $M = \Omega\left(k \log\left(\frac{N}{k}\right)\right)$.*

Proof. Theorem 3.22 proves that if $(\mathbf{\Phi}, \mathcal{A}_{\mathbf{\Phi}})$ provides an ℓ_2/ℓ_1 guarantee, then it also provides an ℓ_1/ℓ_1 guarantee. Therefore, it follows immediately from Theorem 3.23 that $M = \Omega\left(k \log\left(\frac{N}{k}\right)\right)$ measurements are necessary. □

Remark 3.25. *In Section 3.2 we saw that the $2k \times N$ Vandermonde construction of Akcakaya and Tarokh [5] can efficiently recover any vector which is exactly k-sparse. However, since $2k = o\left(k \log\left(\frac{N}{k}\right)\right)$, there is no hope of finding a robust sparse reconstruction algorithm with ℓ_1/ℓ_1 or ℓ_2/ℓ_1 guarantee for this construction. This gives another explanation why the proposed algebraic decoding is not robust against noise.*

Remark 3.26. *Theorem 3.7 with $\boldsymbol{e_M} = \mathrm{O}_{\mathrm{M}}$, implies that if $\mathbf{\Phi}$ is a $(2k, \sqrt{2}-1)$-RIP, then $(\mathbf{\Phi}, \textit{Basis Pursuit})$ provides ℓ_2/ℓ_1 best k-term approximation guarantee. Therefore by invoking Corollary 3.24, any $(2k, \sqrt{2}-1)$-RIP matrix requires $M = \Omega\left(k \log\left(\frac{N}{k}\right)\right)$ measurements. In other words, one cannot expect to find RIP matrices with smaller number of measurements $M = o\left(k \log\left(\frac{N}{k}\right)\right)$. On the other hand, Theorem 3.13 proves that as long as $M = O\left(k \log\left(\frac{N}{k}\right)\right)$, a Gaussian (or Rademacher) sensing matrix is $(2k, \sqrt{2}-1)$-RIP, and by which the ℓ_2/ℓ_1 guarantees are obtainable. This shows that the lower bound of Corollary 3.24 is tight.*

Remark 3.27. *In Chapter 8 we will introduce examples of RIP-less sensing matrices with optimal $M = O\left(k \log\left(\frac{N}{k}\right)\right)$ measurements that provide ℓ_1/ℓ_1 sparse approximation guarantees. Our proposed matrices are sparse and have deterministic constructions, and do not suffer from the storage and computational limitations of RIP Gaussian or Rademacher matrices.*

Thus far we have explored the connections between the ℓ_2/ℓ_1 and ℓ_1/ℓ_1 guarantees, and seen that $\Omega\left(k \log\left(\frac{N}{k}\right)\right)$ measurements are necessary and sufficient to achieve these guarantees. Next, we will see whether it is possible to extend these results to achievability of ℓ_2/ℓ_2 approximation guarantees or not.

An argument similar to the one used in Theorem 3.22 is used by Cohen et. al [75], to prove that the ℓ_2/ℓ_2 guarantee also implies ℓ_1/ℓ_1 guarantees (with the same constant C); however ℓ_2/ℓ_2 does not necessarily imply the ℓ_2/ℓ_1 guarantee.

Nevertheless, Cohen et. al [75] also showed that the ℓ_2/ℓ_2 approximation is *impossible* in general in the compressed sensing framework unless $M = \Omega(N)$.

Theorem 3.28 ([75])**.** *Let $\mathbf{\Phi}$ be an $M \times N$ sensing matrix and let $\mathcal{A}_{\mathbf{\Phi}}$ be a reconstruction algorithm such that $(\mathbf{\Phi}, \mathcal{A}_{\mathbf{\Phi}})$-provides an ℓ_2/ℓ_2 sparse approximation guarantee, then $M = \Omega(N)$.*

Theorem 3.28 directly implies that ℓ_1/ℓ_1 and ℓ_2/ℓ_1 guarantees cannot imply ℓ_2/ℓ_2. Otherwise, one could use a RIP matrix with $M = O(k \log\left(\frac{N}{k}\right)) = o(N)$ measurements and the ℓ_1-minimization algorithm and get the ℓ_2/ℓ_2 guarantee from the provided ℓ_2/ℓ_1 guarantee of Theorem 3.7, or from the ℓ_1/ℓ_1 guarantee of Theorem 3.22.

Finally we emphasize that the result of Theorem 3.28 is only existential. It only shows that if $\mathbf{\Phi}$ is an $M \times N$ sensing matrix with $M = o(N)$, then for each decoding algorithm $\mathcal{A}_{\mathbf{\Phi}}$, there exists one particular vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ (that may depend on the choice of $\mathcal{A}_{\mathbf{\Phi}}$) such that $\|\boldsymbol{\alpha}^* - \mathcal{A}_{\mathbf{\Phi}}(\mathbf{\Phi}\boldsymbol{\alpha}^*)\|_2$ is significantly large. In contrast, in Chapter 12 we will provide examples of deterministic sensing matrices $\mathbf{\Phi}$ with $M = O(k \log N)$ and efficient reconstruction algorithms $\mathcal{A}_{\mathbf{\Phi}}$, such that $(\mathbf{\Phi}, \mathcal{A}_{\mathbf{\Phi}})$ provides ℓ_2/ℓ_2 guarantees for *most* (in contrast to *all*) vectors.

Chapter 4

Applications

4.1 Compressive Imaging

4.1.1 Image Compression

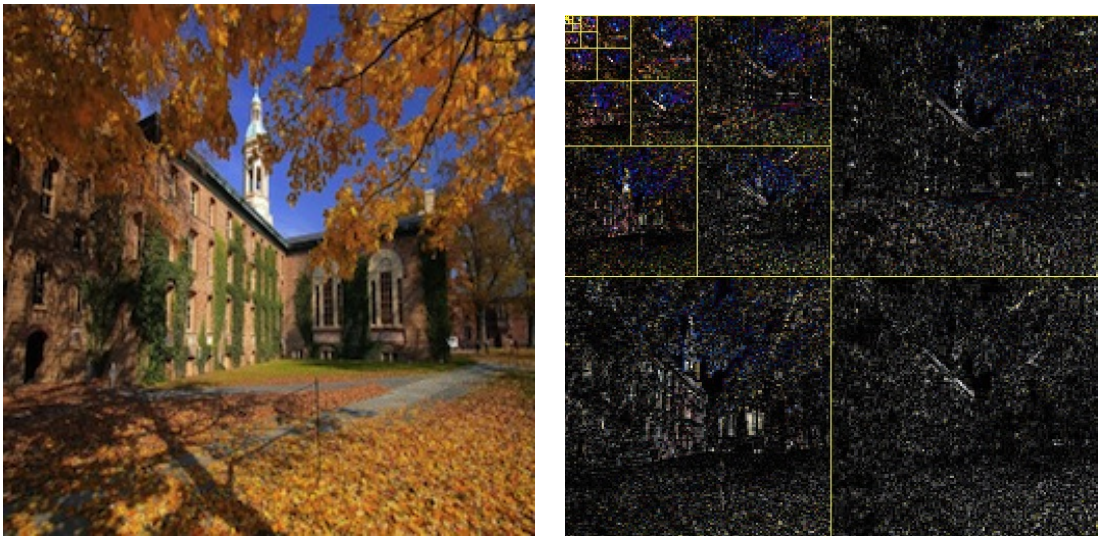
A fundamental assumption in digital image processing is that natural images are *piecewise smooth* in the pixel basis. That is, there are very few edges in the image, and therefore, the differences between the values of adjacent pixels are usually zero or almost zero. The *wavelet transform* can be used to map images from the pixel domain to the wavelet domain in which they have sparse (or approximately sparse) representations [84].

For example, Figure 4.1(b) shows the representation of a natural image in the pixel domain, and Figure 4.1(b) shows the representation of the same image in the wavelet domain. As you can see from the figure, there are very few significant (light) coefficients in the wavelet representation of this image, whereas most wavelet coefficients are almost zero (black).

The wavelet sparsity of images is used in the image compression application [131]. In order to compress a $\sqrt{N} \times \sqrt{N}$ image, the camera first treats the image as a high N-dimensional vector and calculates its wavelet representation. Finally it stores the positions and values of the $k \ll N$ significant wavelet coefficients and throws away the remaining information. The decoding can therefore be done efficiently by forming the (sparsified) wavelet vector, and applying the inverse wavelet transform to restore the image.

Since images are approximately sparse in the wavelet basis, the sparsified image still provides a good approximation of the original image. For instance, Figure 4.2 shows the resulting images when only the largest 1%, 3% or 10% of the wavelet coefficients of an 256×256 image are used.

Even though this image compression protocol provides precise sparse approximation



(a) A 256 image in the pixel domain

(b) The representation of the same image in the wavelet domain. Here the intensities correspond to wavelet coefficient magnitudes.

Figure 4.1: Example of sparse approximations in a wavelet basis.

to most natural images, it is inefficient and costly. It is inefficient because we ultimately throw away most of the calculated wavelet coefficients. We calculate N wavelet coefficients but then we just keep the k largest coefficients and discard the rest of them. It is also costly as the camera requires N sensors, whereas only $k \ll N$ coefficients are ultimately stored.

Here, compressed sensing can be used as a new data acquisition framework, to overcome the inefficiencies of the classical image compression approach [216]. In contrast to the classical approach, which involves sensing a high-resolution signal and then compressing it by throwing away part of the sensed data, compressed sensing attempts to develop methods to sense signals directly into compressed form [53].

To see how compressed sensing works for image compression, let $\boldsymbol{\Psi}$ denote the $N \times N$ wavelet transform matrix. Also let $\boldsymbol{l}$ denote an N dimensional image. Then the vector $\boldsymbol{\alpha}^* = \boldsymbol{\Psi l}$ is the wavelet representation of image and is approximately k-sparse (with $k \ll N$).

Let $\boldsymbol{\Phi}$ be an $M \times N$ sensing matrix (with $M \approx k \ll N$), and let $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$. The *measurement* matrix $\boldsymbol{A}$ is obtained by combining the sensing matrix $\boldsymbol{\Phi}$ and the wavelet transform matrix $\boldsymbol{\Psi}$. In standard compressed sensing, the compressed vector $\boldsymbol{f}$ is obtained by finding the k largest wavelet coefficients of the image (i.e. $\boldsymbol{f} = \mathrm{H}_k(\boldsymbol{\Psi l})$). This approach requires $O(N^2)$ operations. In contrast, in compressed sensing we use the $M \times N$ matrix $\boldsymbol{A}$ to compress the image. That is

$$\boldsymbol{f} = \boldsymbol{Al} = \boldsymbol{\Phi\Psi l} = \boldsymbol{\Phi\Psi\Psi}^{-1}\boldsymbol{\alpha}^* = \boldsymbol{\Phi\alpha}^*. \tag{4.1.1}$$

(a) Image obtained by using the 1% largest coefficients.

(b) mage obtained by using the 3% largest coefficients.

(c) mage obtained by using the 10% largest coefficients.n

Figure 4.2: Resulting images when only the largest $1\%, 3\%$, or 10% largest db2 coefficients are used.

Therefore, if the measurement matrix $\boldsymbol{A}$ is precomputed, the encoding $\boldsymbol{f} = \boldsymbol{A}\boldsymbol{l}$ can be performed efficiently using $O(MN)$ operations.

A digital camera usually has limited computational resources. However, the image recovery is usually done once the camera is connected to a computer that has more powerful computation resources. Therefore, given the measurement vector $\boldsymbol{f}$, and the prior information that $\boldsymbol{\alpha}^*$ is (approximately) k-sparse, sparse approximation algorithms can be used to find a sparse approximation $\hat{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha}^*$. Subsequently, the inverse wavelet transform can find an approximation $\hat{\boldsymbol{l}}$ for the image $\boldsymbol{l}$ in the pixel domain. Since the wavelet transform is unitary

$$\|\boldsymbol{l} - \hat{\boldsymbol{l}}\|_2 = \|\boldsymbol{\Psi}^{-1}(\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}})\|_2 = \|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_2.$$

The sparse approximation error in the pixel domain is the same as the error in the wavelet domain.

4.1.2 Single-Pixel Camera

Compressed sensing addresses the computational issue central to classic image compression. However, it still first measures the whole image in the pixel domain using N sensors and then performs the compression. To overcome this final challenge, new hardware, called the *single-pixel camera* was developed at Rice University [232, 102].

The camera uses a small array of chip mirrors, each mirror corresponding to a pixel of the image. These mirrors can be independently rotated to either reflect the light towards a lens (on state) or away from it (off state). The mirrors can turn on and off very quickly, and thus one pixel can be partially reflected as determined by the ratio between the on and off time. A photodiode is then put in the cannon of the lens to convert the accumulated light intensity into a quantitative measurement. By

repeating this process M times, we can sense a number M of linear measurements $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$.

In summary, in the single pixel camera, the linear measurements are performed efficiently by nature, and only one sensor (photodiode) is required for the whole procedure. This is in particular advantageous if the arrays of N high-resolution sensors (as used in classics digital cameras) are too expensive or even not available, for instance in infrared imaging.

4.1.3 Biomedical Imaging

Another important application for compressed sensing is in reducing the sampling rate in magnetic resonance imaging (MRI) [132]. Traditional MRI scanners sequentially sample Fourier coefficients of the human brain's image. Unfortunately, this traditional MRI approach is very time costly, as the speed of data collection is limited by physical and physiological constraints However, most MRI images are sparse in the Fourier domain. As a result compressed sensing can be used to significantly decrease the number measurements without reducing the accuracy of the MRI image [181].

4.2 Data Streaming

In data streaming applications, devices with limited memory process massive streams of data [196, 8]. For instance, in a network with 2^{32} addressees, a monitoring table counts the number of packets going from each source address to each destination address. The monitoring table is therefore a $2^{32} \times 2^{32}$ table, and the entry at row i and column j of the table shows the number of packets going from the source address i to the destination address j.

Storing the whole table requires 2^{64} memory and is not practically feasible. However, this monitoring table is often approximately sparse. There are a few source/destination pairs with a significant number of packets, whereas most pairs communicate no or very few packets. If we are interested in the most traffic-heavy pairs, our aim is to obtain (approximately) the heaviest elements of the table.

This problem can be viewed as a compressed sensing application if we represent the monitoring table as a high-dimensional vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ (e.g. $N = 2^{64}$.). The goal is then to design an efficient $M \times N$ matrix (with $k \approx M \ll N$), such that for every table $\boldsymbol{\alpha}^*$, the M-dimensional sketch $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$ captures most information regarding the significant entries of $\boldsymbol{\alpha}^*$. That is, we aim is to obtain a k-sparse approximation to $\boldsymbol{\alpha}^*$ from the sketch vector $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$.

It is easy to see that the encoding can be done efficiently in real-time thanks to the linearity of the update operation. Let $\boldsymbol{\alpha}^{*t}$ denote the monitoring table at time t, for which we only have access to the sketch $\boldsymbol{f}^t = \boldsymbol{\Phi}\boldsymbol{\alpha}^{*t}$. Also, $\boldsymbol{\Delta}^{\boldsymbol{t}} \in \mathbb{R}^N$ be the vector

that contains the number of packets that have arrived in the interval $[t, t+1)$. The sketch of the new table $\boldsymbol{\alpha}^{*t+1} = \boldsymbol{\alpha}^{*t} + \boldsymbol{\Delta}^{\boldsymbol{t}}$ is $\boldsymbol{\Phi}\boldsymbol{\alpha}^{*t+1} = \boldsymbol{\Phi}(\boldsymbol{\alpha}^{*t} + \boldsymbol{\Delta}^{\boldsymbol{t}}) = \boldsymbol{\Phi}\boldsymbol{\alpha}^{*t} + \boldsymbol{\Phi}\boldsymbol{\Delta}^{\boldsymbol{t}}$. Thus we can directly update the sketch by calculating $\boldsymbol{\Phi}\boldsymbol{\Delta}^{\boldsymbol{t}}$. At the end of the day, given the final sketch $\boldsymbol{\Phi}\boldsymbol{\alpha}^{*}$, one can use efficient sparse approximation algorithms to find a k-sparse monitoring table $\hat{\boldsymbol{\alpha}}$ that closely approximates the true monitoring table $\boldsymbol{\alpha}^{*}$.

4.3 Digital Communications

The problem of configuring wireless networks to enable network communication in the presence of inference is one of the major challenges facing communication research [245]. One important case is managing inference in peer to peer networks and in an uplink where multiple sensors look to communicate with an access point.

the interference-mitigation for downlink communications in which a single transmitter (e.g. a cellular base station) communicates simultaneously with multiple (N) receivers [2, 10].

The key idea connecting compressed sensing to wireless communication is that at each time only a small ($k \ll N$)) number of receivers are active. The sender then maintains an $M \times N$ sensing matrix $\boldsymbol{\Phi}$, such that the i^{th} columns of $\boldsymbol{\Phi}$ is associated with the i^{th} user.

At each transmission time, the transmitted signal is constructed as the sum of individual signals, each intended to a different receiver. That is, the transmitted signal is a superposition of at most k columns of the matrix. With this strategy, each receiver can also invoke sparse reconstruction algorithms and decode its own information.

4.4 Group Testing

Group testing is the problem of devising tests to efficiently identify members of a group with a certain property [101, 74]. The group testing applications range from the blood testing problem which was used in World War II for identifying men who carry a certain disease [100], to the problem of testing the impacts of new drugs on human genes [83, 164].

In group testing the aim is to avoid individual testing of all candidates by repeatedly pooling up a subgroup of multiple individuals and testing this subgroup instead [74]. It is often assumed that there are only a few people sharing some specified property, and the goal is to design an $M \times N$ test matrix describing the M subgroup tests, so that it is possible to efficiently recover the sparse special members of the group from the tests [124] (see also [122, 175]).

4.5 Machine Learning

Compressed sensing has been recently used for solving the face classification problem, in which the goal is to predict whose face a new (test) face is given a large collection of labeled training faces [256].

A key assumption in face classification is that all faces of most human-beings lie in a low dimensional subspace, and much fewer degrees of freedom (compared to the total number of pixels), govern the structure of all possible faces [246, 254]. As a result, given a sufficiently rich training set of faces for a particular person, any new (test) face can be represented by a linear combination of her training faces, and therefore by a *sparse* linear combination of all training faces of all people in the training repository.

Therefore, sparse approximation can be used to identify the person whose training faces form the largest contribution in approximating the test face [255]. Using compressed sensing and sparse approximation has provided significant improvements over the existing state-of-the-art methods that use support vector machines [250], or principal component analysis [170].

A similar approach has been used in speaker identification and speech recognition applications [260, 165, 225]. Here using compressed sensing and sparse approximation has given classification accuracy improvements on the standard datasets after more than 20 years [220].

Compressed sensing has also been used for efficiently solving the multi-label classification problems with large label space size N [151]. It has been shown both theoretically and experimentally that under the reasonable assumption that each example has at most $k \ll N$ associated labels, the compressed sensing approach is more efficient and robust compared to other multi-label classification approaches.

Dictionary learning is another machine learning application in which using compressed sensing is advantageous. Dictionary learning is a powerful tool in machine learning with applications in source separation in music [236], object recognition in computer vision [235], and image denoising in digital image processing [184]. In dictionary learning, the $M \times N$ matrix (also called the overcomplete dictionary) is *learned* from the available training examples. It has been shown that one approach for solving the dictionary learning problem is to solve a series of non-convex optimization problems iteratively, where each non-convex optimization consists of solving a sparse coding problem followed by a convex optimization problem [103, 104]. Devising efficient sparse approximation algorithms for sparse coding can facilitate the task of learning overcomplete dictionaries.

4.6 Quantum Computing

A major obstacle to engineer quantum devices such as quantum computers had been lack of an effective scheme for noise characterization in many component systems. The number of parameters required to represent the state of a quantum system grows exponentially with the number of its components in contrast to a classical system. As a result the number of measurements needed for full characterization of the noisy dynamics of a quantum system becomes astronomically large.

In [226], Shabani et. al. have developed a CS theory to estimate the effect of noise on a quantum system dynamics. They show a linear relation, $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$, between the parameters of a noisy quantum dynamics, $\boldsymbol{\alpha}^*$, and measurement outcomes, $\boldsymbol{f}$. The sparsity property is assumed for the signal $\boldsymbol{\alpha}^*$ holds under some common physical conditions.

Chapter 5

Thesis Outline

5.1 Thesis Statement

In this thesis, we shall see that our proposed deterministic sensing framework is significantly more powerful from many practical applications, compared to the conventional "random projections followed by ℓ_1-minimization" framework used in compressed sensing.

5.2 Main Contributions

In this section we outline the main contributions of the thesis. We also provide references to the papers that cover the main materials of each chapter. The central objective of this thesis is to provide efficient deterministic sensing frameworks that avoid the performance, storage and computational limitations of the random sensing framework. Towards this end, we will first introduce efficient and *generic* recovery algorithms that do not rely on non-verifiable properties, such as the restricted isometry property. We focus on two important complementary tasks of *sparse approximation* and *model selection*. In sparse approximation the goal is to find a sparse vector sufficiently close to the sparse target vector in some metric, whereas in the model selection the objective is to recover the support of the sparse target vector in the presence of noise.

The first half of this thesis focuses on the sparse approximation problem. In Chapter 6.1, we will show that the sparse approximation problem can always be reformulated as a zero-sum game. Then, in Chapter 7, we will introduce the Bregman divergence as a generalization of the Euclidean distance. We will also propose and analyze an efficient algorithm, called the GAME algorithm, that approximately solves the sparse approximation problem by simulating a repeated game between the two players of the zero-sum game. The algorithm is generic and does not assume any

non-verifiable assumption regarding the sensing matrix $\mathbf{\Phi}$ [158].

In Chapter 8, we will introduce the expander-based compressed sensing as our first deterministic sensing framework. We will also propose an efficient recovery algorithm capable of recovering any k sparse vector in at most $2k$ simple iterations in the noiseless settings [159]. In Chapter 9 we focus on bounded ℓ_1-norm noise model, and show that in that model, if the an expander-based sensing matrix is used, then it is possible to significantly tighten the generic bounds of the GAME algorithm. Empirical results support the fidelity of the GAME algorithm [156].

In Chapter 10, we consider the problem of expander-based compressed sensing in the presence of Poisson noise. Poisson noise is an important noise model in applications such as low-light imaging and data streaming. We will show that in the expander-based compressed sensing framework, a Bayesian reconstruction algorithm can *provably* recover a close approximation to any sparse target vector. This means that in the Poisson noise model, expander-based compressed sensing not only provides storage and computational advantages over the dense random sensing, but it also gains sparse approximation guarantees that are not directly obtainable in the dense sensing framework [211, 212, 157].

The second half of the thesis investigates the model-selection problem. In Chapter 11, we will introduce two fundamental measures of coherence between the columns of a sensing matrix. We will further show that as long as the sensing matrix satisfies a verifiable *coherence property*, a simple and efficient *One-Step Thresholding* algorithm is capable of finding the support of most sparse vectors [17, 14, 16].

Reed-Muller sensing is the second proposed deterministic sensing framework. Chapter 12 introduces the Delsarte-Goethals frames, as a family of deterministic sensing with optimal measures of coherence. The Delsarte-Goethals frames are generated from the Delsarte-Goethals codes, which are a properly chosen subset of the second order Reed-Muller codes. We will show how the coherence-optimality of the DG frames relates to model-selection optimality of the OST algorithm in the Reed-Muller Sensing framework. To demonstrate the efficiency of the OST algorithm, we also show that in our C++ implementation, it only takes about one minute for the OST algorithm to recover sparse 2^{32} dimensional vectors from 2^{12} DG frame-based measurements [44, 45, 46, 47, 174].

Finally in Chapter 13 we describe the model-based compressed sensing problem. In model-based compressed sensing, some extra prior knowledge (e.g., positivity, block sparsity, etc) is also available about the target sparse vector. In this setting, we will introduce an iterative algorithm, called the NIHT algorithm, which can incorporate the available extra prior knowledge and approximately solve the model-based sparse approximation problem. The NIHT algorithm can be considered as a generalization of the OST algorithm, as the OST algorithm is equivalent to the NIHT algorithm run for only one iteration. We will provide several different experiments to show that NIHT can empirically outperform ℓ_1-minimization methods in different compressed

sensing settings [69]. Figure 5.1 summarizes the main contributions of this thesis.

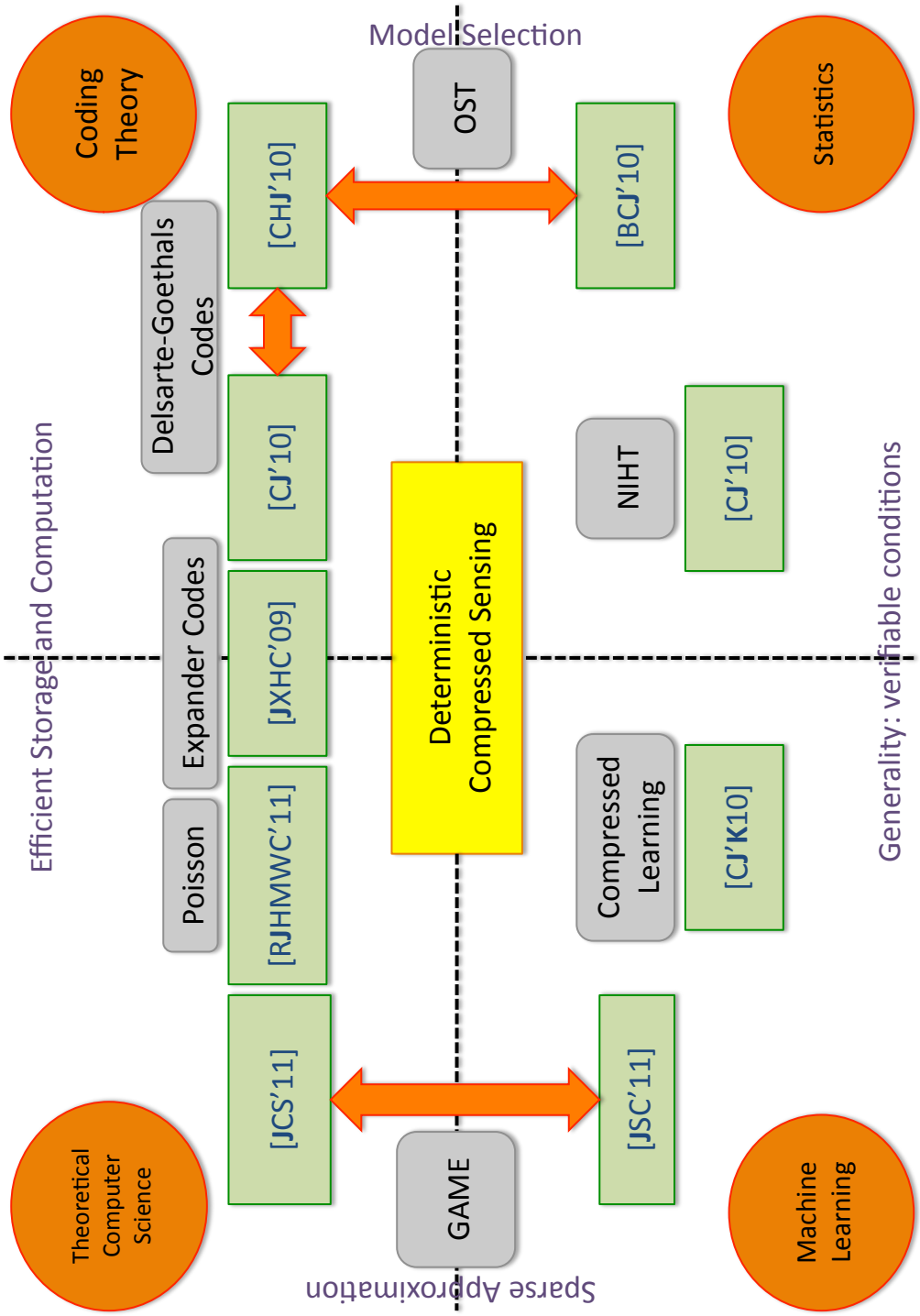


Figure 5.1: The deterministic sensing map: summary of the main contributions of this thesis.

Part II

Sparse Approximation for Compressed Sensing

Chapter 6

Game Theory Meets Compressed Sensing

6.1 Game Theoretic Reformulation of Sparse Approximation

Sparse approximation is a fundamental problem in compressed sensing (see Section 3.1), as well as in many other signal processing and machine learning applications including variable selection in regression [233, 247, 193], graphical model selection [213, 191], and sparse principal components analysis [209, 162]. In sparse approximation, one is provided with a dimensionality reducing measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M \times N}$ ($M < N$), and a low dimensional vector $\boldsymbol{f} \in \mathbb{R}^M$. The goal is to find a sparse vector $\hat{\boldsymbol{\alpha}}$ such that $\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$ is sufficiently close to $\boldsymbol{f}$.

In this chapter, we consider the sparse approximation problem in the ℓ_q norm, where q is a positive integer. Let k be a positive integer, and let τ be an arbitrary positive number. Define

$$\Delta(\tau) \doteq \{\boldsymbol{\alpha} \in \mathbb{R}^N : \|\boldsymbol{\alpha}\|_1 \leq \tau\}, \tag{6.1.1}$$

as the set of all vectors inside the hyper-diamond of radius τ, and define

$$\Delta(k, \tau) \doteq \{\boldsymbol{\alpha} \in \mathbb{R}^N : \|\boldsymbol{\alpha}\|_0 \leq k \text{ and } \|\boldsymbol{\alpha}\|_1 \leq \tau\}, \tag{6.1.2}$$

as the set of all k-sparse vectors in $\Delta(\tau)$.

We shall prove that for every dimension reducing matrix $\boldsymbol{\Phi}$, and every measurement vector $\boldsymbol{f}$, one can a find vector $\hat{\boldsymbol{\alpha}} \in \Delta(k, \tau)$ with

$$\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - f\|_q \leq \min_{\boldsymbol{\alpha} \in \Delta(k,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q + \tilde{O}\left(\frac{1}{\sqrt{k}}\right). \tag{6.1.3}$$

This sparse approximation framework works for any matrix $\boldsymbol{\Phi}$, and not just for matrices satisfying the RIP. Later on, In Chapter 9 we will see that if $\boldsymbol{\Phi}$ is a deterministic

matrix constructed from expander graphs, then the provided bounds of this chapter can be further tightened to an ℓ_1/ℓ_1 data-domain sparse approximation guarantee.

Note that since $\Delta(k,\tau)$ is not convex, the optimization problem

$$\text{minimize}_{\boldsymbol{\alpha}\in\Delta(k,\tau)}\|\boldsymbol{\Phi\alpha}-\boldsymbol{f}\|_q \tag{6.1.4}$$

is not a convex optimization. This optimization problem is actually NP-hard in general [197], and cannot be solved precisely. However, in this chapter we will show that there exist efficient algorithms that can provide an *approximate* solution.

We reformulate this sparse-approximation problem as a zero-sum game, and then propose a computationally efficient algorithm to obtain a sparse approximation for the optimal game solution. The proposed algorithms employ a primal-dual scheme, and require $\tilde{O}(k)$ iterations in order to find a k-sparse vector with $O\left(k^{-0.5}\right)$ additive approximation error.

We start by defining a zero-sum game and then proving that the sparse approximation problem of Equation (6.1.4) can be reformulated as a zero-sum game.

Definition 6.1 (Zero-sum games [207])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two closed sets. Let $\mathcal{L}:\mathcal{A}\times\mathcal{B}\to\mathbb{R}$ be a function. The value of a zero sum game, with domains $\mathcal{A}$ and $\mathcal{B}$ with respect to a function $\mathcal{L}$ is defined as*

$$\min_{\boldsymbol{a}\in\mathcal{A}}\max_{\boldsymbol{b}\in\mathcal{B}}\mathcal{L}(\boldsymbol{a},\boldsymbol{b}). \tag{6.1.5}$$

The function $\mathcal{L}$ is usually called the *loss function*. A zero-sum game can be viewed as a game between two players Mindy and Max in the following way. First, Mindy finds a vector $\boldsymbol{a}$, and then Max finds a vector $\boldsymbol{b}$. The loss that Mindy suffers[1] is $\mathcal{L}(\boldsymbol{a},\boldsymbol{b})$. The game-value of a zero-sum game is then the loss that Mindy suffers if both Mindy and Max play with their optimal strategies.

Von Neumann's well-known Minimax Theorem [206, 116] states that if both $\mathcal{A}$ and $\mathcal{B}$ are convex compact sets, and if the loss function $\mathcal{L}(\boldsymbol{a},\boldsymbol{b})$ is convex with respect to $\boldsymbol{a}$, and concave with respect to $\boldsymbol{b}$, then the game-value is independent of the ordering of the game players.

Theorem 6.2 (Von Neumann's Minimax Theorem [206])**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be closed convex sets, and let $\mathcal{L}:\mathcal{A}\times\mathcal{B}\to\mathbb{R}$ be a function which is convex with respect to its first argument, and concave with respect to its second argument. Then*

$$\inf_{\boldsymbol{a}\in\mathcal{A}}\sup_{\boldsymbol{b}\in\mathcal{B}}\mathcal{L}(\boldsymbol{a},\boldsymbol{b})=\sup_{\boldsymbol{b}\in\mathcal{B}}\inf_{\boldsymbol{a}\in\mathcal{A}}\mathcal{L}(\boldsymbol{a},\boldsymbol{b}).$$

For the history of the Minimax Theorem see [173]. The Minimax Theorem tells us that for a large class of functions $\mathcal{L}$, the values of the min-max game in which Mindy

[1] which is equal to the gain that Max obtains as the game is zero-sum.

goes first is identical to the value of the max-min game in which Max starts the game. The proof of the Minimax Theorem is provided in [117].

Having defined a zero-sum game, and the Von Neumann Minimax Yheorem, we next show how the sparse approximation problem of Equation (6.1.4) can be reformulated as a zero-sum game. Let $p \doteq \frac{q}{q-1}$, and define

$$\Xi_p \doteq \{\mathbf{P} \in \mathbb{R}^M : \|\mathbf{P}\|_p \leq 1\}. \tag{6.1.6}$$

Define the loss function $\mathcal{L} : \Xi_p \times \Delta(\tau) \to \mathbb{R}$ as

$$\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) \doteq \langle \mathbf{P}, (\boldsymbol{\Phi}\boldsymbol{\alpha} - f)\rangle. \tag{6.1.7}$$

Observe that the loss-function is bilinear. Now it follows from Holder inequality (Theorem 2.1) that for every $\boldsymbol{\alpha}$ in $\Delta(k, \tau)$, and for every $\mathbf{P}$ in Ξ_p

$$\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) = \langle \mathbf{P}, (\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})\rangle \leq \|\mathbf{P}\|_p \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q \leq \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q. \tag{6.1.8}$$

The inequality of Equation (6.1.8) becomes equality for

$$P_i^* = \frac{(\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})_i^{q/p}}{\left(\sum_{i=1}^M (\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})_i^q\right)^{1/p}}.$$

Therefore

$$\max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) = \max_{\mathbf{P}\in\Xi_p} \langle \mathbf{P}, (\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})\rangle = \langle \mathbf{P}^*, (\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})\rangle = \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q. \tag{6.1.9}$$

Equation (6.1.9) is true for every $\boldsymbol{\alpha} \in \Delta(\tau)$. As a result, by taking the minimum over $\Delta(k, \tau)$ we get

$$\min_{\boldsymbol{\alpha}\in\Delta(k,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q = \min_{\boldsymbol{\alpha}\in\Delta(k,\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}).$$

Similarly by taking the minimum over $\Delta(\tau)$ we get

$$\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q = \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{6.1.10}$$

Solving the sparse approximation problem of Equation (6.1.4) is therefore equivalent to finding the optimal strategies of the game

$$\min_{\boldsymbol{\alpha}\in\Delta(k,\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{6.1.11}$$

In the next section we provide a primal-dual algorithm that approximately solves this min-max game. Observe that since $\Delta(k, \tau)$ is a subset of $\Delta(\tau)$, we always have

$$\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) \leq \min_{\boldsymbol{\alpha}\in\Delta(k,\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}),$$

and therefore, in order to approximately solve the game of Equation (6.1.11), it is sufficient to find $\hat{\boldsymbol{\alpha}} \in \Delta(k, \tau)$ with

$$\max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \hat{\boldsymbol{\alpha}}) \approx \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{6.1.12}$$

Chapter 7

A Primal-Dual Approach for Sparse Approximation

In this chapter we provide an efficient algorithm for approximately solving the sparse approximation problem of Equation (6.1.4). Our approximation algorithm highly relies on Bregman projections [65]. Therefore, before introducing the GAME algorithm, we first provide a few important properties of Bregman projections.

7.1 Bregman Distances and Projections

Bregman divergences or Bregman distances are an important family of distances that all share similar properties [65, 40].

Definition 7.1 (Bregman Distance)**.** *Let $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ be a continuously-differentiable real-valued and strictly convex function defined on a closed convex set $\mathcal{S}$. The Bregman distance associated with $\mathcal{R}$ for points $\mathbf{P}$ and $\mathbf{Q}$ is:*

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}) = \mathcal{R}(\mathbf{P}) - \mathcal{R}(\mathbf{Q}) - \langle (\mathbf{P} - \mathbf{Q}), \nabla \mathcal{R}(\mathbf{Q}) \rangle.$$

Intuitively, the Bregman distance measures the strictness of convexity of the function $\mathcal{R}$. and its geometric significance is illustrated in Figure 7.1. The Bregman divergence is the vertical distance at $\mathbf{P}$ between the graph of $\mathcal{R}$ and the line tangent to the graph of $\mathcal{R}$ in $\mathbf{Q}$. Table 7.1 summarizes examples of the most widely used Bregman functions and the corresponding Bregman distances.

Note that the Bregman distance is not a metric. It is not symmetric, and it does not satisfy the triangle inequality. However, it has several important properties that we will use later in analyzing our sparse approximation algorithm.

Theorem 7.2. *Bregman distance satisfies the following properties:*

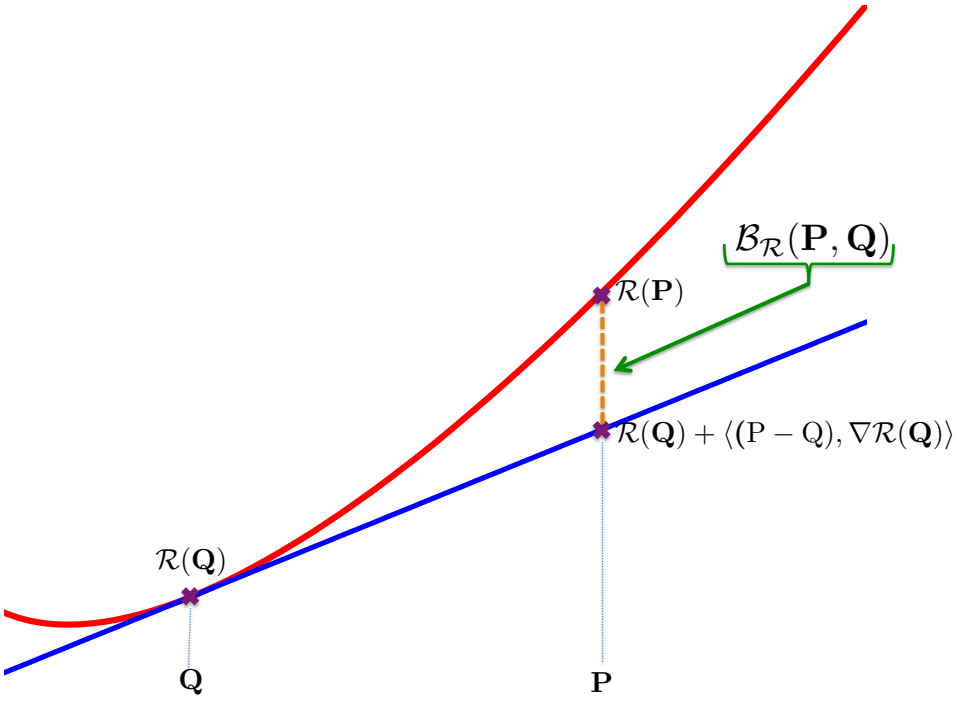


Figure 7.1: The Bregman divergence associated with a continuously-differentiable real-valued and strictly convex function $\mathcal{R}$ is the vertical distance at $\mathbf{P}$ between the graph of $\mathcal{R}$ and the line tangent to the graph of $\mathcal{R}$ in $\mathbf{Q}$.

- *(P1).* $\mathcal{B}_{\mathcal{R}}(\mathbf{P},\mathbf{Q}) \geq 0$, *and the equality holds if and only if* $\mathbf{P} = \mathbf{Q}$.
- *(P2). For every fixed* $\mathbf{Q}$ *if we define* $\mathcal{G}(\mathbf{P}) = \mathcal{B}_{\mathcal{R}}(\mathbf{P},\mathbf{Q})$, *then*

$$\nabla\mathcal{G}(\mathbf{P}) = \nabla\mathcal{R}(\mathbf{P}) - \nabla\mathcal{R}(\mathbf{Q}).$$

- *(P3). Three point property: For every* $\mathbf{P},\mathbf{Q}$ *and* $\mathbf{T}$ *in* $\mathcal{S}$

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P},\mathbf{Q}) = \mathcal{B}_{\mathcal{R}}(\mathbf{P},\mathbf{T}) + \mathcal{B}_{\mathcal{R}}(\mathbf{T},\mathbf{Q}) + \langle(\mathbf{P}-\mathbf{T}), \nabla\mathcal{R}(\mathbf{Q}) - \nabla\mathcal{R}(\mathbf{T})\rangle.$$

- *(P4). For every* $\mathbf{P},\mathbf{Q} \in \mathcal{S}$,

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P},\mathbf{Q}) + \mathcal{B}_{\mathcal{R}}(\mathbf{Q},\mathbf{P}) = \langle(\mathbf{P}-\mathbf{Q}), (\nabla\mathcal{R}(\mathbf{P}) - \nabla\mathcal{R}(\mathbf{Q}))\rangle.$$

Proof. All four properties follow directly from Definition 7.1. □

Now that we have introduced important properties of Bregman distances, we are ready to define Bregman projections of points into convex sets.

Table 7.1: Summary of the most popular Bregman functions and their corresponding Bregman distances. Here $\mathbf{\Phi}$ is a positive semidefinite matrix.

Name	Bregman Function ($\mathcal{R}(\mathbf{P})$)	Bregman Distance ($\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q})$)
Squared Euclidean	$\|\mathbf{P}\|_2^2$	$\|\mathbf{P} - \mathbf{Q}\|_2^2$
Squared Mahalanobis	$\langle \mathbf{P}, \mathbf{\Phi P} \rangle$	$\langle (\mathbf{P} - \mathbf{Q}), \mathbf{\Phi}(\mathbf{P} - \mathbf{Q}) \rangle$
Kullback-Leibler	$\sum_i \mathrm{P}_i \log \mathrm{P}_i - \sum_i \mathrm{P}_i$	$\sum_i \mathrm{P}_i \log \frac{\mathrm{P}_i}{\mathrm{Q}_i} - \sum_i \mathrm{P}_i + \sum_i \mathrm{Q}_i$
Itakura-Saito	$\sum_i -\log \mathrm{P}_i$	$\sum_i \left(\frac{\mathrm{P}_i}{\mathrm{Q}_i} - \log \frac{\mathrm{P}_i}{\mathrm{Q}_i} + 1 \right)$

Definition 7.3 (Bregman Projection)**.** *Let $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ be a continuously-differentiable real-valued and strictly convex function defined on a closed convex set $\mathcal{S}$. Let Ω be a closed subset of $\mathcal{S}$. Then, for every point $\mathbf{Q}$ in $\mathcal{S}$, the Bregman projection of $\mathbf{Q}$ into Ω, denoted as $\mathcal{P}_\Omega(\mathbf{Q})$ is*

$$\mathcal{P}_\Omega(\mathbf{Q}) \doteq \arg\min_{\mathbf{P} \in \Omega} \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}).$$

Bregman projections satisfy a generalized Pythagorean Theorem.

Theorem 7.4 (Generalized Pythagorean Theorem [65])**.** *Let $\mathcal{R} : \mathcal{S} \to \mathbb{R}$ be a continuously-differentiable real-valued and strictly convex function defined on a closed convex set $\mathcal{S}$. Let Ω be a closed subset of $\mathcal{S}$. Then for every $\mathbf{P} \in \Omega$ and $\mathbf{Q} \in \mathcal{S}$*

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}) \geq \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathcal{P}_\Omega(\mathbf{Q})) + \mathcal{B}_{\mathcal{R}}(\mathcal{P}_\Omega(\mathbf{Q}), \mathbf{Q}), \tag{7.1.1}$$

and in particular

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}) \geq \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathcal{P}_\Omega(\mathbf{Q})). \tag{7.1.2}$$

The generalized Pythagorean Theorem is illustrated in Figure 7.2. We refer the reader to [65], or [66] for a proof of this theorem and further discussions.

7.2 GAME Algorithm for Sparse Approximation

In this section we provide an efficient algorithm for approximately solving the problem of sparse approximation in ℓ_q norm, defined by Equation (6.1.3). Let $\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha})$ be the loss function defined by Equation (6.1.7), and recall that in order to approximately solve Equation (6.1.3), it is sufficient to find a sparse vector $\hat{\boldsymbol{\alpha}} \in \Delta(k, \tau)$ such that

$$\max_{\mathbf{P} \in \Xi_p} \mathcal{L}(\mathbf{P}, \hat{\boldsymbol{\alpha}}) \approx \min_{\boldsymbol{\alpha}' \in \Delta(\tau)} \max_{\mathbf{P} \in \Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{7.2.1}$$

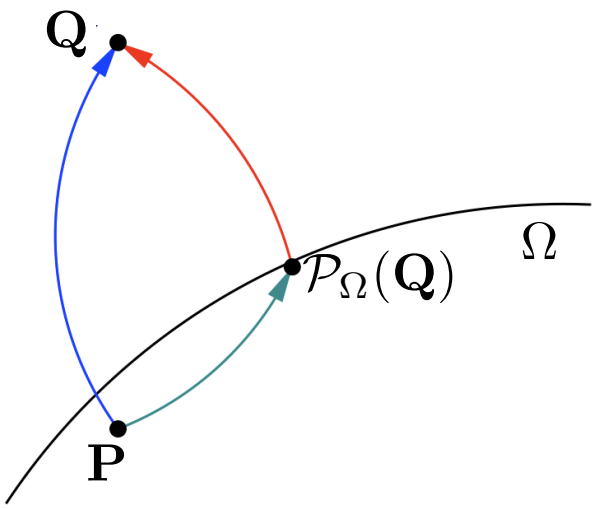


Figure 7.2: Generalized Pythagorean Theorem states that if $\mathcal{P}_\Omega(\mathbf{Q})$ is the Bregman projection of $\mathbf{Q}$ into Ω, then every other point $\mathbf{P}$ in Ω has larger Bregman distance to $\mathbf{Q}$ than to $\mathcal{P}_\Omega(\mathbf{Q})$. That is $\mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{Q}) \geq \mathcal{B}_\mathcal{R}(\mathbf{P}, \mathcal{P}_\Omega(\mathbf{Q}))$.

The original sparse approximation problem of Equation (6.1.3) is NP-complete, but it is computationally feasible to compute the value of the min-max game

$$\min_{\boldsymbol{\alpha}' \in \Delta(\tau)} \max_{\mathbf{P} \in \Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{7.2.2}$$

The reason is that the loss function $\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha})$ of Equation (6.1.7) is a bilinear function, and the sets $\Delta(\tau)$, and Ξ_p are both convex and closed.

Therefore, finding the game values and optimal strategies of the game of Equation (7.2.2) is equivalent to solving a convex optimization problem and can be done using off-the-shelf non-smooth convex optimization methods [204, 203]. However, if an off-the-shelf convex optimization method is used, then there is no guarantee that the recovered strategy $\hat{\boldsymbol{\alpha}}$ is also sparse. We need an approximation algorithm that finds near-optimal strategies $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{P}}$ for Mindy and Max with the additional guarantee that Mindy's near optimal strategy $\hat{\boldsymbol{\alpha}}$ is sparse.

Here we introduce the *Game-theoretic Approximate Matching Estimator* (GAME) algorithm which finds a sparse *approximation* to the min-max optimal solution of the game defined in Equation (7.2.2). The GAME algorithm relies on the general primal-dual approach which was originally applied to developing strategies for repeated games [117] (see also [144] and [135]). The pseudocode of the GAME Algorithm is provided in Algorithm 3.

Algorithm 3 GAME Algorithm for Sparse Approximation in ℓ_q-norm.

Inputs: M-dimensional vector $\boldsymbol{f}$, $M \times N$ matrix $\boldsymbol{\Phi}$, number of iterations T, sparse approximation norm q, Bregman function $\mathcal{R}$ and regularization parameter η.
Output: N-dimensional vector $\hat{\boldsymbol{\alpha}}$

0. Find a point $\mathbf{Q^1} \in \Xi_p$ such that $\nabla\mathcal{R}(\mathbf{P^1}) = \mathbf{0}_M$, and set $\mathbf{P}^1 = \mathcal{P}_{\Xi_p}(\mathbf{Q}^1)$.
for $t = 1, \ldots, T$ **do**
1. Let $\boldsymbol{r^t} \doteq \boldsymbol{\Phi}^\top \mathbf{P^t}$ (Requires one matrix-vector multiplication)
2. Find the index i of one largest (in magnitude) element of $\boldsymbol{r^t}$.
3. Let $\boldsymbol{\alpha^t}$ be a 1-sparse vector with

$$\mathrm{Supp}(\boldsymbol{\alpha}^t) = \{i\}, \text{and } \alpha_i^t = -\tau \,\mathrm{Sign}\left(r_i^t\right).$$

(Lemma 7.5: $\boldsymbol{\alpha}^t = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$.)

4. Choose a $\mathbf{Q}^{t+1}$ such that

$$\nabla\mathcal{R}\left(\mathbf{Q^{t+1}}\right) = \nabla\mathcal{R}(\mathbf{P}^t) + \eta\left(\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}\right).$$

5. Project $\mathbf{Q}^{t+1}$ into Ξ_p:

$$\mathbf{P}^{t+1} \doteq \mathcal{P}_{\Xi_p}(\mathbf{Q}^{t+1}) = \arg\min_{\mathbf{P}\in\Xi_p} \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}^{t+1}).$$

end for
6. Output $\hat{\boldsymbol{\alpha}} \doteq \frac{1}{T}\sum_{t=1}^{t} \boldsymbol{\alpha}^t$.

The GAME Algorithm can be viewed as a repeated game between two players Mindy and Max who iteratively update their current strategies $\mathbf{P}^t$ and $\boldsymbol{\alpha}^t$, with the aim of ultimately finding near-optimal strategies based on a T-round interaction with each other. Here we briefly explain how each player updates his/her current strategy based on the new update from the other player.

Recall that the ultimate goal is to find the solution of the game

$$\min_{\boldsymbol{\alpha}'\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}).$$

At the begining of each iteration t, Mindy receives the updated value $\mathbf{P}^t$ from Max. A greedy Mindy only focuses on Max's current strategy, and updates her current strategy to $\boldsymbol{\alpha}^t = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$. In the following lemma we show that this is indeed what our Mindy does in the first three steps of the main loop.

Lemma 7.5. *Let $\mathbf{P}^t$ denote Max's strategy at the begining of iteration t. Let $\boldsymbol{r}^t = \boldsymbol{\Phi}^\top\mathbf{P}^t$, and let i denote the index of a largest (in magnitude) element of $\boldsymbol{r}^t$. Let $\boldsymbol{\alpha}^t$ be a 1-sparse vector with $\mathrm{Supp}(\boldsymbol{\alpha}^t) = \{i\}$ and with $\alpha_i^t = -\tau\,\mathrm{Sign}\,(r_i^t)$. Then $\boldsymbol{\alpha}^t = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$.*

Proof. Let $\tilde{\boldsymbol{\alpha}}$ be any solution $\tilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$. It follows from the bilinearity of the loss function (Equation (6.1.7)) that

$$\tilde{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}) = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \langle \mathbf{P}^t, \boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\rangle = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \langle \boldsymbol{\Phi}^\top \mathbf{P}^t, \boldsymbol{\alpha}\rangle.$$

Hence, Holder inequality yields that for every $\boldsymbol{\alpha}^\# \in \Delta(\tau)$,

$$\langle \boldsymbol{\Phi}^\top \mathbf{P}^t, \boldsymbol{\alpha}^\#\rangle \geq -\|\boldsymbol{\alpha}^\#\|_1 \|\boldsymbol{\Phi}^\top \mathbf{P}^t\|_\infty \geq -\tau \|\boldsymbol{\Phi}^\top \mathbf{P}^t\|_\infty. \tag{7.2.3}$$

Now let $\boldsymbol{\alpha}^t$ be a 1-sparse vector with $\mathrm{Supp}(\boldsymbol{\alpha}^t) = \{i\}$ and $\boldsymbol{\alpha}^t_i = -\tau\,\mathrm{Sign}\,(r^t_i)$. Then $\boldsymbol{\alpha}^t \in \Delta(\tau)$, and

$$\langle \boldsymbol{\Phi}^\top \mathbf{P}^t, \boldsymbol{\alpha}^t\rangle = -\tau \|\boldsymbol{\Phi}^\top \mathbf{P}^t\|_\infty.$$

In other words, for $\boldsymbol{\alpha}^t$ the Holder inequality of Equation (7.2.3) is an equality. Hence $\boldsymbol{\alpha}^t$ is a minimizer of $\langle \boldsymbol{\Phi}^\top \mathbf{P}^t, \boldsymbol{\alpha}\rangle$. □

Thus far we have seen that at each iteration Mindy always finds a 1-sparse solution $\boldsymbol{\alpha}^t = \arg\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$. Mindy then sends her updated strategy $\boldsymbol{\alpha}^t$ to Max, and now it is Max's turn to update his strategy. A greedy Max would prefer to update his strategy as $\mathbf{P}^{t+1} = \arg\max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t)$. However, our Max is more conservative and prefers to stay close to his previous value $\mathbf{P}^t$. In other words, Max has two competing objectives

1. Maximizing $\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t)$, or equivalently minimizing $-\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t)$.
2. Remaining close to the previous strategy $\mathbf{P}^t$, by minimizing $\mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{P}^{t-1})$.

Let

$$\mathcal{L}_\mathcal{R}(\mathbf{P}) \doteq -\eta\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) + \mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{P}^t),$$

be a regularized loss function which is a linear combination of the two objectives above.

A conservative Max then tries to minimize a combination of the two objectives above by minimizing the regularized loss function

$$\mathbf{P}^{t+1} = \arg\min_{\mathbf{P}\in\Xi_p} \mathcal{L}_\mathcal{R}(\mathbf{P}) = \arg\min_{\mathbf{P}\in\Xi_p} -\eta\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) + \mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{P}^t). \tag{7.2.4}$$

Unfortunately, it is not so easy to efficiently solve the optimization problem of Equation (7.2.4) at every iteration. To overcome this difficulty, our Max first ignores the constraint $\mathbf{P}^{t+1} \in \Xi_p$, and instead finds a global optimizer of $\mathcal{L}_\mathcal{R}(\mathbf{P})$ by setting $\nabla\mathcal{L}_\mathcal{R}(\mathbf{P}) = \mathbf{0}_M$, and then projects back the result to Ξ_p via a Bregman projection.

More precisely, it follows from the Property (P2) of Bregman distance (Theorem 7.2) that for every $\mathbf{P}$

$$\nabla\mathcal{L}_\mathcal{R}(\mathbf{P}) = -\eta(\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}) + \nabla\mathcal{R}(\mathbf{P}) - \nabla\mathcal{R}(\mathbf{P}^t),$$

and therefore if $\mathbf{Q}^t$ is a point with

$$\nabla\mathcal{R}(\mathbf{Q}^t) = \nabla\mathcal{R}(\mathbf{P}^{t-1}) + \eta(\mathbf{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}),$$

then $\nabla\mathcal{L}_{\mathcal{R}}(\mathbf{Q}^t) = \mathbf{0}_M$.

The vector $\mathbf{Q}^t$ is finally projected back to Ξ_p via a Bregman projection to ensure that Max's new strategy is in the feasible set Ξ_p.

7.3 Analysis of the GAME Algorithm

In this section we prove that the GAME algorithm finds a near-optimal solution for the sparse approximation problem of Equation (6.1.3). The analysis of the GAME algorithm relies heavily on the analysis of the generic primal-dual approach. This approach originates from the *link-function methodology* in computational optimization [135, 172], and is related to the *mirror descent* approach in the optimization community [202, 26] . The primal-dual Bregman optimization approach is widely used in online optimization applications including portfolio selection [80, 145], online learning [1], and boosting [176, 76].

Let $\mathcal{A}$ and $\mathcal{B}$ be two convex sets, and let $\mathcal{L} : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ be a loss function which is convex with respect to $\mathcal{A}$, and concave with respect to $\mathcal{B}$. In online convex optimization, an online player chooses a point $\boldsymbol{a} \in \mathcal{A}$. After the point is chosen, an adversary chooses a point $\boldsymbol{b} \in \mathcal{B}$, and the online player receives payoff $\mathcal{L}(\boldsymbol{a}, \boldsymbol{b})$. This scenario is repeated for T iterations, and the goal of the online player is to minimize the regret loss [143, 262]

$$\max_{\boldsymbol{b^1},\ldots,\boldsymbol{b^T}} \left[\sum_{t=1}^{T} \mathcal{L}(\boldsymbol{a^t}, \boldsymbol{b^t}) - \min_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{a}, \boldsymbol{b^t}) \right].$$

However, there is a major difference between the sparse approximation problem and the problem of online convex optimization. In the sparse approximation problem, the set $\mathcal{A} = \Delta(k, \tau)$ is not convex anymore; therefore, there is no guarantee that an online convex optimization algorithm outputs a sparse strategy $\hat{\boldsymbol{\alpha}}$. Hence, it is not possible to directly translate the bounds from the online convex optimization scheme to the sparse approximation scheme.

Moreover, as discussed in Lemma 7.5 there is also a major difference between the Mindy players of the GAME algorithm and the general Mindy of general online convex optimization games. In the GAME algorithm Mindy is not a *blackbox adversary* that responds with an update to her strategy based on Max's update. Here Mindy always performs a greedy update and finds the *best* strategy as a response to Max's update. Moreover, our Mindy always finds a 1-*sparse* new strategy. That is, she looks among all best responses to Max's update, and finds a 1-sparse strategy among them.

As we will see next, the combination of cooperativeness by Mindy, and standard ideas for bounding the regret in online convex optimization schemes, enables us to analyze the GAME algorithm for sparse approximation. The following lemma bounds the regret loss of the primal-dual strategy in online convex optimization problems and is proved in [144].

Theorem 7.6. *Let q and T be positive integers, and let $p = \frac{q}{q-1}$. Suppose that $\mathcal{R}$ is such that for every $\mathbf{P}, \mathbf{Q} \in \Xi_p$, $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}) \geq \|\mathbf{P} - \mathbf{Q}\|_p^2$, and let*

$$\mathrm{G} = \max_{\boldsymbol{\alpha} \in \Delta(1,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q. \tag{7.3.1}$$

Also assume that for every $\mathbf{P} \in \Xi_p$, we have $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^1) \leq \mathrm{D}^2$. Suppose $\langle (\mathbf{P}^1, \boldsymbol{\alpha}^1), \cdots, (\mathbf{P}^T, \boldsymbol{\alpha}^T) \rangle$ is the sequence of pairs generated by the GAME *Algorithm after T iterations with $\eta = \frac{2\mathrm{D}}{\mathrm{G}\sqrt{T}}$. Then*

$$\max_{\mathbf{P} \in \Xi_p} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) \leq \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) + \frac{\mathrm{DG}}{2\sqrt{T}}.$$

Proof. Let $\mathbf{P}$ be an arbitrary point in Ξ_p. We have

$$\begin{aligned}
\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) - \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) &= \langle (\mathbf{P} - \mathbf{P}^t), \boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f} \rangle \\
&=^a \frac{1}{\eta} \langle (\mathbf{P} - \mathbf{P}^t), \nabla\mathcal{R}(\mathbf{Q}^{t+1}) - \nabla\mathcal{R}(\mathbf{P}^t) \rangle \\
&=^b \frac{1}{\eta} \left(\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^t) - \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}^{t+1}) + \mathcal{B}_{\mathcal{R}}(\mathbf{P}^t, \mathbf{Q}^{t+1}) \right) \\
&\leq^c \frac{1}{\eta} \left(\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^t) - \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^{t+1}) + \mathcal{B}_{\mathcal{R}}(\mathbf{P}^t, \mathbf{Q}^{t+1}) \right).
\end{aligned} \tag{7.3.2}$$

Equality (a) follows from the definition of $\mathbf{Q}^{t+1}$ (Step 4 of Algorithm 3). Equality (b) is the three point property of Bregman distances (Property (P3) of Theorem 7.2), and inequality (c) follows from the generalized Pythagorean theorem for Bregman projections (Theorem 7.4) as $\mathbf{P}^{t+1}$ is the Bregman projection of $\mathbf{Q}^{t+1}$ into Ξ_p (Step 5 of Algorithm 3).

Therefore, from the telescoping trick we have

$$\sum_{t=1}^{T} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) - \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) = \frac{1}{\eta} \left(\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^1) - \mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^{T+1}) + \sum_{t=1}^{T} \mathcal{B}_{\mathcal{R}}(\mathbf{P}^t, \mathbf{Q}^{t+1}) \right) \tag{7.3.3}$$

$$\leq^d \frac{\mathrm{D}^2}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} \mathcal{B}_{\mathcal{R}} \left(\mathbf{P}^t, \mathbf{Q}^{t+1} \right).$$

The inequality (d) follows from the facts that $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^{T+1})$ is non-negative and $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^1) \leq \mathrm{D}^2$. Our next step is to bound $\mathcal{B}_{\mathcal{R}}(\mathbf{P}^t, \mathbf{Q}^{t+1})$. From Property (P4) of Theorem 7.2 we have

$$\mathcal{B}_{\mathcal{R}}\left(\mathbf{P}^t, \mathbf{Q}^{t+1})\right) + \mathcal{B}_{\mathcal{R}}\left(\mathbf{Q}^{t+1}, \mathbf{P}^t)\right) = \langle (\mathbf{Q}^{t+1} - \mathbf{P}^t), (\nabla \mathcal{R}(\mathbf{Q}^{t+1}) - \nabla \mathcal{R}(\mathbf{P}^t))\rangle \quad (7.3.4)$$

$$= \eta \langle (\mathbf{Q}^{t+1} - \mathbf{P}^t), (\mathbf{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f})\rangle. \quad (7.3.5)$$

Now from Holder inequality we get

$$\eta \langle (\mathbf{Q}^{t+1} - \mathbf{P}^t), (\mathbf{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f})\rangle \leq 2\frac{\eta}{2} \|\mathbf{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}\|_q \|\mathbf{Q}^{t+1} - \mathbf{P}^t\|_p \leq \left(\frac{\eta^2}{4}\mathrm{G}^2 + \|\mathbf{Q}^{t+1} - \mathbf{P}^t\|_p^2\right). \quad (7.3.6)$$

Thus, by plugging back Equation (7.3.6) into Equation (7.3.4), and recalling the assumption $\mathcal{B}_{\mathcal{R}}(\mathbf{Q}^{t+1}, \mathbf{P}^t) \geq \|\mathbf{Q}^{t+1} - \mathbf{P}^t\|_p^2$, we get

$$\mathcal{B}_{\mathcal{R}}\left(\mathbf{P}^t, \mathbf{Q}^{t+1})\right) \leq \frac{\eta^2}{4}\mathrm{G}^2. \quad (7.3.7)$$

Finally plugging Equation (7.3.7) into (7.3.3) and summing over all T yields

$$\sum_{t=1}^{T} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) - \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) \leq \frac{\mathrm{D}^2}{\eta} + \frac{\eta T \mathrm{G}^2}{4}. \quad (7.3.8)$$

Equation 7.3.8 is valid for any η and every $\mathbf{P} \in \Xi_p$. In particular by setting $\eta = \frac{2\mathrm{D}}{\mathrm{G}\sqrt{T}}$, taking the maximum over $\mathbf{P}$ we get

$$\max_{\mathbf{P} \in \Xi_p} \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}^t) \leq \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) + \frac{\mathrm{DG}}{2\sqrt{T}}. \quad (7.3.9)$$

□

Finally we use Theorem 7.6 to show that the GAME algorithm after T iterations finds a T-sparse vector $\hat{\boldsymbol{\alpha}}$ with near-optimal value $\|\mathbf{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_q$.

Theorem 7.7. *Let q and T be positive integers, and let $p = \frac{q}{q-1}$. Suppose that for every $\mathbf{P}, \mathbf{Q} \in \Xi_p$, the function $\mathcal{R}$ satisfies $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{Q}) \geq \|\mathbf{P} - \mathbf{Q}\|_p^2$, and let*

$$\mathrm{G} = \max_{\boldsymbol{\alpha} \in \Delta(1,\tau)} \|\mathbf{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q. \quad (7.3.10)$$

Also assume that for every $\mathbf{P} \in \Xi_p$, we have $\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^1) \leq \mathrm{D}^2$. Suppose $\langle (\mathbf{P}^1, \boldsymbol{\alpha}^1), \cdots, (\mathbf{P}^T, \boldsymbol{\alpha}^T)\rangle$ is the sequence of pairs generated by the GAME *Algorithm after T iterations with $\eta = \frac{2\mathrm{D}}{\mathrm{G}\sqrt{T}}$. Let $\hat{\boldsymbol{\alpha}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^t$ be the output of the* GAME *algorithm. Then $\hat{\boldsymbol{\alpha}}$ is a T-sparse vector with $\|\hat{\boldsymbol{\alpha}}\|_1 \leq \tau$ and*

$$\|\mathbf{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_q \leq \min_{\boldsymbol{\alpha} \in \Delta(T,\tau)} \|\mathbf{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q + \frac{\mathrm{DG}}{2\sqrt{T}}. \quad (7.3.11)$$

Proof. It follows from Step 2. of Algorithm 3 that every $\boldsymbol{\alpha}^t$ is 1-sparse and $\|\boldsymbol{\alpha}^t\|_1 = \tau$. Therefore, $\hat{\boldsymbol{\alpha}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^t$ can have at most T non-zero entries and moreover $\|\hat{\boldsymbol{\alpha}}\|_1 \leq \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{\alpha}^t\|_1 \leq \tau$. Therefore $\hat{\boldsymbol{\alpha}}$ is in $\Delta(T, \tau)$.

Next we show that the Equation 7.3.11 holds for $\hat{\boldsymbol{\alpha}}$. Let $\hat{\mathbf{P}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{P}^t$. Observe that

$$\min_{\boldsymbol{\alpha}\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) =^{e} \max_{\mathbf{P}\in\Xi_p} \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) \geq^{f} \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}\left(\hat{\mathbf{P}}, \boldsymbol{\alpha}\right) \geq^{g} \frac{1}{T} \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$$
$$\geq^{h} \frac{1}{T}\sum_{t=1}^{T} \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}) =^{i} \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha}^t) \geq^{j} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}\left(\mathbf{P}, \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^t\right) - \frac{\mathrm{DG}}{2\sqrt{T}}.$$

Equality (e) is the minimax Theorem (Theorem 6.2). Inequality (f) follows from the definition of the max function. Inequalities (g) and (h) are consequences of the bilinearity of $\mathcal{L}$ and concavity of the min function. Equality (i) is valid by the definition of $\boldsymbol{\alpha}^t$, and Inequality (j) follows from Theorem 7.6. As a result

$$\begin{aligned} \|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_q &= \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \hat{\boldsymbol{\alpha}}) \qquad (7.3.12) \\ &\leq \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \max_{\mathbf{P}\in\Xi_p} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) + \frac{\mathrm{DG}}{2\sqrt{T}} = \min_{\boldsymbol{\alpha}\in\Delta(\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_q + \frac{\mathrm{DG}}{2\sqrt{T}}. \end{aligned}$$

□

Remark 7.8. *In general, different choices for the Bregman function may lead to different convergence bounds with different running times to perform the new projections and updates. For instance, a multiplicative update version of the algorithm can be derived by using the Bregman divergence based on the Kullback-Leibler function, and an additive update version of the algorithm can be derived by using the Bregman divergence based on the squared Euclidean function.*

Part III

Expander-Based Compressed Sensing

Chapter 8

Efficient Compressed Sensing using Optimized Expander Graphs

In Chapter 3 we introduced *compressed sensing* with the goal of replacing conventional sampling with a more general combination of linear measurement and non-linear reconstruction in order to acquire certain kinds of signals at a rate significantly below Nyquist.

Recall that the original approach employed Gaussian and Rademacher random matrices satisfying the RIP. However, as we discussed in Section 3.4, in order to store the whole matrix in memory we still need $O(MN)$ units of storage which is inefficient. Also with these matrices each matrix-vector multiplication requires $O(MN)$ operations. We then introduced the partial Fourier/Hadamard matrices as another family of random matrices satisfying the RIP. The partial Fourier/Hadamard matrices are obtained by randomly sampling rows of the Fourier/Hadamard matrix. These matrices require $O\left(M \log N\right)$ units of storage, but now the number of required measurements is suboptimal, $M = \Omega\left(k \log^5 N\right)$ compared to $M = O\left(k \log\left(\frac{N}{k}\right)\right)$ for Gaussian and Rademacher matrices. Moreover, there is no efficient algorithm to verify whether a given random matrix satisfies the RIP or not.

In this chapter, we will introduce efficient and sparse sensing matrices that are constructed from the adjacency matrices of expander graphs. We will see that expander graphs do not have the storage and computational issues of dense random matrices, and moreover explicit constructions for such matrices exist.

The idea of using expander graphs in compressed sensing is based on the connections between coding theory and compressed sensing (See Section 3.2.) In 1996, Sipser and Spielman [228] used expander graphs to build a family of linear error-correcting codes with linear decoding time complexity. These codes belong to class of error correcting codes called *Low Density Parity Check (LDPC) Codes*. Later, Xu and Hassibi [258, 259] generalized the decoding of expander codes to the field of real numbers and proposed the first reconstruction algorithm for expander-based compressed sensing.

Following [258, 259], we will show how random dense matrices can be replaced by the adjacency matrices of an optimized family of expander graphs, thereby reducing the space complexity of matrix storage and the time complexity of recovery to a few simple iterations. The main idea is that we study expander graphs with expansion coefficient beyond the $\frac{3}{4}$ that was considered in [258, 259]. Our results have interesting connection with the sequential decoding of expander codes, and generalize the results of Sipser and Spielman[228], and Xu and Hassibi [258, 259].

In the remainder of this chapter, we first formally define expander graphs, and describe a few key properties that we later use in expander-based compressed sensing. In Section 8.3 we propose an efficient combinatorial algorithm that efficiently recovers any k-sparse vector $\boldsymbol{\alpha}^*$ from $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$ after at most $2k$ simple iterations.

Our proposed algorithm generalizes the algorithm of [258, 259] to expander graphs with expansion coefficient beyond $\frac{3}{4}$. The key difference is that now the progress in each iteration is proportional to $\log N$, as opposed to a constant in [258, 259] We then describe how the algorithm can be implemented efficiently using simple data structures.

In Section 8.4 we describe the connections between our proposed algorithm and the SMP algorithm, which was proposed subsequently by Berinde, Indyk and Ruzic [31], and can recover a sparse approximation to any vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ in the ℓ_1/ℓ_1 approximation settings of Section 3.5.

8.1 What is an Expander Graph?

We start by defining an *unbalanced bipartite vertex-expander graph* [148].

Definition 8.1. *Let $\mathcal{G}$ be a bipartite graph with variable (left-side) nodes $\boldsymbol{V}$, check (right-side) nodes $\boldsymbol{C}$, and edges $\boldsymbol{E}$. We say that $\mathcal{G}$ is a* (k, ϵ, d)-expander *if*

1. *$\mathcal{G}$ is left regular with left degree d. That is, every variable node is connected to exactly d check nodes,*
2. *for any subset $\boldsymbol{S}$ of the variable nodes $\boldsymbol{V}$ with $|\boldsymbol{S}| \leq k$, the set of neighbors $\mathcal{N}(\boldsymbol{S})$ of $\boldsymbol{S}$ has size $|\mathcal{N}(\boldsymbol{S})| > (1-\epsilon)d\,|\boldsymbol{S}|$.*

Figure 8.1 illustrates such a graph. Intuitively a bipartite graph is an expander if any sufficiently small subset of its variable nodes has a sufficiently large neighborhood. In the compressed sensing setting, $\boldsymbol{V}$ (respectively $\boldsymbol{C}$) will correspond to the components of the original signal (respectively its compressed representation). Hence, for a given $N = |\boldsymbol{V}|$ and sparsity level k, an "optimized" expander should have $M = |\boldsymbol{C}|$, d, and ϵ as small as possible, while k should be as close as possible to M. The following proposition, proved using the probabilistic method [9, 24], is well-known in the literature on expanders:

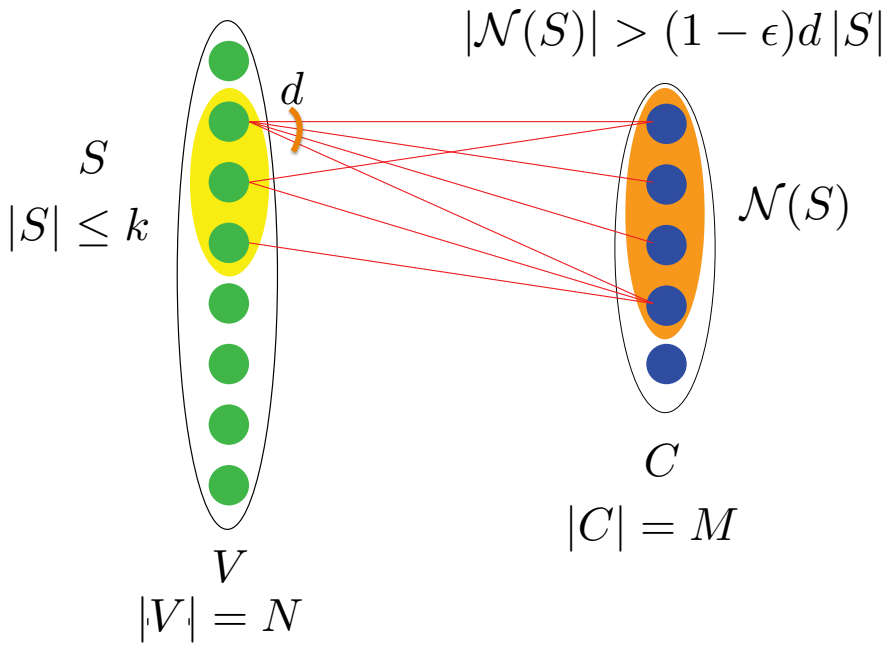


Figure 8.1: A (k, ϵ, d)-expander. In this example, the green nodes correspond to $\boldsymbol{V}$, the blue nodes correspond to $\boldsymbol{C}$, the yellow oval corresponds to the set $\boldsymbol{S} \subset \boldsymbol{V}$, and the orange oval corresponds to the set $\mathcal{N}(\boldsymbol{S}) \subset \boldsymbol{C}$.

Proposition 8.2 (Existence of optimized expanders)**.** *For any* $1 \leq k \leq \frac{N}{2}$ *and any* $\epsilon \in (0, 1)$*, there exists a* (k, ϵ, d)*-expander with left degree* $d = O\left(\frac{\log(N/k)}{\epsilon}\right)$ *and right set size* $M = O\left(\frac{k \log(N/k)}{\epsilon^2}\right)$.

As a result, by using such expander graph we can get optimality $M = O\left(\frac{k \log(N/k)}{\epsilon^2}\right)$ in the number of measurements. Unfortunately, the problem of deterministic construction of expanders from Definition 8.1 is only solved in the special case that $k = \Theta(N)$ [61]. However, it can be shown that, with high probability, any d-regular random graph with

$$d = O\left(\frac{\log(N/k)}{\epsilon}\right) \text{ and } M = O\left(\frac{k \log(N/k)}{\epsilon^2}\right)$$

satisfies the required expansion property [28]. Thus, on the one hand, it may suffice to use random bipartite regular graphs in many practical applications. On the other hand, there exists an explicit construction for a class of expander graphs that comes very close to the guarantees of Proposition 8.2. This construction, due to Guruswami,

Umans, and Vadhan. [139], uses Parvaresh-Vardy codes [208] and has the following guarantees:

Proposition 8.3 (Explicit construction of high-quality expanders)**.** *For any positive constant β, and any N, k, ϵ, there exists a deterministic construction of a (k, ϵ, d)-expander graph with $d = O\left(\left(\frac{\log n}{\epsilon}\right)^{\frac{1+\beta}{\beta}}\right)$ and $M = O(d^2 k^{1+\beta})$.*

Next we define a few combinatorial properties of the expander graphs that we will use later in our analysis.

Theorem 8.4 (Unique neighbor nodes)**.** *Let $\mathcal{G}$ be a (k, ϵ, d)-expander graph. Let $\boldsymbol{S}$ be a subset of the variable nodes with $|\boldsymbol{S}| \leq k$, and let $\mathcal{N}(\boldsymbol{S})$ denote the set of neighbors $\mathcal{N}(\boldsymbol{S})$ of $\boldsymbol{S}$. Let $\mathcal{N}_{=1}(\boldsymbol{S})$ denote the set of those nodes in $\mathcal{N}(\boldsymbol{S})$ that are connected to a single node in $\boldsymbol{S}$. Then*

$$|\mathcal{N}_{=1}(\boldsymbol{S})| > (1 - 2\epsilon)d\,|\boldsymbol{S}|.$$

Proof. Let $\mathcal{N}_{>1}(\boldsymbol{S})$, consist of those nodes in $\mathcal{N}(\boldsymbol{S})$ that are connected to more than a single node in $\boldsymbol{S}$. First observe that any node in $\mathcal{N}(\boldsymbol{S})$ is either in $\mathcal{N}_{=1}(\boldsymbol{S})$ or in $\mathcal{N}_{>1}(\boldsymbol{S})$. Therefore, it follows from the expansion property of the graph that

$$|\mathcal{N}_{=1}(\boldsymbol{S})| + |\mathcal{N}_{>1}(\boldsymbol{S})| = |\mathcal{N}(\boldsymbol{S})| > (1 - \epsilon)d\,|\boldsymbol{S}|.$$

Furthermore, since every node in $\mathcal{N}_{=1}(\boldsymbol{S})$ is connected to exactly one node in $\boldsymbol{S}$, and every node in $\mathcal{N}_{>1}(\boldsymbol{S})$ is connected to more than one node in $\boldsymbol{S}$, by counting the edges connecting $\boldsymbol{S}$ and $\mathcal{N}_(\boldsymbol{S})$, we have

$$|\mathcal{N}_{=1}(\boldsymbol{S})| + 2|\mathcal{N}_{>1}(\boldsymbol{S})| \leq d|\boldsymbol{S}|.$$

The result then follows by combining the latter two inequalities. □

The set $\mathcal{N}_{=1}(\boldsymbol{S})$ is called the set of unique neighbor nodes of $\boldsymbol{S}$

Corollary 8.5. *Let $\mathcal{G}$ be a (k, ϵ, d)-expander graph. Let $\boldsymbol{S}$ be a subset of the variable nodes with $|\boldsymbol{S}| \leq k$. Let $\mathcal{N}_{=1}(\boldsymbol{S})$ denote the set of those nodes in $\mathcal{N}(\boldsymbol{S})$ that are connected to a single node in $\boldsymbol{S}$. Then, there exists a node in $\boldsymbol{S}$ that is connected to more than $(1 - 2\epsilon)d$ nodes in $\mathcal{N}_{=1}(\boldsymbol{S})$.*

Proof. Assume that every node in $\boldsymbol{S}$ has at most $(1 - 2\epsilon)d$ unique neighbors. Then $|\mathcal{N}_{=1}(\boldsymbol{S})| \leq (1 - \epsilon)d\,|\boldsymbol{S}|$, which contradicts Theorem 8.4. □

Remark 8.6. *Corollary 8.5 guarantees that every subset $\boldsymbol{S}$ of variable nodes of an expander graph of size at most k has at least one node with more than $(1-2\epsilon)d$ unique neighbors with respect to $\boldsymbol{S}$. We will use this corollary in Section 8.3 in the analysis of our reconstruction algorithm.*

The following theorem is a generalization of the unique neighborhood property, and proves that in every set $\boldsymbol{S}$ of size at most k there are many nodes with a significantly large number of unique neighbors with respect to $\boldsymbol{S}$.

Theorem 8.7. *Let $\mathcal{G}$ be a (k, ϵ, d)-expander graph. Let $\boldsymbol{S}$ be a subset of the variable nodes with $|\boldsymbol{S}| \leq k$, and let $\mathcal{N}(\boldsymbol{S}) \subset \boldsymbol{C}$ denote the set of neighbors $\mathcal{N}(\boldsymbol{S})$ of $\boldsymbol{S}$. Let $\mathcal{N}_{=1}(\boldsymbol{S})$ denote the set of unique neighbor nodes of $\boldsymbol{S}$. For every $\kappa \geq 2\epsilon$ define*

$$\boldsymbol{S'} = \{v \in \boldsymbol{S} : |\mathcal{N}(v) \cap \mathcal{N}_{=1}(\boldsymbol{S})| \geq (1-\kappa)d. \tag{8.1.1}$$

Then $|\boldsymbol{S}'| > \left(1 - \frac{2\epsilon}{\kappa}\right)|\boldsymbol{S}|$.

Proof. We prove Theorem 8.7 by double-counting the size of $\mathcal{N}_{=1}(\boldsymbol{S})$. Observe that every node in $\boldsymbol{S} - \boldsymbol{S}'$ has at most $(1-\kappa)d$ unique neighbors. Moreover, since the graph is d-regular, every node in $\boldsymbol{S}'$ has at most d unique neighbors. Therefore,

$$(|\boldsymbol{S}| - |\boldsymbol{S}'|)(1-\kappa)d + |\boldsymbol{S}'|d \geq |\mathcal{N}_{=1}(\boldsymbol{S})|.$$

Now by using Theorem 8.4 we get

$$(|\boldsymbol{S}| - |\boldsymbol{S}'|)(1-\kappa)d + |\boldsymbol{S}'|d > (1-2\epsilon)d\,|\boldsymbol{S}|. \tag{8.1.2}$$

By simplifying Equation (8.1.2) we get $\kappa d|\boldsymbol{S}'| < (2\epsilon - \kappa)d|\boldsymbol{S}|$. □

8.2 Compressed Sensing and RIP-1 Property

Let $\boldsymbol{\Phi}$ be the $M \times N$ adjacency matrix of a (k, ϵ, d) expander graph. Expander-based compressed sensing uses the matrix $\boldsymbol{\Phi}$ as the sensing matrix. As the expander graph is d-regular, every column of $\boldsymbol{\Phi}$ has at most d non-zero entries which all have value 1. Therefore, storing the whole matrix requires only $O(dN) = O\left(N \log \frac{N}{k}\right)$ random bits. Moreover, each forward matrix-vector multiplication $\boldsymbol{u} = \boldsymbol{\Phi v}$ requires only $O(dN)$ operations, as every entry of $\boldsymbol{v}$ updates at most d entries of $\boldsymbol{u}$. Similarly, the adjoint multiplication $\boldsymbol{v} = \boldsymbol{\Phi}^\top \boldsymbol{u}$ also requires only $O(dN)$ operations. Therefore, sparse sensing matrices constructed from expander graphs have significant storage and computational advantages over dense Gaussian and Rademacher matrices.

Sensing matrices based on random expander graphs of Proposition 8.2 have the extra advantage that their number of measurements is optimal $M = O\left(k \log \frac{N}{k}\right)$. In contrast, sensing matrices based on explicit expander graphs of Proposition 8.3 have deterministic constructions.

Since every entry of $\boldsymbol{\Phi}$ is either zero or one, a result of Chandar [70] states that $\boldsymbol{\Phi}$ cannot satisfy the RIP property of Definition 3.5. However, a similar Restricted Isometry Property in the ℓ_1 norm (known as the RIP-1 property) can be derived from the expansion property and will guarantee the uniqueness of sparse representation. The RIP-1 property is proved by Berinde et. al. in [29].

Lemma 8.8 (RIP-1 property of the expander graphs [29]). *Let $\mathbf{\Phi}$ be the $M \times N$ adjacency matrix of a (k, ϵ, d)expander graph $\mathcal{G}$. Then for any k-sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^N$ we have:*

$$(1 - 2\epsilon)d\|\boldsymbol{\alpha}\|_1 \leq \|\mathbf{\Phi}\boldsymbol{\alpha}\|_1 \leq d\|\boldsymbol{\alpha}\|_1 \tag{8.2.1}$$

The full recovery property now follows immediately from Lemma 8.8 and guarantees that expander-based compressed sensing is at least information-theoretically possible.

Theorem 8.9 (Full recovery). *Let m be a positive integer. Suppose $\mathbf{\Phi}_{M\times N}$ is the adjacency matrix of an $((m+1)k, \epsilon, d)$ expander graph. Suppose $\boldsymbol{\alpha}$ is a k-sparse and $\boldsymbol{\alpha}'$ is an mk-sparse vector, such that $\mathbf{\Phi}\boldsymbol{\alpha} = \mathbf{\Phi}\boldsymbol{\alpha}'$. Then $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$.*

Proof. Let $\boldsymbol{z} = \boldsymbol{\alpha} - \boldsymbol{\alpha}'$. We have

$$\|\boldsymbol{z}|_0 \leq \|\boldsymbol{\alpha}\|_0 + \|\boldsymbol{\alpha}'\|_0 \leq (m+1)k.$$

Now from Lemma 8.8 we have:

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_1 \leq \frac{1}{(1-2\epsilon)}\|\mathbf{\Phi}\boldsymbol{\alpha} - \mathbf{\Phi}\boldsymbol{\alpha}'\|_1 = 0,$$

hence $\boldsymbol{\alpha} = \boldsymbol{\alpha}'$. □

Note that the proof of the above theorem essentially says that the adjacency matrix of a $((m+1)k, \epsilon, d)$ expander graph does not have a null vector that is $(m+1)k$ sparse. If $m = 2$ then Theorem 8.9 guarantees that no two k-sparse vectors can be mapped to the same measurement vector, and compressed sensing is information theoretically possible. We will also give a direct proof of this result (which does not appeal to RIP-1) since it gives a flavor of the main arguments of the next section.

Lemma 8.10 (Null space of $\mathbf{\Phi}$). *Let m be a positive integer, Suppose $\mathbf{\Phi}_{M\times N}$ is the adjacency matrix of an $((m+1)k, \epsilon, d)$ expander graph with $\epsilon \leq \frac{1}{2}$. Then any nonzero vector in the null space of $\mathbf{\Phi}$, i.e., any $\boldsymbol{z} \neq 0$ such that $\mathbf{\Phi}\boldsymbol{z} = 0$, has more than $(m+1)k$ nonzero entries.*

Proof. Define $\boldsymbol{S}$ to be the support set of z. Suppose that z has at most $(m+1)k$ nonzero entries, i.e., that $|\boldsymbol{S}| \leq (m+1)k$. Let $\mathcal{N}_{=1}(\boldsymbol{S})$ denotes the set of those nodes in $\mathcal{N}(\boldsymbol{S})$ that are connected to a single node in $\boldsymbol{S}$. Then from Theorem 8.4 we have $|\mathcal{N}_{=1}(\boldsymbol{S})| > (1-2\epsilon)d|\boldsymbol{S}| \geq 0$. This implies that there is at least one entry of $\mathbf{\Phi}\boldsymbol{z}$ which is only connected to one entry of the support of $\boldsymbol{z}$, and therefore has non-zero value. However, this contradicts the fact that $\mathbf{\Phi}\boldsymbol{z} = 0$ and so $\boldsymbol{z}$ must have more than $(m+1)k$ nonzero entries. □

Table 8.1 compares expander-based sensing matrices satisfying the RIP-1 property, with dense sensing matrices satisfying the RIP-2 property. Expander-based matrices

Table 8.1: Comparison between different properties of sensing matrices satisfying the RIP-2 property with the same properties of expander-based matrices satisfying the RIP-1 property. All bounds ignore the $O(.)$ constants.

Matrix	Number of measurements	Memory (random bits)	Matrix-vector multiplication	Random vs. Deterministic	RIP
Gaussian [21]	$k \log\left(\frac{N}{k}\right)$	$kN \log\left(\frac{N}{k}\right)$	$kN \log\left(\frac{N}{k}\right)$	Random	RIP-2
Partial (Fourier)[218]	$k \log^5 N$	$k \log^6 N$	$N \log N$	Random	RIP-2
Incoherent Frames [11, 15]	k^2	-	$N \log N$	Deterministic	RIP-2
Expander Graphs [159, 29]	$k \log\left(\frac{N}{k}\right)$	$N \log\left(\frac{N}{k}\right)$	$N \log\left(\frac{N}{k}\right)$	Random	RIP-1
Expander Graphs [139]	$k^{1+\beta} (\log N)^{\frac{2(1+\beta)}{\beta}}$	-	$N (\log N)^{\frac{(1+\beta)}{\beta}}$	Deterministic	RIP-1

are superior in terms of the number of measurements, storage, computation, and having explicit constructions. In the next sections we show that expander-based compressed sensing is also computationally possible by providing efficient and robust sparse recovery algorithms.

8.3 Efficient Sparse Recovery in the Noiseless Regime

In this section we propose a simple recovery algorithm which recovers any k-sparse vector $\boldsymbol{\alpha}^*$ from the low-dimensional vector $\boldsymbol{\Phi}\boldsymbol{\alpha}^*$ in only $2k$ simple iterations and a total running time of $O\left(N \log \frac{N}{k}\right)$. The proposed algorithm is a simple iterative algorithm that starts by setting the all-zero vector as its initial guess. At every iteration the algorithm updates only one coordinate of the guess vector by selecting a coordinate whose neighbors mostly have the same gap value (defined bellow). The algorithm keeps on updating the guess vector for at most $2k$ iterations. The pseudocode of our proposed algorithm is provided in Algorithm 4.

Before proving the result, we introduce some notations used in the recovery algorithm and in the proof.

In Algorithm 4 the gap is defined as follows.

Definition 8.11 (gap)**.** *Let $\boldsymbol{\alpha}^*$ be the original signal and $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$. Furthermore, let $\boldsymbol{\alpha}^t$ be our estimate for $\boldsymbol{\alpha}^*$ after t iterations of Algorithm 4 . For each variable*

Algorithm 4 Expander Recovery Algorithm for Sparse Signals

Inputs: An M dimensional vector $\boldsymbol{f}$, and the $M \times N$ matrix $\boldsymbol{\Phi}$ which is the adjacency of an expander graph.

Output: An N dimensional vector $\hat{\boldsymbol{\alpha}}$.

1: Initialize $\hat{\boldsymbol{\alpha}}^1 \doteq \mathbf{0}_N$ and $\boldsymbol{f}^1 \doteq \boldsymbol{f}$.
2: **for** $t = 1, \cdots, 2k$ **do**
3: **if** $\boldsymbol{f}^t = \mathbf{0}_M$ **then**
4: output $\hat{\boldsymbol{\alpha}}^t$ and exit.
5: **else**
6: For each variable node j, set

$$g_j^t \doteq \text{Median}\left(\{f_i^t : i \in \mathcal{N}(j)\}\right).$$

7: Find a variable node j such that at least $(1 - 2\epsilon)\ d$ of the measurements it participates in, have identical non-zero gap value g_j^t.
8: Set

$$\hat{\alpha_{j'}}^{t+1} \doteq \left\{ \begin{array}{ll} \hat{\alpha_{j'}}^t + g_j^t & \text{if } j' = j \\ \hat{\alpha_{j'}}^t & \text{Otherwise} \end{array} \right\}.$$

9: Set

$$\boldsymbol{f}^{t+1} \doteq \boldsymbol{f}^t - g_j^t \boldsymbol{\Phi}_j.$$

10: **end if**
11: **end for**

node j, we define the gap value g_j^t as:

$$g_j^t = \text{Median}\left(\{f_i^t : i \in \mathcal{N}(j)\}\right).$$

That is, each vertex j selects the entries f_i^t where i is a neighbor of j in $\mathcal{G}$, and then computes the median g_j^t of those d entries.

Definition 8.12. *At each iteration t, $\boldsymbol{G^t}$ is the support of the residual vector $\boldsymbol{f}^t$:*

$$\boldsymbol{G^t} = \text{Supp}(\boldsymbol{f}^t) = \text{Supp}(\boldsymbol{f} - \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t),$$

similarly $\boldsymbol{S^t}$ is the support of the difference between the true (unknown) vector $\boldsymbol{\alpha^*}$, and our candidate $\hat{\boldsymbol{\alpha}}^t$:

$$\boldsymbol{S^t} = \text{Supp}(\hat{\boldsymbol{\alpha}}^{\text{t}} - \boldsymbol{\alpha^*}) = \{\text{j} : \hat{\alpha}_{\text{j}}^{\text{t}} \neq \alpha_{\text{j}}^*\}.$$

Now we are ready to state the main result:

Theorem 8.13 (Expander Recovery Algorithm)**.** *Let $\boldsymbol{\Phi}_{m \times n}$ be the adjacency matrix of a $(3k, \epsilon, d)$ expander graph, where $\epsilon \leq 1/4$. Then, for any k-sparse signal $\boldsymbol{\alpha^*}$, given*

$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^*$, the expander recovery algorithm (Algorithm 4) recovers $\boldsymbol{\alpha}^*$ successfully in at most $2k$ iterations.

The proof consists of the following lemmas.

- The algorithm never gets stuck, and one can always find a coordinate j that is connected to at least $(1-2\epsilon)d$ parity nodes with identical non-zero gaps.
- With certainty the algorithm will stop after at most $2k$ rounds. Furthermore, by choosing ϵ small enough the number of iterations can be made arbitrarily close to k.

We need the following lemmas to prove Theorem 8.13.

Lemma 8.14 (progress)**.** *Suppose at each iteration t, $|\boldsymbol{S^t}| < 2k$. Then always there exists a variable node j such that at least $(1-2\epsilon)d$ of its neighbor check nodes have the same non-zero gap value g_j^t.*

Proof. Since $\boldsymbol{\Phi}$ is the normalized matrix of a $(2k, \epsilon, d)$ expander graph, and $|\boldsymbol{S^t}| < 2k$, it follows from Corollary 8.5 that there exists a coordinate j in $\boldsymbol{S^t}$ that is uniquely connected to at least $(1-2\epsilon)d$ check nodes, in other words no other non-zero variable node in $\boldsymbol{S^t}$ is connected to these nodes. This immediately implies the lemma. □

Lemma 8.15 (gap elimination)**.** *At each step t if $|\boldsymbol{S^t}| < 2k$ then $|\boldsymbol{G^{t+1}}| < |\boldsymbol{G^t}| - (1-4\epsilon)d$*

Proof. By the previous lemma, if $|\boldsymbol{S^t}| < 2k$, there always exists a node j that is connected to at least $(1-2\epsilon)d$ nodes with identical nonzero gap, and hence to at most $2\epsilon d$ nodes possibly with zero gaps. Adding the gap value g_j^t to the current value of this variable node sets the gaps on these uniquely connected neighbors of j to zero, but it may make some zero gaps on the remaining $2\epsilon d$ neighbors non-zero. So at least $(1-2\epsilon)d$ coordinates of $\boldsymbol{G^t}$ will become zero, and at most $2\epsilon d$ its zero coordinates may become non-zero. Hence

$$|\boldsymbol{G^{t+1}}| < |\boldsymbol{G^t}| - (1-2\epsilon)d + 2\epsilon d = |\boldsymbol{G^t}| - (1-4\epsilon)d. \tag{8.3.1}$$

□

The following lemma provides a direct connection between the size of $\boldsymbol{G^t}$ and the size of $\boldsymbol{S^t}$.

Lemma 8.16 (connection)**.** *If at iteration t, $|\boldsymbol{S^t}| < 2k$, then $(1-2\epsilon)d|\boldsymbol{S^t}| < |\boldsymbol{G^t}|$.*

Proof. Since $|\boldsymbol{S^t}| < 2k$, by Theorem 8.4

$$|\mathcal{N}_{=1}(\boldsymbol{S^t})| > (1-2\epsilon)d|\boldsymbol{S^t}|.$$

Also, each node in $\mathcal{N}_{=1}(\boldsymbol{S^t})$ has non-zero gap and so is a member of $\boldsymbol{G^t}$. □

Lemma 8.17 (preservation)**.** *At each step t if $|\boldsymbol{S^t}| < 2k$, after running the algorithm we have $|\boldsymbol{S^{t+1}}| < 2k$.*

Proof. Since at each step we are only changing one coordinate of $\hat{\boldsymbol{\alpha}}^t$, we have $|\boldsymbol{S^{t+1}}| = |\boldsymbol{S^t}| + 1$, so we only need to prove that $\boldsymbol{S^{t+1}} \neq 2k$.

Suppose for a contradiction that $|\boldsymbol{S^{t+1}}| = 2k$, and partition $\mathcal{N}(\boldsymbol{S^{t+1}})$ into two disjoint sets:

1. $\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})$: The vertices in $\mathcal{N}(\boldsymbol{S^{t+1}})$ that are connected only to one vertex in $\boldsymbol{S^{t+1}}$.
2. $\mathcal{N}_{>1}(\boldsymbol{S^{t+1}})$: The other vertices (that are connected to more than one vertex in $\boldsymbol{S^{t+1}}$).

The argument is similar to that given in Theorem 8.4; by double counting the number of vertices in $\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})$ and $\mathcal{N}_{>1}(\boldsymbol{S^{t+1}})$ one can show that

$$|\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})| > (1-2\epsilon)\ d\ |\boldsymbol{S^{t+1}}| = (1-2\epsilon)\ d\ 2k \tag{8.3.2}$$

Now we have the following facts:

- $|\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})| \leq |\boldsymbol{G^{t+1}}|$: Coordinates in $\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})$ are connected uniquely to coordinates in $\boldsymbol{S^{t+1}}$, hence each coordinate in $\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})$ has non-zero gap.
- $|\boldsymbol{G^{t+1}}| > |\boldsymbol{G^1}|$: gap elimination from Lemma 8.15.
- $|\boldsymbol{G^1}| \leq kd$: since $\boldsymbol{\alpha^*}$ is k-sparse and $\hat{\boldsymbol{\alpha}}^1$ is the all-zero vector, $\hat{\boldsymbol{\alpha}}^1$ and $\boldsymbol{\alpha^*}$ differ in at most k coordinates. Therefore, since the graph is d-regular, $\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^1$ and $\boldsymbol{\Phi\alpha^*}$ can differ in at most kd coordinates.

As a result we have:

$$(1-2\epsilon)2\ dk < |\mathcal{N}_{=1}(\boldsymbol{S^{t+1}})| \leq |\boldsymbol{G^{t+1}}| < |\boldsymbol{G^1}| \leq kd \tag{8.3.3}$$

This implies $\epsilon > \frac{1}{4}$ which contradicts the assumption $\epsilon < \frac{1}{4}$. □

Proof of the Theorem 8.13. Preservation (Lemma 8.17) and Progress (Lemma 8.14) together immediately imply that the algorithm will never get stuck. Also by Lemma 8.15 we had shown that $|\boldsymbol{G^1}| \leq kd$ and $|\boldsymbol{G^{t+1}}| < |\boldsymbol{G^t}| - (1-4\epsilon)d$. Hence after at most $T = \frac{k}{(1-4\epsilon)}$ steps we will have $|\boldsymbol{G^T}| = 0$ and this together with the Connection Lemma implies that $|\boldsymbol{S^T}| = 0$, which is the exact recovery of the original signal.

Note that we have to choose $\epsilon < \frac{1}{4}$, and as an example, by setting $\epsilon = \frac{1}{8}$ the recovery needs at most $2k$ iterations. □

Algorithm 4 is an iterative algorithm that consists of at most $2k$ simple iterations. Each iteration can be implemented efficiently (see [258]) since the adjacency matrix of the expander graph is sparse with all entries 0 or 1.

The efficiency of the algorithm can also be improved by using a priority queue data-structure. The idea is to use preprocessing as follows: For each variable node j compute the median of its neighbors $g_j^1 \doteq \text{Median}\left(\{f_i : i \in \mathcal{N}(j)\}\right)$ and also compute the number of neighbors with the same value g_j^1. Note that if a node has $(1-2\epsilon)d$ unique neighbors, their median should also be among them. Then construct the priority queue based on the values g_j^1, and at each iteration extract the root node from the queue, perform the gap elimination on it, and then, if required, make the correction on corresponding dD variable nodes, where d is the left-degree and D is the maximum right degree of the expander. The main computational cost of this variation of the algorithm will be the cost of building the priority queue which is $O(N \log \frac{N}{k})$; finding the median of d elements can be done in $O(d)$ and building a priority queue requires linear computational time.

8.4 Sparse Matching Pursuit

Thus far we have introduced efficient iterative algorithms for recovery of (almost) k-sparse vectors. An important next step is to generalize these algorithms to provide sparse approximations for *arbitrary* vectors in $\mathbb{R}^N$ in the settings of Section 3.5. More precisely, let $\boldsymbol{\Phi}$ be the adjacency matrix of an expander graph. The goal is to design an efficient recovery algorithm such that for every data vector $\boldsymbol{\alpha^*} \in \mathbb{R}^N$, and noise vector $\boldsymbol{e_M} \in \mathbb{R}^M$, given $\boldsymbol{f} = \boldsymbol{\Phi\alpha^*} + \boldsymbol{e_M}$, the algorithm can find a sparse vector $\hat{\boldsymbol{\alpha}}$ close to the best k-term approximation $\mathrm{H}_k(\boldsymbol{\alpha^*})$ of $\boldsymbol{\alpha^*}$.

Expander Matching Pursuit (EMP) [153] was the first expander-based recovery algorithm capable of solving the general sparse approximation problem. The key feature of the EMP algorithm is that sparse recovery requires near linear $O\left(N \log \frac{N}{k}\right)$ operations, while still using $O\left(k \log \frac{N}{k}\right)$ measurements. Moreover, the algorithm provides ℓ_1/ℓ_1 sparse approximation guarantees. That is, for every $\boldsymbol{\alpha^*} \in \mathbb{R}^N$ and $\boldsymbol{e_M} \in \mathbb{R}^M$, given $\boldsymbol{f} = \boldsymbol{\Phi\alpha^*} + \boldsymbol{e_M}$, the algorithm finds a k-sparse vector $\hat{\boldsymbol{\alpha}}$ with

$$\|\boldsymbol{\alpha^*} - \hat{\boldsymbol{\alpha}}\|_1 = O\left(\|\boldsymbol{\alpha^*} - \mathrm{H}_k(\boldsymbol{\alpha^*})\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d}\right).$$

However, the empirical performance of EMP is less impressive. The algorithm requires $M = 5000$ measurements to recover random signed 50-sparse signals of length $N = 20000$, whereas the convex optimization method with random Gaussian matrices requires only about $M = 450$ measurements [31, 98].

Sparse Matching Pursuit (SMP) [31] is an iterative message-passing algorithm designed to overcome the empirical suboptimality of the EMP algorithm. The SMP

algorithm shares significant similarities with Algorithm 4 provided in Section 8.3 for recovering sparse vectors, and can be viewed as a generalization of Algorithm 4.

In each iteration, the algorithm estimates the difference between the current approximation $\hat{\boldsymbol{\alpha}}^t$ and the true vector $\boldsymbol{\alpha}^*$ from the measurement error $\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}$. The estimation is obtained by using the median estimator $\boldsymbol{g}^t$ as in Algorithm 4. The data-domain approximation $\hat{\boldsymbol{\alpha}}^t$ is updated by $\boldsymbol{g}^t$, and the process is repeated. However, the key difference between the SMP algorithm and Algorithm 4 is that in contrast to Algorithm 4 which only updates one coordinate of $\hat{\boldsymbol{\alpha}}^t$ at each iteration, SMP updates $2k$ coordinates of $\hat{\boldsymbol{\alpha}}^t$ simultaneously in a greedy manner. The pseudocode of the SMP algorithm is provided in Algorithm 5.

Algorithm 5 Sparse Matching Pursuit (SMP)

Inputs: An M dimensional vector $\boldsymbol{f}$, the $M \times N$ expander matrix $\boldsymbol{\Phi}$, and the number of iterations T.

Output: An N dimensional vector $\hat{\boldsymbol{\alpha}}$.

1: Initialize $\hat{\boldsymbol{\alpha}}^1 \doteq \mathbf{0}_N$ and $\boldsymbol{f}^1 \doteq \boldsymbol{f}$.
2: **for** $t = 1, \cdots, T$ **do**
3: For each variable node j, set $g_j^t \doteq \text{Median}\left(\{f_i^t : i \in \mathcal{N}(j)\}\right)$.
4: Set $\boldsymbol{h^t} \doteq \text{H}_{2k}(\boldsymbol{g}^t)$.
5: Set $\hat{\boldsymbol{\alpha}}^{t+1} \doteq \text{H}_k(\hat{\boldsymbol{\alpha}}^t + \boldsymbol{h^t})$.
6: Set $\boldsymbol{f}^{t+1} \doteq \boldsymbol{f} - \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^{t+1}$.
7: **end for**
8: Output $\hat{\boldsymbol{\alpha}} \doteq \hat{\boldsymbol{\alpha}}^{T+1}$.

The following theorem summarizes the performance guarantees of the SMP algorithm.

Theorem 8.18 ([31]). *Let $\boldsymbol{\Phi}$ be the adjacency of an $(O(k), \epsilon, d)$ expander graph with sufficiently small ϵ. Then, there exists a constant κ_{SMP} depending only on ϵ, such that for any data vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ and noise vector $\boldsymbol{e_M} \in \mathbb{R}^M$, given $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$, the SMP algorithm after T iterations finds a k-sparse vector $\hat{\boldsymbol{\alpha}}$ with*

$$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_1 = \frac{\|\boldsymbol{\alpha}^*\|_1}{2^T} + 4\kappa_{\text{SMP}}\left(\|\boldsymbol{\alpha}^* - \text{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d}\right).$$

Corollary 8.19. *Let $\boldsymbol{\Phi}$ be the adjacency of an $(O(k), \epsilon, d)$ expander graph with sufficiently small ϵ. Then, there exists a constant κ_{SMP} depending only on ϵ, such that for any data vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ and noise vector $\boldsymbol{e_M} \in \mathbb{R}^M$, given $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$, the SMP algorithm after*

$$T = \log_2\left(\frac{\|\boldsymbol{\alpha}^*\|_1}{\left(\|\boldsymbol{\alpha}^* - \text{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d}\right)}\right)$$

iterations finds a k-sparse vector $\hat{\boldsymbol{\alpha}}$ *with*

$$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_1 = (1 + 4\kappa_{\mathrm{SMP}}) \left(\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d} \right).$$

Therefore, the SMP algorithm provides an ℓ_1/ℓ_1 best k-term approximation guarantee. The running time of the SMP algorithm is by a logarithmic factor higher than EMP; however, it has better empirical performance and requires significantly fewer measurements. For instance, $M = 2000$ measurements are enough successfully recover 50-sparse signals of dimension 20000.

Empirical performance can be further improved via the *Sequential Sparse Matching Pursuit (SSMP)* [30] which is a recent greedy variant of SMP. In contrast to the original SMP algorithm, SSMP performs sequential updates by always selecting the best update first. In the above setting $M = 1400$ measurements, rather than $M = 2000$, are sufficient for successful recovery. Nevertheless, the running time of the SSMP algorithm is by a logarithmic factor slower than SMP.

Remark 8.20. *Note that more efficient sparse approximation algorithms exist for the special case where the non-zero entries of the sparse signal have all positive values [169, 71].*

Expander-based compressed sensing with the reconstruction algorithms mentioned in this chapter provide *for all* sparse approximation guarantees. This means that the deterministic (random) expander sensing matrix combined with any of the reconstruction algorithms mentioned above guarantees ℓ_1/ℓ_1 best k-term approximation for every vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ surely (or with high probability). In contrast, another sparse recovery setting which was evolved in the context of data-streaming, provides a weaker *for-each* guarantee [123].

In this setting, the sensing matrix $\boldsymbol{\Phi}$ is a sparse matrix chosen at random from some distribution and for each vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$, the recovery algorithm successfully finds a sparse approximation to $\boldsymbol{\alpha}^*$ with probability at least $1 - \frac{1}{N}$. Examples of such algorithms which only require $O(k \log N)$ measurements are [79, 72] which provide instance optimal ℓ_2/ℓ_2 guarantees in $O(N \log N)$ running time, and [125] which provides instance optimal ℓ_1/ℓ_1 guarantees in $O\left(\mathrm{Poly}(k, \log N)\right)$ running time.

We note that even though these *for-each* algorithms provided tighter instance optimal guarantees or are faster, they are not typically resilient to measurement noise. This imposes an important difficulty on using them in compressed sensing applications other than the data-streaming application.

Chapter 9

A Game Theoretic Approach to Expander-based Compressive Sensing

9.1 RIP-1 and ℓ_1-Minimization

In Chapter 8 we introduced expander graphs and proposed efficient algorithms for recovery of (almost) sparse signals. We also described *combinatorial* sparse recovery algorithms with ℓ_1/ℓ_1 best k-term approximation guarantees. However, there is a major drawback with these algorithms if the signal is not almost sparse. While they are efficient and rather easy to implement, their approximation guarantees are meaningful only in extremely large dimensions. The big O notation in their ℓ_1/ℓ_1 guarantees hides large constants, making these algorithms only suitable for extremely high-dimensions or when the SNR is significantly high.

On the other hand, in Section 3.3 we mentioned the ℓ_1-minimization method as a robust sparse recovery scheme that depends on the *geometry* of sensing matrices satisfying the RIP in the ℓ_2 norm. It is natural to ask is whether it is possible to design stable sparse recovery algorithms, such as ℓ_1-minimization, that rely on the *geometry* of expander graphs.

In this section we will see that the answer to the above question is *yes*, and like the RIP-2 case, the RIP-1 property of the expander graphs is sufficient to guarantee that the ℓ_1-minimization methods stably recover every sparse vector.

In Lemma 8.8 we saw that the adjacency matrix of any (k, ϵ, d) expander graph almost preserves the ℓ_1 norm of any k-sparse vector. This property is known as the *Restricted Isometry Property for ℓ_1 norms* or the *RIP-1* property.

The following proposition is a direct consequence of the above RIP-1 property. It states that for any vector $\boldsymbol{u}$, if there exists a vector $\boldsymbol{v}$ whose ℓ_1 norm is close to that

of $\boldsymbol{u}$, then if $\boldsymbol{\Phi v}$ approximates $\boldsymbol{\Phi u}$ in the ℓ_1 norm, then $\boldsymbol{v}$ also approximates $\boldsymbol{u}$ in the ℓ_1 norm.

Theorem 9.1. *Let $\boldsymbol{\Phi}$ be the adjacency matrix of a $(2k, \epsilon, d)$-expander and $\boldsymbol{u}, \boldsymbol{v}$ be two vectors in $\mathbb{R}^n$, such that*

$$\|\boldsymbol{u}\|_1 \geq \|\boldsymbol{v}\|_1 - \Delta$$

for some $\Delta > 0$. Then

$$\|\boldsymbol{u} - \boldsymbol{v}\|_1 \leq \frac{(1-2\epsilon)}{(1-6\epsilon)} \left(2\|\boldsymbol{u} - \mathrm{H}_k(\boldsymbol{u})\|_1 + \Delta\right) + \frac{2}{d(1-6\epsilon)} \|\boldsymbol{\Phi u} - \boldsymbol{\Phi v}\|_1.$$

Proof. Let $S = \mathrm{Supp}(\mathrm{H}_k(\boldsymbol{u}))$, and let $\langle S^1, \cdots, S^t \rangle$ be a decreasing partitioning of $\overline{S}$ (with respect to coefficient magnitudes), such that all sets but (possibly) S^t have size k. Note that $S^0 = S$. Let $\boldsymbol{\Phi}_{\mathcal{N}(S)}$ be a submatrix of $\boldsymbol{\Phi}$ containing rows from $\mathcal{N}(S)$. Finally let $\boldsymbol{y} = \boldsymbol{u} - \boldsymbol{v}$. Then, following the argument of Berinde et al. [29], which also appears in Sipser and Spielman [228], we have the following chain of inequalities:

$$\begin{aligned}
\|\boldsymbol{\Phi y}\|_1 \geq \|\boldsymbol{\Phi}_{\mathcal{N}(S)} \boldsymbol{y}\|_1 &\geq \|\boldsymbol{\Phi}_{\mathcal{N}(S)} \boldsymbol{y}_S\|_1 - \sum_{i=1}^{t} \sum_{\substack{j \in S^i, l \in \mathcal{N}(S) \\ (j,l)\text{: edge}}} |y_j| \qquad (9.1.1) \\
&\geq d(1-2\epsilon)\|\boldsymbol{y}_S\|_1 - \sum_{i=1}^{t} \sum_{\substack{j \in S^i, l \in \mathcal{N}(S) \\ (j,l)\text{: edge}}} \frac{\|\boldsymbol{y}_{S^{i-1}}\|_1}{k} \\
&\geq d(1-2\epsilon)\|\boldsymbol{y}_S\|_1 - 2kd\epsilon \sum_{i=1}^{t} \frac{\|\boldsymbol{y}_{S^{i-1}}\|_1}{k} \geq d(1-2\epsilon)\|\boldsymbol{y}_S\|_1 - 2d\epsilon\|\boldsymbol{y}\|_1.
\end{aligned}$$

Hence

$$\|\boldsymbol{\Phi y}\|_1 + 2d\epsilon\|\boldsymbol{y}\|_1 \geq (1-2\epsilon)d\|\boldsymbol{y}_S\|_1. \qquad (9.1.2)$$

Now, from the triangle inequality we have

$$\begin{aligned}
\|\boldsymbol{u}\|_1 &\geq \|\boldsymbol{v}\|_1 - \Delta \geq \|\boldsymbol{u}\|_1 - \Delta - \|\boldsymbol{u} - \boldsymbol{v}\|_1 \qquad (9.1.3) \\
&= \|\boldsymbol{u}\|_1 - \Delta + \|\boldsymbol{u} - \boldsymbol{v}\|_1 - 2\left(\|(\boldsymbol{u} - \boldsymbol{v})_S\|_1 + \|(\boldsymbol{u} - \boldsymbol{v})_{\overline{S}}\|_1\right) \\
&\geq \|\boldsymbol{u}\|_1 - \Delta + \|\boldsymbol{u} - \boldsymbol{v}\|_1 - 2\left(\|(\boldsymbol{u} - \boldsymbol{v})_S\|_1 + \|\boldsymbol{u} - \mathrm{H}_k(\boldsymbol{u})\|_1\right).
\end{aligned}$$

Therefore, from Equation (9.1.2), we obtain

$$\|\boldsymbol{u}\|_1 \geq \|\boldsymbol{u}\|_1 - 2\|\boldsymbol{u} - \mathrm{H}_k(\boldsymbol{u})\|_1 - \Delta + \|\boldsymbol{u} - \boldsymbol{v}\|_1 - \frac{2\|\boldsymbol{\Phi u} - \boldsymbol{\Phi v}\|_1 + 4d\epsilon\|\boldsymbol{u} - \boldsymbol{v}\|_1}{(1-2\epsilon)d}.$$

Rearranging the inequality completes the proof. □

Using Theorem 9.1, Berinde et al. have shown that the RIP-1 property is sufficient for sparse recovery using ℓ_1 minimization [29].

Theorem 9.2. *Let ϵ be a positive number smaller than $1/6$, and let $\boldsymbol{\Phi}$ be a sensing matrix satisfying $(2k, \epsilon, d)$-RIP1. Let $\boldsymbol{\alpha}^*$ be any arbitrary vector in $\mathbb{R}^N$, and let $\mathrm{H}_k(\boldsymbol{\alpha}^*)$ denote the best k-term approximation of $\boldsymbol{\alpha}^*$ defined by Equation (2.1.1). Finally let $\boldsymbol{e_M}$ be an arbitrary noise vector in $\mathbb{R}^M$, and let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$. Then the solution $\hat{\boldsymbol{\alpha}}$ of the Basis Pursuit problem*

$$\begin{aligned} &\textit{minimize } \|\boldsymbol{\alpha}'\|_1 \\ &\textit{subject to } \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}'\|_1 \leq \|\boldsymbol{e_M}\|_1, \end{aligned}$$

satisfies the following ℓ_1/ℓ_1 sparse approximation guarantee:

$$\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_1 \leq c_1 \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + c_2 \|\boldsymbol{e_M}\|_2, \tag{9.1.4}$$

with $c_1 = \frac{2(1-2\epsilon)}{(1-6\epsilon)}$, and $c_2 = \frac{2}{(1-6\epsilon)d}$.

The Optimization-based method of Theorem 9.2 exploits the *geometry* of the expander graphs, and performs significantly better than the combinatorial approach in practical applications. Figure 9.1 compares the performance of the geometric BP algorithm and the combinatorial SSMP algorithm. This comparison is borrowed from experiments performed by Berinde and Indyk [30], and shows that the ℓ_1 minimization method can recover signals with significantly higher sparsity level than the SSMP algorithm. For instance, about 1000 measurements are sufficient to recover a 100-sparse signals of dimensions $20,000$ using the BP algorithm, whereas at least 2200 measurements are required for the successful recovery using the SSMP algorithm.

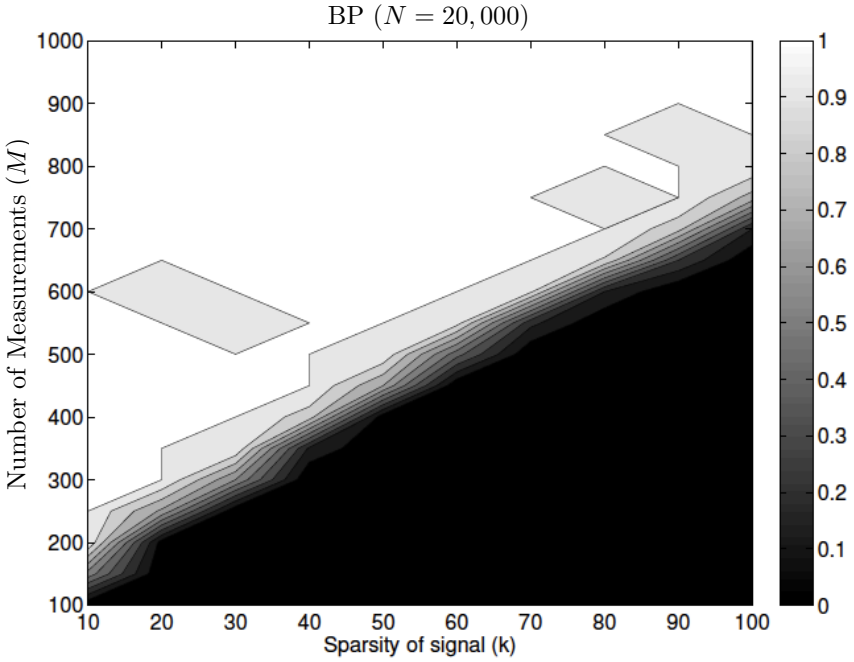
Even though the ℓ_1-minimization approach has the best practical performance, the computational cost of solving sparse recovery using the interior point method is typically $O(N^{1.5}M^2)$. Moreover, since the feasible set $\|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1$ is not even differentiable, most gradient-based optimization methods are not directly applicable. In the next section we propose an alternative efficient algorithm that *approximately* solves the objective of the geometric approach.

9.2 Sparse Approximation in ℓ_1 Norm

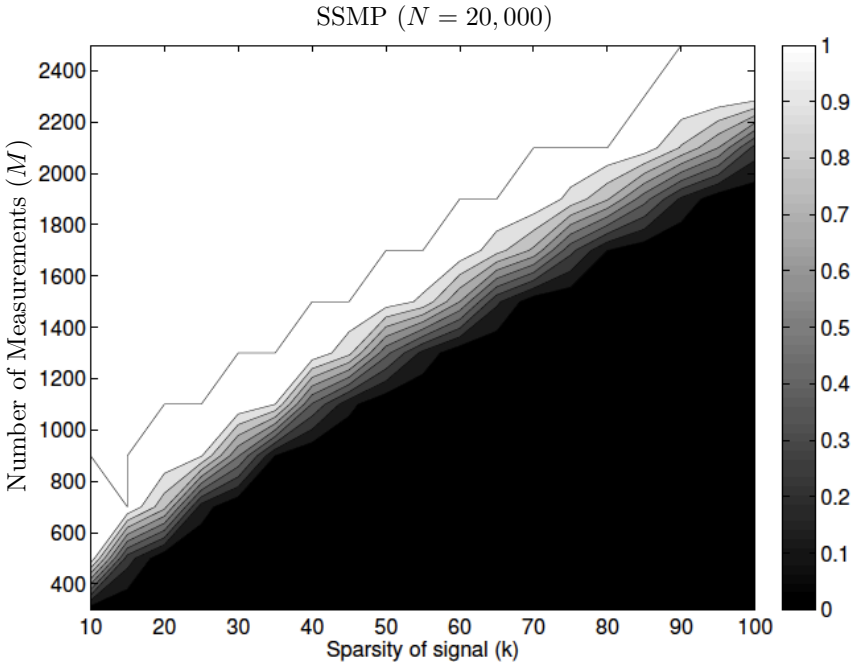
In this section, we propose the ℓ_1-GAME algorithm for solving the problem of sparse approximation in the ℓ_1-norm. The ℓ_1-GAME algorithm is a special instance of the GAME algorithm introduced in Section 7.2. Later on, in Section 9.3 we will use the ℓ_1-GAME algorithm to approximately solve the non-smooth ℓ_1-minimization of Theorem 9.2.

Let $\boldsymbol{\Phi}$ be any $M \times N$ matrix with $M \ll N$, let $\boldsymbol{\alpha}^*$ be a sparse vector in $\Delta(k, \tau)$, defined by Equation (6.1.2),and let $\boldsymbol{e_M}$ be any vector in $\mathbb{R}^M$. Let $\boldsymbol{f} \doteq \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$ denote the measurement vector. Sparse approximation in the ℓ_1-norm refers to the following problem

$$\text{minimize}_{\boldsymbol{\alpha} \in \Delta(k,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1. \tag{9.2.1}$$



(a) $M = 1000$ measurements are sufficient for successful recovery of 100-sparse signals using the Basis Pursuit algorithm.



(b) $M = 2200$ measurements are necessary for successful recovery of 100-sparse signals using the SSMP algorithm.

Figure 9.1: Comparisons of exact recovery experiments for SSMP and BP algorithms as provided in [30]. All plots are for the same signal length $N = 20000$, and left degree $d = 8$. Each experiment is repeated independently 100 times.

Algorithm 6 The ℓ_1-GAME algorithm

Inputs: $\boldsymbol{f}$, $\boldsymbol{\Phi}$, and parameters T, τ and $\eta > 0$.
Output: A T-sparse vector $\hat{\boldsymbol{\alpha}}$.

1: Set $\mathbf{P}^1 = [\mathbf{0}]_M$.
2: **for** $t = 1, \cdots, T$ **do**
3: Let $\boldsymbol{r^t} \doteq \boldsymbol{\Phi}^\top \mathbf{P^t}$
4: Find the index i of one largest (in magnitude) element of $\boldsymbol{r^t}$.
5: Let $\boldsymbol{\alpha^t}$ be a 1-sparse vector with

$$\mathrm{Supp}(\boldsymbol{\alpha}^t) = \{i\}, \text{and } \alpha_i^t = -\tau \,\mathrm{Sign}\left(r_i^t\right).$$

6: Update $\mathbf{Q}^{t+1} = \mathbf{P}^t + \frac{\eta}{2}(\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f})$.
7: For each $j \in [M]$, let $\mathrm{P}_j^{t+1} = \mathrm{S}\left(\mathrm{Q}_j^{t+1}, 1\right)$.
($\mathrm{S}\left(\mathrm{Q}_j^{t+1}, 1\right)$ uses the soft-thresholding operator of Definition 2.3 with $\theta = 1$).
8: **end for**
9: Output $\hat{\boldsymbol{\alpha}} \doteq \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^t$.

Unfortunately, since $\Delta(\tau, k)$ is not sparse, solving the problem of Equation (9.2.1) is intractable. However, this optimization problem is equivalent to sparse approximation in the ℓ_q norm with $q = 1$. Therefore the results of Chapter 6.1 can be applied to approximately solve Equation (9.2.1) efficiently.

Remark 9.3. *Note that throughout this section we assume that an upper-bound τ on the ℓ_1-norm of $\boldsymbol{\alpha^*}$ is known a priori. While this assumption is directly valid in many applications, we will still provide a way to efficiently compute an estimate in the expander-based compressed sensing problem in Section 9.3.*

Following Equation (6.1.6), let

$$\Xi_\infty \doteq [1, 1]^M = \{\mathbf{P} \in \mathbb{R}^M : \|\mathbf{P}\|_\infty \leq 1\},$$

and let $\Delta(\tau)$ be as in Equation (6.1.1). Define the loss function $\mathcal{L} : \Xi_\infty \times \Delta(\tau) \to \mathbb{R}$ as

$$\mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}) \doteq \langle \mathbf{P}, (\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f})\rangle.$$

It follows from Equation (6.1.10) with $q = 1$ and $p = \infty$ that the problem of sparse approximation in the ℓ_1 norm can be viewed as a min-max game:

$$\min_{\boldsymbol{\alpha} \in \Delta(k,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1 = \min_{\boldsymbol{\alpha} \in \Delta(k,\tau)} \max_{\mathbf{P} \in \Xi_\infty} \mathcal{L}(\mathbf{P}, \boldsymbol{\alpha}). \tag{9.2.2}$$

Since sparse approximation in the ℓ_1 norm is a special case of sparse approximation in the ℓ_q norm, the GAME Algorithm (Algorithm 3) can be used to approximate the

optimal solution of Equation (9.2.1). In order to obtain guarantees on the performance of the GAME algorithm, Theorem 9.4 requires that the Bregman function is properly chosen so that

$$\forall \mathbf{P}, \mathbf{Q} \in \Xi_\infty,\ \mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{Q}) \geq \|\mathbf{P} - \mathbf{Q}\|_\infty^2.$$

It is easy to verify that this requirement is satisfied if the squared Euclidean norm $\mathcal{R}(\mathbf{P}) = \|\mathbf{P}\|_2^2$, with $\mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{Q}) = \|\mathbf{P} - \mathbf{Q}\|_2^2$ is used in the GAME algorithm. The pseudocode of the ℓ_1-GAME algorithm is provided in Algorithm 6, and describes a special GAME Algorithm which exploits the choice of Euclidean distance as the Bregman function.

Other choices for the Bregman function may lead to different convergence bounds and different running times for the new projections and updates. For instance, a multiplicative update version of the algorithm (MU-GAME) can be derived by using the Bregman divergence based on the relative entropy function. Surprisingly, the derived guarantees for GAME can be shown to also hold for MU-GAME in a straightforward manner.

The general ℓ_1-GAME algorithm starts by finding a $\mathbf{P}^1$ such that $\nabla\mathcal{R}(\mathbf{P}^1) = 2\mathbf{P}^1 = \mathbf{0}_M$. Then at every iteration, in step 6, the algorithm finds a $\mathbf{Q}^{t+1}$ with

$$\nabla\left(\mathcal{B}_\mathcal{R}(\mathbf{Q}^{t+1}, \mathbf{P}^t) - \eta\langle\mathbf{Q}^{t+1}, (\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f})\rangle\right) = 2(\mathbf{Q}^{t+1} - \mathbf{P}^t) - \eta(\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}) = \mathbf{0}_M,$$

and then updates $\mathbf{P}^{t+1}$ via the Bregman projection

$$\mathbf{P}^{t+1} \doteq \arg\min_{\mathbf{P}\in[-1,1]^M} \mathcal{B}_\mathcal{R}(\mathbf{P}, \mathbf{Q}^{t+1}) = \mathrm{S}\left(\mathbf{Q}^{t+1}, 1\right).$$

The theorem below is based on Theorem 9.4 in Section 7.3, and shows that for every positive ε, as long as $T = O\left(\frac{M\tau^2}{(\varepsilon\|\boldsymbol{e_M}\|_1)^2}\right)$, the GAME algorithm after T iterations finds a T-sparse vector $\hat{\boldsymbol{\alpha}}$ with $\varepsilon\|\boldsymbol{e_M}\|_2$ multiplicative approximation error in the measurement domain.

Theorem 9.4. *Let $\boldsymbol{\Phi}$ be an $M \times N$ matrix, and let $\|\boldsymbol{\Phi}\|_1$ denote the ℓ_1 norm of the matrix, which is defined in Equation 2.2.1. Suppose $\boldsymbol{\alpha^*}$ is a vector in $\Delta(k, \tau)$, and let $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha^*} + \boldsymbol{e_M}$, where $\boldsymbol{e_M}$ is the measurement noise vector. Let ε be any number in $(0, 1]$, and let $\hat{\boldsymbol{\alpha}}$ denote the output of the* GAME *algorithm after*

$$T = \max\left\{k, M\left(\frac{\|\boldsymbol{e_M}\|_1 + 2\|\boldsymbol{\Phi}\|_1\tau}{2\varepsilon\|\boldsymbol{e_M}\|_1}\right)^2\right\} \tag{9.2.3}$$

iterations with regularization parameter

$$\eta = \frac{4\varepsilon\|\boldsymbol{e_M}\|_1}{\left(\|\boldsymbol{e_M}\|_1 + 2\|\boldsymbol{\Phi}\|_1\tau\right)^2}.$$

Then, $\hat{\boldsymbol{\alpha}}$ is a vector in $\Delta(\tau, T)$ with

$$\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_1 \leq (1 + \varepsilon)\|\boldsymbol{e_M}\|_1. \tag{9.2.4}$$

Proof. At every iteration t, $\boldsymbol{\alpha}^t$ is a 1-sparse solution of the minimization problem minimize$\mathcal{L}(\mathbf{P}^t, \boldsymbol{\alpha})$. Moreover, for every $\boldsymbol{\alpha}$ in $\Delta(1, \tau)$ we have

$$\|\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}\|_2 \leq \|\boldsymbol{\Phi}\boldsymbol{\alpha}^t - \boldsymbol{f}\|_1 \leq \left(\|\boldsymbol{e_M}\|_1 + \|\boldsymbol{\Phi}\|_1 \|\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^*\|_1\right) \leq \|\boldsymbol{e_M}\|_1 + 2\|\boldsymbol{\Phi}\|_1 \tau.$$

Moreover, for every $\mathbf{P} \in \Xi_\infty$ we have

$$\mathcal{B}_{\mathcal{R}}(\mathbf{P}, \mathbf{P}^1) = \|\mathbf{P}\|_2^2 = \sum_{j=1}^{M} |P_j|^2 \leq M.$$

As a result, by setting $\mathrm{G} = \|\boldsymbol{e_M}\|_1 + 2\|\boldsymbol{\Phi}\|_1 \tau$, and $\mathrm{D} = \sqrt{M}$, from Theorem 9.4 it follows that $\hat{\boldsymbol{\alpha}}$ is a vector in $\Delta(T, \tau)$ with

$$\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_1 \leq \min_{\boldsymbol{\alpha} \in \Delta(k,\tau)} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1 + \frac{\mathrm{GD}}{2\sqrt{T}} \leq \|\boldsymbol{e_M}\|_1 + \varepsilon\|\boldsymbol{e_M}\|_1 = (1+\varepsilon)\|\boldsymbol{e_M}\|_1.$$

□

Remark 9.5. *In this section, we assumed that the vector $\boldsymbol{\alpha}^*$ is exactly k-sparse. However, this assumption is without loss of generality. To see this, observe that every vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ satisfies*

$$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M} = \boldsymbol{\Phi}\mathrm{H}_k(\boldsymbol{\alpha}^*) + \left[\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)) + \boldsymbol{e_M}\right],$$

and

$$\|\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*))\|_1 \leq \|\boldsymbol{\Phi}\|_1 \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1.$$

Therefore, one can always assume that the original vector is exactly k-sparse, by assuming that the measurement noise is $\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)) + \boldsymbol{e_M}$, with

$$\|\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)) + \boldsymbol{e_M}\|_1 \leq \|\boldsymbol{\Phi}\|_1 \|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \|\boldsymbol{e_M}\|_1.$$

In particular, if $\boldsymbol{\Phi}$ is the adjacency of an expander graph, then $\|\boldsymbol{\Phi}\|_1 = d$ and

$$\|\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)) + \boldsymbol{e_M}\|_1 \leq d\,\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \|\boldsymbol{e_M}\|_1. \tag{9.2.5}$$

9.3 Expander-based GAME Algorithm

In Section 9.2 we showed that if $\boldsymbol{\Phi}$ is the adjacency of an expander graph, then for every vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$, given $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$, the ℓ_1-GAME algorithm can efficiently find a vector $\hat{\boldsymbol{\alpha}}$ with bounded measurement-domain error

$$\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_1 = O\left(d\,\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \|\boldsymbol{e_M}\|_1\right).$$

In this section, we combine the results of Section 9.2 and Theorem 9.2, and propose an efficient algorithm, called e-GAME , that finds an estimate $\hat{\boldsymbol{\alpha}}$ with the ℓ_1/ℓ_1 guarantee $\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_1 = O(\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d})$.

Similar to the previous section, throughout this section without loss of generality we assume that the vector $\boldsymbol{\alpha}^*$ is exactly k-sparse by adding the residual term $\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*))$ to the measurement noise.

The pseudocode of the e-GAME algorithm is shown in Algorithm 7. The following lemma is key in establishing the guarantees of e-GAME .

Lemma 9.6. *Let $\boldsymbol{\Phi}$ be the adjacency of a (k, ϵ, d) expander graph with $\|\boldsymbol{\Phi}\|_1 = d$. Let $\langle \hat{\boldsymbol{\alpha}}^1, \cdots, \hat{\boldsymbol{\alpha}}^\Theta \rangle$ be the vectors generated by the* e-GAME *algorithm. Then at least one of the following two conditions holds. That is, either*

(C1). there exists an index t with $\|\hat{\boldsymbol{\alpha}}^t\|_1 \leq \|\boldsymbol{\alpha}^*\|_1$ and $\|\boldsymbol{\Phi}(\hat{\boldsymbol{\alpha}}^t - \boldsymbol{\alpha}^*)\|_1 \leq (2+\varepsilon)\|\boldsymbol{e_M}\|_1$; or

(C2). for every iteration t, $\mathrm{Lo}^t \leq \|\boldsymbol{\alpha}^*\|_1 \leq \mathrm{Up}^t$.

Proof. We prove Lemma 9.6 by induction. First consider $t = 0$. Since $\boldsymbol{\alpha}^*$ is k-sparse, it follows from the RIP-1 property of expander graphs (Lemma 8.8), and the triangle inequality that

$$(1 - 2\epsilon)d\|\boldsymbol{\alpha}^*\|_1 \leq \|\boldsymbol{\Phi}\boldsymbol{\alpha}^*\|_1 = \|\boldsymbol{f} - \boldsymbol{e_M}\|_1 \leq \|\boldsymbol{f}\|_1 + \|\boldsymbol{e_M}\|_1.$$

Assume that Condition (C2) holds for $t - 1$, we now show that it is also valid for index t via two different cases:

Case 1: $\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}\|_1 > (1 + \varepsilon)\|\boldsymbol{e_M}\|_1$. If $\|\boldsymbol{\alpha}^*\|_1 \leq \tau^t$ then

$$\min_{\|\boldsymbol{\alpha}\|_1 \leq \tau^t} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1 \leq \|\boldsymbol{\Phi}\boldsymbol{\alpha}^* - \boldsymbol{f}\|_1 = \|\boldsymbol{e_M}\|_1 < \frac{\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}\|_1}{(1 + \varepsilon)},$$

which contradicts the $(1 + \varepsilon)$ approximation guarantee of the ℓ_1-GAME algorithm. Therefore we must have $\|\boldsymbol{\alpha}^*\|_1 \geq \tau^t = \mathrm{Lo}^t$. It also follows from the induction hypothesis that $\|\boldsymbol{\alpha}^*\|_1 \leq \mathrm{Up}^{t-1} = \mathrm{Up}^t$.

Case 2: $\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}\|_1 \leq (1 + \varepsilon)\|\boldsymbol{e_M}\|_1$. In this case, if $\tau^t \leq \|\boldsymbol{\alpha}^*\|_1$, then we have $\|\hat{\boldsymbol{\alpha}}^t\|_1 \leq \|\boldsymbol{\alpha}^*\|_1$ and

$$\|\boldsymbol{\Phi}(\hat{\boldsymbol{\alpha}}^t - \boldsymbol{\alpha}^*)\|_1 \leq \|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}\|_1 + \|\boldsymbol{\Phi}\boldsymbol{\alpha}^* - \boldsymbol{f}\|_1 \leq (2 + \varepsilon)\|\boldsymbol{e_M}\|_1,$$

which is Condition (C1). Therefore, if (C1) is not valid then we must have $\mathrm{Up}^t = \tau^t > \|\boldsymbol{\alpha}^*\|_1$. Also again from the induction hypothesis we get $\|\boldsymbol{\alpha}^*\|_1 \geq \mathrm{Lo}^{t-1} = \mathrm{Lo}^t$. □

The following theorem proves that at least one estimate $\hat{\boldsymbol{\alpha}}^t$ is sufficiently close to $\boldsymbol{\alpha}^*$.

Algorithm 7 The e-GAME Algorithm

Inputs: $\boldsymbol{f}$, $\boldsymbol{\Phi}$, and parameters ε, and δ.
Output: An approximation $\hat{\boldsymbol{\alpha}}$ for the vector $\boldsymbol{\alpha}^*$.

Set $\mathrm{Lo}^0 = 0$, $\mathrm{Up}^0 = \frac{\|\boldsymbol{f}\|_1 + \|\boldsymbol{e_M}\|_1}{d(1-2\epsilon)}$, and $\Theta = \log\left(\frac{4}{\delta}\right)$.
for $t = 1, \cdots, \Theta$ **do**
  Let $\hat{\boldsymbol{\alpha}}^t$ be the solution of the ℓ_1-GAME algorithm with $\tau^t = \frac{\mathrm{Lo}^{t-1} + \mathrm{Up}^{t-1}}{2}$ and T^t of Equation (9.2.3).
  if $\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^t - \boldsymbol{f}\|_1 \leq (1+\varepsilon)\|\boldsymbol{e_M}\|_1$ **then**
    Set $\mathrm{Up}^t = \tau^t$, and $\mathrm{Lo}^t = \mathrm{Lo}^{t-1}$.
  else
    Set $\mathrm{Lo}^t = \tau^t$, and $\mathrm{Up}^t = \mathrm{Up}^{t-1}$.
  end if
end for
Output $\hat{\boldsymbol{\alpha}} \doteq \arg\min_{t\in[\Theta]} \|\boldsymbol{\Phi}\mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t) - \boldsymbol{f}\|_1$.

Theorem 9.7. *Let $\boldsymbol{\Phi}$ be the adjacency of a $(2k, \epsilon, d)$ expander graph. Let ε and δ be any two positive numbers, and let $\langle \hat{\boldsymbol{\alpha}}^1, \cdots, \hat{\boldsymbol{\alpha}}^\Theta \rangle$ be the vectors generated by the* e-GAME *algorithm. Then there exists an index t with*

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}^t\|_1 \leq \frac{(1-2\epsilon)\delta\|\boldsymbol{\alpha}^*\|_1}{(1-6\epsilon)} + \frac{2(2+\varepsilon)\|\boldsymbol{e_M}\|_1}{(1-6\epsilon)d}.$$

Proof. The proof of Theorem 9.7 relies on Lemma 9.6. If Condition (C1) is satisfied for some index t, then Theorem 9.1 implies that

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}^t\|_1 \leq \frac{2(2+\varepsilon)\|\boldsymbol{e_M}\|_1}{(1-6\epsilon)d},$$

whereas if Condition (C2) is satisfied, then at every iteration t we have

$$\|\hat{\boldsymbol{\alpha}}^t\|_1 - \|\boldsymbol{\alpha}^*\|_1 \leq \mathrm{Up}^t - \|\boldsymbol{\alpha}^*\|_1 \leq \mathrm{Up}^t - \mathrm{Lo}^t \leq \frac{\mathrm{Up}^0 - \mathrm{Lo}^0}{2^t}.$$

In this case, if $\Theta = \log_2\left(\frac{4}{\delta}\right)$, then Lemma 8.8 implies that

$$\begin{aligned} \|\hat{\boldsymbol{\alpha}}^\Theta\|_1 - \|\boldsymbol{\alpha}^*\|_1 &\leq \frac{\mathrm{Up}^0 - \mathrm{Lo}^0}{2^\theta} = \frac{(\|\boldsymbol{f}\|_1 + \|\boldsymbol{e_M}\|_1)\delta}{4d(1-2\epsilon)} \\ &\leq \frac{(\|\boldsymbol{f}\|_1 - \|\boldsymbol{e_M}\|_1)\delta}{d} \leq \frac{\|\boldsymbol{\Phi}\boldsymbol{\alpha}^*\|_1\delta}{d} \leq \|\boldsymbol{\alpha}^*\|_1\delta. \end{aligned} \tag{9.3.1}$$

Furthermore, since both $\|\boldsymbol{\alpha}^*\|_1$ and $\|\hat{\boldsymbol{\alpha}}\|_1$ are smaller than τ^Θ, Theorem 9.4 guarantees that

$$\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}^\Theta - \boldsymbol{f}\|_1 \leq (1+\varepsilon) \min_{\|\boldsymbol{\alpha}\|_1 \leq \tau^\Theta} \|\boldsymbol{\Phi}\boldsymbol{\alpha} - \boldsymbol{f}\|_1 \leq (1+\varepsilon)\|\boldsymbol{e_M}\|_1.$$

Therefore, from Theorem 9.1 we get

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}^{\Theta}\|_1 \leq \frac{(1-2\epsilon)\delta\|\boldsymbol{\alpha}^*\|_1}{(1-6\epsilon)} + \frac{2(2+\varepsilon)\|\boldsymbol{e_M}\|_1}{(1-6\epsilon)d}.$$

□

So far, we have shown that at least one of the estimates $\langle \hat{\boldsymbol{\alpha}}^1, \cdots, \hat{\boldsymbol{\alpha}}^{\theta} \rangle$ is sufficiently close to $\boldsymbol{\alpha}^*$. However, since $\boldsymbol{\alpha}^*$ is not known *a priori*, we cannot directly estimate which $\hat{\boldsymbol{\alpha}}^t$ is close enough to $\boldsymbol{\alpha}^*$. Fortunately, the RIP-1 property of the expander graphs allows us to use the measurement domain accuracy as a proxy to measure the data-domain accuracy of the estimates. More precisely, we show that the estimate

$$\hat{\boldsymbol{\alpha}} \doteq \arg \min_{t \in \{1, \cdots, \Theta\}} \|\boldsymbol{\Phi} \mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t) - \boldsymbol{f}\|_1$$

is sufficiently close to $\boldsymbol{\alpha}^*$

Theorem 9.8. *Let $\boldsymbol{\Phi}$ be the adjacency of a $(2k, \epsilon, d)$ expander graph. Let ε and δ be any two positive numbers, and let $\hat{\boldsymbol{\alpha}}$ be the output of the* e-GAME *algorithm. Then*

$$\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}})\|_1 \leq \frac{2\delta\|\boldsymbol{\alpha}^*\|_1}{(1-6\epsilon)} + \frac{1}{(1-2\epsilon)}\left(2 + \frac{4(2+\varepsilon)}{(1-6\epsilon)}\right)\frac{\|\boldsymbol{e_M}\|_1}{d}.$$

Proof. Let $\boldsymbol{b}$ be any vector in $\mathbb{R}^N$. Since $\boldsymbol{\alpha}^*$ is k-sparse, from the triangle inequality and the definition of the best k-term approximation we have

$$\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{b})\|_1 \leq \|\boldsymbol{\alpha}^* - \boldsymbol{b}\|_1 + \|\boldsymbol{b} - \mathrm{H}_k(\boldsymbol{b})\|_1 \leq 2\|\boldsymbol{\alpha}^* - \boldsymbol{b}\|_1.$$

Now, observe that for every t, $\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t)$ is always $2k$-sparse. As a result, Lemma 8.8 yields that at every iteration t

$$\begin{aligned} &\|\boldsymbol{f} - \boldsymbol{\Phi}\mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t)\|_1 \leq \|\boldsymbol{e_M}\|_1 + \|\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t))\|_1 \\ &\leq \|\boldsymbol{e_M}\|_1 + d\|(\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}}^t))\|_1 \leq \|\boldsymbol{e_M}\|_1 + 2d\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}^t\|_1. \end{aligned} \tag{9.3.2}$$

Therefore, it follows from Theorem 9.7 that

$$\|\boldsymbol{f} - \boldsymbol{\Phi}\mathrm{H}_k(\hat{\boldsymbol{\alpha}})\|_1 \leq \frac{2d(1-2\epsilon)\delta\|\boldsymbol{\alpha}^*\|_1}{(1-6\epsilon)} + \left(1 + \frac{4(2+\varepsilon)}{(1-6\epsilon)}\right)\|\boldsymbol{e_M}\|_1. \tag{9.3.3}$$

Moreover, since $\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}})$ is $2k$-sparse, the RIP1 property implies that

$$\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}})\|_1 \leq \frac{\|\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\hat{\boldsymbol{\alpha}})\|_1}{(1-2\epsilon)d} \leq \frac{\|\boldsymbol{f} - \boldsymbol{\Phi}\mathrm{H}_k(\hat{\boldsymbol{\alpha}})\|_1 + \|\boldsymbol{e_M}\|_1}{(1-2\epsilon)d},$$

which completes the proof. □

The following corollary is a direct consequence of Theorem 9.8.

Corollary 9.9. *Let $\boldsymbol{\Phi}$ be the adjacency of a $(2k, \epsilon, d)$ expander graph, where ϵ is a constant less than $\frac{1}{6}$. Let $\boldsymbol{\alpha}^*$ be any vector in $\mathbb{R}^N$, and let $\boldsymbol{e_M}$ be any noise vector in $\mathbb{R}^M$. Let*

$$\mathrm{SNR}_1 \doteq \frac{\|\mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1}{\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d}}. \tag{9.3.4}$$

Then the e-GAME *algorithm with $\delta = \frac{1}{\mathrm{SNR}_1}$ and $\varepsilon = 1$ recovers a vector $\hat{\boldsymbol{\alpha}}$ with*

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1 = \mathcal{O}\left(\|\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)\|_1 + \frac{\|\boldsymbol{e_M}\|_1}{d}\right).$$

Moreover, the overall recovery time is $O\left(MN\,d\,\mathrm{SNR}_1^2 \log \mathrm{SNR}_1\right)$.

Proof. By treating $\boldsymbol{\Phi}(\boldsymbol{\alpha}^* - \mathrm{H}_k(\boldsymbol{\alpha}^*)) + \boldsymbol{e_M}$ as the measurement domain noise, we can always without loss of generality assume that $\boldsymbol{\alpha}^*$ is exactly k-sparse. The data-domain sparse approximation bound then follows from Theorem 9.8 by setting $\delta = \frac{1}{\mathrm{SNR}_1}$. and $\varepsilon = 1$.

To calculate the overall running time of the algorithm note that the e-GAME algorithm requires $\Theta = \mathcal{O}(\log \mathrm{SNR}_1)$ iterations. At each such iteration, the ℓ_1-GAME algorithm requires $T = \mathcal{O}(M\mathrm{SNR}_1^2)$ iterations (Equation (9.2.3)) in which the bottleneck is one matrix-vector multiplication (i.e., calculating $\boldsymbol{\Phi}^\top \mathbf{P}^t$). This multiplication can be calculated efficiently using $\mathcal{O}(N\,d)$ operations as the graph is d regular. □

Remark 9.10. *An alternative approach is to use Nesterov's smoothing method for approximately solving non-smooth objective functions [204, 203]. We omit the details of this implementation. With the Nesterov method, we still need $\mathcal{O}(\log \mathrm{SNR}_1)$ outer iterations, while the number of inner iterations can be reduced to $T = \mathcal{O}(M\mathrm{SNR}_1)$. However, each inner iteration of the Nesterov method requires solving three smooth convex optimization problems, and is much more complicated than calculating one matrix-vector multiplication.*

Tables 9.1 and 9.2 compare different properties of various sparse recovery algorithms which use sparse matrices constructed from the adjacencies of bipartite graphs. The algorithms are categorized as either *combinatorial*, which exploit the combinatorial properties (e.g. the unique neighborhood property) of the graph, or *geometric*, which exploit the geometric properties (e.g. the RIP-1 property) of that graph. The geometric algorithms are often capable of recovering signals with higher sparsity level, whereas the combinatorial algorithms are computationally more efficient.

The *for all* sparse recovery model corresponds to recovery algorithms that (surely or with high probability) provide a close sparse approximation to every vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$, whereas the *for each* model focuses on the recovery algorithms that can recover a sparse approximation to each vector $\boldsymbol{\alpha}^*$ with high probability. The *almost sparse*

model assumes that the vector has k-significant entries and the remaining entries are sufficiently close to zero. The *positive* model corresponds to recovery of sparse vectors with non-negative entries.

9.4 Experimental Results

In this section, we provide experimental results to empirically investigate the fidelity of the algorithms proposed in this chapter. Throughout these experiments we used $N = 1000$, $M = 200$ and $k = 20$ for illustration to demonstrate the typical behavior of the algorithms for other N, M and k. We first generated a 200×1000 random expander matrix $\mathbf{\Phi}$, and then repeated the following experiment 100 times. We generated a sparse vector with random support, random sign, and unit magnitudes, generated compressive measurements, and then recovered a sparse estimate for the original signal.

Figure 9.2(a) plots the measurement-domain error of the ℓ_1-GAME algorithm as a function of the number of iterations. Here we let the algorithm continue for $100,000$ iterations. The Figure shows that with ℓ_1-GAME , the measurement-domain error consistently decreases. Moreover, after some initial burn-in, the rates of convergence are approximately $\frac{1}{T}$ (as opposed to the slower rate of$\frac{1}{\sqrt{T}}$, which was expected from theory). The $\frac{1}{T}$ rate of convergence matches the best known first-order optimization results [203].

Figure 9.2(b) compares the performance of the e-GAME algorithm with Basis Pursuit algorithm, and with SSMP [30] algorithm in terms of their stability against the measurement noise. As above, we set $N = 1000$, $M = 200$ and $k = 20$, and repeated each experiment independently 100 times. The signal is generated in the same process as above, and we used white Gaussian measurement noise with standard deviation ranging from 10^{-5} to 10^{-2}.

In this experiment we compared the average reconstruction error $\frac{\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1}{\|\boldsymbol{\alpha}^*\|_1}$ of the three algorithms above as a function of the noise level. We used the CVX package [134, 133] which is a standard convex optimization package, for directly solving the Basis Pursuit algorithm of Theorem 9.2. We also used the SSMP code provided by Berinde and Indyk [30] with 40 outer iterations, 100 inner iterations, and threshold 25 to solve the SSMP optimization [1].

It is interesting that the approximation error of the e-GAME algorithm is very close to the approximation error of the Basis Pursuit algorithm, and significantly lower than the error of the SSMP algorithm. This experiment, and many other similar experiments confirm the advantages of the geometric reconstruction algorithms over

[1] We observed that if the SSMP algorithm is provided exactly with the sparsity level $k = 20$ then the algorithm has a much better performance; however, the performance of the SSMP algorithm is significantly decreased even if the threshold is slightly higher than the true sparsity level.

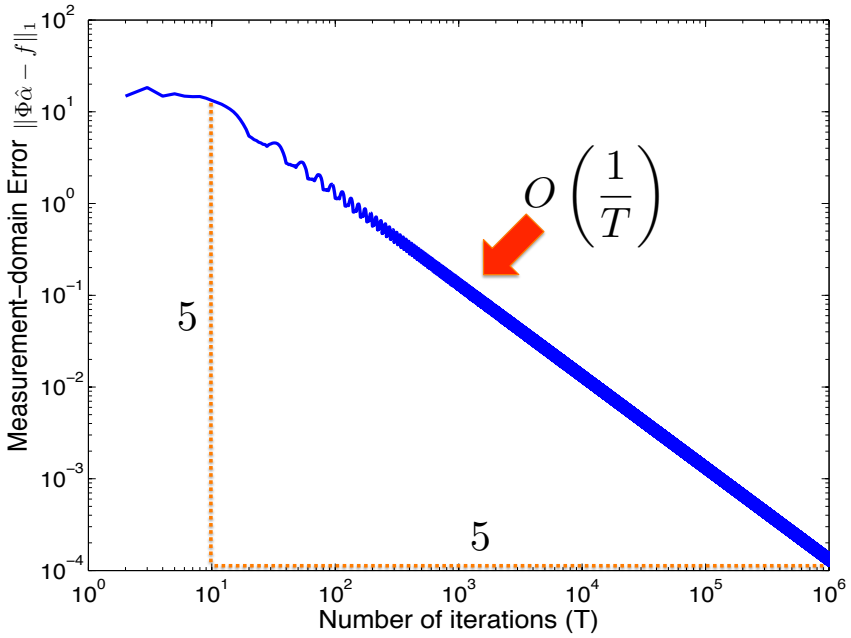
Table 9.1: Summary of sparse recovery algorithms that use geometric properties of matrices constructed from sparse bipartite graphs. All bounds ignore the $O()$ constants. β is a constant, possibly different in each row, and SNR_1 is defined by Equation (9.3.4). The rows of the table are sorted first by the signal model, then by the number of measurements, and finally by recovery time in a decreasing order.

Approach	Number of measurements	Decoding time	Signal Model	Noise Tolerance
Expander-codes Alg. 4 [159, 258, 259]	$k \log \frac{N}{k}$	$N \log \frac{N}{k}$	Almost sparse	No
Minimal Expansion [169]	$k \log \frac{N}{k}$	$N \log \frac{N}{k}$	Positive	Yes
Count-Min [78, 79, 72]	$k \log N$ $k \log^{\beta} N$	$N \log N$ $k \log^{\beta} N$	For each	No
LDPC-codes [125]	$k \log \frac{N}{k}$	$k \log^{\beta} N$	For each	Yes
SSMP [153, 31, 30]	$k \log \frac{N}{k}$	$N \log^2 \frac{N}{k} \log \mathrm{SNR}_1$	For all	Yes

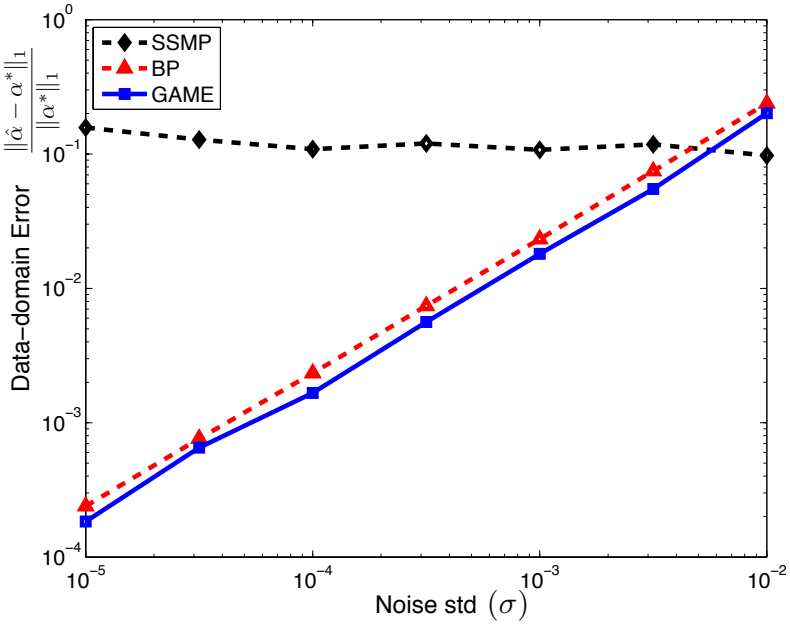
Table 9.2: Summary of sparse recovery algorithms that use combinatorial properties of matrices constructed from sparse bipartite graphs. All bounds ignore the $O()$ constants. β is a constant, possibly different in each row, and SNR_1 is defined by Equation (9.3.4). All algorithms provide guarantees in the *for all* signal model. The rows of the table are sorted first by the number of measurements, and then by recovery time in a decreasing order.

Approach	Number of measurements	Decoding time	Noise Tolerance
Expander-codes [137, 138]	$k(\log N)^{\beta}$	$N^{1.5} k^2 (\log N)^{2\beta}$	No
Basis Pursuit [29]	$k \log \frac{N}{k}$	$N^{1.5} k^2 \left(\log \frac{N}{k}\right)^2$	Yes
e-GAME Algorithm 7 [156]	$k \log \frac{N}{k}$	$kN \log \left(\frac{N}{k}\right)^2 \mathrm{SNR}_1^2 \log \mathrm{SNR}_1$	Yes

the combinatorial algorithms in terms of stable signal recovery in expander-based compressed sensing.



(a) The dependency between the measurement-domain error $\|\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} - \boldsymbol{f}\|_1$ and the number of iterations of the ℓ_1-GAME Algorithm. The empirical rate of convergence is approximately $\frac{1}{T}$ (as opposed to slower rate $\frac{1}{\sqrt{T}}$ expected from theory).



(b) Approximate recovery experiments with SSMP, Basis Pursuit, and e-GAME algorithms for expander-based compressed sensing. The measurement noise standard deviation ranges from 10^{-5} to 10^{-2}, and the approximation error is measured as $\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1 / \|\boldsymbol{\alpha}^*\|_1$.

Figure 9.2: Empirical performance of the e-GAME algorithm. Here we fixed $N = 1000$, $M = 200$ and $k = 20$, and repeated each experiment independently 100 times using a random expander graph with left-degree $d = 8$.

Chapter 10

Expander-based Compressed Sensing in the Presence of Poisson Noise

10.1 Introduction

In Chapter 4 we introduced different applications of compressed sensing, and showed that the compressed sensing framework is particularly appealing whenever the measurement is costly or constrained in some sense. For example, in the context of photon-limited applications (such as low-light imaging), the photo-multiplier tubes used within sensor arrays are physically large and expensive. Similarly, when measuring network traffic flows, the high-speed memory used in packet counters is cost-prohibitive [8]. These problems appear ripe for the application of CS.

However, photon-limited measurements [229] and arrivals/departures of packets at a router [33] are commonly modeled with a Poisson probability distribution, posing significant theoretical and practical challenges in the context of CS. One of the key challenges is the fact that the measurement error variance scales with the true intensity of each measurement, so that we cannot assume constant noise variance across the collection of measurements. Futhermore, measurements, underlying true intensities, and system models are all subject to certain physical constraints which play a significant role in performance.

Recent works [215, 155, 63, 177] explore methods for CS reconstruction in the presence of impulsive, sparse or exponential family noise, but do not account for the physical constraints associated with a typical Poisson setup and do not contain the related performance bounds emphasized in this chapter. In previous work [252, 210], Willett and Raginsky showed that a Poisson noise model combined with conventional dense CS sensing matrices (properly scaled) yielded performance bounds that were

somewhat sobering relative to bounds typically found in the literature. In particular, they found that if the number of photons (or packets) available to sense were held constant, and if the number of measurements M, was above some critical threshold, then larger M in general led to larger bounds on the error between the true and the estimated signals. This can intuitively be understood as resulting from the fact that dense CS measurements in the Poisson case cannot be zero-mean, and the DC offset used to ensure physical feasibility adversely impacts the noise variance.

The approach considered in this chapter hinges on reconstructing a signal from compressive measurements by optimizing a sparsity-regularized goodness-of-fit objective function. In contrast to many CS approaches, however, we measure the fit of an estimate to the data using the Poisson log-likelihood instead of a squared error term. This chapter demonstrates that the bounds developed in previous work can be improved for some sparsity models by considering alternatives to dense sensing matrices with random entries. In particular, we show that *sparse* sensing matrices given by scaled adjacency matrices of expander graphs have important theoretical characteristics that are ideally suited to controlling the performance of Poisson CS.

Formally, suppose we have a signal $\boldsymbol{\alpha}^* \in \mathbb{R}_+^N$ with known ℓ_1 norm $\|\boldsymbol{\alpha}^*\|_1$ (or a known upper bound on $\|\boldsymbol{\alpha}^*\|_1$). We aim to find a matrix $\boldsymbol{A} \in \mathbb{R}_+^{M \times N}$ with M, the number of measurements, as small as possible, so that $\boldsymbol{\alpha}^*$ can be recovered efficiently from the measured vector $\boldsymbol{f} \in \mathbb{R}_+^M$, which is related to $\boldsymbol{A}\boldsymbol{\alpha}^*$ through a Poisson observation model. The restriction that elements of $\boldsymbol{A}$ be nonnegative reflects the physical limitations of many sensing systems of interest (e.g., packet routers and counters or linear optical systems).

In Section 8.1 we introduced the adjacency matrices of expander graphs as an alternative to dense random matrices within the compressed sensing framework, leading to computationally efficient recovery algorithms. Subsequently, we saw that variations of the standard recovery approaches such as *basis pursuit* (Theorem 9.2) and *matching pursuit* (Corollary 8.19) are consistent with the expander sensing approach and can recover the original sparse signal successfully. In the presence of Gaussian or sparse noise, random dense sensing and expander sensing are known to provide similar performance in terms of the number of measurements and recovery computation time. Furthermore, expander sensing requires less storage whenever the signal is sparse in the canonical basis.

The approach described in this chapter consists of the following key elements:

- expander sensing matrices and the RIP-1 associated with them;
- a reconstruction objective function which explicitly incorporates the Poisson likelihood;
- a countable collection of candidate estimators; and

- a penalty function defined over the collection of candidates, which satisfies the Kraft inequality and which can be used to promote sparse reconstructions.

In general, the penalty function is selected to be small for signals of interest, which leads to theoretical guarantees that errors are small with high probability for such signals. In this chapter, exploiting the RIP-1 property and the non-negativity of the expander-based sensing matrices, we show that, in contrast to random dense sensing, expander sensing empowered with a maximum *a posteriori* (MAP) algorithm can approximately recover the original signal in the presence of Poisson noise, and we prove bounds which quantify the MAP performance. As a result, in the presence of Poisson noise, expander graphs not only provide general storage and computational advantages, but they also allow devising efficient MAP recovery methods with performance guarantees comparable to the best k-term approximation of the original signal. Finally, the bounds are tighter than specific dense matrices proposed by Willett and Raginsky [252, 210] whenever the signal is sparse in the canonical domain, in that a log term in the bounds in [210] is absent from the bounds presented in this chapter.

10.1.1 Dense sensing matrices for Poisson CS

In recent work, Willett and Raginsky established performance bounds for CS in the presence of Poisson noise using dense sensing matrices based on appropriately shifted and scaled Rademacher ensembles [252, 210]. Several features distinguish that work from the present chapter:

- The dense sensing matrices used in [252, 210] require more memory to store and more computational resources to apply to a signal in a reconstruction algorithm. As explained in Table 8.1, the expander-based approach, in contrast, is more efficient.

- The expander-based approach described in this chapter works *only* when the signal of interest is sparse in the canonical basis. In contrast, the dense sensing matrices used in [252, 210] can be applied to arbitrary sparsity bases (though the proof technique there needs to be altered slightly to accommodate sparsity in the canonical basis).

- The bounds in *both* this chapter and [252, 210] reflect a sobering tradeoff between performance and the number of measurements collected. In particular, more measurements (after some critical minimum number) can actually *degrade* performance as a limited number of events (e.g., photons) are distributed among a growing number of detectors, impairing the SNR of the measurements.

10.2 Compressed sensing in the presence of Poisson Noise

10.2.1 Problem statement

We wish to recover an unknown vector $\boldsymbol{\alpha}^* \in \mathbb{R}_+^N$ of Poisson intensities from a measured vector $\boldsymbol{f} \in \mathbb{Z}_+^M$, sensed according to the Poisson model

$$\boldsymbol{f} \sim \text{Poisson}(\boldsymbol{A}\boldsymbol{\alpha}^*), \tag{10.2.1}$$

where $\boldsymbol{A} \in \mathbb{R}_+^{M\times N}$ is a positivity-preserving sensing matrix. That is, for each $j \in \{1,\dots,M\}$, f_j is sampled independently from a Poisson distribution with mean $(A\alpha^*)_j$:

$$\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f}) = \prod_{j=1}^{M} \mathbb{P}_{(A\alpha^*)_j}(f_j), \tag{10.2.2}$$

where, for any $z \in \mathbb{Z}_+$ and $\lambda \in \mathbb{R}_+$, we have

$$\mathbb{P}_\lambda(z) \triangleq \begin{cases} \dfrac{\lambda^z}{z!}e^{-\lambda} & \text{if } \lambda > 0 \\ 1_{\{z=0\}} & \text{otherwise} \end{cases}, \tag{10.2.3}$$

where the $\lambda = 0$ case is a consequence of the fact that

$$\lim_{\lambda\to 0} \frac{\lambda^z}{z!}e^{-\lambda} = 1_{\{z=0\}}.$$

We assume that the ℓ_1 norm of $\boldsymbol{\alpha}^*$ is known, $\|\boldsymbol{\alpha}^*\|_1 = L$ (although later we will show that this assumption can be relaxed). We are interested in designing a sensing matrix $\boldsymbol{A}$ and an estimator $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\alpha}}(\boldsymbol{f})$, such that $\boldsymbol{\alpha}^*$ can be recovered with small expected ℓ_1 risk

$$R\left(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\right) \triangleq \mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_1,$$

where the expectation is taken w.r.t. the distribution $\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}$.

10.2.2 The proposed estimator and its performance

For future convenience, we introduce the following notation. Given N and $1 \le k \le N/4$, we denote by $G_{k,N}$ a $(2k, 1/16)$-expander with left set size N whose existence is guaranteed by Proposition 8.2. Then $G_{k,N} = (\boldsymbol{V}, \boldsymbol{C}, \boldsymbol{E})$ has

$$|\boldsymbol{V}| = N, \quad |\boldsymbol{C}| = M = O(k\log(N/k)), \quad d = O(\log(N/k)).$$

Moreover, since $G_{k,N}$ is regular, there exists a minimal set $\Omega \subset \boldsymbol{V}$ of size at most $M = |\boldsymbol{C}|$, such that its neighborhood covers all of $\boldsymbol{C}$, i.e $\mathcal{N}(\Omega) = \boldsymbol{C}$. Hence, $\boldsymbol{\Phi} I_\Omega \succeq 1$.

To recover $\boldsymbol{\alpha}^*$, we will use a penalized Maximum Likelihood Estimation (pMLE) approach. Let us choose a convenient $1 \leq k \leq N/4$ and take $\boldsymbol{A}$ to be the normalized adjacency matrix of the expander $G_{k,N}$: $\boldsymbol{A} \triangleq \boldsymbol{\Phi}/d$. Moreover, let us choose a finite or countable set Θ_L of candidate estimators $\boldsymbol{\alpha} \in \mathbb{R}_+^N$ with $\|\boldsymbol{\alpha}\|_1 \leq L$, and a *penalty* $\mathrm{pen} : \Theta_L \to \mathbb{R}_+$, satisfying the *Kraft inequality*[1]

$$\sum_{\boldsymbol{\alpha} \in \Theta_L} e^{-\mathrm{pen}(\boldsymbol{\alpha})} \leq 1.$$

For instance, we can impose less penalty on sparser signals or construct a penalty based on any other prior knowledge about the underlying signal.

With these definitions, we consider the following *penalized maximum likelihood estimator (pMLE)*:

$$\hat{\boldsymbol{\alpha}} \triangleq \operatorname*{argmin}_{\boldsymbol{\alpha} \in \Theta_L} \left[-\log \mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}(\boldsymbol{f}) + 2\,\mathrm{pen}(\boldsymbol{\alpha}) \right] \tag{10.2.4}$$

One way to think about the procedure in (10.2.4) is as a Maximum *a posteriori* Probability (MAP) algorithm over the set of estimates Θ_L, where the likelihood is computed according to the Poisson model (10.2.3) and the penalty function corresponds to a negative log prior on the candidate estimators in Θ_L.

Our main bound on the performance of the pMLE is as follows:

Theorem 10.1. *Let $\boldsymbol{\Phi}$ be the normalized adjacency matrix of $G_{k,N}$, let $\boldsymbol{\alpha}^* \in \mathbb{R}_+^N$ be the original signal compressively sampled in the presence of Poisson noise, and let $\hat{\boldsymbol{\alpha}}$ be obtained through* (10.2.4)*. Then*

$$\begin{aligned} R\left(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\right) &\leq 4\sigma_k(\boldsymbol{\alpha}^*) \\ &\quad + 8\sqrt{L \min_{\boldsymbol{\alpha} \in \Theta_L} \left[KL(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \parallel \mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}) + 2\,\mathrm{pen}(\boldsymbol{\alpha}) \right]}, \end{aligned} \tag{10.2.5}$$

where

$$KL(\mathbb{P}_{\boldsymbol{g}} \| \mathbb{P}_{\boldsymbol{h}}) \triangleq \sum_{\boldsymbol{y} \in \mathbb{Z}_+^M} \mathbb{P}_{\boldsymbol{g}}(\boldsymbol{y}) \log \frac{\mathbb{P}_{\boldsymbol{g}}(\boldsymbol{y})}{\mathbb{P}_{\boldsymbol{h}}(\boldsymbol{y})}$$

is the Kullback–Leibler divergence (relative entropy) between $\mathbb{P}_{\boldsymbol{g}}$ and $\mathbb{P}_{\boldsymbol{h}}$, and $\sigma_k(\boldsymbol{\alpha}^*)$ is the best k-term approximation to $\boldsymbol{\alpha}^*$, defined in Equation (2.1.2)*.*

Proof. Since $\hat{\boldsymbol{\alpha}} \in \Theta_L$, we have $L = \|\boldsymbol{\alpha}^*\|_1 \geq \|\hat{\boldsymbol{\alpha}}\|_1$. Hence, using Theorem 9.1 with $\Delta = 0$, we can write

$$\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1 \leq 4\sigma_k(\boldsymbol{\alpha}^*) + 4\|\boldsymbol{A}(\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}})\|_1.$$

[1] Many penalization functions can be modified slightly (e.g. scaled appropriately) to satisfy the Kraft inequality. All that is required is a finite collection of estimators (i.e. Θ_L) and an associated prefix code for each candidate estimate in Θ_L. For instance, this would certainly be possible for a total variation penalty, though the details are beyond the scope of this paper.

Taking expectations, we obtain

$$\begin{aligned} R\left(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\right) &\leq 4\sigma_k(\boldsymbol{\alpha}^*) + 4\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\boldsymbol{A}(\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}})\|_1 \\ &\leq 4\sigma_k(\boldsymbol{\alpha}^*) + 4\sqrt{\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\boldsymbol{A}(\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}})\|_1^2} \end{aligned} \tag{10.2.6}$$

where the second step uses Jensen's inequality. Using Lemmas 10.5 and 10.6 in Section 10.2.4, we have

$$\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\boldsymbol{A}(\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}})\|_1^2 \leq 4L \min_{\boldsymbol{\alpha}\in\Theta_L} \left[\mathrm{KL}(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \parallel \mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}) + 2\,\mathrm{pen}(\boldsymbol{\alpha})\right]$$

Substituting this into (10.2.6), we obtain (10.2.5). □

The bound of Theorem 10.1 is an *oracle inequality*: it states that the ℓ_1 error of $\hat{\boldsymbol{\alpha}}$ is (up to multiplicative constants) the sum of the k-term approximation error of $\boldsymbol{\alpha}^*$ plus $\sqrt{L}$ times the minimum penalized relative entropy error over the set of candidate estimators Θ_L. The first term in (10.2.5) is smaller for sparser $\boldsymbol{\alpha}^*$, and the second term is smaller when there exists $\boldsymbol{\alpha} \in \Theta_L$ which is simultaneously a good approximation to $\boldsymbol{\alpha}^*$ (in the sense that the distributions $\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}$ and $\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}$ are close) and has a low penalty.

Remark 10.2. *So far we have assumed that the ℓ_1 norm of $\boldsymbol{\alpha}^*$ is known. However, if $\|\boldsymbol{\alpha}^*\|_1$ is not known a priori, we can still estimate it with high accuracy using noisy compressive measurements. Observe that, since each measurement f_j is a Poisson random variable with mean $(A\alpha^*)_j$, $\sum_j f_j$ is Poisson with mean $\|\boldsymbol{A}\boldsymbol{\alpha}^*\|_1$. Therefore, $\sqrt{\sum_j f_j}$ is approximately normally distributed with mean $\approx \sqrt{\|\boldsymbol{A}\boldsymbol{\alpha}^*\|_1}$ and variance $\approx \frac{1}{4}$ [189, Sec. 6.2].*[2] *Hence, Mill's inequality [249, Thm. 4.7] guarantees that, for every positive t,*

$$\Pr\left[\left|\sqrt{\sum_j f_j} - \sqrt{\|\boldsymbol{A}\boldsymbol{\alpha}^*\|_1}\right| > t\right] \lesssim \frac{e^{-2t^2}}{\sqrt{2\pi}t},$$

where $\lesssim$ is meant to indicate the fact that this is only an approximate bound, with the approximation error controlled by the rate of convergence in the central limit theorem. Now we can use the RIP-1 property of the expander graphs obtain the estimates

$$\left(\sqrt{\sum_j f_j} - t\right)^2 \leq \|\boldsymbol{A}\boldsymbol{\alpha}^*\|_1 \leq \|\boldsymbol{\alpha}^*\|_1,$$

and

$$\frac{\left(\sqrt{\sum_j f_j} + t\right)^2}{(1-2\epsilon)} \geq \frac{\|\boldsymbol{A}\boldsymbol{\alpha}^*\|_1}{(1-2\epsilon)} \geq \|\boldsymbol{\alpha}^*\|_1$$

that hold with probability (approximately) at least $1 - (\sqrt{2\pi}t)^{-1}e^{-2t^2}$.

[2] This observation underlies the use of variance-stabilizing transforms.

10.2.3 A bound in terms of ℓ_1 error

The bound of Theorem 10.1 is not always useful since it bounds the ℓ_1 risk of the pMLE in terms of the relative entropy. A bound purely in terms of ℓ_1 errors would be more desirable. However, this is not easy to obtain without imposing extra conditions either on $\boldsymbol{\alpha}^*$ or on the candidate estimators in Θ_L. This follows from the fact that the divergence $\mathrm{KL}(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}})$ may take the value $+\infty$ if there exists some $\boldsymbol{f}$ such that $\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}(\boldsymbol{f}) = 0$ but $\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f}) > 0$.

One way to eliminate this problem is to impose an additional requirement on the candidate estimators in Θ_L: There exists some $c > 0$, such that

$$\boldsymbol{A}\boldsymbol{\alpha} \succeq c, \qquad \forall \boldsymbol{\alpha} \in \Theta_L \tag{10.2.7}$$

Under this condition, we will now develop a risk bound for the pMLE purely in terms of the ℓ_1 error.

Theorem 10.3. *Suppose that all the conditions of Theorem 10.1 are satisfied. In addition, suppose that the set Θ_L satisfies the condition* (10.2.7)*. Then*

$$R\left(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\right) \leq 4\sigma_k(\boldsymbol{\alpha}^*) + 8\sqrt{L \min_{\boldsymbol{\alpha}\in\Theta_L}\left[\frac{1}{c}\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\|_1^2 + \mathrm{pen}(\boldsymbol{\alpha})\right]}. \tag{10.2.8}$$

Proof. Using Lemma 10.7 in Section 10.2.4, we get the bound

$$\mathrm{KL}(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}) \leq \frac{1}{c}\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}\|_1^2, \qquad \forall \boldsymbol{\alpha} \in \Theta_L.$$

Substituting this into Eq. 10.2.5, we get (10.2.8). □

Remark 10.4. *Because every $\boldsymbol{\alpha} \in \Theta_L$ satisfies $\|\boldsymbol{\alpha}\|_1 \leq L$, the constant c cannot be too large. In particular, if* (10.2.7) *holds, then for every $\boldsymbol{\alpha} \in \Theta_L$ we must have*

$$\|\boldsymbol{A}\boldsymbol{\alpha}\|_1 \geq M \min_j (A\alpha^*)_j \geq Mc.$$

On the other hand, since $\|\boldsymbol{\Phi}\|_1 = 1$, we have $\|\boldsymbol{A}\boldsymbol{\alpha}\|_1 \leq \|\boldsymbol{\alpha}\|_1 \leq L$. Thus, a necessary condition for (10.2.7) *to hold is $c \leq L/M$. Since $M = O(k\log(N/k))$, the best risk we may hope to achieve under some condition like* (10.2.7) *is on the order of*

$$\begin{aligned} R\left(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}^*\right) &\leq 4\sigma_k(\boldsymbol{\alpha}^*) \\ &\quad + C\sqrt{\min_{\boldsymbol{\alpha}\in\Theta_L}\left[k\log(N/k)\|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|_1^2 + L\,\mathrm{pen}(\boldsymbol{\alpha})\right]} \end{aligned} \tag{10.2.9}$$

for some constant C, e.g., by choosing $c \propto \frac{L}{k\log(N/k)}$. Effectively, this means that, under the positivity condition (10.2.7)*, the ℓ_1 error of $\hat{\boldsymbol{\alpha}}$ is the sum of the k-term approximation error of $\boldsymbol{\alpha}^*$ plus $\sqrt{M} = \sqrt{k\log(N/k)}$ times the best penalized ℓ_1 approximation error. The first term in* (10.2.9) *is smaller for sparser $\boldsymbol{\alpha}^*$, and the second term is smaller when there is a $\boldsymbol{\alpha} \in \Theta_L$ which is simultaneously a good ℓ_1 approximation to $\boldsymbol{\alpha}^*$ and has a low penalty.*

10.2.4 Technical lemmas

Lemma 10.5. *Any $\boldsymbol{\alpha} \in \Theta_L$ satisfies the bound*

$$\|\boldsymbol{A}(\boldsymbol{\alpha}^* - \boldsymbol{\alpha})\|_1^2 \le 4L \sum_{i=1}^{M} \left| (\boldsymbol{A}\boldsymbol{\alpha}^*)_i^{1/2} - (\boldsymbol{A}\boldsymbol{\alpha})_i^{1/2} \right|^2 .$$

Proof. Since $\boldsymbol{A}$ is the *normalized* adjacency of the d-regular expander graph, $\|\boldsymbol{A}\|_1 = 1$. Therefore,

$$\|\boldsymbol{A}\boldsymbol{\alpha}\|_1 \le \|\boldsymbol{\alpha}\|_1 \le L, \qquad \forall \boldsymbol{\alpha} \in \Theta_L. \tag{10.2.10}$$

Let $\boldsymbol{\beta}^* \triangleq \boldsymbol{A}\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta} \triangleq \boldsymbol{A}\boldsymbol{\alpha}$. Then

$$\begin{aligned}
\|\boldsymbol{\beta}^* - \boldsymbol{\beta}\|_1^2 &= \left(\sum_{i=1}^{M} |\boldsymbol{\beta}^*{}_i - \boldsymbol{\beta}_i| \right)^2 = \left(\sum_{i=1}^{M} \left| \boldsymbol{\beta}^{*1/2}_i - \boldsymbol{\beta}_i^{1/2} \right| . \left| \boldsymbol{\beta}^{*1/2}_i + \hat{\boldsymbol{\beta}}_i^{1/2} \right| \right)^2 \\
&\le \sum_{i,j=1}^{M} \left| \boldsymbol{\beta}^{*1/2}_i - \boldsymbol{\beta}_i^{1/2} \right|^2 . \left| \boldsymbol{\beta}^{*1/2}_j + \boldsymbol{\beta}_j^{1/2} \right|^2 \le 2 \sum_{i=1}^{M} \left| \boldsymbol{\beta}^{*1/2}_i - \boldsymbol{\beta}_i^{1/2} \right|^2 . \sum_{j=1}^{M} \left| \boldsymbol{\beta}^*{}_j + \boldsymbol{\beta}_j \right| \\
&= 2 \sum_{i=1}^{M} \left| \boldsymbol{\beta}^{*1/2}_i - \boldsymbol{\beta}_i^{1/2} \right|^2 . (\|\boldsymbol{\beta}^*\|_1 + \|\boldsymbol{\beta}\|_1) \le 4L \sum_{i=1}^{M} \left| \boldsymbol{\beta}^{*1/2}_i - \boldsymbol{\beta}_i^{1/2} \right|^2 . \\
&= 4L \sum_{i=1}^{M} \left| (\boldsymbol{A}\boldsymbol{\alpha}^*)_i^{1/2} - (\boldsymbol{A}\boldsymbol{\alpha})_i^{1/2} \right|^2 .
\end{aligned}$$

The first and the second inequalities are by Cauchy–Schwarz, while the third inequality is a consequence of Eq. (10.2.10). □

Lemma 10.6. *Let $\hat{\boldsymbol{\alpha}}$ be a minimizer in Eq.* (10.2.4)*. Then*

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \left[\sum_{i=1}^{M} \left| (\boldsymbol{A}\boldsymbol{\alpha}^*)_i^{1/2} - (\boldsymbol{A}\hat{\boldsymbol{\alpha}})_i^{1/2} \right|^2 \right] \\
&\quad \le \min_{\boldsymbol{\alpha} \in \Theta_L} \left[KL(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \parallel \mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}) + 2\,\mathrm{pen}(\boldsymbol{\alpha}) \right] .
\end{aligned} \tag{10.2.11}$$

Proof. Using Lemma 10.8 below with $\boldsymbol{g} = \boldsymbol{A}\boldsymbol{\alpha}^*$ and $\boldsymbol{h} = \boldsymbol{A}\hat{\boldsymbol{\alpha}}$ we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \left[\sum_{i=1}^{M} \left| (\boldsymbol{A}\boldsymbol{\alpha}^*)_i^{1/2} - (\boldsymbol{A}\hat{\boldsymbol{\alpha}})_i^{1/2} \right|^2 \right] \\
&\quad = \mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*} \left[2 \log \frac{1}{\int \sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f}) \mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})} d\nu(\boldsymbol{f})} \right] .
\end{aligned}$$

Clearly

$$\int \sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}d\nu(\boldsymbol{f}) = \mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right].$$

We now provide a bound for this expectation. Let $\tilde{\boldsymbol{\alpha}}$ be a minimizer of $\mathrm{KL}(\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\|\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}}) + 2\,\mathrm{pen}(\boldsymbol{\alpha})$ over $\boldsymbol{\alpha} \in \Theta_L$. Then, by definition of $\hat{\boldsymbol{\alpha}}$, we have

$$\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})} \geq \sqrt{\mathbb{P}_{\boldsymbol{A}\tilde{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\tilde{\boldsymbol{\alpha}})}$$

for every $\boldsymbol{f}$. Consequently,

$$\frac{1}{\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]} \leq \frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\tilde{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\tilde{\boldsymbol{\alpha}})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]},$$

We can split the quantity

$$2\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\log\left(\frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\tilde{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\tilde{\boldsymbol{\alpha}})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]}\right)\right]$$

into three terms:

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\log\left(\frac{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\tilde{\boldsymbol{\alpha}}}(\boldsymbol{f})}\right)\right] + 2\,\mathrm{pen}(\tilde{\boldsymbol{\alpha}})\\ &+ 2\mathbb{E}\left[\log\left(\frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]}\right)\right]\end{aligned}$$

We show that the third term is always nonpositive, which completes the proof. Using Jensen's inequality,

$$\begin{aligned}&\mathbb{E}\left[\log\left(\frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]}\right)\right]\\ &\leq \log\left(\mathbb{E}\left[\frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]}\right]\right).\end{aligned}$$

Now

$$\mathbb{E}\left[\frac{\sqrt{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}e^{-\mathrm{pen}(\hat{\boldsymbol{\alpha}})}}{\sqrt{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}\mathbb{E}_{\boldsymbol{A}\boldsymbol{\alpha}^*}\left[\sqrt{\frac{\mathbb{P}_{\boldsymbol{A}\hat{\boldsymbol{\alpha}}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A}\boldsymbol{\alpha}^*}(\boldsymbol{f})}}\right]}\right] \leq \sum_{\boldsymbol{\alpha}\in\Theta_L} e^{-\mathrm{pen}(\boldsymbol{\alpha})} \leq 1.$$

Since $\mathbb{E}_{\boldsymbol{A\alpha^*}}\left[\log\left(\frac{\mathbb{P}_{\boldsymbol{A\alpha^*}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A\tilde{\alpha}}}(\boldsymbol{f})}\right)\right] = \mathrm{KL}(\mathbb{P}_{\boldsymbol{A\alpha^*}}\|\mathbb{P}_{\boldsymbol{A\tilde{\alpha}}})$, we obtain

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{A\alpha^*}}\left[\sum_{i=1}^{M}\left|(\boldsymbol{A\alpha^*})_i^{1/2} - (\boldsymbol{A\hat{\alpha}})_i^{1/2}\right|^2\right] \\
&\quad\le \mathrm{KL}(\mathbb{P}_{\boldsymbol{A\alpha^*}}\|\mathbb{P}_{\boldsymbol{A\tilde{\alpha}}}) + 2\,\mathrm{pen}(\tilde{\boldsymbol{\alpha}}) \\
&\quad= \min_{\boldsymbol{\alpha}\in\Theta_L}\left[\mathrm{KL}(\mathbb{P}_{\boldsymbol{A\alpha^*}}\|\mathbb{P}_{\boldsymbol{A\alpha}}) + 2\,\mathrm{pen}(\boldsymbol{\alpha})\right],
\end{aligned}$$

which proves the lemma. □

Lemma 10.7. *If the estimators in Θ_L satisfy the condition* (10.2.7)*, then the following inequality holds:*

$$KL(\mathbb{P}_{\boldsymbol{A\alpha^*}} \parallel \mathbb{P}_{\boldsymbol{A\alpha}}) \le \frac{1}{c}\|\boldsymbol{\alpha^*} - \boldsymbol{\alpha}\|_1^2, \qquad \forall \boldsymbol{\alpha} \in \Theta_L.$$

Proof. By definition of the KL divergence,

$$\begin{aligned}
&\mathrm{KL}(\mathbb{P}_{\boldsymbol{A\alpha^*}}\|\mathbb{P}_{\boldsymbol{A\alpha}}) = \mathbb{E}_{\boldsymbol{A\alpha^*}}\left[\log\left(\frac{\mathbb{P}_{\boldsymbol{A\alpha^*}}(\boldsymbol{f})}{\mathbb{P}_{\boldsymbol{A\alpha}}(\boldsymbol{f})}\right)\right] \\
&= \sum_{j=1}^{M}\mathbb{E}_{(A\alpha^*)_j}\left[\boldsymbol{f}_j\log\left(\frac{(A\alpha^*)_j}{(A\alpha)_j}\right)\right] - \sum_{j=1}^{M}\mathbb{E}_{(A\alpha^*)_j}\left[(A\alpha^*)_j - (A\alpha)_j\right] \\
&= \sum_{j=1}^{M}\left[(A\alpha^*)_j\log\left(\frac{(A\alpha^*)_j}{(A\alpha)_j}\right) - (A\alpha^*)_j + (A\alpha)_j\right] \\
&\le \sum_{j=1}^{M}(A\alpha^*)_j\left(\frac{(A\alpha^*)_j}{(A\alpha)_j} - 1\right) - (A\alpha^*)_j + (A\alpha)_j \\
&= \sum_{j=1}^{M}\frac{1}{(A\alpha)_j}\left|(\boldsymbol{A\alpha^*} - \boldsymbol{A\alpha})_j\right|^2 \le \frac{1}{c}\|\boldsymbol{A}(\boldsymbol{\alpha^*} - \boldsymbol{\alpha})\|_2^2 \\
&\le \frac{1}{c}\|\boldsymbol{A}(\boldsymbol{\alpha^*} - \boldsymbol{\alpha})\|_1^2 \le \frac{1}{c}\|\boldsymbol{\alpha^*} - \boldsymbol{\alpha}\|_1^2.
\end{aligned}$$

The first inequality uses $\log t \le t-1$, the second is by (10.2.7), the third uses the fact that the ℓ_1 norm dominates the ℓ_2 norm, and the last one is by the RIP-1 property (Lemma 8.8). □

Lemma 10.8. *Given two Poisson parameter vectors $\boldsymbol{g}, \boldsymbol{h} \in \mathbb{R}_+^M$, the following equality holds:*

$$2\log\frac{1}{\int\sqrt{\mathbb{P}_{\boldsymbol{g}}(\boldsymbol{f})\mathbb{P}_{\boldsymbol{h}}(\boldsymbol{f})}d\mu(\boldsymbol{f})} = \sum_{j=1}^{M}\left|\boldsymbol{g}_j^{1/2} - \boldsymbol{h}_j^{1/2}\right|^2,$$

where μ denotes the counting measure on $\mathbb{R}_+^M$.

Proof.

$$\begin{aligned}
&\int \sqrt{\mathbb{P}_{\boldsymbol{g}}(\boldsymbol{f})\mathbb{P}_{\boldsymbol{h}}(\boldsymbol{f})}d\mu(\boldsymbol{f}) \\
&= \prod_{j=1}^{M}\sum_{\boldsymbol{f}_j=0}^{\infty}\frac{(\boldsymbol{g}_j\boldsymbol{h}_j)^{\boldsymbol{f}_j/2}}{\boldsymbol{f}_j!}e^{-(\boldsymbol{g}_j+\boldsymbol{h}_j)/2} \\
&= \prod_{j=1}^{M}e^{-\frac{1}{2}(\boldsymbol{g}_j-2(\boldsymbol{g}_j\boldsymbol{h}_j)^{1/2}+\boldsymbol{h}_j)}\sum_{\boldsymbol{f}_j=0}^{\infty}\frac{(\boldsymbol{g}_j\boldsymbol{h}_j)^{\boldsymbol{f}_j/2}}{\boldsymbol{f}_j!}e^{-(\boldsymbol{g}_j\boldsymbol{h}_j)^{1/2}} \\
&= \prod_{j=1}^{M}e^{-\frac{1}{2}(\boldsymbol{g}_j-2(\boldsymbol{g}_j\boldsymbol{h}_j)^{1/2}+\boldsymbol{h}_j)}\underbrace{\int \mathbb{P}_{(\boldsymbol{g}_j\boldsymbol{h}_j)^{1/2}}(\boldsymbol{f}_j)d\nu_j(\boldsymbol{f}_j)}_{=1} \\
&= \prod_{j=1}^{M}e^{-\frac{1}{2}\left(\boldsymbol{g}_j^{1/2}-\boldsymbol{h}_j^{1/2}\right)^2}
\end{aligned}$$

Taking logs, we obtain the lemma. □

10.2.5 Empirical performance

Here we present a simulation study that corroborates our proposed method. In this experiment, compressive Poisson observations are collected of a randomly generated sparse signal passed through the sensing matrix generated using the proposed expander graph method. We then reconstruct the signal by utilizing an algorithm that minimizes the proposed objective function in (10.2.4), and assess the accuracy of this estimate. We repeat this procedure over several trials to estimate the average performance of the method.

More specifically, we generate our length N sparse signal $\boldsymbol{\alpha}^*$ through a two-step procedure. First we select k elements uniformly at random, then we assign these elements an intensity I. All other components of the signal are set to zero. For these experiments, we chose a length $N = 100{,}000$ and varied the sparsity k among three different choices of 100, 500, and 1,000 for two intensity levels I of 10,000 and 100,000. We then vary the number M of Poisson observations from 100 to 20,000 using an expander graph sensing matrix with degree $d = 8$. Recall that the sensing matrix is normalized such that the total signal intensity is divided amongst the measurements, hence the seemingly high choices of I.

To reconstruct the signal, we utilize the SPIRAL-ℓ_1 algorithm [141] which solves (10.2.4) when $\text{pen}(\boldsymbol{\alpha}) = \tau\|\boldsymbol{\alpha}\|_1$. This algorithm utilizes a sequence of quadratic subproblems derived by using a second-order Taylor expansion of the Poisson log-likelihood at each iteration. These subproblems are made easier to solve by using a separable approximation whereby the second-order Hessian matrix is approximated by

a scaled identity matrix. For the particular case of the ℓ_1 penalty, these subproblems can be solved quickly, exactly, and noniteratively by a soft-thresholding rule.

After reconstruction, we assess the estimate $\hat{\boldsymbol{\alpha}}$ according to the normalized ℓ_1 error $\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1 / \|\boldsymbol{\alpha}^*\|_1$. We select the regularization weighting τ in the SPIRAL-ℓ_1 algorithm to minimize this quantity for each randomly generated experiment indexed by (I, k, M). To assure that the results are not biased in our favor by only considering a single random experiment for each (I, k, M), we repeat this experiment several times. The averaged reconstruction accuracy over 10 trials is presented in Figure 10.1.

These results show that the proposed method is able to accurately estimate sparse signals when the signal intensity is sufficiently high; however, the performance of the method degrades for lower signal strengths. More interesting is the behavior as we vary the number of measurements. There is a clear phase transition where accurate signal reconstruction becomes possible, however the performance gently degrades with the number of measurements since there is a lower signal-to-noise ratio per measurement. This effect is more pronounced at lower intensity levels, as we more quickly enter the regime where only a few photons are collected per measurement. Both of these results support the error bounds developed in Section 10.2.2.

10.3 Application: Estimating packet arrival rates

This section describes an application of the pMLE estimator of Section 10.2: an indirect approach for reconstructing average packet arrival rates and instantaneous packet counts for a given number of streams (or flows) at a router in a communication network, where the arrivals of packets in each flow are assumed to follow a Poisson process. All packet counting must be done in hardware at the router, and any hardware implementation must strike a delicate balance between speed, accuracy, and cost. For instance, one could keep a dedicated counter for each flow, but, depending on the type of memory used, one could end up with an implementation that is either fast but expensive and unable to keep track of a large number of flows (e.g., using SRAMs, which have low access times, but are expensive and physically large) or cheap and high-density but slow (e.g., using DRAMs, which are cheap and small, but have longer access times) [108, 180].

However, there is empirical evidence [109, 110] that flow sizes in IP networks follow a *power-law* pattern: just a few flows (say, 10%) carry most of the traffic (say, 90%). Based on this observation, several investigators have proposed methodologies for estimating flows using a small number of counters by either (a) keeping track only of the flows whose sizes exceed a given fraction of the total bandwidth (the approach suggestively termed "focusing on the elephants, ignoring the mice") [108] or (b) using sparse random graphs to aggregate the raw packet counts and recovering flow sizes using a message passing decoder [180].

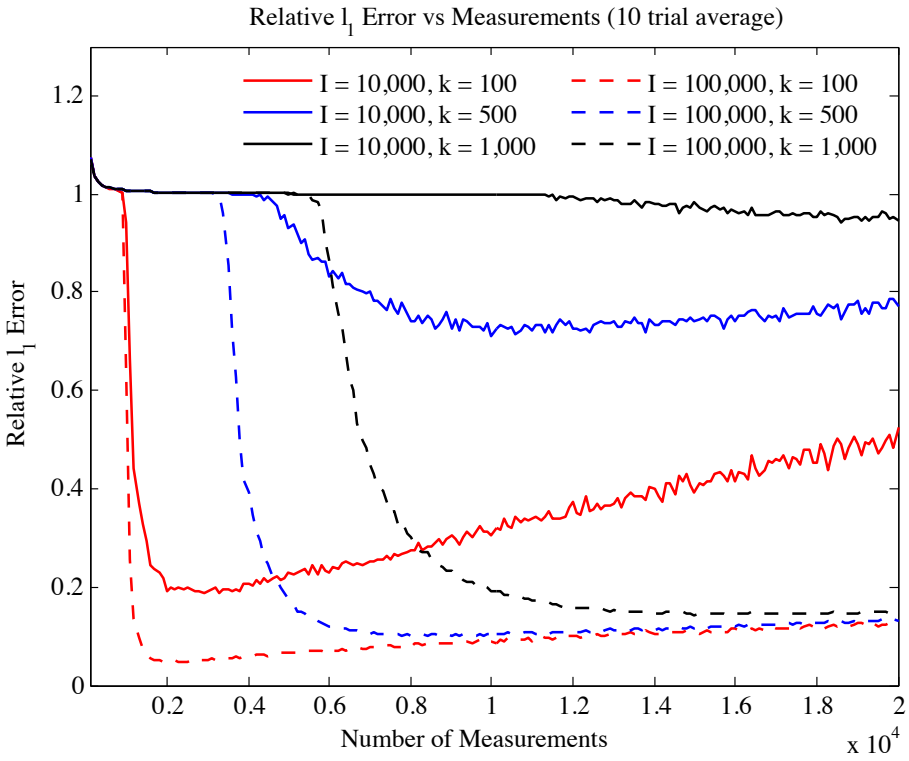


Figure 10.1: Average performance (as measured by the normalized ℓ_1 error $\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_1/\|\boldsymbol{\alpha}^*\|_1$) for the proposed expander-based observation method for recovering sparse signals under Poisson noise. In this experiment, we sweep over a range of measurements and consider a few sparsity (k) and intensity (I) levels of the true signal.

We consider an alternative to these approaches based on Poisson CS, assuming that the underlying Poisson rate vector is sparse or approximately sparse — and, in fact, it is the approximate sparsity of the rate vector that mathematically describes the power-law behavior of the average packet counts. The goal is to maintain a compressed summary of the process sample paths using a small number of counters, such that it is possible to reconstruct both the total number of packets in each flow and the underlying rate vector. Since we are dealing here with Poisson streams, we would like to push the metaphor further and say that we are "focusing on the whales, ignoring the minnows."

10.3.1 Problem Formulation

We wish to monitor a large number N of packet flows using a much smaller number M of counters. Each flow is a homogeneous Poisson process (cf. [33] for details pertaining to Poisson processes and networking applications). Specifically, let $\boldsymbol{\lambda}^* \in \mathbb{R}_+^N$ denote the vector of rates, and let U denote the random process $U = \{U_t\}_{t\in\mathbb{R}_+}$ with sample paths in $\mathbb{Z}_+^N$. In other words, for each $i \in \{1, \ldots, N\}$, the ith component of U, which we will denote by $U^{(i)}$, is a homogeneous Poisson process with the rate of λ_i arrivals per unit time, and all the $U^{(i)}$'s are mutually conditionally independent given $\boldsymbol{\lambda}$.

The goal is to estimate the unknown rate vector $\boldsymbol{\lambda}$ based on $\boldsymbol{f}$. We will focus on performance bounds for power-law network traffic, i.e., for $\boldsymbol{\lambda}^*$ belonging to the class

$$\Sigma_{\beta,L_0} \triangleq \left\{\boldsymbol{\lambda} \in \mathbb{R}_+^N : \|\boldsymbol{\lambda}\|_1 = L_0; \sigma_k(\boldsymbol{\lambda}) = O(k^{-\beta})\right\} \tag{10.3.1}$$

for some $L_0 > 0$ and $\beta \geq 1$, where the constant hidden in the $O(\cdot)$ notation may depend on L_0. Here, β is the power-law exponent that controls the tail behavior; in particular, the extreme regime $\beta \to +\infty$ describes the fully sparse setting. As in Section 10.2, we assume the total arrival rate $\|\boldsymbol{\lambda}^*\|_1$ to be known (and equal to a given L_0) in advance, but this assumption can be easily dispensed with (cf. Remark 10.2).

As before, we evaluate each candidate estimator $\widehat{\boldsymbol{\lambda}} = \widehat{\boldsymbol{\lambda}}(\boldsymbol{f})$ based on its expected ℓ_1 risk,

$$R\left(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^*\right) = \mathbb{E}_{\boldsymbol{\lambda}^*}\|\widehat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^*\|_1.$$

10.3.2 Two estimation strategies

We consider two estimation strategies. In both cases, we let our measurement matrix $\boldsymbol{\Phi}$ be the adjacency matrix of the expander $G_{k,N}$ for a fixed $k \leq N/4$ (see Section 10.2.2 for definitions). The first strategy, which we call the *direct method*, uses standard expander-based CS to construct an estimate of $\boldsymbol{\lambda}^*$. The second is the pMLE strategy, which relies on the machinery presented in Section 10.2 and can be used when only the rates are of interest.

The direct method

In this method, which will be used as a "baseline" for assessing the performance of the pMLE, the counters are updated in discrete time, every τ time units. Let $\boldsymbol{x} = \{\boldsymbol{x}_\nu\}_{\nu\in\mathbb{Z}_+}$ denote the sampled version of U, where $\boldsymbol{x}_\nu \triangleq U_{\nu\tau}$. The update takes place as follows. We have a binary matrix $\boldsymbol{\Phi} \in \{0,1\}^{M\times N}$, and at each time ν let $\boldsymbol{f}_\nu = \boldsymbol{\Phi}\boldsymbol{x}_\nu$. In other words, $\boldsymbol{f}$ is obtained by passing a sampled N-dimensional homogeneous Poisson process with rate vector $\boldsymbol{\lambda}$ through a linear transformation $\boldsymbol{\Phi}$. We emphasize the fact that this observation model is *not* equivalent to sampling a Poisson process with rate $\boldsymbol{\Phi}\boldsymbol{\lambda}$.

The direct method uses expander-based CS to obtain an estimate $\widehat{\boldsymbol{x}}_\nu$ of $\boldsymbol{x}_\nu$ from $\boldsymbol{f}_\nu = \boldsymbol{\Phi}\boldsymbol{x}_\nu$, followed by letting

$$\widehat{\boldsymbol{\lambda}}_\nu^{\text{dir}} = \frac{\widehat{\boldsymbol{x}}_\nu^+}{\nu\tau}. \tag{10.3.2}$$

This strategy is based on the observation that $\boldsymbol{x}_\nu/(\nu\tau)$ is the maximum-likelihood estimator of $\boldsymbol{\lambda}^*$. To obtain $\widehat{\boldsymbol{x}}_\nu$, we need to solve the convex program

$$\text{minimize } \|\boldsymbol{u}\|_1 \qquad \text{subject to } \boldsymbol{\Phi}\boldsymbol{u} = \boldsymbol{f}_\nu$$

which can be cast as a linear program [29]. The resulting solution $\widehat{\boldsymbol{x}}_\nu$ may have negative coordinates,[3] hence the use of the $(\cdot)^+$ operation in (10.3.2). We then have the following result:

Theorem 10.9.

$$R\left(\widehat{\boldsymbol{\lambda}}_\nu^{\text{dir}}, \boldsymbol{\lambda}^*\right) \le 4\sigma_k(\boldsymbol{\lambda}^*) + \frac{\|(\boldsymbol{\lambda}^*)^{1/2}\|_1}{\sqrt{\nu\tau}}, \tag{10.3.3}$$

where $(\boldsymbol{\lambda}^*)^{1/2}$ *is the vector with components* $\sqrt{\boldsymbol{\lambda}_i^*}, \forall i$.

Remark 10.10. *Note that the error term in* (10.3.3) *is* $O(1/\sqrt{\nu})$*, assuming everything else is kept constant, which coincides with the optimal rate of the* ℓ_1 *error decay in parametric estimation problems.*

Proof. We first observe that, by construction, $\widehat{\boldsymbol{x}}_\nu$ satisfies the relations $\boldsymbol{\Phi}\widehat{\boldsymbol{x}}_\nu = \boldsymbol{\Phi}\boldsymbol{x}_\nu$ and $\|\widehat{\boldsymbol{x}}_\nu\|_1 \le \|\boldsymbol{x}_\nu\|_1$. Hence,

$$\begin{aligned}\mathbb{E}\|\widehat{\boldsymbol{x}}_\nu - \nu\tau\boldsymbol{\lambda}^*\|_1 &\le \mathbb{E}\|\widehat{\boldsymbol{x}}_\nu - \boldsymbol{x}_\nu\|_1 + \mathbb{E}\|\boldsymbol{x}_\nu - \nu\tau\boldsymbol{\lambda}^*\|_1 \\ &\le 4\mathbb{E}\sigma_k(\boldsymbol{x}_\nu) + \mathbb{E}\|\boldsymbol{x}_\nu - \nu\tau\boldsymbol{\lambda}^*\|_1 \end{aligned} \tag{10.3.4}$$

where the first step uses the triangle inequality, while the second step uses Proposition 9.1 with $\Delta = 0$. To bound the first term in (10.3.4), let $S \subset \{1, \dots, N\}$ denote the positions of the k largest entries of $\boldsymbol{\lambda}^*$. Then, by definition of the best k-term representation,

$$\sigma_k(\boldsymbol{x}_\nu) \le \|\boldsymbol{x}_\nu - (\boldsymbol{x}_\nu)_S\|_1 = \sum_{i\in S^c} |\boldsymbol{x}_{\nu,i}| = \sum_{i\in S^c} \boldsymbol{x}_{\nu,i}.$$

Therefore,

$$\mathbb{E}\sigma_k(\boldsymbol{x}_\nu) \le \mathbb{E}\left[\sum_{i\in S^c} \boldsymbol{x}_{\nu,i}\right] = \nu\tau \sum_{i\in S^c} \boldsymbol{\lambda}_i^\star \equiv \nu\tau\sigma_k(\boldsymbol{\lambda}^*).$$

[3] Khajehnejad et al. [169] have recently proposed the use of perturbed adjacency matrices of expanders to recover nonnegative sparse signals.

To bound the second term, we can use concavity of the square root, as well as the fact that each $\boldsymbol{x}_{\nu,i} \sim \text{Poisson}(\nu\tau\boldsymbol{\lambda}_i^\star)$, to write

$$\begin{aligned}\mathbb{E}\|\boldsymbol{x}_\nu - \nu\tau\boldsymbol{\lambda}^*\|_1 &= \mathbb{E}\left[\sum_{i=1}^N |\boldsymbol{x}_{n,i} - \nu\tau\boldsymbol{\lambda}_i^\star|\right] \\ &= \mathbb{E}\left[\sum_{i=1}^N \sqrt{(\boldsymbol{x}_{n,i} - \nu\tau\boldsymbol{\lambda}_i^\star)^2}\right] \\ &\le \sum_{i=1}^N \sqrt{\mathbb{E}(\boldsymbol{x}_{\nu,i} - \nu\tau\boldsymbol{\lambda}_i^\star)^2} = \sum_{i=1}^N \sqrt{\nu\tau\boldsymbol{\lambda}_i^\star}.\end{aligned}$$

Now, it is not hard to show that $\|\widehat{\boldsymbol{x}}_\nu^+ - \nu\tau\boldsymbol{\lambda}^\star\|_1 \le \|\widehat{\boldsymbol{x}}_\nu - \nu\tau\boldsymbol{\lambda}^\star\|_1$. Therefore,

$$R\left(\widehat{\boldsymbol{\lambda}}_\nu^{\text{dir}}, \boldsymbol{\lambda}^*\right) \le \frac{\mathbb{E}\|\widehat{\boldsymbol{x}}_\nu - \nu\tau\boldsymbol{\lambda}^*\|_1}{\nu\tau} \le 4\sigma_k(\boldsymbol{\lambda}^*) + \frac{\|(\boldsymbol{\lambda}^*)^{1/2}\|_1}{\sqrt{\nu\tau}},$$

which proves the theorem. □

The penalized MLE approach

In the penalized MLE approach the counters are updated in a slightly different manner. Here the counters are still updated in discrete time, every τ time units; however, each counter $i \in \{1, \cdots, M\}$ is updated at times $\left(\nu\tau + \frac{i}{M}\tau\right)_{\nu\in\mathbb{Z}_+}$, and only aggregates the packets that have arrived during the time period $\left[\nu\tau + \frac{i-1}{M}\tau, \nu\tau + \frac{i}{M}\tau\right)$. Therefore, in contrast to the direct method, here each arriving packet is registered by at most one counter. Furthermore, since the packets arrive according to a homogeneous Poisson process, conditioned on the vector $\boldsymbol{\lambda}^*$, the values measured by distinct counters are independent[4]. Therefore, the vector of counts at time ν obeys

$$\boldsymbol{f}_\nu \sim \text{Poisson}(\boldsymbol{A}\boldsymbol{\alpha}^*) \qquad \text{where} \qquad \boldsymbol{\alpha}^* = \frac{\nu\tau d}{M}\boldsymbol{\lambda}^*$$

which is precisely the sensing model we have analyzed in Section 10.2.

Now assume that the total average arrival rate $\|\boldsymbol{\lambda}^*\|_1 = L_0$ is known. Let Λ be a finite or a countable set of candidate estimators with $\|\boldsymbol{\lambda}\|_1 \le L_0$ for all $\boldsymbol{\lambda} \in \Lambda$, and let $\text{pen}(\cdot)$ be a penalty functional satisfying the Kraft inequality over Λ. Given ν and τ, consider the scaled set

$$\Lambda_{\nu,\tau} \triangleq \frac{\nu\tau d}{M}\Lambda \equiv \left\{\frac{\nu\tau d}{M}\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \Lambda\right\}$$

[4]The independence follows from the fact that if $X_1, \cdots, X_M$ are conditionally independent random variables, then for any choice of functions $g_1, \cdots, g_M$, the random variables $g_1(X_1), \cdots, g_M(X_M)$ are also conditionally independent.

with the same penalty function, $\mathrm{pen}\left(\frac{\nu\tau d}{M}\boldsymbol{\lambda}\right) = \mathrm{pen}(\boldsymbol{\lambda})$ for all $\boldsymbol{\lambda} \in \Lambda$. We can now apply the results of Section 10.2. Specifically, let

$$\widehat{\boldsymbol{\lambda}}_{\nu}^{\mathrm{pMLE}} \triangleq \frac{M\,\hat{\boldsymbol{\alpha}}}{\nu\tau d},$$

where $\hat{\boldsymbol{\alpha}}$ is the corresponding pMLE estimator obtained according to (10.2.4). The following theorem is a consequence of Theorem 10.3 and the remark following it:

Theorem 10.11. *If the set Λ satisfies the strict positivity condition* (10.2.7)*, then there exists some absolute constant $C > 0$, such that*

$$R\left(\widehat{\boldsymbol{\lambda}}_{\nu}^{\mathrm{pMLE}}, \boldsymbol{\lambda}^*\right) \leq 4\sigma_k(\boldsymbol{\lambda}^*) + C\sqrt{\min_{\boldsymbol{\lambda}\in\Lambda}\left[k\log(N/k)\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\|_1^2 + \frac{k\,L_0\,\mathrm{pen}(\boldsymbol{\lambda})}{\nu\tau}\right]}. \tag{10.3.5}$$

We now develop risk bounds under the power-law condition. To this end, let us suppose that $\boldsymbol{\lambda}^*$ is a member of the power-law class $\Sigma_{L_0,\beta}$ defined in (10.3.1). Fix a small positive number δ, such that $L_0/\sqrt{\delta}$ is an integer, and define the set

$$\Lambda \triangleq \left\{\boldsymbol{\lambda} \in \mathbb{R}_+^N : \|\boldsymbol{\lambda}\|_1 \leq L_0; \boldsymbol{\lambda}_i \in \{s\sqrt{\delta}\}_{s=0}^{L_0/\sqrt{\delta}}, \forall i\right\}$$

These will be our candidate estimators of $\boldsymbol{\lambda}^*$. We can define the penalty function $\mathrm{pen}(\boldsymbol{\lambda}) \doteq \|\boldsymbol{\lambda}\|_0 \log(\delta^{-1})$. For any $\boldsymbol{\lambda} \in \Sigma_{\beta,L_0}$ and any $1 \leq r \leq N$ we are able to find some $\boldsymbol{\lambda}^{(r)} \in \Lambda$, such that $\|\boldsymbol{\lambda}^{(r)}\|_0 \asymp r$ and

$$\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(r)}\|_1^2 \asymp r^{-2\beta} + r\,\delta.$$

Here we assume that δ is sufficiently small, so that the penalty term $\frac{k\,r\log(\delta^{-1})}{\nu\tau}$ dominates the quantization error $r\,\delta$. In order to guarantee that the penalty function satisfies Kraft's inequality, we need to ensure that

$$\sum_{r=1}^{n} \sum_{\substack{\boldsymbol{\lambda}^{(r)}\in\Lambda \\ \|\boldsymbol{\lambda}^{(r)}\|_0 = r}} \delta^r \leq 1.$$

For every fixed r, there are exactly $\binom{N}{r}$ subspaces of dimension r, and each subspace contains exactly $\left(\frac{L_0}{\sqrt{\delta}}\right)^r$ distinct elements of Λ. Therefore, if

$$\delta \leq (2N\,L_0)^{-2}, \tag{10.3.6}$$

then

$$\sum_{r=1}^{N}\binom{N}{r}\left(L_0\sqrt{\delta}\right)^r \leq \sum_{r=0}^{N}\left(n\,L_0\sqrt{\delta}\right)^r \leq \sum_{r=1}^{n}\frac{1}{2^r} \leq 1,$$

and Kraft's inequality is satisfied.

Using the fact that $k \log(N/k) = O(kd)$, we can bound the minimum over $\boldsymbol{\lambda} \in \Lambda$ in (10.3.5) from above by

$$\begin{aligned}
&\min_{1 \le r \le N} \left[k d r^{-2\beta} + \frac{r\, k \log(\delta^{-1})}{\nu\tau} \right] \\
&\qquad = O\left(k\, d^{\frac{1}{2\beta+1}} \right) \left(\frac{\log(\delta^{-1})}{\nu\tau} \right)^{\frac{2\beta}{2\beta+1}} \\
&\qquad = O\left(k\, d^{\frac{1}{2\beta+1}} \right) \left(\frac{\log N}{\nu\tau} \right)^{\frac{2\beta}{2\beta+1}}
\end{aligned}$$

We can now particularize Theorem 10.11 to the power-law case:

Theorem 10.12.

$$\begin{aligned}
&\sup_{\boldsymbol{\lambda}^* \in \Sigma_{\beta, L_0}} R\left(\widehat{\boldsymbol{\lambda}}_{\nu}^{\mathrm{pMLE}}, \boldsymbol{\lambda}^* \right) \\
&\qquad = O(k^{-\beta}) + O\left(k^{\frac{1}{2}}\, d^{\frac{1}{4\beta+2}} \right) \left(\frac{\log N}{\nu\tau} \right)^{\frac{\beta}{2\beta+1}},
\end{aligned}$$

where the constants implicit in the $O(\cdot)$ notation depend on L_0 and β.

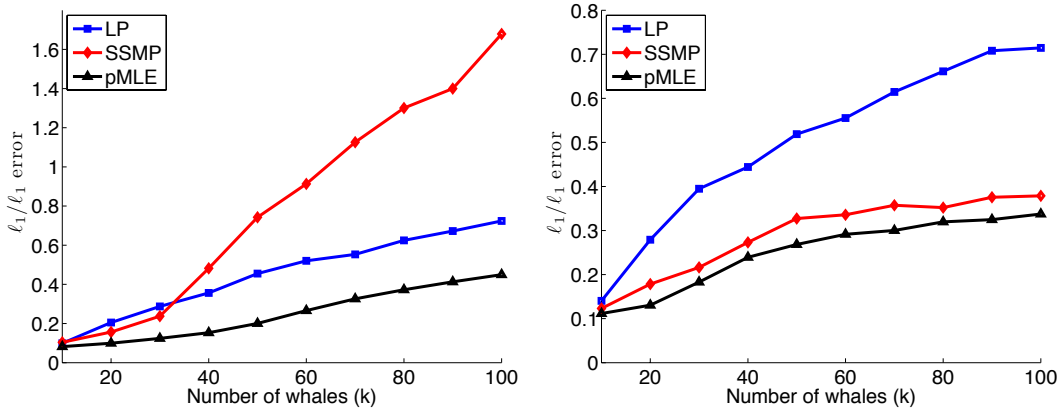
Note that the risk bound here is slightly worse than the benchmark bound of Theorem 10.9. However, as we will see in Section 10.3.3, the pMLE approach obtains higher empirical accuracy.

10.3.3 Experimental Results

Here we compare penalized MLE with ℓ_1-magic [52], a universal ℓ_1 minimization method, and with SSMP [30], an alternative method that employs combinatorial optimization. ℓ_1-magic and SSMP both compute the "direct" estimator. For the ease of computation, the candidate set Λ is approximated by the convex set of all positive vectors with bounded ℓ_1 norm, and the CVX package [134, 133] is used to directly solve the pMLE objective function with $\mathrm{pen}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$.

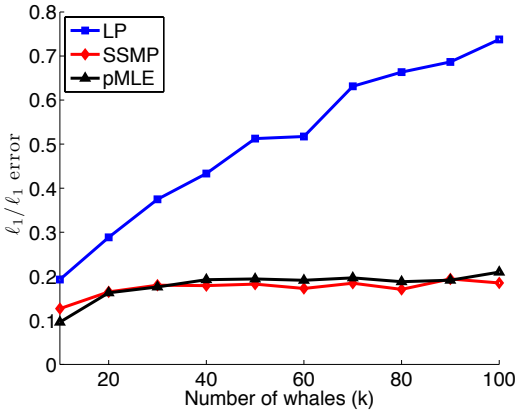
Figures 10.2(a) through 10.4(b) report the result of numerical experiments, where the goal is to identify the k largest entries in the rate vector from the measured data. Since a random graph is, with overwhelming probability, an expander graph, each experiment was repeated 30 times using independent sparse random graphs with $d = 8$.

We also used the following process to generate the rate vector. First, given the power-law exponent β, the magnitude of the k whales where chosen according to a power-law distribution with parameter β. The positions of the k whales were then chosen



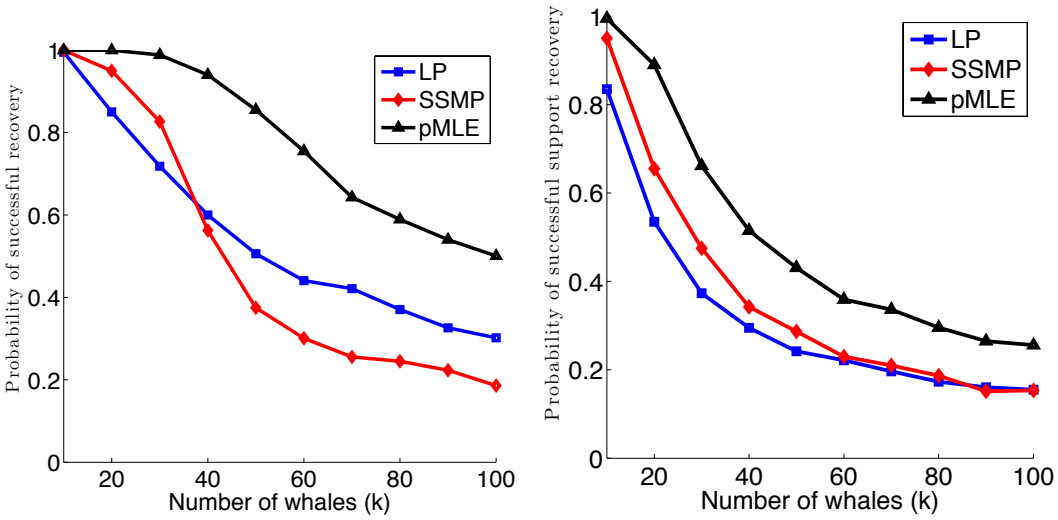
(a) $\beta = 1$

(b) $\beta = 1.5$



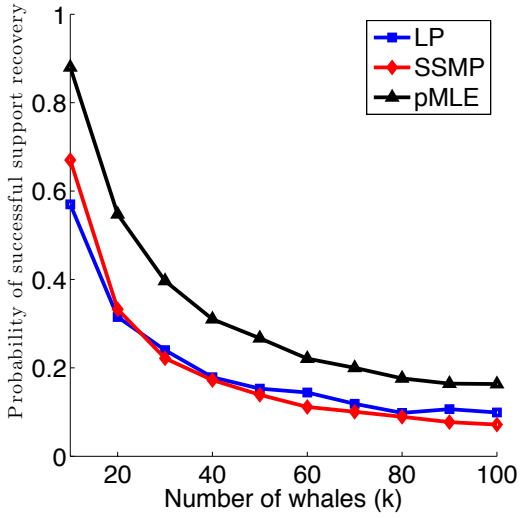
(c) $\beta = 2$

Figure 10.2: Relative ℓ_1 error as a function of number of whales k, for ℓ_1-magic (LP), SSMP and pMLE for different choices of the power-law exponent β. The number of flows $N = 5000$, the number of counters $M = 800$, and the number of updates is 40.



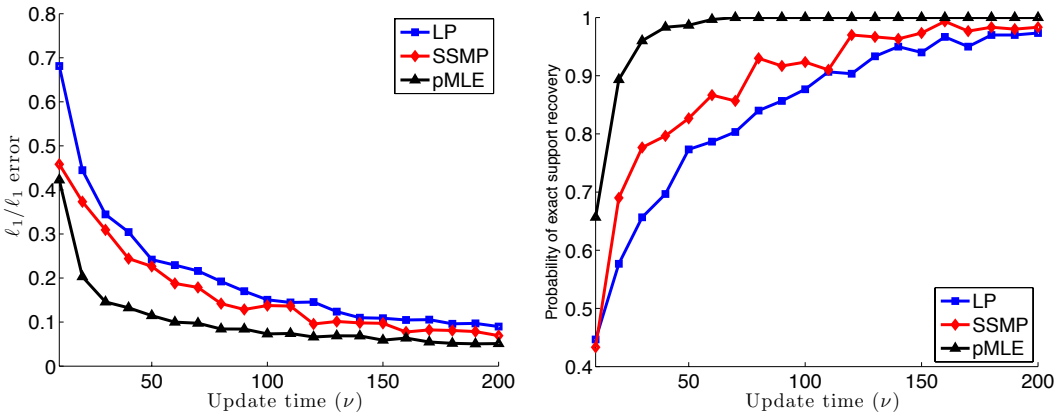
(a) $\beta = 1$

(b) $\beta = 1.5$



(c) $\beta = 2$

Figure 10.3: Probability of successful support recovery as a function of number of whales k, for ℓ_1-magic (LP), SSMP and pMLE for different choices of the power-law exponent β. The number of flows $N = 5000$, the number of counters $M = 800$, and the number of updates is 40.



(a) Relative ℓ_1 error as a function of number of updates ν.

(b) Probability of successful support recovery as a function of number of updates ν.

Figure 10.4: Performance of ℓ_1-magic, SSMP and pMLE algorithms as a function of the number of updates ν. The number of flows is $N = 5000$, the number of counters is $M = 800$, and the number of whales is $k = 30$. There are k whales whose magnitudes are assigned according to a power-law distribution with $\beta = 1$, and the remaining entries are minnows with magnitudes determined by a $\mathcal{N}(0, 10^{-6})$ random variable.

uniformly at random. Finally the $N-k$ minnows were sampled independently from a $\mathcal{N}(0, 10^{-6})$ distribution. Figure 10.2 shows the relative ℓ_1 error ($\|\boldsymbol{\lambda} - \widehat{\boldsymbol{\lambda}}_\nu\|_1/\|\boldsymbol{\lambda}\|_1$) of the three above algorithms as a function of k. Note that in all cases $\beta = 1$, $\beta = 1.5$, and $\beta = 2$, the pMLE algorithm provides lower ℓ_1 errors. Similarly, Figure 10.3 reports the probabilities of exact recovery as a function of k. Again, it turns out that in all three cases the pMLE algorithm has higher probability of exact support recovery compared to the two direct algorithms. We also analyzed the impact of changing the number of updates on the accuracy of the three above algorithms. The results are demonstrated in Figure 10.4. Here we fixed the number of whales to $k = 30$, and changed the number of updates from 10 to 200. It turned out that as the number of updates ν increases, the relative ℓ_1 errors of all three algorithms decrease and their probability of exact support recovery consistently increase. Moreover, the pMLE algorithm always outperforms the ℓ_1-magic (LP), and SSMP algorithms.

Part IV

Optimal Model-Selection via the Reed-Muller Frames

Chapter 11

Two Fundamental Measures of Coherence and Their Role in Model Selection

11.1 What is Model Selection?

11.1.1 Background

In compressed sensing, and in many other information processing and statistics problems involving high-dimensional data, the *curse of dimensionality* can often be broken by exploiting the fact that real-world data tend to live on low-dimensional manifolds. This phenomenon is exemplified by the important special case in which a data vector $\boldsymbol{\alpha}^* \in \mathbb{R}^N$ satisfies $\|\boldsymbol{\alpha}^*\|_0 \doteq \sum_{i=1}^N 1_{\{|\alpha_i^*|>0\}} \leq k \ll N$ and is observed according to the linear measurement model $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$. Here, $\boldsymbol{\Phi}$ is an $M \times N$ (real- or complex-valued) matrix called the *sensing or design* matrix, while $\boldsymbol{e_M} \in \mathbb{R}^M$ represents noise in the measurement system.

Fundamentally, given a measurement vector $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$ in the compressed setting, there are three complementary—but nonetheless distinct—questions that might be asked.

[**Sparse Approximation**] Under what conditions can we obtain a reliable estimate of a k-sparse $\hat{\boldsymbol{\alpha}}$ from $\boldsymbol{f}$?

[**Regression**] Under what conditions can we reliably approximate $\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}$ corresponding to a k-sparse $\hat{\boldsymbol{\alpha}}$ from $\boldsymbol{f}$?

[**Model Selection**] Under what conditions can we reliably recover the locations of the nonzero entries of a k-sparse $\hat{\boldsymbol{\alpha}}$ (in other words, the model $\mathcal{S} \doteq \{i \in \{1, \ldots, N\} : |\hat{\alpha}_i| > 0\}$) from $\boldsymbol{f}$?

Algorithm 8 The One-Step Thresholding (OST) Algorithm for Model Selection

Input: An $M \times N$ matrix $\boldsymbol{\Phi}$, a vector $\boldsymbol{f} \in \mathbb{C}^M$, and a threshold $\lambda > 0$
Output: An estimate $\widehat{\mathcal{S}} \subset \{1, \ldots, p\}$ of the true model $\mathcal{S}$

$\tilde{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\Phi}^\top \boldsymbol{f}$ {Form signal proxy}
$\widehat{\mathcal{S}} \leftarrow \{i \in \{1, \ldots, N\} : |\tilde{\alpha}_i| > \lambda\}$ {Select model via OST}

In Parts II and III of this thesis, we focused on efficient algorithms for (approximately) solving the sparse approximation and the regression problems. In many application areas, however, the model-selection question is equally—if not more—important than the other two questions. In particular, the problem of model selection (sometimes also known as *variable selection* or *sparsity pattern recovery*) arises indirectly in a number of contexts, such as subset selection in linear regression [193], estimation of structures in graphical models [192], and signal denoising [73]. In addition, solving the model-selection problem in some (but not all) cases also enables one to solve the sparse approximation and/or the regression problem.

11.1.2 Main Contributions

Model Selection: One of the primary objectives of this chapter is to study the problem of polynomial time, model-order agnostic *model selection* in a compressed setting for the general case of arbitrary (random or deterministic) design matrices and arbitrary nonzero entries of the signal. In order to accomplish this task, we introduce two fundamental measures of coherence among the (normalized) columns $\{\boldsymbol{\varphi_i} \in \mathbb{C}^M\}$ of the $M \times N$ design matrix $\boldsymbol{\Phi}$, namely,[1]

- *Worst-Case Coherence*: $\mu(\boldsymbol{\Phi}) \doteq \max\limits_{i,j:i \neq j} \left|\langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle\right|$, and
- *Average Coherence*: $\nu(\boldsymbol{\Phi}) \doteq \frac{1}{N-1} \max\limits_{i} \left| \sum\limits_{j:j \neq i} \langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle \right|$.

Roughly speaking, worst-case coherence—which has been introduced in Section 3.4.2—is a similarity measure between the columns of a design matrix: the smaller the worst-case coherence, the less similar the columns. On the other hand, average coherence is a measure of the spread of the columns of a design matrix within the M-dimensional unit ball: the smaller the average coherence, the more spread out the column vectors.

Our main contribution in the area of model selection is that we make use of these two measures of coherence to propose and analyze a model-order agnostic threshold

[1] Here, and throughout the rest of this chapter, we assume without loss of generality that $\boldsymbol{\Phi}$ has unit ℓ_2-norm columns. This is because deviations to this assumption can always be accounted for by appropriately scaling the entries of the data vector $\boldsymbol{\alpha}^*$ instead.

for the *one-step thresholding* (OST) algorithm (see Algorithm 8) for model selection. Specifically, we characterize in Section 11.2 both the exact and the partial model-selection performance of OST in a *non-asymptotic* setting in terms of μ and ν. In particular, we establish in Section 11.2 that if $\mu(\mathbf{\Phi}) \asymp M^{-1/2}$ and $\nu(\mathbf{\Phi}) \precsim M^{-1}$ then OST—despite being computationally primitive—can perform near-optimally for the case when either (i) the energy of any nonzero entry of $\boldsymbol{\alpha}^*$ is not too far away from the average signal energy per nonzero entry $\|\boldsymbol{\alpha}^*\|_2^2/k$ or (ii) the signal-to-noise ratio (SNR) in the measurement system is not too high. Equally importantly, in contrast to some of the existing literature on model selection, this analysis holds for arbitrary values of the nonzero entries of $\boldsymbol{\alpha}^*$ and it does not require the $M \times k$ submatrices of the design matrix $\mathbf{\Phi}$ to have full column rank.

11.1.3 Relationship to Previous Work

The problems of model selection and sparse-signal recovery in general and the use of OST (also known as *simple thresholding* [90] and *marginal regression* [120]) to solve these problems in particular have a rich history in the literature. In the context of model selection in the compressed setting, Mallow's C_N selection procedure [187] and the *Akaike information criterion* (AIC) [3]—both of which essentially attempt to solve a complexity-regularized version of the least-squares criterion—are considered to be seminal works, and are known to perform well empirically as well as theoretically; see, e.g., [188] and the references therein. These two procedures have been modified by numerous researchers over the years in order to improve their performance—the most notable variants being the *Bayesian information criterion* (BIC) [224] and the *risk inflation criterion* (RIC) [115]. Solving model-selection procedures such as C_N, AIC, BIC, and RIC, however, is known to be an NP-hard problem [197] even if the true model order k is made available to these procedures.

In order to overcome the computational intractability of these model-selection procedures, several methods based on convex optimization have been proposed by various researchers in recent years. Among these proposed methods, the LASSO [233] has arguably become the standard tool for model selection, which can be partly attributed to the theoretical guarantees provided for the LASSO in [192, 261, 247, 50]. In particular, the results reported in [192, 261] establish that the LASSO asymptotically identifies the correct model under certain conditions on the design matrix $\mathbf{\Phi}$ and the sparse vector $\boldsymbol{\alpha}^*$. Later, Wainwright in [247] strengthens the results of [192, 261] and makes explicit the dependence of exact model selection using the LASSO on the smallest (in magnitude) nonzero entry of $\boldsymbol{\alpha}^*$. However, apart from the fact that the results reported in [192, 261, 247, 221] are for exact model selection and are only asymptotic in nature, the main limitation of these works is that explicit verification of the conditions (such as the *irrepresentable condition* of [261] and the *incoherence condition* of [247]) that a generic design matrix $\mathbf{\Phi}$ needs to satisfy is computationally intractable for $k \succsim \mu^{-1}$. The most general (and non-asymptotic) model-selection results using the

LASSO for arbitrary design matrices have been reported in [50]. Specifically, Candès and Plan have established in [50] that the LASSO correctly identifies most models with probability $1 - O(N^{-1})$ under certain conditions on the smallest nonzero entry of $\boldsymbol{\alpha}^*$ provided: (i) the spectral norm (the largest singular value) and the worst-case coherence of $\boldsymbol{\Phi}$ are not too large, and (ii) the values of the nonzero entries of $\boldsymbol{\alpha}^*$ are independent and statistically symmetric around zero. Despite these recent theoretical triumphs of the LASSO, it is still desirable to study alternative solutions to the problem of polynomial time, model-order agnostic model selection in a compressed setting. This is because

1. LASSO solves a *detection* problem by solving a (more complicated) *estimation* problem.

2. LASSO requires the minimum singular values of the submatrices of $\boldsymbol{\Phi}$ corresponding to the true models to be bounded away from zero [192, 261, 247, 50]. While this is a plausible condition for the case when one is interested in estimating $\boldsymbol{\alpha}^*$, it is arguable whether this condition is necessary for the case of model selection.

3. The current literature on model selection using the LASSO lacks guarantees beyond $k \gtrsim \mu^{-1}$ for the case of generic design matrices and arbitrary nonzero entries. In particular, given an arbitrary design matrix $\boldsymbol{\Phi}$, [192, 261, 247, 50] do not provide any guarantees beyond $k \gtrsim \sqrt{M}$ for even the simple case of $\boldsymbol{\alpha}^* \in \mathbb{R}_+^N$.

4. The computational complexity of the LASSO for generic design matrices tends to be $O(N^3)$ [120]. This makes the LASSO computationally demanding for large-scale model-selection problems.

Recently, a few researchers have raised somewhat similar concerns about the LASSO and revisited the much older (and oft-forgotten) method of thresholding for model selection [223, 113, 214, 120], which has computational complexity of only one matrix-vector multiplication. Algorithmically, this makes our approach to model selection similar to that of [223, 113, 214, 120]. Nevertheless, the OST algorithm presented in this chapter differs from [223, 113, 214, 120] in five key aspects:

1. *Model-Order Agnostic Model Selection:* Unlike [223, 113, 214, 120], the OST algorithm presented in this chapter is completely agnostic to both the true model order k and any estimate of k.

2. *Generic Design Matrices and Arbitrary Nonzero Entries:* The results reported in this chapter hold for arbitrary (random or deterministic) design matrices and do not assume any statistical prior on the values of the nonzero entries of $\boldsymbol{\alpha}^*$ even when k scales linearly with M. In contrast, [113] only studies

the problem of Gaussian design matrices whereas the most influential results reported in [223, 214, 120] assume that the values of the nonzero entries of $\boldsymbol{\alpha}^*$ are independent and statistically symmetric around zero.

3. *Verifiable Sufficient Conditions:* In contrast to [223, 113, 214, 120], we relate the model-selection performance of OST to two global parameters of $\boldsymbol{\Phi}$, namely, μ and ν, which are trivially computable in polynomial time: $\mu(\boldsymbol{\Phi}) = \|\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{I}\|_\infty$ and $\nu(\boldsymbol{\Phi}) = \frac{1}{N-1}\|(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{I})\mathbf{1}\|_\infty$.

4. *Non-Asymptotic Theory:* Similar to [113, 214, 120], the analysis in this chapter can be used to establish that OST achieves (asymptotically) consistent model selection under certain conditions. However, the results reported in this chapter are completely non-asymptotic in nature (with explicit constants) and thereby shed light on the rate at which OST achieves consistent model selection.

5. *Partial Model Selection:* In addition to the exact model-selection performance of OST, we also characterize in the chapter its partial model-selection performance. In this regard, we establish that the *universal threshold* proposed in Section 11.2 for OST guarantees $\widehat{\mathcal{S}} \subset \mathcal{S}$ with high probability and we quantify the cardinality of the estimate $\widehat{\mathcal{S}}$. On the other hand, both [223] and [113] study only exact model selection, whereas [120, 214] study approximate (though not partial) model selection only for Gaussian design matrices [120] and assuming Gaussian (resp. statistical) priors on the nonzero entries of $\boldsymbol{\alpha}^*$ [214] (resp. [120]).

11.2 Model Selection Using One-Step Thresholding

11.2.1 Assumptions

Before presenting our results on model selection using OST, we need to be mathematically precise about our problem formulation. To this end, we begin by reconsidering the measurement model $\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e_M}$ and assume that $\boldsymbol{\Phi}$ is an $M \times N$ real- or complex-valued design matrix having unit ℓ_2-norm columns, $\boldsymbol{\alpha}^* \in \mathbb{C}^N$ is a k-sparse signal ($\|\boldsymbol{\alpha}^*\|_0 \leq k$), and $k < M \leq N$. Here, we allow $\boldsymbol{\Phi}$ to be either a random or a deterministic design matrix, while we take $\boldsymbol{e_M}$ to be a complex additive white Gaussian noise vector. It is worth mentioning that Gaussianity of $\boldsymbol{e_M}$ is just a simplified assumption for the sake of this exposition; in particular, the results presented in this section are readily generalizable to other noise distributions as well as perturbations having bounded ℓ_2-norms. Finally, the main assumption that we make here is that the true model $\mathcal{S} \doteq \{i \in \{1, \ldots, N\} : |\alpha_i^*| > 0\}$ is a uniformly random k-subset of $\{1, \ldots, N\}$. In other words, we have a uniform prior on the *support* of the data vector $\boldsymbol{\alpha}^*$.

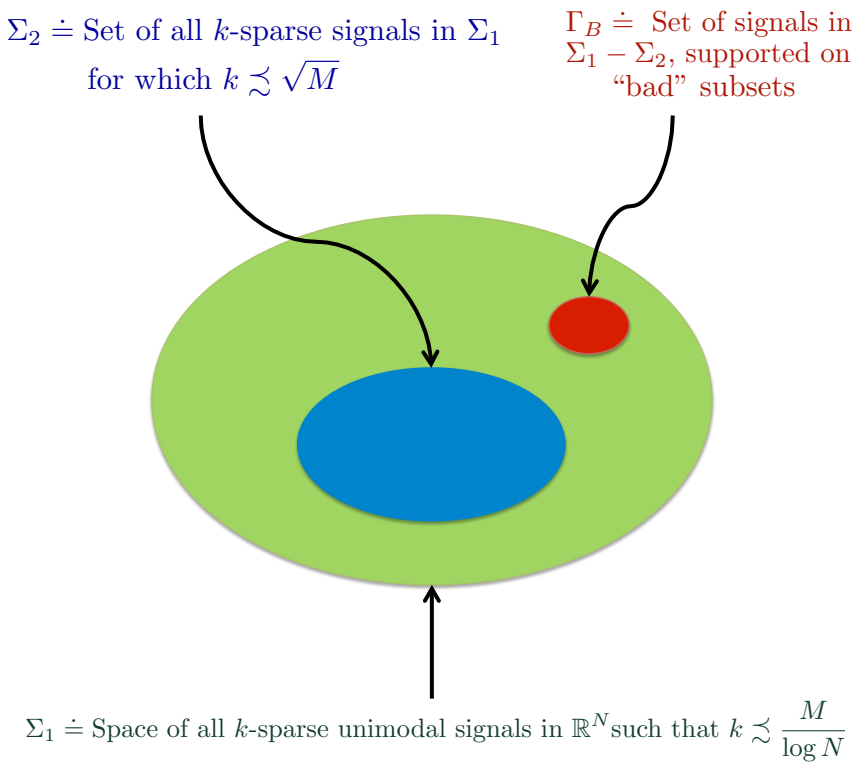


Figure 11.1: A Venn digram used to illustrate the major difference between the BP-based recovery guarantees and the OST-based recovery guarantees for k-sparse unimodal signals in $\mathbb{R}^N$ measured using Alltop Gabor frames. The OST algorithm is guaranteed to recover $\boldsymbol{\alpha}^* \in \Sigma_1 - \Gamma_B$. But BP, unlike OST, is only guaranteed to recover $\boldsymbol{\alpha}^* \in \Sigma_2$ in this case.

11.2.2 Main Results

Intuitively speaking, successful model selection requires the columns of the design matrix to be *incoherent*. In the case of the LASSO, this notion of incoherence has been quantified in [261] and [247] in terms of the *irrepresentable condition* and the *incoherence condition*, respectively (see also [50]). In contrast to earlier work on model selection, however, we formulate this idea of incoherence in terms of the *coherence property*.

Definition 11.1 (The Coherence Property)**.** *An $M \times N$ design matrix $\boldsymbol{\Phi}$ having unit ℓ_2-norm columns is said to obey the coherence property if the following two conditions*

hold:

$$\mu(\boldsymbol{\Phi}) \leq \frac{0.1}{\sqrt{2\log N}}, \quad \textit{and} \tag{CP-1}$$

$$\nu(\boldsymbol{\Phi}) \leq \frac{\mu}{\sqrt{M}}\,. \tag{CP-2}$$

In words, (CP-1) roughly states that the columns of $\boldsymbol{\Phi}$ are not too similar, while (CP-2) roughly states that the columns of $\boldsymbol{\Phi}$ are somewhat distributed within the M-dimensional unit ball. Note that the coherence property is superior to other measures of incoherence such as the irrepresentable condition in two key aspects. First, it does not require the singular values of the submatrices of $\boldsymbol{\Phi}$ to be bounded away from zero. Second, it can be easily verified in polynomial time since it simply requires checking that

$$\|\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{I}\|_\infty \leq (200\log N)^{-1/2} \text{ and } \|(\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{I})\mathbf{1}\|_\infty \leq (N-1)M^{-1/2}\|\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \boldsymbol{I}\|_\infty.$$

Below, we describe the implications of the coherence property for both the exact and the partial model-selection performance of OST. Before proceeding further, however, it is instructive to first define some fundamental quantities pertaining to the problem of model selection as follows:

$$\|\boldsymbol{\alpha}\|_{\min} \doteq \min_{i\in\mathcal{S}} |\alpha_i^*|\,, \qquad \text{MAR} \doteq \frac{{\|\boldsymbol{\alpha}\|_{\min}}^2}{\|\boldsymbol{\alpha}^*\|_2^2/k}\,,$$

$$\text{SNR}_{\min} \doteq \frac{{\|\boldsymbol{\alpha}\|_{\min}}^2}{\mathbb{E}[\|\boldsymbol{e_M}\|_2^2]/k}\,, \qquad \text{SNR} \doteq \frac{\|\boldsymbol{\alpha}^*\|_2^2}{\mathbb{E}[\|\boldsymbol{e_M}\|_2^2]}\,.$$

In words, $\|\boldsymbol{\alpha}\|_{\min}$ is the magnitude of the smallest nonzero entry of $\boldsymbol{\alpha}^*$, while MAR—which is termed the *minimum-to-average ratio* [113]—is the ratio of the energy in the smallest nonzero entry of $\boldsymbol{\alpha}^*$ and the *average signal energy per nonzero entry* of $\boldsymbol{\alpha}^*$. Likewise, $\text{SNR}_{\min}$ is the ratio of the energy in the smallest nonzero entry of $\boldsymbol{\alpha}^*$ and the average *noise* energy per nonzero entry, while SNR simply denotes the usual signal-to-noise ratio in the system. It is easy to see that $\text{SNR}_{\min} = \text{SNR}\cdot\text{MAR}$. We are now ready to state the first main result of this chapter that concerns the performance of OST in terms of exact model selection.

Theorem 11.2 (Exact Model Selection Using OST)**.** *Suppose that the design matrix* $\boldsymbol{\Phi}$ *satisfies the coherence property and let* $\boldsymbol{e_M}$ *be distributed as* $\mathcal{N}(\mathbf{0}, \sigma^2 I)$*. Next, choose the threshold* $\lambda = \max\left\{\frac{1}{t}10\mu\sqrt{M\cdot\text{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2\log N}$ *for any* $t \in (0,1)$*. Then, if we write* $\mu(\boldsymbol{\Phi})$ *as* $\mu = c_1 M^{-1/\gamma}$ *for some* $c_1 > 0$ *(which may depend on* N*) and* $\gamma \in \{0\}\cup[2,\infty)$*, the OST algorithm (Algorithm 8) satisfies* $\Pr(\widehat{\mathcal{S}} \neq \mathcal{S}) \leq 6N^{-1}$

provided $N \geq 128$ *and the number of measurements satisfies*

$$M > \max\left\{2k\log N, \frac{c_2 k\log N}{\text{SNR}_{\min}}, \left(\frac{c_3 k\log N}{\text{MAR}}\right)^{\gamma/2}\right\}$$
$$\equiv \max\left\{2k\log N, \frac{c_2 k\log N}{\text{SNR}\cdot\text{MAR}}, \left(\frac{c_3 k\log N}{\text{MAR}}\right)^{\gamma/2}\right\}. \tag{11.2.1}$$

Here, the quantities $c_2, c_3 > 0$ *are defined as* $c_2 \doteq 16(1-t)^{-2}$ *and* $c_3 \doteq 800c_1^2 t^{-2}$*, while the probability of failure is with respect to the true model* $\mathcal{S}$ *and the noise vector* $\boldsymbol{e_M}$*.*

The proof of this theorem is provided in Section 11.3. Note that the parameter t in Theorem 11.2 can always be fixed a priori (say $t = 1/2$) without affecting the scaling relation in (11.2.1). In practice, however, t should be chosen so as to reduce the total number of measurements needed to ensure successful model selection; the optimal choice of t in this regard is $t_{opt} = \arg\min_t \left(\max\left\{\frac{c_2 k\log N}{\text{SNR}\cdot\text{MAR}}, \left(\frac{c_3 k\log N}{\text{MAR}}\right)^{\gamma/2}\right\}\right)$. Notice also that Theorem 11.2 is best suited for applications where one is interested in quantifying the minimum number of measurements needed to guarantee exact model selection for a given class of signals. Alternatively, it might be the case in some other applications that the problem dimensions are fixed and one is instead interested in specifying the class of signals that leads to successful model selection. The following variant of Theorem 11.2 is best suited in such situations.

Theorem 11.3. *Suppose that the design matrix* $\boldsymbol{\Phi}$ *satisfies the coherence property and let the noise vector* $\boldsymbol{e_M}$ *be distributed as* $\mathcal{N}(\mathbf{0}, \sigma^2 I)$*. Next, let* $N \geq 128$ *and choose the threshold* $\lambda = \max\left\{\frac{1}{t}10\mu\sqrt{M\cdot\text{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2\log N}$ *for any* $t \in (0,1)$*. Then the OST algorithm (Algorithm 8) satisfies* $\Pr(\widehat{\mathcal{S}} \neq \mathcal{S}) \leq 6N^{-1}$ *as long as we have that* $k \leq M/(2\log N)$ *and*

$$\text{MAR} > \max\left\{\frac{c_2 k\log N}{M\cdot\text{SNR}}, \frac{c_3' k\log N}{\mu^{-2}}\right\}. \tag{11.2.2}$$

Here, $c_2 > 0$ *is as defined in Theorem 11.2,* $c_3' > 0$ *is defined as* $c_3' \doteq 800t^{-2}$*, and the probability of failure is with respect to the true model* $\mathcal{S}$ *and the noise vector* $\boldsymbol{e_M}$*.*

Note that the proof of Theorem 11.3 follows directly from the proof of Theorem 11.2. There are a few important remarks that need to be made at this point concerning the threshold proposed in Theorem 11.2 and Theorem 11.3 for the OST algorithm. First, it is easy to see that the proposed threshold is completely agnostic to the model order k and only requires knowledge of the SNR and the noise variance. Second, extensive simulations suggest that the absolute constant 10 in the proposed threshold is somewhat conservative and can be reduced through the use of more sophisticated analytical tools. Finally, while estimating the true model order k tends to be harder

Algorithm 9 The Sorted One-Step Thresholding (SOST) Algorithm for Model Selection

Input: An $M \times N$ matrix $\boldsymbol{\Phi}$, a vector $\boldsymbol{f} \in \mathbb{C}^M$, and model order k
Output: An estimate $\widehat{\mathcal{S}} \subset \{1, \ldots, N\}$ of the true model $\mathcal{S}$

$\tilde{\boldsymbol{\alpha}} \leftarrow \mathrm{H}_k(\boldsymbol{\Phi}^\top \boldsymbol{f})$ {Form signal proxy}
$\widehat{\mathcal{S}} \leftarrow \mathrm{Supp}(\tilde{\boldsymbol{\alpha}})$

than estimating the SNR and the noise variance σ^2 in majority of the situations, it might be the case that estimating k is easier in some applications. It is better in such situations to work with a slight variant of the OST algorithm (see Algorithm 9) that relies on knowledge of the model order k instead and returns an estimate $\widehat{\mathcal{S}}$ corresponding to the k largest (in magnitude) entries of $\boldsymbol{\Phi}^\top \boldsymbol{f}$. We characterize the performance of this algorithm—which we call *sorted one-step thresholding* (SOST) algorithm—in terms of the following theorem.

Theorem 11.4 (Exact Model Selection Using SOST)**.** *Suppose that the design matrix $\boldsymbol{\Phi}$ satisfies the coherence property and let the noise vector $\boldsymbol{e_M}$ be distributed as $\mathcal{N}(\boldsymbol{0}, \sigma^2 I)$. Next, write $\mu(\boldsymbol{\Phi})$ as $\mu = c_1 M^{-1/\gamma}$ for some $c_1 > 0$ (which may depend on N) and $\gamma \in \{0\} \cup [2, \infty)$. Then the SOST algorithm (Algorithm 9) satisfies $\Pr(\widehat{\mathcal{S}} \neq \mathcal{S}) \leq 6N^{-1}$ as long as $N \geq 128$ and the number of measurements satisfies*

$$M > \min_{t \in (0,1)} \max \left\{ 2k \log N, \frac{c_2 k \log N}{\mathrm{SNR}_{\min}}, \left(\frac{c_3 k \log N}{\mathrm{MAR}} \right)^{\gamma/2} \right\}$$
$$\equiv \min_{t \in (0,1)} \max \left\{ 2k \log N, \frac{c_2 k \log N}{\mathrm{SNR} \cdot \mathrm{MAR}}, \left(\frac{c_3 k \log N}{\mathrm{MAR}} \right)^{\gamma/2} \right\}. \tag{11.2.3}$$

Here, the quantities $c_2, c_3 > 0$ are as defined in Theorem 11.2, while the probability of failure is with respect to the true model $\mathcal{S}$ and the noise vector $\boldsymbol{e_M}$.

The final result that we present in this section concerns the partial model-selection performance of OST. Specifically, note that our focus in this section has so far been on specifying conditions for either the number of measurements or the MAR of the signal that ensure exact model selection. In many real-world applications, however, the parameters of the problem are fixed and it is not always possible to ensure that either the number of measurements or the MAR of the signal satisfy the aforementioned conditions. A natural question to ask then is whether the OST algorithm completely fails in such circumstances or whether any guarantees can still be provided for its performance. We address this aspect of the OST algorithm in the following and show that, even if the MAR of $\boldsymbol{\alpha}^*$ is very small, OST has the ability to identify the locations of the nonzero entries of $\boldsymbol{\alpha}^*$ whose energies are greater than both the noise power and the average signal energy per nonzero entry. In order to make this notion mathematically precise, we first define the *l-th largest-to-average ratio* (LAR_l) of $\boldsymbol{\alpha}^*$

as the ratio of the *energy in the l-th largest (in magnitude) nonzero entry of* $\boldsymbol{\alpha}^*$ and the average signal energy per nonzero entry of $\boldsymbol{\alpha}^*$; that is,

$$\text{LAR}_l \doteq \frac{|\alpha^*_{(l)}|^2}{\|\boldsymbol{\alpha}^*\|_2^2/k}$$

where $\alpha^*_{(l)}$ denotes the l-th largest nonzero entry of $\boldsymbol{\alpha}^*$ (note that $\text{MAR} \equiv \text{LAR}_k$). We are now ready to specify the partial model-selection performance of the OST algorithm.

Theorem 11.5 (Partial Model Selection Using OST)**.** *Suppose that the design matrix* $\boldsymbol{\Phi}$ *satisfies the coherence property. Next, let* $N \geq 128$ *and* $\boldsymbol{e_M}$ *be distributed as* $\mathcal{N}(\mathbf{0}, \sigma^2 I)$*. Finally, fix a parameter* $t \in (0,1)$ *and choose the threshold* $\lambda = \max\left\{\frac{1}{t}10\mu\sqrt{M \cdot \text{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2 \log N}$*. Then, under the assumption that* $k \leq M/(2\log N)$*, the OST algorithm (Algorithm 8) guarantees with probability exceeding* $1 - 6N^{-1}$ *that* $\widehat{\mathcal{S}} \subset \mathcal{S}$ *and* $\left|\mathcal{S} - \widehat{\mathcal{S}}\right| \leq (k - L)$*, where* L *is the largest integer for which the following inequality holds:*

$$\text{LAR}_L > \max\left\{\frac{c_2 k \log N}{M \cdot \text{SNR}}, \frac{c_3' k \log N}{\mu^{-2}}\right\}. \tag{11.2.4}$$

Here, the quantities $c_2, c_3' > 0$ *are as defined in Theorem 11.3, while the probability of failure is with respect to the true model* $\mathcal{S}$ *and the noise vector* $\boldsymbol{e_M}$*.*

11.2.3 LASSO versus OST

Historically, OST (and its variants) was preferred over the LASSO because of its low computational complexity. The results reported in this chapter, however, bring forth another important aspect of OST (also see [120]): *OST can lead to successful model selection even when the LASSO fails.* Specifically, model selection using the LASSO is in fact a byproduct of signal reconstruction, whereas the OST results do not guarantee signal reconstruction without imposing additional constraints on $\boldsymbol{\Phi}$.

In Chapter 12 we will introduce the Reed-Muller frames as examples of design matrices with optimal coherence parameters. We will then show cases in which LASSO completely fails in recovering the support of a sparse vector, whereas the OST algorithm can successfully recover the support of the same sparse vector. In other words, *model selection is inherently an easier problem than signal reconstruction.*

11.3 Proofs of Main Results

In this section, we provide detailed proofs of the main results reported in Section 11.2. Before proceeding further, however, it is advantageous to develop some notation that

will facilitate our forthcoming analysis. In this regard, recall that the true model $\mathcal{S}$ is taken to be a uniformly random k-subset of $[N] \doteq \{1, \ldots, N\}$. We can therefore write the data vector $\boldsymbol{\alpha}^*$ under this assumption as concatenation of a random permutation matrix and a deterministic k-sparse vector. Specifically, let $\bar{\boldsymbol{z}} \in \mathbb{C}^N$ be a *deterministic* k-sparse vector that we write (without loss of generality) as

$$\bar{\boldsymbol{z}} \doteq \Big(\underbrace{z_1, \ldots, z_k}_{\doteq \boldsymbol{z} \in \mathbb{C}^k}, \underbrace{0, \ldots, 0}_{(N-k) \text{ times}} \Big)^{\mathrm{T}} \tag{11.3.1}$$

and let $\boldsymbol{P}_\pi$ be an $N \times N$ random permutation matrix; in other words,

$$\boldsymbol{P}_\pi \doteq \begin{bmatrix} \mathbf{e}_{\boldsymbol{\pi_1}} & \mathbf{e}_{\boldsymbol{\pi_2}} & \ldots & \mathbf{e}_{\boldsymbol{\pi_N}} \end{bmatrix}^\top \tag{11.3.2}$$

where $\mathbf{e}_{\boldsymbol{j}}$ denotes the j-th column of the canonical basis I and $\bar{\Pi} \doteq (\pi_1, \ldots, \pi_N)$ is a random permutation of $[N]$. Then the assumption that the model $\mathcal{S}$ is a random subset of $[N]$ is equivalent to stating that the data vector $\boldsymbol{\alpha}^*$ can be written as $\boldsymbol{\alpha}^* = \boldsymbol{P}_{\boldsymbol{\pi}} \bar{\boldsymbol{z}}$. In other words, the measurement vector $\boldsymbol{f}$ can be expressed as

$$\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{\alpha}^* + \boldsymbol{e}_{\boldsymbol{M}} = \boldsymbol{\Phi}\boldsymbol{P}_{\boldsymbol{\pi}}\bar{\boldsymbol{z}} + \boldsymbol{e}_{\boldsymbol{M}} = \boldsymbol{\Phi}_\Pi \boldsymbol{z} + \boldsymbol{e}_{\boldsymbol{M}} \tag{11.3.3}$$

where $\Pi \doteq (\pi_1, \ldots, \pi_k)$ denotes the first k elements of the random permutation $\bar{\Pi}$, $\boldsymbol{\Phi}_\Pi$ denotes the $M \times k$ submatrix obtained by collecting the columns of $\boldsymbol{\Phi}$ corresponding to the indices in Π, and the vector $\boldsymbol{z} \in \mathbb{C}^k$ represents the k nonzero entries of $\boldsymbol{\alpha}^*$.

Proof of Theorem 11.2

The general road map for the proof of Theorem 11.2 is as follows. Below, we first introduce the notion of (k, ϵ, δ)-*statistical orthogonality condition* (StOC). We next establish the relationship between the StOC parameters and the worst-case and average coherence of $\boldsymbol{\Phi}$ in Lemma 11.8 and Lemma 11.9. We then provide a proof of Theorem 11.2 by first showing that if $\boldsymbol{\Phi}$ satisfies the StOC then OST recovers $\mathcal{S}$ with high probability and then relating the results of Lemma 11.8 and Lemma 11.9 to the coherence property.

Definition 11.6 (**(k, ϵ, δ)**-Statistical Orthogonality Condition)**.** *Let $\bar{\Pi} = (\pi_1, \ldots, \pi_N)$ be a random permutation of $[N]$, and define $\Pi \doteq (\pi_1, \ldots, \pi_k)$ and $\Pi^c \doteq (\pi_{k+1}, \ldots, \pi_N)$ for any $k \in [N]$. Then the $M \times N$ (normalized) design matrix $\boldsymbol{\Phi}$ is said to satisfy the (k, ϵ, δ)-statistical orthogonality condition if there exist $\epsilon, \delta \in [0, 1)$ such that the inequalities*

$$\|(\boldsymbol{\Phi}_\Pi^\top \boldsymbol{\Phi}_\Pi - \boldsymbol{I})\boldsymbol{z}\|_\infty \leq \epsilon \|\boldsymbol{z}\|_2 \tag{StOC-1}$$

$$\|\boldsymbol{\Phi}_{\Pi^c}^\top \boldsymbol{\Phi}_\Pi \boldsymbol{z}\|_\infty \leq \epsilon \|\boldsymbol{z}\|_2 \tag{StOC-2}$$

hold for every fixed $\boldsymbol{z} \in \mathbb{C}^k$ *with probability exceeding* $1 - \delta$ *(with respect to the random permutation* $\bar{\Pi}$*).*

Remark 11.7. *Note that the StOC derives its name from the fact that if* $\boldsymbol{\Phi}$ *is a* $N \times N$ *orthonormal matrix then it trivially satisfies the StOC for every* $k \in [N]$ *with* $\epsilon = \delta = 0$. *In addition, although we will not use this fact explicitly in the chapter, it can be checked that if* $\boldsymbol{\Phi}$ *satisfies* (k, ϵ, δ)*-StoC then it* approximately *preserves the* ℓ_2*-norms of* k*-sparse signals with probability exceeding* $1 - \delta$ *as long as* $k < \epsilon^{-2}$.

Having defined StOC, our goal in the next two lemmas is to relate the StOC parameters k, ϵ, and δ to the worst-case and average coherence of the design matrix $\boldsymbol{\Phi}$.

Lemma 11.8. *Let* $\Pi = (\pi_1, \ldots, \pi_k)$ *denote the first* k *elements of a random permutation of* $[N]$ *and choose a parameter* $a \geq 1$. *Then, for any* $\epsilon \in [0, 1)$, $k \leq \min\big\{\epsilon^2 \nu^{-2}, (1+a)^{-1} N\big\}$, *and fixed* $\boldsymbol{z} \in \mathbb{C}^k$, *we have*

$$\Pr\Big(\big\{\boldsymbol{\Phi} \text{ does not satisfy (StOC-1)}\big\}\Big) \leq 4k \exp\bigg(-\frac{(\epsilon - \sqrt{k}\,\nu)^2}{16(2 + a^{-1})^2 \mu^2}\bigg).$$

Proof. The proof of this lemma relies heavily on the so-called *method of bounded differences* (MOBD) [190]. Specifically, we begin by noting that $\big\|(\boldsymbol{\Phi}_\Pi^\top \boldsymbol{\Phi}_\Pi - I)\boldsymbol{z}\big\|_\infty = \max_i \big|\sum_{j \neq i} z_j \langle \boldsymbol{\varphi}_{\boldsymbol{\pi_i}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle\big|$. Therefore for a fixed index i, and conditioned on the event $\mathcal{A}_{i'} \doteq \big\{\pi_i = i'\big\}$, we have the following equality from basic probability theory

$$\Pr\bigg(\Big|\sum_{\substack{j=1 \\ j \neq i}}^{k} z_j \langle \boldsymbol{\varphi}_{\boldsymbol{\pi_i}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle\Big| > \epsilon \|\boldsymbol{z}\|_2 \bigg| \mathcal{A}_{i'}\bigg) = \Pr\bigg(\Big|\sum_{\substack{j=1 \\ j \neq i}}^{k} z_j \langle \boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle\Big| > \epsilon \|\boldsymbol{z}\|_2 \bigg| \mathcal{A}_{i'}\bigg). \tag{11.3.4}$$

Next, in order to apply the MOBD to obtain an upper bound for (11.3.4), we first define a random $(k-1)$-tuple $\Pi^{-i} \doteq (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_k)$ and then construct a Doob martingale $(Z_0, Z_1, \ldots, Z_{k-1})$ as follows:

$$\begin{aligned} Z_0 &= \mathbb{E}\Big[\sum_{\substack{j=1 \\ j \neq i}}^{k} z_j \langle \boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle \Big| \mathcal{A}_{i'}\Big], \quad \text{and} \\ Z_\ell &= \mathbb{E}\Big[\sum_{\substack{j=1 \\ j \neq i}}^{k} z_j \langle \boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle \Big| \pi_{1\to\ell}^{-i}, \mathcal{A}_{i'}\Big], \ \ell = 1, \ldots, k-1 \end{aligned} \tag{11.3.5}$$

where $\pi_{1\to\ell}^{-i}$ denotes the first ℓ elements of Π^{-i}. The first thing to note here is that we have from the linearity of (conditional) expectation

$$\big|Z_0\big| = \Big|\sum_{j \neq i} z_j \mathbb{E}\big[\langle \boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle | \mathcal{A}_{i'}\big]\Big| \leq \sum_{j \neq i} \big|z_j\big| \Big|\mathbb{E}\big[\langle \boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}} \rangle | \mathcal{A}_{i'}\big]\Big|$$

$$\overset{(a)}{\leq} \sum_{j\neq i} \big|z_j\big| \left| \sum_{\substack{q=1\\ q\neq i'}}^{N} \frac{1}{N-1} \langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_q} \rangle \right| \overset{(b)}{\leq} \nu \, \|\boldsymbol{z}\|_1 \leq \sqrt{k}\, \nu \, \|\boldsymbol{z}\|_2$$

where (a) follows from the fact that, conditioned on $\mathcal{A}_{i'}$, π_j has a uniform distribution over $[N]-\{i'\}$, while (b) is mainly a consequence of the definition of average coherence. In addition, if we use π_ℓ^{-i} to denote the ℓ-th element of Π^{-i} and define

$$Z_\ell(r) \doteq \mathbb{E}\Big[\sum_{\substack{j=1\\ j\neq i}}^{k} z_j \langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}} \rangle \Big| \pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = r, \mathcal{A}_{i'} \Big] \tag{11.3.6}$$

for $\ell = 1, \ldots, k-1$ then, since $(Z_0, Z_1, \ldots, Z_{k-1})$ is a Doob martingale, it can be easily verified that $|Z_\ell - Z_{\ell-1}|$ is upper bounded by $\sup_{r,s} \big[Z_\ell(r) - Z_\ell(s)\big]$ (see, e.g., [195]).

Now in order to upper bound $\sup_{r,s} \big[Z_\ell(r) - Z_\ell(s)\big]$, notice that we can bound $|Z_\ell(r) - Z_\ell(s)|$:

$$\begin{aligned}
&\Big|Z_\ell(r) - Z_\ell(s)\Big| \\
&= \left| \sum_{j\neq i} z_j \left(\mathbb{E}\Big[\langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}} \rangle \Big| \pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = r, \mathcal{A}_{i'} \Big] - \mathbb{E}\Big[\langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}} \rangle \Big| \pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = s, \mathcal{A}_{i'} \Big] \right) \right| \\
&\leq \sum_{j\neq i} \big|z_j\big| \Bigg| \underbrace{\mathbb{E}\Big[\langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}} \rangle \Big| \pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = r, \mathcal{A}_{i'} \Big] - \mathbb{E}\Big[\langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}} \rangle \Big| \pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = s, \mathcal{A}_{i'} \Big]}_{\doteq d_{\ell,j}} \Bigg| \\
&= \sum_{\substack{j\leq \ell+1\\ j\neq i}} \big|z_j\big|\big|d_{\ell,j}\big| + \sum_{\substack{j> \ell+1\\ j\neq i}} \big|z_j\big|\big|d_{\ell,j}\big|.
\end{aligned} \tag{11.3.7}$$

In addition, we have that for every $j > \ell+1, j \neq i$, the random variable π_j has a uniform distribution over $[N] - \{\pi_{1\to\ell-1}^{-i}, r, i'\}$ when conditioned on $\{\pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = r, i'\}$, whereas π_j has a uniform distribution over $[N]-\{\pi_{1\to\ell-1}^{-i}, s, i'\}$ when conditioned on $\{\pi_{1\to\ell-1}^{-i}, \pi_\ell^{-i} = s, i'\}$. Therefore, we get $\forall\ j > \ell+1, j \neq i$,

$$|d_{\ell,j}| = \frac{1}{N-\ell-1} \Big| \langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_r} \rangle - \langle \boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_s} \rangle \Big| \leq \frac{2\mu}{N-\ell-1}. \tag{11.3.8}$$

Similarly, it can be shown that

$$\sum_{\substack{j\leq \ell+1\\ j\neq i}} \big|z_j\big|\big|d_{\ell,j}\big| \leq \big|z_{\ell+1}\big| 2\mu \text{ when } i \leq \ell, \quad \sum_{\substack{j\leq \ell+1\\ j\neq i}} \big|z_j\big|\big|d_{\ell,j}\big| \leq \big|z_\ell\big| 2\mu \text{ when } i = \ell+1,$$

and $\sum_{\substack{j\le \ell+1\\ j\neq i}} \big|z_j\big|\big|d_{\ell,j}\big| \le (|z_\ell| + \frac{|z_{\ell+1}|}{N-\ell-1})2\mu$ when $i > \ell+1$. Consequently, regardless of the initial choice of i, we obtain

$$\sup_{r,s}\big[Z_\ell(r) - Z_\ell(s)\big] \le 2\mu\Big(\underbrace{|z_\ell| + |z_{\ell+1}| + \frac{1}{N-\ell-1}\sum_{j>\ell+1}|z_j|}_{\doteq d_\ell}\Big). \tag{11.3.9}$$

We have now established that $(Z_0, Z_1, \ldots, Z_{k-1})$ is a (real- or complex-valued) bounded-difference martingale sequence with $|Z_\ell - Z_{\ell-1}| \le 2\mu d_\ell$ for $\ell = 1, \ldots, k-1$. Therefore, under the assumption that $k \le \epsilon^2\nu^{-2}$ and since it has been established in (11.3.5) that $|Z_0| \le \sqrt{k}\,\nu\,\|\boldsymbol{z}\|_2$, it is easy to see that

$$\begin{aligned}\Pr\left(\Big|\sum_{\substack{j=1\\ j\neq i}}^{k} z_j\langle\boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right) &\le \Pr\left(\big|Z_{k-1} - Z_0| > \epsilon\|\boldsymbol{z}\|_2 - \sqrt{k}\,\nu\,\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right)\\ &\overset{(c)}{\le} 4\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2\|\boldsymbol{z}\|_2^2}{16\mu^2\sum_{\ell=1}^{k-1} d_\ell^2}\right)\end{aligned} \tag{11.3.10}$$

where (c) follows from the complex Azuma inequality for bounded-difference martingale sequences (see Theorem 2.16 in Chapter 2). Further, it can be established through routine calculations from (11.3.9) that $\sum_{\ell=1}^{k-1} d_\ell^2 \le (2 + a^{-1})^2\|\boldsymbol{z}\|_2^2$ since $k \le N/(1+a)$. Combining all these facts together, we finally obtain that

$$\begin{aligned}\Pr\left(\big\|(\boldsymbol{\Phi}_\Pi^\top\boldsymbol{\Phi}_\Pi - \boldsymbol{I})\boldsymbol{z}\big\|_\infty > \epsilon\|\boldsymbol{z}\|_2\right) &\overset{(d)}{\le} k\,\Pr\left(\Big|\sum_{\substack{j=1\\ j\neq i}}^{k} z_j\langle\boldsymbol{\varphi_{\pi_i}}, \boldsymbol{\varphi_{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\right)\\ &= k\sum_{i'=1}^{p}\Pr\left(\Big|\sum_{\substack{j=1\\ j\neq i}}^{k} z_j\langle\boldsymbol{\varphi_{i'}}, \boldsymbol{\varphi_{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right)\Pr\left(\mathcal{A}_{i'}\right) \overset{(e)}{\le} 4k\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2}{16(2+a^{-1})^2\mu^2}\right)\end{aligned}$$

where (d) follows from the union bound and the fact that the π_i's are identically (though not independently) distributed, while (e) follows from (11.3.10) and the fact that π_i has a uniform distribution over $[N]$. □

Lemma 11.9. *Let $\Pi = (\pi_1, \ldots, \pi_k)$ and $\Pi^c = (\pi_{k+1}, \ldots, \pi_p)$ denote the first k and the last $(N-k)$ elements of a random permutation of $[N]$, respectively, and choose a parameter $a \ge 1$. Then, for any $\epsilon \in [0, 1)$, $k \le \min\big\{\epsilon^2\nu^{-2}, (1+a)^{-1}N\big\}$, and fixed $\boldsymbol{z} \in \mathbb{C}^k$, we have*

$$\Pr\Big(\big\{\boldsymbol{\Phi} \text{ does not satisfy (StOC-2)}\big\}\Big) \le 4(N-k)\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2}{8(1+a^{-1})^2\mu^2}\right).$$

Proof. The proof of this lemma is very similar to that of Lemma 11.8 and also relies on the MOBD. To begin with, we note that $\left\|\boldsymbol{\Phi}_{\Pi^c}^{\top}\boldsymbol{\Phi}_{\Pi}\boldsymbol{z}\right\|_{\infty} = \max_{i\in[N-k]}\left|\sum_j z_j\langle\boldsymbol{\varphi}_{\boldsymbol{\pi}_i^c},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\right|$, where $[N-k]\doteq\{1,\ldots,N-k\}$ and π_i^c denotes the i-th element of Π^c. Then for a fixed index $i\in[N-k]$, and conditioned on the event $\mathcal{A}_{i'}\doteq\{\pi_i^c=i'\}$, we again have the following simple equality

$$\Pr\left(\left|\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{\pi}_i^c},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\right|>\epsilon\|\boldsymbol{z}\|_2\middle|\mathcal{A}_{i'}\right)=\Pr\left(\left|\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\right|>\epsilon\|\boldsymbol{z}\|_2\middle|\mathcal{A}_{i'}\right). \tag{11.3.11}$$

Next, as in the case of Lemma 11.8, we construct a Doob martingale sequence $(Z_0,Z_1,\ldots,Z_k)$ as follows:

$$\begin{aligned} Z_0 &= \mathbb{E}\Big[\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\Big|\mathcal{A}_{i'}\Big] \quad \text{and} \\ Z_\ell &= \mathbb{E}\Big[\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\Big|\pi_{1\to\ell},\mathcal{A}_{i'}\Big],\ \ell=1,\ldots,k \end{aligned}$$

where $\pi_{1\to\ell}$ now denotes the first ℓ elements of Π. Then, since π_j has a uniform distribution over $[N]-\{i'\}$ when conditioned on $\mathcal{A}_{i'}$, we once again have the bound $\left|Z_0\right|\le\sqrt{k}\,\nu\,\|\boldsymbol{z}\|_2$. Therefore, the only remaining thing that we need to show in order to be able to apply the complex Azuma inequality to the constructed martingale $(Z_0,Z_1,\ldots,Z_k)$ is that $|Z_\ell-Z_{\ell-1}|$ is suitably bounded.

In this regard, we make use of the notation

$$Z_\ell(r)\doteq\mathbb{E}\big[\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\Big|\pi_{1\to\ell-1},\pi_\ell=r,\mathcal{A}_{i'}\big]$$

and note that $|Z_\ell(r)-Z_\ell(s)|$ can be bounded as

$$\begin{aligned} &\Big|Z_\ell(r)-Z_\ell(s)\Big| \\ &=\left|\sum_j z_j\left(\mathbb{E}\Big[\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\Big|\pi_{1\to\ell-1},\pi_\ell=r,\mathcal{A}_{i'}\Big]-\mathbb{E}\Big[\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{\pi}_j}\rangle\Big|\pi_{1\to\ell-1},\pi_\ell=s,\mathcal{A}_{i'}\Big]\right)\right| \\ &\le\big|z_\ell\big|\Big|\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{r}}\rangle-\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{s}}\rangle\Big|+\frac{\Big|\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{r}}\rangle-\langle\boldsymbol{\varphi}_{\boldsymbol{i'}},\boldsymbol{\varphi}_{\boldsymbol{s}}\rangle\Big|}{N-\ell-1}\sum_{j>\ell}\big|z_j\big|\le 2\mu\Big(\underbrace{|z_\ell|+\frac{\sum_{j>\ell}|z_j|}{N-\ell-1}}_{\doteq d_\ell}\Big). \end{aligned} \tag{11.3.12}$$

which implies that $\sup_{r,s}\big[Z_\ell(r) - Z_\ell(s)\big] \leq 2\mu d_\ell,\ \ell = 1,\ldots,k$. Consequently, we have now established that $(Z_0, Z_1, \ldots, Z_k)$ is a bounded-difference martingale with $|Z_\ell - Z_{\ell-1}| \leq 2\mu d_\ell$. Therefore, since $k \leq \epsilon^2\nu^{-2}$ and $|Z_0| \leq \sqrt{k}\,\nu\,\|\boldsymbol{z}\|_2$, we once again have from the complex Azuma inequality that

$$\begin{aligned}\Pr\left(\Big|\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right) &\leq \Pr\left(\big|Z_k - Z_0| > \epsilon\|\boldsymbol{z}\|_2 - \sqrt{k}\,\nu\,\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right)\\ &\overset{(a)}{\leq} 4\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2}{8(1+a^{-1})^2\mu^2}\right) \qquad (11.3.13)\end{aligned}$$

where (a) follows by noting that $\sum_{\ell=1}^{k} d_\ell^2 \leq (1+a^{-1})^2\|\boldsymbol{z}\|_2^2$ since $k \leq N/(1+a)$. Combining all these facts together, we finally obtain the claimed result as follows

$$\begin{aligned}&\Pr\left(\left\|\boldsymbol{\Phi}_{\Pi^c}^{\top}\boldsymbol{\Phi}_{\Pi}\boldsymbol{z}\right\|_\infty > \epsilon\|\boldsymbol{z}\|_2\right)\\ &\overset{(b)}{\leq} (N-k)\,\Pr\left(\Big|\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{\pi_i^c}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\right)\\ &\leq (N-k)\sum_{i'=1}^{N}\Pr\left(\Big|\sum_{j=1}^{k} z_j\langle\boldsymbol{\varphi}_{\boldsymbol{i'}}, \boldsymbol{\varphi}_{\boldsymbol{\pi_j}}\rangle\Big| > \epsilon\|\boldsymbol{z}\|_2\Big|\mathcal{A}_{i'}\right)\Pr\left(\mathcal{A}_{i'}\right)\\ &\overset{(c)}{\leq} 4(N-k)\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2}{8(1+a^{-1})^2\mu^2}\right) \qquad (11.3.14)\end{aligned}$$

where (b) follows from the union bound and the fact that the π_i^c's are identically (though not independently) distributed, while (c) follows from (11.3.13) and the fact that π_i^c has a uniform distribution over $[N]$. □

Note that Lemma 11.8 and Lemma 11.9 collectively prove through a simple union bound argument that an $M \times N$ design matrix $\boldsymbol{\Phi}$ satisfies (k,ϵ,δ)-StOC for any $\epsilon \in [0,1)$ with $\delta \leq 4N\exp\left(-\frac{(\epsilon-\sqrt{k}\,\nu)^2}{16(2+a^{-1})^2\mu^2}\right)$ for any $a \geq 1$ as long as we have that $k \leq \min\left\{\epsilon^2\nu^{-2}, (1+a)^{-1}N\right\}$. We are now ready to provide a proof of Theorem 11.2.

Proof of Theorem 11.2

We begin by making use of the notation developed at the start of this section and writing the signal proxy $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\Phi}^{\top}\boldsymbol{f}$ as $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}_{\Pi}\boldsymbol{z} + \boldsymbol{\Phi}^{\top}\boldsymbol{e}_{\boldsymbol{M}}$. Now, let $\Pi^c = (\pi_{k+1}, \ldots, \pi_N)$ denote the last $(N-k)$ elements of $\bar{\Pi}$ and note that we need to show that $\|\tilde{\boldsymbol{\alpha}}_{\Pi^c}\|_\infty \leq \lambda$ and $\min_{i\in\{1,\ldots,k\}}|\tilde{\boldsymbol{\alpha}}_{\pi_i}| > \lambda$ in order to establish that $\widehat{\mathcal{S}} = \mathcal{S}$.

In this regard, we first assume that $\boldsymbol{\Phi}$ satisfies (k,ϵ,δ)-StOC and define

$$\lambda_\epsilon \doteq \max\left\{\frac{1}{t}\epsilon\|\boldsymbol{z}\|_2, \frac{1}{1-t}2\sqrt{\sigma^2\log N}\right\}$$

for any $t \in (0,1)$. Next, it can be verified through Theorem 2.13 in Chapter 2 that $\tilde{\boldsymbol{e_M}} \doteq \boldsymbol{\Phi}^\top \boldsymbol{e_M}$ satisfies $\|\tilde{\boldsymbol{e_M}}\|_\infty \leq 2\sqrt{\sigma^2 \log N}$ with probability exceeding $1 - 2(\sqrt{2\pi \log N} \cdot N)^{-1}$. Now define the probability event

$$\mathcal{G} \doteq \left\{ \left\{ \boldsymbol{\Phi} \text{ satisfies (StOC-1) and (StOC-2)} \right\} \bigcap \left\{ \|\tilde{\boldsymbol{e_M}}\|_\infty \leq 2\sqrt{\sigma^2 \log N} \right\} \right\} \tag{11.3.15}$$

and notice that we have $\Pr(\mathcal{G}) > 1 - \delta - 2(\sqrt{2\pi \log N} \cdot N)^{-1}$. Further, conditioned on the event $\mathcal{G}$, we have

$$\|\tilde{\boldsymbol{\alpha}}_{\Pi^c}\|_\infty \overset{(a)}{\leq} \|\boldsymbol{\Phi}_{\Pi^c}^\top \boldsymbol{\Phi}_\Pi \boldsymbol{z}\|_\infty + \|\boldsymbol{\Phi}_{\Pi^c}^\top \boldsymbol{e_M}\|_\infty \overset{(b)}{\leq} \epsilon \|\boldsymbol{z}\|_2 + 2\sqrt{\sigma^2 \log N} \overset{(c)}{\leq} \lambda_\epsilon \tag{11.3.16}$$

where (a) follows from the triangle inequality, (b) is mainly a consequence of the conditioning on the event $\mathcal{G}$, and (c) follows from the definition of λ_ϵ. Next, we define $r = (\boldsymbol{\Phi}_\Pi^\top \boldsymbol{\Phi}_\Pi - \boldsymbol{I})\boldsymbol{z}$ and notice that, conditioned on the event $\mathcal{G}$, we have for any $i \in [k]$ the following inequality:

$$\begin{aligned} |f_{\pi_i}| &= |z_i + r_i + \tilde{e_{M}}_{\pi_i}| \geq |z_i| - \|\boldsymbol{r}\|_\infty - \|\tilde{\boldsymbol{e_M}}\|_\infty \\ &\overset{(d)}{\geq} \|\boldsymbol{\alpha}\|_{\min} - \epsilon \|\boldsymbol{z}\|_2 - 2\sqrt{\sigma^2 \log N} \overset{(e)}{\geq} \|\boldsymbol{\alpha}\|_{\min} - \lambda_\epsilon \,. \end{aligned} \tag{11.3.17}$$

Here, (d) follows from the conditioning on $\mathcal{G}$, while (e) is a simple consequence of the choice of λ_ϵ. It can therefore be concluded from (11.3.16) and (11.3.17) that if $\boldsymbol{\Phi}$ satisfies (k, ϵ, δ)-StOC and the OST algorithm uses the threshold λ_ϵ then we have $\Pr(\widehat{\mathcal{S}} \neq \mathcal{S}) \leq \Pr(\mathcal{G}^c)$ as long as $\|\boldsymbol{\alpha}\|_{\min} > 2\lambda_\epsilon$.

Finally, to complete the proof of this theorem, we let $k \leq M/(2 \log N)$ and fix $\epsilon = 10\mu\sqrt{2 \log N}$. Then the claim is that $\boldsymbol{\Phi}$ satisfies (k, ϵ, δ)-StOC with $\delta \leq 4N^{-1}$. In order to establish this claim, we only need to ensure that the chosen parameters satisfy the assumptions of Lemma 11.8 and Lemma 11.9. In this regard, note that (i) $\epsilon < 1$ because of (CP-1), and (ii) $\sqrt{k}\,\nu \leq \frac{\epsilon}{9}$ because of the assumption that $k \leq M/(2 \log N)$ and (CP-2). Therefore, since the assumption $N \geq 128$ together with $k \leq M/(2 \log N)$ implies that $16(2 + a^{-1})^2 < 72$, we obtain $\exp\left(-\frac{(\epsilon - \sqrt{k}\,\nu)^2}{16(2+a^{-1})^2 \mu^2}\right) \leq N^{-2}$. We can now combine this fact with the previously established facts to see that the threshold $\lambda = \max\left\{\frac{1}{t} 10\mu\sqrt{M \cdot \text{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2 \log N}$ guarantees that $\Pr(\widehat{\mathcal{S}} \neq \mathcal{S}) \leq 6N^{-1}$ as long as $M \geq 2k \log N$ and $\|\boldsymbol{\alpha}\|_{\min} > 2\lambda$. Finally, note that

$$\|\boldsymbol{\alpha}\|_{\min} > \frac{1}{1-t} 4\sqrt{\sigma^2 \log N} \iff M > \frac{c_2 k \log N}{\text{SNR}_{\min}}$$

and

$$\|\boldsymbol{\alpha}\|_{\min} > \frac{1}{t} 20\mu\sqrt{2M\sigma^2 \log N \cdot \text{SNR}} \iff M > \left(\frac{c_3 k \log N}{\text{MAR}}\right)^{\gamma/2}.$$

This completes the proof of the theorem.

Proof of Theorem 11.5

We begin by making use of the notation developed earlier in this section and conditioning on the event $\mathcal{G}$ defined in (11.3.15) with $\epsilon = 10\mu\sqrt{2\log N}$. Then it is easy to see from the proof of Theorem 11.2 that the estimate $\widehat{\mathcal{S}}$ is a subset of $\mathcal{S}$ because of the fact that $\|\tilde{\boldsymbol{\alpha}}_{\Pi^c}\|_\infty \leq \lambda$.

Next, assume without loss of generality that $z_i \equiv \alpha^*_{(i)}$ and note from (11.3.17) that $|\tilde{\alpha}_{\pi_i}| \geq |\alpha^*_{(i)}| - \lambda$ for any $i \in \{1, \ldots, k\}$. Then, since $\pi_i \in \widehat{\mathcal{S}}$ if and only if $|\tilde{\alpha}_{\pi_i}| > \lambda$, we have that $\left|\alpha^*_{(i)}\right| > 2\lambda \;\Rightarrow\; \pi_i \in \widehat{\mathcal{S}}$. Now define L to be the largest integer for which $\left|\alpha^*_{(L)}\right| > 2\lambda$ holds and note that $\left|\alpha^*_{(L)}\right| > 2\lambda \;\Rightarrow\; \alpha^*_{(i)} > 2\lambda \;\Rightarrow\; \pi_i \in \widehat{\mathcal{S}}$ for every $i \in \{1, \ldots, L\}$, which in turn implies $\left|\mathcal{S} - \widehat{\mathcal{S}}\right| \leq (k - L)$. Finally, note that

$$\left|\alpha^*_{(L)}\right| > \frac{1}{1-t} 4\sqrt{\sigma^2 \log N} \iff \text{LAR}_L > \frac{c_2 k \log N}{M \cdot \text{SNR}}$$

and

$$\left|\alpha^*_{(L)}\right| > \frac{1}{t} 20\mu\sqrt{2n\sigma^2 \log N \cdot \text{SNR}} \iff \text{LAR}_L > \frac{c_3' k \log N}{\mu^{-2}}.$$

This completes the proof of the theorem since the event $\mathcal{G}$ holds with probability exceeding $1 - 6N^{-1}$.

11.4 Near-Optimal Design Matrices for One-Step Thresholding: Some Examples

Section 11.2 establishes that design matrices with small worst-case coherence (and consequently small average coherence) are particularly well-suited for model selection and recovery of sparse signals using OST. Moreover, in the next chapter we will see the implications of the spectral norm on the uniqueness of sparse representation. Further, since the Welch bound [251] dictates that $\mu \gtrsim M^{-1/2}$ for $N \gg 1$ and since we have from elementary linear algebra that $\|\boldsymbol{\Phi}\|_2 \geq \sqrt{\frac{N}{M}}$, we are particularly interested in design matrices that approximately satisfy the scaling relations $\mu(\boldsymbol{\Phi}) \asymp M^{-1/2}$, $\nu(\boldsymbol{\Phi}) \precsim M^{-1}$, and $\|\boldsymbol{\Phi}\|_2 \asymp \sqrt{\frac{N}{M}}$. In the following, we provide some examples of both random and deterministic design matrices that are nearly-optimal in terms of these requisite conditions (also, see Table 11.1 for an overview of the results reported in here).

11.4.1 Random Design Matrices

Random matrices are perhaps the most well-studied design matrices in the literature on high-dimensional, linear inference problems. This is in part due to the fact that geometric concepts such as the irrepresentable condition [261] and the restricted isometry property (RIP) [49] have, to date, been shown to hold near-optimally only for the case of random matrices. The following two lemmas make precise the intuition that traditional random design matrices such as Gaussian matrices and (random) partial Fourier matrices also tend to be near-optimal in terms of the geometric measures of μ, ν, and/or $\|\boldsymbol{\Phi}\|_2$.

Lemma 11.10 (Geometry of Gaussian Matrices)**.** *Let $\boldsymbol{\Phi}$ be an $M \times N$ design matrix with independent and identically distributed (i.i.d.) $\mathcal{N}(0, 1/M)$ entries and let $M \geq 60 \log N$. Then, we have that $\boldsymbol{\Phi}$ satisfies (i) $\mu(\boldsymbol{\Phi}) \leq \sqrt{\frac{15 \log N}{M}}$, (ii) $\nu(\boldsymbol{\Phi}) \leq \frac{\sqrt{15 \log N}}{M}$, and (iii) $\|\boldsymbol{\Phi}\|_2 \leq 1 + 2\sqrt{\frac{N}{M}}$ with probability exceeding $1 - 2(N^{-1} + N^{-2} + e^{-N/2})$.*[2]

Note that the worst-case coherence bound in this lemma follows from bounds on the inner product of independent Gaussian vectors (see, e.g., [17, Appendix A]) and a simple union bound argument, the proof of the average coherence bound is provided in [17, Lemma 2], and the spectral norm bound follows from [219, (2.3)]. It is worth pointing out here that similar results can also be obtained for sub-Gaussian design matrices using standard concentration inequalities and [219, Proposition 2.4].

Lemma 11.11 (Geometry of Partial Fourier Matrices)**.** *Let $\boldsymbol{U}$ be an N-point (non-normalized) discrete Fourier transform matrix such that $\boldsymbol{U}^\top \boldsymbol{U} = N\boldsymbol{I}$. Next, populate Ω by sampling M times with replacement from the set $\{1, \ldots, N\}$ and construct $\boldsymbol{\Phi}$ by collecting the rows of $\boldsymbol{U}$ corresponding to the indices in Ω and normalizing the resulting matrix by $1/\sqrt{M}$. Then $\boldsymbol{\Phi}$ satisfies (i) $\mu(\boldsymbol{\Phi}) \leq \sqrt{\frac{12 \log N}{M}}$ and (ii) $\nu(\boldsymbol{\Phi}) \leq \max\left\{\frac{1}{N-1}, \frac{N-M}{M(N-1)}\right\}$ with probability exceeding $1 - 2N^{-1}$.*

In this lemma, the worst-case coherence bound follows by noting that the columns of $\boldsymbol{U}$ form a group under pointwise multiplication and then making use of Hoeffding's inequality [38]. On the other hand, the average coherence expression in it follows from the definition of the average coherence and the fact that $\mathbf{1}$ is in the null space of any partial Fourier matrix that does not include the first row of $\boldsymbol{U}$. Finally, note that the fact that sampling in Lemma 11.11 is carried out with replacement, which makes it difficult to specify the spectral norm of $\boldsymbol{U}$. In practice, however, one would not construct partial Fourier matrices with identical rows and the spectral norm of partial Fourier matrices in such cases would be $\sqrt{\frac{N}{M}}$ for the simple reason that the rows of $\boldsymbol{U}$ are mutually orthogonal.

[2] Note that the results (and the definition of the coherence property) presented earlier remain valid if $\mu(\boldsymbol{\Phi})$ is replaced with an upperbound $\bar{\mu}(\boldsymbol{\Phi})$.

Table 11.1: Comparisons between different classes of random and deterministic design matrices. All bounds ignore the $O()$ constants.

Matrix	N	$\mu(\mathbf{\Phi})$	$\nu(\mathbf{\Phi})$	$\|\mathbf{\Phi}\|_2$	Randomness	Complexity
Gaussian Matrices	–	$\sqrt{\frac{\log N}{M}}$	$\frac{\sqrt{\log N}}{M}$	$\sqrt{\frac{N}{M}}$	MN	MN
Partial Fourier Matrices	–	$\sqrt{\frac{\log N}{M}}$	$\frac{N-M}{M(N-1)}$	–	$M \log N$	$N \log N$
Alltop Gabor Frames	M^2	$\frac{1}{\sqrt{M}}$	$\frac{1}{M}$	$\sqrt{\frac{N}{M}}$	–	$N \log N$
Discrete-Chirp Matrices	M^2	$\frac{1}{\sqrt{M}}$	$\frac{N-M}{M(N-1)}$	$\sqrt{\frac{N}{M}}$	–	$N \log N$
Dual BCH Sensing Matrices	N^2	$\sqrt{\frac{2}{M}}$	$\frac{N-M}{M(N-1)}$	$\sqrt{\frac{N}{M}}$	–	$N \log N$
Delsarte–Goethals Frames	M^{2+r}	$\frac{2^r}{\sqrt{M}}$	$\frac{1}{N-1}$	$\sqrt{\frac{N}{M}}$	–	$N \log N$

11.4.2 Deterministic Design Matrices

Having described the geometry of Gaussian matrices and partial Fourier matrices, we now show that there in fact exist many classes of deterministic design matrices that are quite similar to these random design matrices in terms of the geometric measures of μ, ν, and $\|\mathbf{\Phi}\|_2$. This is in stark contrast to the best known results for the RIP of deterministic matrices and has important implications from an implementation viewpoint since multiplications with the deterministic matrices described below (and their adjoints) can be efficiently carried out using algorithms such as the *fast Fourier transform* (FFT) and the *fast Hadamard transform* (FHT).

Geometry of Gabor Frames and Its Implications

A (finite) frame for $\mathbb{C}^M$ is defined as any collection of $N \geq M$ vectors that span the M-dimensional Hilbert space $\mathbb{C}^M$. Gabor frames for $\mathbb{C}^M$ constitute an important class of frames, having applications in areas such as communications [18] and radar [146], that are constructed from time- and frequency-shifts of a nonzero seed vector in $\mathbb{C}^M$. Specifically, let $\boldsymbol{g} \in \mathbb{C}^N$ be a unit-norm seed vector and define $\boldsymbol{T}$ to be an

$M \times M$ *time-shift matrix* that is generated from $\boldsymbol{g}$ as follows

$$\boldsymbol{T}(\boldsymbol{g}) \doteq \begin{bmatrix} g_1 & g_n & & g_2 \\ g_2 & g_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & g_n \\ g_n & g_{n-1} & & g_1 \end{bmatrix} \tag{11.4.1}$$

where we write $\boldsymbol{T} = \boldsymbol{T}(\boldsymbol{g})$ to emphasize that $\boldsymbol{T}$ is a matrix-valued function on $\mathbb{C}^M$. Next, denote the collection of M samples of a discrete sinusoid with frequency $2\pi\frac{m}{M}, m \in \{0, \ldots, M-1\}$ as $\omega_m \doteq \begin{bmatrix} \mathrm{e}^{j2\pi\frac{m}{M}0} & \ldots & \mathrm{e}^{j2\pi\frac{m}{M}(M-1)} \end{bmatrix}^\top$. Finally, define the corresponding $M \times M$ diagonal *modulation matrices* as $\boldsymbol{W_m} = \mathrm{diag}(\omega_m)$. Then the Gabor frame generated from $\boldsymbol{g}$ is an $M \times M^2$ block matrix of the form

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{W_0 T} & \boldsymbol{W_1 T} & \ldots & \boldsymbol{W_{M-1} T} \end{bmatrix}. \tag{11.4.2}$$

In words, columns of the Gabor frame $\boldsymbol{\Phi}$ are given by downward circular shifts and modulations (frequency shifts) of the seed vector $\boldsymbol{g}$. We are now ready to state the first main result concerning the geometry of Gabor frames, which follows directly from [178].

Theorem 11.12 (Spectral Norm of Gabor Frames [178])**.** *Gabor frames generated from nonzero (unit-norm) seed vectors are tight frames; in other words, we have that* $\|\boldsymbol{\Phi}\|_2 = \sqrt{\frac{N}{M}}$.

Theorem 11.12 implies that Gabor frames are the best that one can hope for in terms of the spectral norm. The next result that we prove concerns the average coherence of Gabor frames.

Theorem 11.13 (Average Coherence of Gabor Frames)**.** *Let* $\boldsymbol{\Phi}$ *be a Gabor frame generated from a unit-norm seed vector* $\boldsymbol{g} \in \mathbb{C}^M$*. Then, using the notation* $g_{\max} \doteq \max_i |g_i|$ *and* $g_{\min} \doteq \min_i |g_i|$*, the average coherence of* $\boldsymbol{\Phi}$ *can be bounded from the above as follows:*

$$\nu(\boldsymbol{\Phi}) \leq \frac{M\, g_{\max}(\sqrt{M} - g_{\min}) + 1 - M\, g_{\min}^2}{M^2 - 1}. \tag{11.4.3}$$

Proof. In order to facilitate the proof of this theorem, we first map the indices of the columns of $\boldsymbol{\Phi}$ from $\{1, \ldots, M^2\}$ to $\mathcal{C} \doteq \{0, \ldots, M-1\} \times \{0, \ldots, M-1\}$ as follows

$$\kappa : i \mapsto \left((i \bmod M) - 1, \left\lfloor \frac{i-1}{M} \right\rfloor \right). \tag{11.4.4}$$

In words, $\kappa(i) = (\ell, m)$ signifies that the i-th column of $\boldsymbol{\Phi}$ corresponds to the $(\ell+1)$-th column of $W_m T$. Next, fix an index i (resp. $\kappa(i) = (\ell, m)$) and make use of the

above reindexing to write

$$\sum_{\substack{j=1\\ j\neq i}}^{M^2}\langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{i})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{j})}\rangle = \sum_{\substack{(\ell',m')\in\mathcal{C}\\ (\ell',m')\neq(\ell,m)}} \langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell},\boldsymbol{m})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell'},\boldsymbol{m'})}\rangle$$
$$= \sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{M-1}\sum_{m'=0}^{M-1}\langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell},\boldsymbol{m})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell'},\boldsymbol{m'})}\rangle + \sum_{\substack{m'=0\\ m'\neq m}}^{M-1}\langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell},\boldsymbol{m})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell'},\boldsymbol{m'})}\rangle. \tag{11.4.5}$$

Finally, note that we can explicitly write the columns of $\boldsymbol{\Phi}$ using (11.4.2) for any $(\ell, m) \in \mathcal{C}$ as follows

$$\boldsymbol{\varphi}_{\boldsymbol{(\ell,m)}} \doteq \begin{bmatrix} g_{(1-\ell)_M}\mathrm{e}^{j2\pi\frac{m}{M}0} & \dots & g_{(M-\ell)_M}\mathrm{e}^{j2\pi\frac{m}{M}(M-1)}\end{bmatrix}^\top \tag{11.4.6}$$

where we use the notation $g_{(q)_M}$ as a shorthand for $g_{q \bmod M}$. The rest of the proof now follows from simple algebraic manipulations. Specifically, it follows from (11.4.6) that the first term in (11.4.5) can be simplified as

$$\sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{N-1}\sum_{m'=0}^{N-1}\langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell},\boldsymbol{m})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell'},\boldsymbol{m'})}\rangle$$
$$= \sum_{q=1}^{M}\sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{M-1} g^*_{(q-\ell)_M} g_{(q-\ell')_M}\sum_{m'=0}^{M-1}\mathrm{e}^{j2\pi\frac{q-1}{M}(m'-m)}$$
$$= \sum_{q=2}^{M}\sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{M-1} g^*_{(q-\ell)_M} g_{(q-\ell')_M}\sum_{m'=0}^{M-1}\mathrm{e}^{j2\pi\frac{q-1}{M}(m'-m)} +$$
$$+ M\sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{M-1} g^*_{(1-\ell)_M} g_{(1-\ell')_M} \overset{(a)}{=} M\, g^*_{(1-\ell)_M}\sum_{\substack{\ell'=0\\ \ell'\neq\ell}}^{M-1} g_{(1-\ell')_M} \tag{11.4.7}$$

where (a) in the above expression is a consequence of the fact that $\sum_{m'=0}^{M-1}\mathrm{e}^{j2\pi\frac{q-1}{M}(m'-m)} = 0$ for any fixed $q \in \{2, \dots, M\}$. Likewise, we can simplify the second term in (11.4.5)

as follows

$$\begin{aligned}
&\sum_{\substack{m'=0\\ m'\neq m}}^{N-1} \big\langle \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell},\boldsymbol{m})}, \boldsymbol{\varphi}_{\boldsymbol{\kappa}(\boldsymbol{\ell}',\boldsymbol{m}')} \big\rangle \\
&\quad = \sum_{q=1}^{M} g^*_{(q-\ell)_M} g_{(q-\ell)_M} \sum_{\substack{m'=0\\ m'\neq m}}^{M-1} \mathrm{e}^{j2\pi \frac{q-1}{M}(m'-m)} \\
&\quad = \sum_{q=2}^{M} \big|g_{(q-\ell)_M}\big|^2 \sum_{\substack{m'=0\\ m'\neq m}}^{M-1} \mathrm{e}^{j2\pi \frac{q-1}{M}(m'-m)} + \big|g_{(1-\ell)_M}\big|^2 \sum_{\substack{m'=0\\ m'\neq m}}^{M-1} 1 \\
&\quad \overset{(b)}{=} -\sum_{q=2}^{M} \big|g_{(q-\ell)_M}\big|^2 + (M-1)\big|g_{(1-\ell)_M}\big|^2 \\
&\quad = -1 + M\big|g_{(1-\ell)_M}\big|^2
\end{aligned} \tag{11.4.8}$$

where (b) follows from the fact that $\sum_{m'\neq m} \mathrm{e}^{j2\pi \frac{q-1}{M}(m'-m)} = -1$ for any fixed $q \in \{2,\dots,M\}$.

To conclude the theorem, note from (11.4.5), (11.4.7), and (11.4.8) that we can write

$$\begin{aligned}
&\max_{i\in\{1,\dots,M^2\}} \left| \sum_{\substack{j=1\\ j\neq i}}^{M^2} \langle \boldsymbol{\varphi}_{\boldsymbol{i}}, \boldsymbol{\varphi}_{\boldsymbol{j}} \rangle \right| \\
&\quad = \max_{\ell} \left| M\, g^*_{(1-\ell)_M} \sum_{\substack{\ell'=0\\ \ell'\neq \ell}}^{M-1} g_{(1-\ell')_M} - 1 + M\big|g_{(1-\ell)_M}\big|^2 \right| \\
&\quad \overset{(c)}{\leq} \max_{r\in\{1,\dots,M\}} \left| M\, g_r^* \sum_{\substack{s=1\\ s\neq r}}^{M} g_s \right| + \max_{r\in\{1,\dots,M\}} \Big| -1 + M|g_r|^2 \Big| \\
&\quad \leq M \max_{r\in\{1,\dots,M\}} |g_r| \sum_{\substack{s=1\\ s\neq r}}^{M} |g_s| + 1 - M\, g_{\min}^2 \\
&\quad \overset{(d)}{\leq} M\, g_{\max}(\sqrt{M} - g_{\min}) + 1 - M\, g_{\min}^2.
\end{aligned} \tag{11.4.9}$$

Here, (c) mainly follows from the triangle inequality and a simple reindexing argument, while (d) mainly follows from the Cauchy–Schwarz inequality since $\sum_{\substack{s=1\\ s\neq r}}^{M} |g_s| = \|\boldsymbol{g}\|_1 - |g_r| \leq \sqrt{M} - g_{\min}$. The proof of the theorem now follows by dividing the above expression by $M^2 - 1$. □

In words, Theorem 11.13 states that the average coherence of Gabor frames cannot be too large. In particular, it implies that Gabor frames generated from unimodal

(unit-norm) seed vectors (i.e., seed vectors characterized by $g_{\min} \asymp g_{\max} \asymp M^{-1/2}$) satisfy $\nu(\mathbf{\Phi}) \precsim M^{-1}$. On the other hand, recall that the Welch bound [251] dictates that $\mu(\mathbf{\Phi}) \geq (M+1)^{-1/2}$ for Gabor frames. It is therefore possible to conclude from these two facts that Gabor frames generated from unimodal seed vectors are automatically guaranteed to satisfy the coherence property (resp. strong coherence property) as long as $\mu(\mathbf{\Phi}) \precsim (\log N)^{-1/2}$ (resp. $\nu(\mathbf{\Phi}) \precsim (\log N)^{-1}$). In the context of model selection and sparse-signal recovery, Theorem 11.13 therefore suggests that Gabor frames generated from unimodal seed vectors are the best that one can hope for in terms of the average coherence.

Finally, recall from the discussions in Section 11.2 that—among the class of matrices that satisfy the coherence property—design matrices with small worst-case coherence are particularly well-suited for model selection and sparse-signal recovery. In the context of Gabor frames, the goal then is to design unimodal seed vectors that yield Gabor frames with the smallest-possible worst-case coherence. This, however, is an active area of mathematical research and a number of researchers have looked at this problem in recent years; see, e.g., [230]. As such, we can simply leverage some of the existing research in this area in order to provide explicit constructions of Gabor frames that satisfy the coherence property with nearly-optimal worst-case coherence.

Specifically, let $M \geq 5$ be a prime number and construct a unimodal seed vector $\boldsymbol{g} \in \mathbb{C}^N$ as follows

$$\boldsymbol{g} = \left[\frac{1}{\sqrt{M}} \mathrm{e}^{j2\pi \frac{0^3}{M}} \quad \frac{1}{\sqrt{M}} \mathrm{e}^{j2\pi \frac{1^3}{M}} \quad \ldots \quad \frac{1}{\sqrt{M}} \mathrm{e}^{j2\pi \frac{(M-1)^3}{M}} \right]^{\mathrm{T}}. \tag{11.4.10}$$

The sequence $\left\{ \frac{1}{\sqrt{M}} \mathrm{e}^{j2\pi \frac{q^3}{M}} \right\}_{q=0}^{M-1}$ is termed the *Alltop sequence* [6] in the literature. This sequence has the property that its autocorrelation decays very fast and, therefore, it is particularly well-suited for generating Gabor frames with small worst-case coherence. In particular, it was established recently in [230] that Gabor frames generated from the Alltop seed vector $\boldsymbol{g}$ given in (11.4.10) satisfy

$$\mu(\mathbf{\Phi}) \doteq \max_{i,j:i \neq j} \left| \langle \boldsymbol{\varphi_i}, \boldsymbol{\varphi_j} \rangle \right| \leq \frac{1}{\sqrt{M}}. \tag{11.4.11}$$

In addition, since we have that $g_{\min} = g_{\max} = M^{-1/2}$ for the Alltop seed vector, it is possible to conclude from Theorem 11.13 that the average coherence of Alltop Gabor frames satisfies $\nu(\mathbf{\Phi}) \leq (M+1)^{-1} \leq \mu(\mathbf{\Phi})/\sqrt{M}$. An immediate consequence of this discussion is that all the results reported in Section 11.2 in the context of model selection using OST apply directly to the case of Alltop Gabor frames.

Geometry of Discrete-Chirp Matrices

An M-length chirp signal for any prime M takes the form

$$\boldsymbol{\varphi}_{(m,r)}(\ell) = \frac{1}{\sqrt{M}} e^{j2\pi \frac{m\ell}{M} + j2\pi \frac{r\ell^2}{M}}, \ \ell = 0, \ldots, M-1 \tag{11.4.12}$$

where m is the base frequency and r is the chirp rate of the signal. Discrete-chirp matrices are $M \times M^2$ matrices that are constructed by collecting all possible chirp signals into columns[11]. The columns of the $M \times M^2$ discrete-chirp matrix $\boldsymbol{\Phi}$ are the M^2 distinct chirp signals corresponding to the M^2 possible pairs $(m, r) \in \mathbb{Z}_M \times \mathbb{Z}_M$. The following lemma characterizes the geometry of discrete-chirp matrices.

Lemma 11.14. *Let $\boldsymbol{\Phi}$ be an $M \times M^2$ discrete-chirp matrix for any prime M. Then $\boldsymbol{\Phi}$ satisfies (i) $\mu(\boldsymbol{\Phi}) \equiv \frac{1}{\sqrt{M}}$, (ii) $\nu(\boldsymbol{\Phi}) \equiv \frac{N-M}{M(N-1)}$, and (iii) $\|\boldsymbol{\Phi}\|_2 \equiv \sqrt{\frac{N}{M}}$.*

Here, the worst-case coherence bound and the spectral norm expression follow from [64], while the average coherence expression follows from the fact that $\boldsymbol{\Phi}^\top \boldsymbol{\Phi} \mathbf{1} \equiv \frac{N}{M}\mathbf{1}$. Finally, note that the structure of the discrete-chirp matrix $\boldsymbol{\Phi}$ implies that the multiplications $\boldsymbol{\Phi}\boldsymbol{v}$ and $\boldsymbol{\Phi}^\top \boldsymbol{u}$ can be carried out using the FFT in $O(N \log N)$ time.

Geometry of Dual BCH Sensing Matrices

Dual BCH sensing matrices constitute another class of design matrices that corresponds to exponentiating the codewords of an algebraic code. Specifically, take $m \in \mathbb{Z}_+$ to be an odd number and use $BCH(m, 2)$ to denote the extended 2-error correcting, binary BCH code of length $M = 2^m$ [183]. Then the dual of $BCH(m, 2)$ is a code of length M and dimension $2m + 1$ that is the union of M cosets of the first-order Reed–Muller code $RM(1, m)$ of dimension $m + 1$. The important thing to point out here is that exponentiating codewords in the dual of $BCH(m, 2)$ and scaling the resulting $M \times M^2$ matrix $\boldsymbol{\Phi}$ by $1/\sqrt{M}$ gives a union of M orthonormal basis. This can be seen by noting that exponentiating codewords in $RM(1, m)$ gives Walsh basis vectors (and their negatives, which we discard in here). We also note because of the very same reason that the multiplications $\boldsymbol{\Phi}\boldsymbol{v}$ and $\boldsymbol{\Phi}^\top \boldsymbol{u}$ in the case of dual BCH sensing matrices can also be carried out using the FHT in $ON \log N)$ time. The following lemma characterizes the geometry of dual $BCH(m, 2)$ sensing matrices.

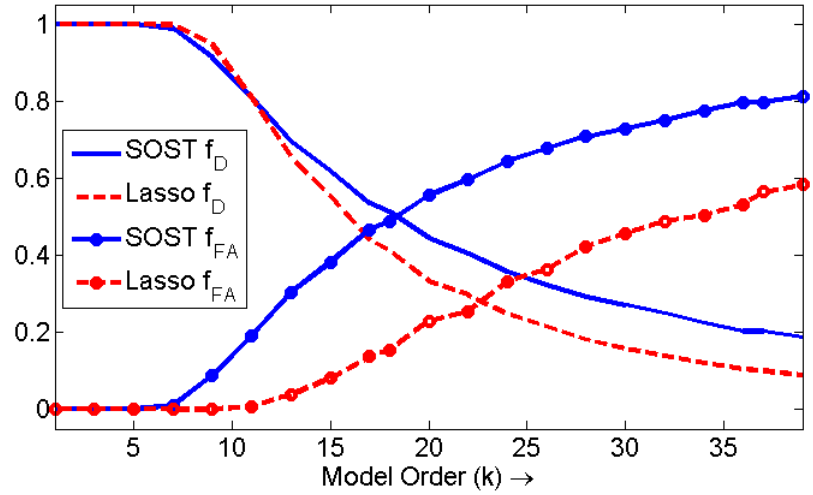
Lemma 11.15. *Let $\boldsymbol{\Phi}$ be an $M \times M^2$ dual BCH sensing matrix obtained from the dual of $BCH(m, 2)$ for some odd m. Then the matrix $\boldsymbol{\Phi}$ satisfies (i) $\mu(\boldsymbol{\Phi}) \equiv \sqrt{\frac{2}{M}}$, (ii) $\nu(\boldsymbol{\Phi}) \equiv \frac{N-M}{M(N-1)}$, and (iii) $\|\boldsymbol{\Phi}\|_2 \equiv \sqrt{\frac{N}{M}}$.*

Remark 11.16. *In this section, we introduced deterministic design matrices with optimal spectral norm* $\|\mathbf{\Phi}\|_2 = \sqrt{\frac{N}{M}}$*, and worst-case coherence* $\mu(\mathbf{\Phi}) \precsim M^{-0.5}$*. The introduced matrices also satisfy the coherence property as* $\nu(\mathbf{\Phi}) \precsim M^{-1} \precsim \mu(\mathbf{\Phi})M^{-0.5}$*. However, these matrices suffer from the restriction that each matrix can have at most* M^2 *columns (i.e.,* $N \leq M^2$*). In the next chapter, we will introduce the Delsarte-Goethals frames (*$\mathrm{DG}(m,r)$*) as another family of design matrices with optimal spectral norm (*$\mathbf{\Phi} = \sqrt{\frac{N}{M}}$*), and worst-case coherence (*$\mu \leq \frac{2^r}{\sqrt{M}}$*). We will see that the Delsarte-Goethals frames also have with much smaller average coherence* $\nu(\mathbf{\Phi}) \leq \frac{1}{N-1} \ll \frac{1}{M}$*, and much larger number of columns* $N = M^{r+2}$*.*

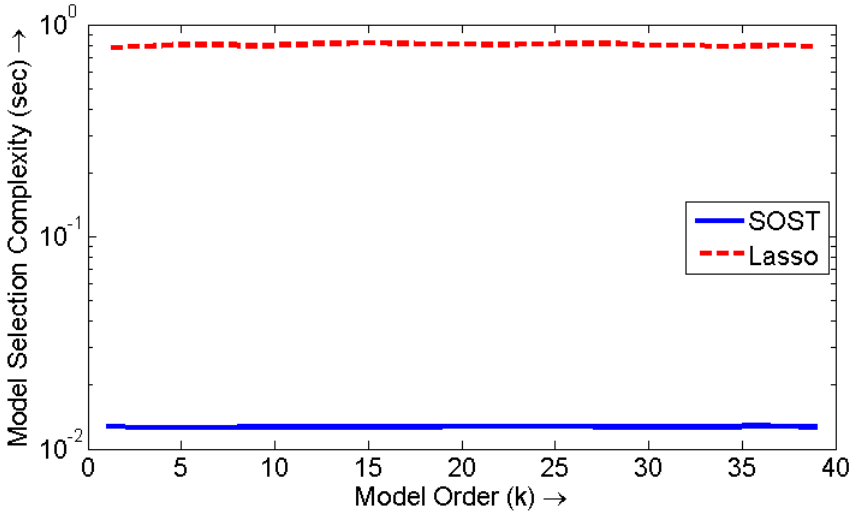
11.5 Conclusion

In this chapter, we have revisited two variants of the often forgotten but extremely fast one-step thresholding (OST) algorithm for model selection. One of the key insights offered by the chapter in this regard is that polynomial-time model selection can be carried out even when signal reconstruction (and thereby the lasso) fails. In addition, we have established in the chapter that if the $M \times N$ design matrix $\mathbf{\Phi}$ satisfies $\mu(\mathbf{\Phi}) \asymp M^{-1/2}$ and $\nu(\mathbf{\Phi}) \precsim M^{-1}$ then OST can perform near-optimally for the case when either (i) the minimum-to-average ratio (MAR) of the signal is not too small or (ii) the signal-to-noise ratio (SNR) in the measurement system is not too high. It is worth pointing out here that some researchers in the past have observed that the sorted variant of the OST (SOST) algorithm at times performs similar to or better than the lasso (see Fig. 11.2 for an illustration of this in the case of an Alltop Gabor frame in $\mathbb{C}^{127}$). One of our main contributions in this regard is that we have taken the mystery out of this observation and explicitly specified in the chapter the four key parameters of the model-selection problem, namely, $\mu(\mathbf{\Phi}), \nu(\mathbf{\Phi}),$ MAR, and SNR, that determine the non-asymptotic performance of the SOST algorithm for generic (random or deterministic) design matrices and data vectors having generic (random or deterministic) nonzero entries; also, see [120] for a comparison of our results with corresponding results recently reported in the literature.

The second main contribution of this chapter—which completely sets it apart from existing work on thresholding for model selection—is that we have proposed and analyzed a model-order agnostic threshold for the OST algorithm. The significance of this aspect of the chapter can be best understood by realizing that in real-world applications it is often easier to estimate the SNR and the noise variance in the system than to estimate the true model order. In particular, we have established in the chapter that the threshold $\lambda = \max\left\{\frac{1}{t}10\mu\sqrt{M \cdot \mathrm{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2 \log N}$ for $t \in (0,1)$ enables the OST algorithm to carry out near-optimal partial model selection. Fig. 11.3 reports the results of an experiment concerning partial model-selection performance of the OST



(a) Plots of the *fraction of detections*, defined as $f_D = |\mathcal{S} \cap \widehat{\mathcal{S}}|/k$, and the *fraction of false alarms*, defined as $f_{FA} = (|\widehat{\mathcal{S}}| - |\mathcal{S} \cap \widehat{\mathcal{S}}|)/|\widehat{\mathcal{S}}|$, versus the model order (averaged over 200 independent trials) for both SOST and the lasso.



(b) Plots of the amount of time (averaged over 200 independent trials) that it takes SOST and the lasso to solve one model-selection problem versus the model order.

Figure 11.2: Numerical comparisons between the performance of the SOST algorithm (Algorithm 9) and the lasso [233] using an Alltop Gabor frame. The $M \times N$ design matrix $\mathbf{\Phi}$ has dimensions $M = 127$ and $N = M^2$, the MAR of the signals is 1, the SNR in the measurement system is 10 dB, and the noise variance is $\sigma^2 = 10^{-2}$.

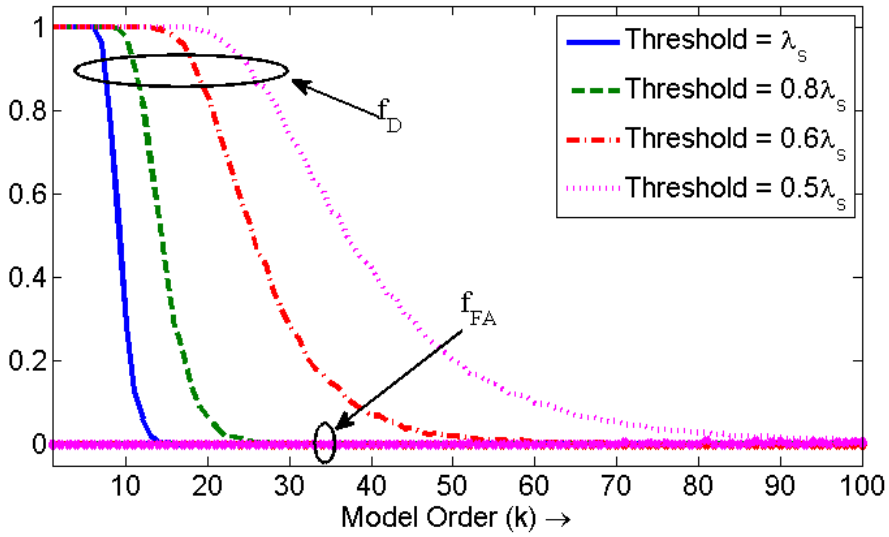


Figure 11.3: Partial model-selection performance of the OST algorithm (averaged over 200 independent trials) corresponding to an Alltop Gabor frame in $\mathbb{C}^{997}$. The MAR of the signals in this experiment is 1, the SNR in the measurement system is 3 dB, and the noise variance is $\sigma^2 = 10^{-2}$.

algorithm in terms of the metrics of *fraction of detections*, $f_D \doteq \frac{|\mathcal{S} \cap \widehat{\mathcal{S}}|}{k}$, and *fraction of false alarms*, $f_{FA} \doteq \frac{|\widehat{\mathcal{S}}| - |\mathcal{S} \cap \widehat{\mathcal{S}}|}{|\widehat{\mathcal{S}}|}$, averaged over 200 independent trials. In this experiment, the $M \times N$ design matrix $\mathbf{\Phi}$ corresponds to an Alltop Gabor frame in $\mathbb{C}^{997}$, the noise variance is $\sigma^2 = 10^{-2}$, the MAR and the SNR are chosen to be 1 and 3 dB, respectively, and the initial threshold is set at $\lambda_s \doteq \max\left\{\frac{1}{t}c'\mu\sqrt{M \cdot \text{SNR}}, \frac{1}{1-t}\sqrt{2}\right\}\sqrt{2\sigma^2 \log N}$ with $t = (\sqrt{2}-1)/\sqrt{2}$ and $c' = 2t$. It can clearly be seen from Fig. 11.3 that OST successfully carries out partial model selection ($f_{FA} \equiv 0$) even when the threshold is set at $0.6\lambda_s$.

Chapter 12

Reed-Muller Based Compressed Sensing

In the previous chapter, we introduced two fundamental measures of coherence between the columns of a sensing matrix, and showed that if the matrix satisfies a *coherence property*, then a simple *One-Step Thresholding* algorithm can successfully recover most k-sparse vectors. In this chapter, we introduce the *spectral norm* of the sensing matrix as a measure of coherence between the *rows* of the matrix, and show that if a sensing also has sufficiently small spectral norm, then most k-sparse vectors have *unique* representations in the measurement domain. This further implies that reconstruction algorithms such as LASSO or OST not only can successfully recover the supports of most k-sparse vectors, but are also capable of providing close approximations to those vectors.

The coherence between rows of a sensing matrix is a measure of the new information provided by an additional measurement. The spectral norm $\|\mathbf{\Phi}\|_2$ measures the maximal coherence between the rows of the frame. The ideal case is when different measurements are orthogonal. Then, provided that the matrix also has sufficiently low worst-case coherence, with high probability a k-sparse vector has a unique sparse representation [240], and this representation can be efficiently recovered using a LASSO program [50].

In this chapter, we consider sensing matrices based on the $\mathbb{Z}_4$-linear representation of Delsarte Goethals codes. The columns are obtained by exponentiating codewords in the quaternary Delsarte-Goethals code; they are uniformly and very precisely distributed over the surface of an M-dimensional sphere. Coherence between columns reduces to properties of these algebraic codes. Section 12.2.1 reviews the construction of Delsarte-Goethals (DG) sets of $\mathbb{Z}_4$-linear quadratic forms which is the starting point for the construction of the corresponding codes; each quadratic form determines a codeword where the entries are the values taken by quadratic form. Section 12.2.1 introduces Delsarte-Goethals frames; the columns of these sensing matrices are ob-

tained by exponentiating DG codewords. We then determine the worst case coherence, average-coherence, and spectral norm for these sensing matrices.

Candès and Plan [50] specified coherence conditions under which a LASSO program will successfully recover a k-sparse signal when the k non-zero entries are above the noise variance. Similarly, in Theorems 11.2 to 11.5 we proved similar coherence conditions for successful recovery of the OST algorithm. We use these results to provide an average case error analysis for stochastic noise in both the data and measurement domains. The Delsarte Goethals (DG) sensing matrices are essentially tight frames so that white noise in the data domain maps to white noise in the measurement domain.

Section 12.3 presents the results of numerical experiments that compare DG frames with random Gaussian matrices of the same size. The SPGL package [244, 243] is used to implement the LASSO recovery algorithm in all cases. It turns out that the DG frames have almost identical performance to random matrices in terms of probability of successful sparse recovery, but in contrast to random matrices, DG frames do not suffer from storage and computational limitations. These matrices have deterministic constructions, and matrix-vector multiplications $\mathbf{\Phi}\boldsymbol{v}$ and $\mathbf{\Phi}^\top \boldsymbol{u}$ can be done efficiently using the Fast Hadamard Transform.

We remark that there are alternative fast reconstruction algorithms that exploit the structure of DG sensing matrices. The Chirp Reconstruction algorithm proposed in [45, 150, 46] requires only $O(kM \log^2 M)$ operations, independent of the data-domain dimension N, and is known to work extremely well in the presence of noise as long as the sparsity level is not too high (See [46] for further discussion of the Chirp Reconstruction algorithm).

12.1 Sparse Reconstruction using Incoherent Tight-Frames

In Definition 2.5 of Chapter 2 we showed that an $M \times N$ dictionary $\mathbf{\Phi}$ that satisfies the condition $\mathbf{\Phi}\mathbf{\Phi}^\top = \frac{N}{M}\mathbf{I}_{M\times M}$ is a tight-frame with redundancy $\frac{N}{M}$. Also in Section 3.4.2 we saw that the mutual coherence of any $M \times N$ dictionary is at least $\frac{1}{\sqrt{M}}$ [251]. Designing dictionaries with small spectral norms (tight frames in the ideal case), and with small coherence $\left(\mu = O\left(\frac{1}{\sqrt{M}}\right) \text{ in the ideal case}\right)$ is useful in compressed sensing for the following reasons.

Uniqueness of Sparse Representation (ℓ_0 minimization) The following results are due to Tropp [240], and Gurevich and Hadani [136]), and show that with overwhelming probability the ℓ_0 minimization program successfully recovers the original k-sparse signal.

Theorem 12.1 ([239, 136]). *Assume the dictionary $\mathbf{\Phi}$ satisfies $\mu \leq \frac{c}{\log N}$, where c is*

an absolute constant. Further assume $k \leq \frac{c\,N}{\|\mathbf{\Phi}\|^2 \log N}$. *Let* S *be a random subset of* $[N]$ *of size* k, *and let* $\mathbf{\Phi}_S$ *be the corresponding* $M \times k$ *submatrix. Then there exists an absolute constant* c_0 *such that*

$$\Pr\left[\left\|\mathbf{\Phi}_S^* \mathbf{\Phi}_S - \boldsymbol{I}\right\| \geq c_0 \left(\mu \log N + 2\sqrt{\frac{\|\mathbf{\Phi}\|^2\, k}{N}}\right)\right] \leq 2\, N^{-1}.$$

Theorem 12.2 ([240, 238])**.** *Assume the dictionary* $\mathbf{\Phi}$ *satisfies* $\mu \leq \frac{c}{\log N}$, *where* c *is an absolute constant. Further assume* $k \leq \frac{c\,N}{\|\mathbf{\Phi}\|^2 \log N}$. *Let* $\boldsymbol{\alpha}^*$ *be a* k*-sparse vector, such that the support of the* k *nonzero coefficients of* $\boldsymbol{\alpha}^*$ *is selected uniformly at random. Further assume that conditioned on the support, the values of the* k *non-zero entries of* $\boldsymbol{\alpha}^*$ *are sampled from a distribution which is absolutely continuous with respect to the Lebesgue measure on* $\mathbb{R}^k$. *Then with probability* $1 - O\left(N^{-1}\right)$, $\boldsymbol{\alpha}^*$ *is the unique* k*-sparse vector mapped to* $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^*$ *by the measurement matrix* $\mathbf{\Phi}$.

Sparse Recovery via LASSO (ℓ_1 minimization) Uniqueness of sparse representation is of limited utility given that ℓ_0 minimization is computationally intractable. However, given modest restrictions on the class of sparse signals, Candès and Plan [50] have shown that with overwhelming probability the solution to the ℓ_0 minimization problem coincides with the solution to a convex LASSO program.

Theorem 12.3. *Assume the dictionary* $\mathbf{\Phi}$ *satisfies* $\mu \leq \frac{c}{\log N}$, *where* c *is an absolute constant. Further assume* $k \leq \frac{c_1\, N}{\|\mathbf{\Phi}\|^2 \log N}$, *where* c_1 *is a constant. Let* $\boldsymbol{\alpha}^*$ *be a* k*-sparse vector, such that*

1. *The support of the* k *nonzero coefficients of* $\boldsymbol{\alpha}^*$ *is selected uniformly at random.*
2. *Conditional on the support, the signs of the nonzero entries of* $\boldsymbol{\alpha}^*$ *are independent and equally likely to be* -1 *or* 1.

Let $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha} + \boldsymbol{e_M}$, *where* $\boldsymbol{e_M}$ *contains* M *iid* $\mathcal{N}(0, \sigma^2)$ *Gaussian elements. Then if* $\|\boldsymbol{\alpha}^*\|_{\min} \geq 8\sigma\sqrt{2 \log N}$, *with probability* $1 - O(N^{-1})$ *the LASSO estimate*

$$\hat{\boldsymbol{\alpha}} \doteq \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^N} \frac{1}{2}\|\boldsymbol{f} - \mathbf{\Phi}\boldsymbol{\alpha}\|^2 + 2\sqrt{2 \log N}\,\sigma\,\|\boldsymbol{\alpha}\|_1$$

has the same support and sign as $\boldsymbol{\alpha}^*$, *and* $\|\mathbf{\Phi}\boldsymbol{\alpha}^* - \mathbf{\Phi}\hat{\boldsymbol{\alpha}}\|^2 \leq c_2\, k\, \sigma^2$, *where* c_2 *is a constant independent of* $\boldsymbol{\alpha}^*$.

Stochastic noise in the data domain. The tight-frame property of the sensing matrix makes it possible to map iid Gaussian noise in the data domain to iid Gaussian noise in the measurement domain:

Lemma 12.4. *Let* $\boldsymbol{e_D}$ *be a vector with* N *iid* $\mathcal{N}(0, \sigma_D^2)$ *entries and* $\boldsymbol{e_M}$ *be a vector with* M *iid* $\mathcal{N}(0, \sigma_M^2)$ *entries. Let* $\hbar = \mathbf{\Phi}\boldsymbol{e_D}$ *and* $\boldsymbol{e} = \hbar + \boldsymbol{e_M}$. *Then* $\boldsymbol{e}$ *contains* M *entries, sampled iid from* $\mathcal{N}(0, \sigma^2)$, *where* $\sigma^2 = \frac{N}{M}\sigma_D^2 + \sigma_M^2$.

Proof. The tight frame property implies

$$\mathbb{E}\left[\hbar\hbar^\top\right] = E[\boldsymbol{\Phi e_D} \boldsymbol{e_D}^\top \boldsymbol{\Phi}^\top] = \sigma_D^2 \boldsymbol{\Phi\Phi}^\top = \frac{N}{M}\sigma_D^2 \boldsymbol{I}.$$

Therefore, $\boldsymbol{e} = \boldsymbol{\Phi e_D} + \boldsymbol{e_M}$ contains iid Gaussian elements with zero mean and variance σ^2. □

Next we construct a family of low-coherence tight frames with optimal coherence parameters using Delsarte-Goethals codes.

12.2 Construction of the Delsarte-Goethals Frames

12.2.1 Delsarte-Goethals Sets of Binary Symmetric Matrices

The finite field $\mathbb{F}_{2^m}$ is obtained from the binary field $\mathbb{F}_2$ by adjoining a root ξ of a primitive irreducible polynomial g of degree m. The elements of $\mathbb{F}_{2^m}$ are polynomials in ξ of degree at most $m-1$ with coefficients in $\mathbb{F}_2$, and we will identify the polynomial $x_0 + x_1\xi + \cdots + x_{m-1}\xi^{m-1}$ with the binary m-tuple $(x_0, \cdots, x_{m-1})$. The *Frobenius map* $f : \mathbb{F}_{2^m} \to \mathbb{F}_{2^m}$ is defined by $f(x) = x^2$ and the *Trace map* $\mathrm{Tr} : \mathbb{F}_{2^m} \to \mathbb{F}_2$ is defined by

$$\mathrm{Tr}(x) \doteq x + x^2 + \cdots + x^{2^{m-1}}.$$

The identity $(x+y)^2 = x^2 + y^2$ implies that $\mathrm{Tr}(x+y) = \mathrm{Tr}(x) + \mathrm{Tr}(y)$; the trace is a linear map over the binary field $\mathbb{F}_2$. The trace inner product given by $(v, w) = \mathrm{Tr}(vw)$ is non-degenerate; if $\mathrm{Tr}(vz) = 0$ for all z in $\mathbb{F}_2^m$ then $v = 0$. Every element a in $\mathbb{F}_{2^m}$ determines a symmetric bilinear form $\mathrm{Tr}[xya]$ to which is associated a binary symmetric matrix $P^0(a)$. That is, $P^0(a)$ is a binary matrix such that for every field elements x and y

$$\mathrm{Tr}[xya] = (x_0 \cdots x_{m-1}) P^0(a) (y_o \cdots y_{m-1})^\top.$$

The *Kerdock set* K_m is the m-dimensional binary vector space formed by the matrices $P^0(a)$. For example, let $m = 3$, and assume the finite field $\mathbb{F}_8$ is generated by adjoining a root ξ of the polynomial $g(x) = x^3 + x + 1$. Then K_3 is spanned by

$$P^0(100) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad P^0(010) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \text{ and } P^0(001) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Theorem 12.5. *Every nonzero matrix in K_m is nonsingular.*

Proof. If $xP^0(a) = 0$ then $\mathrm{Tr}[xya] = 0$ for all $y \in \mathbb{F}_{2^m}$. Now the non-degeneracy of the trace implies $a = 0$. □

Next we define higher order bilinear forms, each associated with a binary symmetric matrix. Given a positive integer t where $0 < t \leq \frac{m-1}{2}$ and given a field element a

$$\mathrm{Tr}\left[\left(xy^{2^t} + x^{2^t}y\right)a\right]$$

defines a symmetric bilinear form that is represented by a binary symmetric matrix $P^t(a)$ as above:

$$\mathrm{Tr}\left[\left(xy^{2^t} + x^{2^t}y\right)a\right] = (x_0 \cdots x_{m-1})P^t(a)(y_o \cdots y_{m-1})^\top \tag{12.2.1}$$

The *Delsarte-Goethals set* $DG(m,r)$ is then defined as

$$DG(m,r) \doteq \left\{ \sum_{t=0}^{r} P^t(a_t) \,|\, a_t \in \mathbb{F}_{2^m},\ t = 0, 1, \cdots, r \right\}.$$

The Delsarte-Goethals sets are nested

$$K_m = DG(m,0) \subset DG(m,1) \subset \cdots \subset DG\left(m, \frac{m-1}{2}\right),$$

and every bilinear form is associated with some matrix in $DG\left(m, \frac{m-1}{2}\right)$.

For example, let $m = 3$ and $g(x) = x^3 + x + 1$, the set $DG(3,1)$ is spanned by K_3, and

$$P^1(100) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad P^1(010) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } P^1(001) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Theorem 12.6. *Every nonzero matrix in $DG(m,r)$ has rank at least $m - 2r$.*

Proof. If x is in the null space of $\sum_{t=0}^{r} P^t(a_t)$, then for all $y \in \mathbb{F}_{2^m}$

$$\mathrm{Tr}\left[xya_0 + \sum_{t=1}^{r}\left(xy^{2^t} + x^{2^t}y\right)a_t\right] = 0.$$

Since $\mathrm{Tr}(x) = \mathrm{Tr}(x^2) = \cdots = \mathrm{Tr}\left(x^{\frac{1}{2}}\right)$ we have

$$\mathrm{Tr}\left[\left((xa_0)^{2^r} + \sum_{t=1}^{r}\left((xa_t)^{2^{r-t}} + a_t^{2^r}\, x^{2^{t+r}}\right)\right)y^{2^r}\right] = 0.$$

Non-degeneracy of the trace now implies

$$(xa_0)^{2^r} + \sum_{t=1}^{r}\left((xa_t)^{2^{r-t}} + a_t^{2^r}\, x^{2^{t+r}}\right) = 0.$$

The LHS is a polynomial of degree at most 2^{2r} so there are at most 2^{2r} solutions. Hence the rank of the binary symmetric matrix $\sum_{t=0}^{r} P^t(a_t)$ is at least $m - 2r$. □

Delsarte-Goethals Frames for Compressed Sensing

In Chapter 11 we introduced two fundamental measures of coherence between the columns of a tight-frame, and showed how these parameters can be related to the performance of the LASSO and OST algorithms in model selection and sparse reconstruction. In this section we construct an explicit sensing matrix (*Delsarte-Goethals frame* [45, 47]) with optimal worst-case coherence μ and average coherence ν. We start by picking an odd number m. The 2^m rows of the Delsarte-Goethals frame $\mathbf{\Phi}$ are indexed by the binary m-tuples x, and the $2^{(r+2)m}$ columns are indexed by the pairs (P, b), where P is an $m \times m$ binary symmetric matrix in the Delsarte-Goethals set $DG(m, r)$, and b is a binary m-tuple. The entry $\varphi_{(P,b),x}$ is given by

$$\varphi_{(P,b),x} = \frac{1}{\sqrt{M}} \, i^{wt(d_P) + 2wt(b)} \, {}_i{}^{xPx^\top + 2bx^\top} \tag{12.2.2}$$

where d_P denotes the main diagonal of P, and wt denotes the *Hamming weight* (the number of 1s in the binary vector). Note that all arithmetic in the expressions $xPx^\top + 2bx^\top$ and $wt(d_P) + 2wt(b)$ takes place in the ring of integers modulo 4, since they appear only as exponents.

The Delsarte-Goethals set $DG(m, r)$, defined in Section 12.2.1, is a binary vector space containing $2^{(r+1)m}$ binary symmetric matrices with the property that the binary sum of any two distinct matrices has rank at least $m - 2r$ (See [140]). The first set $DG(m, 0)$ is the classical Kerdock set, and the last set $DG(m, \frac{(m-1)}{2})$ is the set of all binary symmetric matrices. Given P and b, the vector $xPx^\top + 2bx^\top$ is a codeword in the Delsarte-Goethals code. The set $DG(m, 0)$ corresponds to Kerdock codes, and the set $DG\left(m, \frac{m-1}{2}\right)$ corresponds to all codewords of the second-order Reed-Muller codes. We refer the reader to [41], [43], and [42] for further details.

The r^{th} Delsarte-Goethals frame is determined by $DG(m, r)$ and has $M = 2^m$ rows and $N = 2^{(r+2)m}$ columns. For a fixed matrix P, the 2^m columns $\boldsymbol{\varphi_{(P,b)}}$ $(b \in \mathbb{F}_2^m)$ form an orthonormal basis Γ_P that can also be obtained by postmultiplying the Walsh-Hadamard basis by the unitary transformation diag $\left[i^{xPx^\top}\right]$.

Throughout the rest of this section let $\mathbf{1}$ denote the all-one vector. Also let $\boldsymbol{A}$ denote the *unnormalized* DG frame, i.e., $\mathbf{\Phi} = \frac{1}{\sqrt{M}}\boldsymbol{A}$. We use the following lemmas to show that the Delsarte-Goethals frames are low-coherence tight-frames. First we prove that the columns of the r^{th} Delsarte-Goethals sensing matrix form a group under pointwise multiplication.

Lemma 12.7. *Let $\mathcal{G} = \mathcal{G}(m, r)$ be the set of unnormalized columns $\boldsymbol{A_{(P,b)}}$ where*

$$a_{(P,b),x} = i^{wt(d_P) + 2wt(b)} \, {}_i{}^{xPx^\top + 2bx^\top} \quad , \text{ where } x \in \mathbb{F}_2^m$$

where $b \in \mathbb{F}_2^m$ and where the binary symmetric matrix P varies over the Delsarte-Goethals set $DG(m, r)$. Then $\mathcal{G}$ is a group of order $2^{(r+2)m}$ under pointwise multiplication.

Proof. We have

$$a_{(P,b),x}a_{(P',b'),x} = i^{wt(d_P)+wt(d_{P'})+2wt(b\oplus b')}\, i^{x(P+P')x^\top+2(b\oplus b')x^\top}$$

where $\oplus$ is used to emphasize addition in $\mathbb{F}_2^m$. Write $P+P' = (P\oplus P')+2Q\ (mod\ 4)$ where Q is a binary symmetric matrix. Observe that $2xQx^\top = 2d_Qx^\top (mod\ 4)$, where the diagonal $d_Q = d_P * d_{P'}$ is a pointwise product of d_P and $d_{P'}$.
Thus $a_{[(P,b),x]}a_{[(P',b'),x]}$ equals

$$i^{([wt(d_P)+wt(d_{P'})+2wt(d_P*d_{P'})]+2wt(b\oplus b'\oplus d_P*d_{P'}))}\, i^{x(P+P')x^\top+2(b\oplus b'\oplus d_P*d_{P'})x^\top}, \quad (12.2.3)$$

which is equal to $a_{[(P\oplus P',b\oplus b'\oplus d_P*d_{P'}),t]}$. Therefore, $\mathcal{G}$ is closed under pointwise multiplication, and the possible inner products of columns $\boldsymbol{A}_{(P,b)}, \boldsymbol{A}_{(P',b')}$ are exactly the possible column sums for columns $\boldsymbol{A}_{Q,b''}$ where $Q = P\oplus P'$ and $b'' = b\oplus b'\oplus d_P*d_{P'}$. $\square$

Next we bound the worst-case coherence of the Delsarte-Goethals frames.

Theorem 12.8. *Let Q be a binary symmetric $m\times m$ matrix from the $DG(m,r)$ set, and let $b\in\mathbb{F}_2^m$. If $S \doteq \sum_{x\in\mathbb{F}_2^m} i^{xQx^\top+2bx^\top}$, then either $S=0$, or*

$$S^2 = 2^m + 2r\, i^{v_1Qv_1^\top+2bv_1^\top},\ \textit{where } v_1Q = d_Q.$$

Proof. We have

$$S^2 = \sum_{x,u\in\mathbb{F}_2^m} i^{xQx^\top+uQu^\top+2b(x+u)^\top} = \sum_{x,u\in\mathbb{F}_2^m} i^{(x\oplus u)Q(x\oplus u)^\top+2xQu^\top+2b(x\oplus u)^\top}$$

Changing variables to $v = x\oplus u$ and u gives

$$S^2 = \sum_{v\in\mathbb{F}_2^m} i^{vQv^\top+2bv^\top} \sum_{u\in\mathbb{F}_2^m} (-1)^{(d_Q+vQ)u^\top}.$$

Since the diagonal d_Q of a binary symmetric matrix Q is contained in the row space of Q there exists a solution for the Equation $vQ = d_Q$. Moreover, since Q has rank at least $m-2r$, the solutions to the Equation $vQ=0$ form a vector space E of dimension at most $2r$, and for all $e,f\in E$

$$eQe^\top + fQf^\top = (e+f)Q(e+f)^\top \ \ (mod\ 4).$$

Hence

$$S^2 = 2^m \sum_{e\in E} i^{(v_1+e)Q(v_1+e)^\top+2(v_1+e)b^\top} = 2^m i^{v_1Qz_1^\top+2v_1b^\top} \sum_{e\in E} i^{eQe^\top+2eb^\top}.$$

The map $e \to eQe^\top$ is a linear map from E to $\mathbb{Z}_2$, so the numerator $eQe^\top + 2eb^\top$ also determines a linear map from E to $\mathbb{Z}_2$ (here we identify $\mathbb{Z}_2$ and $2\mathbb{Z}_4$). If this linear map is the zero map then

$$S^2 = 2^m + 2r_i v_1 Q v_1^\top + 2b v_1^\top,$$

and if it is not zero then $S = 0$. □

Corollary 12.9. *Let $\mathbf{\Phi}$ be an $M \times N$ $DG(m,r)$ frame whose column entries are defined by* (12.2.2)*. Then $\mu \leq \frac{2^r}{\sqrt{M}}$.*

Proof. Lemma 12.7 states that the columns of the unnormalized DG frame form a group under pointwise multiplication. Therefore, the inner product between any two columns of this matrix equal to the sum of the entries of another column of the matrix. Consequently, we have

$$\mu = \max_{i \neq j} |\langle \mathbf{\Phi}_i, \mathbf{\Phi}_j \rangle| = \frac{1}{2^m} \max_{i \neq j} |\langle \boldsymbol{A}_i, \boldsymbol{A}_j \rangle| = \frac{1}{2^m} \max_{i \neq 1} |\langle \boldsymbol{A}_i, \mathbf{1} \rangle| \leq \frac{\sqrt{2^{m+2r}}}{2^m} = \frac{2^r}{\sqrt{M}}.$$

□

Next we show that the Delsarte-Goethals frames have significantly smaller average-coherence. In fact, the Delsarte-Goethals frames achieve the lowest average-coherence among all known low-coherence tight-frames (see Table 11.1.)

Lemma 12.10. *Let $\mathbf{\Phi}$ be a $DG(m,r)$ frame with $M = 2^m$, and $N = M^{(r+2)}$. Then $\nu = \frac{1}{N-1}$.*

Proof. We have

$$\nu \doteq \max_i \frac{1}{N-1} \left| \sum_{j \neq i} \langle \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j \rangle \right| = \frac{1}{M(N-1)} \left| \sum_{j \neq i} \langle \boldsymbol{A}_i, \boldsymbol{A}_j \rangle \right| = \frac{1}{M(N-1)} \left| \sum_{i \neq 1} \langle \mathbf{1}, \boldsymbol{A}_i \rangle \right|.$$

Now since the columns of $\boldsymbol{A}$ form a group under pointwise multiplication, and since every row of $\boldsymbol{A}$ has at least one non-identity element, it follows from Lemma 2.17 that every row sum vanishes. Therefore, $\sum_{i \in [N]} \mathbf{1}^\top \boldsymbol{A}_i = 0$, and since $\boldsymbol{A}_1 = \mathbf{1}$ we have

$$\frac{1}{M(N-1)} \left| \sum_{i \neq 0} \langle \mathbf{1}, \boldsymbol{A}_i \rangle \right| = \frac{1}{M(N-1)} \left| -\mathbf{1}^\top \mathbf{1} \right| = \frac{1}{N-1}.$$

□

Lemma 12.11. *Let $\mathbf{\Phi}$ be a $DG(m,r)$ frame. Then $\mathbf{\Phi}$ is a tight-frame with redundancy $\frac{N}{M}$.*

Proof. Let x and x' be two indices in $[M]$. We calculate the inner-product between the rows indexed by x and x'. It follows from Equation (12.2.2) that the inner-product can be written as

$$\begin{aligned}\sum_{P,b} \varphi_{(P,b),x}\overline{\varphi_{(P,b),x}} &= \frac{1}{M}\sum_{P,b} i^{xPx^\top - x'Px'^\top + 2bx^\top - 2bx'^\top} \\ &= \frac{1}{M}\left(\sum_{P} i^{xPx^\top - x'Px'^\top}\right)\left(\sum_{b}(-1)^{b(x\oplus x')^\top}\right).\end{aligned}$$

Therefore, it follows from Lemma 2.17 that if $x \neq x'$ then the inner-product is zero, and is $\frac{N}{M}$ otherwise. □

Putting all the above results together we obtain the following theorem regarding the coherence of the Delsarte-Goethals frames

Theorem 12.12. *Let M be an odd integer, and let r be a positive integer not larger than $\frac{m-1}{2}$. Let $M = 2^m$, and $N = 2^{(r+2)m}$. Let $\mathbf{\Phi}$ be the $M \times N$ DG frame, whose columns are generated by Equation* (12.2.2)*. Then*

1. *$\mathbf{\Phi}$ is a tight-frame with redundancy $\frac{N}{M}$.*
2. *$\mathbf{\Phi}$ is maximally incoherent. That is $\mu(\mathbf{\Phi}) \leq \frac{2^r}{\sqrt{M}}$, and $\nu(\mathbf{\Phi}) = \frac{1}{N-1}$.*

Proof. The tight-frame property follows from Lemma 12.11. Lemma 12.9 bound the worst-case coherence of $\mathbf{\Phi}$, and Lemma 12.10 calculates its average coherence. □

Remark 12.13. *The explicit structure of the DG frames also provides storage and computational advantages over random Gaussian and Rademacher matrices. To see this, observe that each DG frame has the form*

$$\mathbf{\Phi} = \sqrt{\frac{1}{M}}\left[\boldsymbol{D_1 H B},\ \boldsymbol{D_2 H B},\ \cdots,\ \boldsymbol{D_{M^{r+1}} H B}\right], \tag{12.2.4}$$

where $\boldsymbol{B}$ is a diagonal vector with entries $\left[(-1)^{wt(b)}\right]_{b\in\mathbb{F}_2^m}$, $\boldsymbol{H}$ is the unnormalized Hadamard matrix, and each $[\boldsymbol{D_j}]_{j=1}^{M^{r+1}}$ is the diagonal matrix diag $\left[i^{wt(P_j)+xP_jx^\top}\right]_{P_j\in \mathrm{DG(m,r)}}$. As a result, matrix-vector multiplications $\mathbf{\Phi}\boldsymbol{v}$ and $\mathbf{\Phi}^\top\boldsymbol{u}$ only require $O(N\log M)$ running time via the Fast Hadamard Transform.

12.2.2 Real-Valued Delsarte-Goethals Frames

In certain cases, we desire a CS matrix with real-valued entries; there are two possible approaches to adapt the DG frames to a real-valued CS matrix.

First, one can restrict the binary symmetric matrices P to the subset of the DG set of matrices with zero-valued diagonal entries. With such a restriction, the term

$$xPx^{\top} = 2 \sum_{0 \le i < j < 2^m} x_i x_j P_{ij}$$

is an even number, rendering the entries of $\mathbf{\Phi}$ real-valued. However, since only zero-diagonal binary symmetric matrices are used, a zero-diagonal subset of the $DG(m, r+1)$ set is required in order to obtain $N = M^{r+1}$ columns. As a result, Corollary 12.9 implies that the worst-case coherence of the new real-valued sensing matrix is by a factor of $\sqrt{2}$ larger than the worst-case coherence of the complex DG frame.

Alternatively, one can create a CS matrix having twice as many rows as the DG frame by applying the Gray map

$$g : \left\{ \begin{array}{rl} 1 \rightarrow & (1,1) \\ \imath \rightarrow & (1,-1) \\ -1 \rightarrow & (-1,-1) \\ -\imath \rightarrow & (-1,1) \end{array} \right\}$$

to the entries of the complex DG frame.

The Gray map has the property that the norm of the difference between any two powers of $\imath$ is equal to the norm of the difference of their Gray map image vectors. The new Gray-mapped CS matrix, which we denote by $\mathbf{\Phi}^{\boldsymbol{G}}$ has $M = 2^{m+1}$ rows and $N = 2^{(r+2)m}$ columns. The rows of the matrix are indexed by $x \in \mathbb{F}_2^{m+1}$, and its columns are indexed by pairs (G, b), where G is an skew-symmetric matrix, i.e. it has zero diagonals. The entry at row x and column (G, b) of the real-valued DG frame $\mathbf{\Phi}^{\boldsymbol{G}}$ is therefore

$$\varphi^{G}_{(P,b),x} = \frac{1}{\sqrt{M}} \, i^{2wt(b) \; {}_i x G x^{\top} + 2bx^{\top}}, \tag{12.2.5}$$

Calderbank et. al. [140] showed that the correspondence between binary symmetric matrices P in the complex-valued DG frames, and binary skew symmetric matrices G, in the real-valued DG frames is given by

$$G = \left[\begin{array}{cc} 0 & d_P \\ d_P^{\top} & d_P^{\top} d_P + P \end{array} \right].$$

This mapping between $M \times M$ binary symmetric matrices P, and $(m+1) \times (m+1)$ binary skew-symmetric matrices G allows us to prove that real-valued DG frames are incoherent tight-frames.

Theorem 12.14. *Let m be an odd integer, and let r be a positive integer not larger than $\frac{m-1}{2}$. Let $M = 2^{m+1}$, and $N = 2^{(r+2)m}$. Let $\mathbf{\Phi}$ be the $M \times N$ real-valued DG*

frame, whose columns are generated by Equation (12.2.5) *(or equivalently by applying the Gray map to Equation* (12.2.2)*). Then* $\mathbf{\Phi}$ *is a tight-frame with redundancy* $\frac{N}{M}$*, and* $\mu(\mathbf{\Phi}) \leq \frac{2^r}{\sqrt{M}}$.

Proof. The tight-frame property follows from the fact that the real-valued DG frames are unions of orthonormal bases. More precisely, the inner product between any two distinct rows x and y can be written as

$$\begin{aligned} &\frac{1}{M}\sum_{G,b} i^{xGx^\top + 2bx^\top - yGy^\top - 2by^\top} \\ =&\frac{1}{M}\left(\sum_{G} i^{xGx^\top - yGy^\top}\right)\left(\sum_{b}(-1)^{b(x\oplus y)^\top}\right), \end{aligned} \tag{12.2.6}$$

now since b ranges over all field elements, the inner sum is equal to zero unless $x = y$. On the other hand, if $x = y$, then

$$\frac{1}{M}\sum_{G,b} i^{xGx^\top + 2bx^\top - yGy^\top - 2by^\top} = \frac{N}{M}.$$

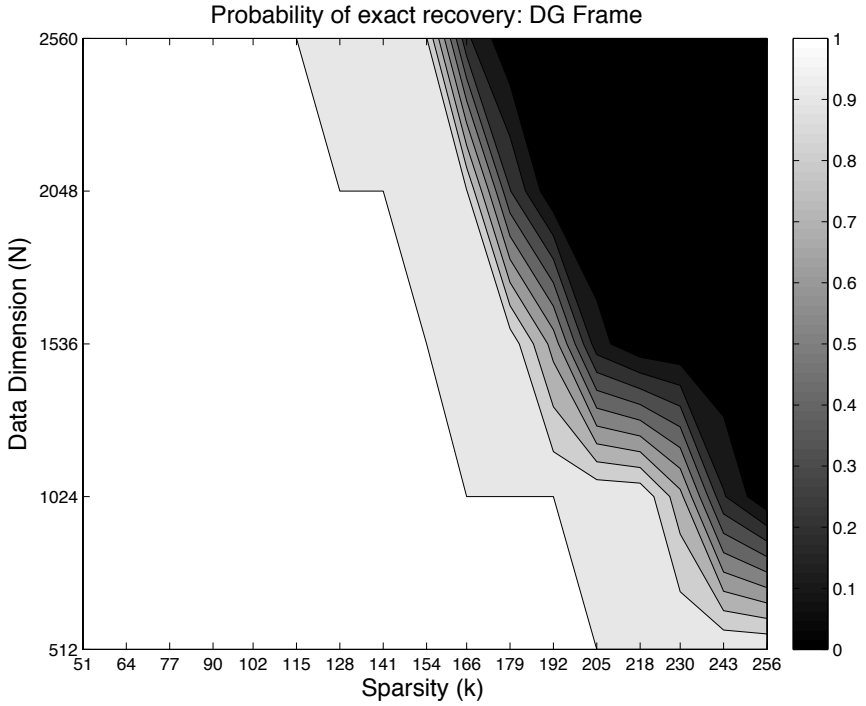
Moreover, it follows from the distance preserving property of the Gray map [41, 42], and Corollary 12.9, that the inner product between any two distinct columns of a real-valued DG frame is at most $\frac{2^r}{\sqrt{M}}$. □

Remark 12.15. *Note that even though the Gray map duplicates the number of required measurements, it also decreases the worst case coherence and the spectral norm of the matrix by a factor* $\frac{1}{\sqrt{2}}$*. In other words, a complexed* $DG(m,r)$ *frame has* $M_{\mathbb{C}} = 2^m$ *rows,* $N_{\mathbb{C}} = 2^{(r+2)m}$ *columns, worst-case coherence* $\mu_{\mathbb{C}} = \frac{1}{2^{\frac{m}{2}-r}}$*, and spectral norm* $\|\mathbf{\Phi}_{\mathbb{C}}\| = \sqrt{2^{(m+1)r}}$*, whereas a real-valued* $DG(m,r)$ *frame has* $M_{\mathbb{R}} = 2^{m+1}$ *rows,* $N_{\mathbb{R}} = 2^{(r+2)m}$ *columns, worst-case coherence* $\mu_{\mathbb{R}} = \frac{1}{2^{\frac{m+1}{2}-r}}$*, and spectral norm* $\|\mathbf{\Phi}_{\mathbb{R}}\| = \sqrt{2^{(m+1)r-1}}$.

12.3 Efficient Compressed Sensing via the Delsarte-Goethals Frames

So far we have proved that Delsarte-Goethals frames are tight-frames with optimal coherence values. On the other hand, in Chapter 11 we proved that the coherence property of Definition 11.1 is a sufficient condition for the fidelity of the OST algorithm (Algorithm 8). Later on, we also showed that if a sensing matrix has sufficiently small worst-case coherence and spectral norm, then most k-sparse vectors have unique low-dimensional representations (Theorem 12.2), and the LASSO algorithm can successfully recover a close approximation to them (Theorem 12.3). Now we combine the

Figure 12.1: Probability of exact signal recovery as a function of the sparsity level k, and the data domain dimension N using a $DG(9,0)$ frame.



low-coherence results of Theorem 12.12 and provide sparse approximation guarantees for LASSO and OST algorithms combined with the Delsarte-Goethals frames.

Theorem 12.16. *Let m be an odd number, and let r be an integer less than $\frac{m-1}{2}$. Let $\mathbf{\Phi}$ be an $M \times N$ DG frame. Assume*

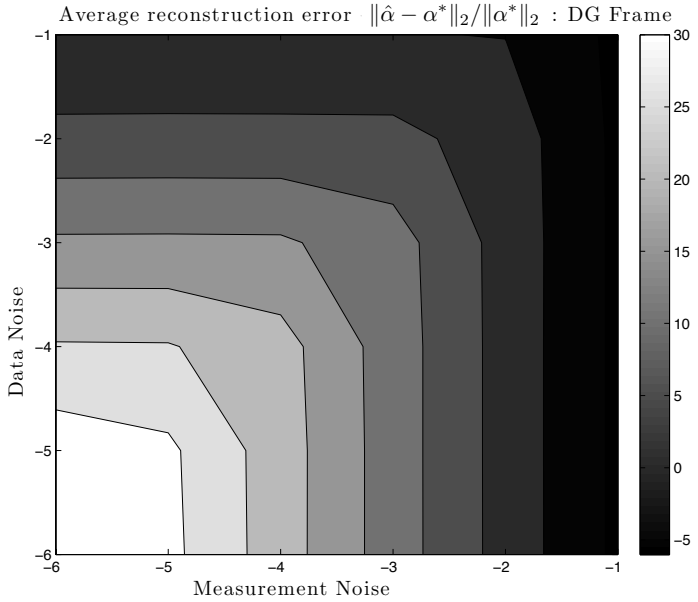
$$k \leq \frac{\kappa_1 M}{2^{2r} \log N},$$

where κ_1 is an absolute constant. Let $\boldsymbol{\alpha}^*$ be a k-sparse vector, such that

1. *The support of the k nonzero coefficients of $\boldsymbol{\alpha}^*$ is selected uniformly at random.*
2. *Conditional on the support, the signs of the nonzero entries of $\boldsymbol{\alpha}^*$ are independent and equally likely to be -1 or 1.*
3. *The distribution of the k non-zero entries of $\boldsymbol{\alpha}^*$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^k$*

Let $\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{\alpha}^* + \mathbf{\Phi}\boldsymbol{e_D} + \boldsymbol{e_M}$, where $\boldsymbol{e_{MD}}$ is the data-domain noise, containing N iid $\mathcal{N}(0, \sigma_D^2)$ Gaussian elements, and $\boldsymbol{e_M}$ is the measurement-domain noise, contain M iid $\mathcal{N}(0, \sigma_M^2)$ Gaussian elements. Let $\sigma = \sqrt{\frac{N}{M}\sigma_D^2 + \sigma_M^2}$. Then if $\|\boldsymbol{\alpha}\|_{\min} \geq$

Figure 12.2: Average reconstruction error as a function of the data domain noise (σ_d), and the measurement domain noise (σ_m) using a $DG(9,0)$ frame.



$8\sigma\sqrt{2\log N}$, *with probability* $1-O(N^{-1})$ *the LASSO estimate*

$$\hat{\boldsymbol{\alpha}} \doteq \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \frac{1}{2}\|\boldsymbol{f}-\boldsymbol{\Phi\alpha}\|^2 + 2\sqrt{2\log N}\,\sigma\,\|\boldsymbol{\alpha}\|_1$$

has the same support and sign as $\boldsymbol{\alpha}$, *and* $\|\boldsymbol{\alpha^*}-\hat{\boldsymbol{\alpha}}\|^2 \le \kappa_2\, k\,\sigma^2$, *where* κ_2 *is a constants independent of* $\boldsymbol{\alpha^*}$.

Proof. First, note that without loss of generality we can assume that the measurement noise is $\boldsymbol{e} \doteq \boldsymbol{\Phi e_D} + \boldsymbol{e_M}$. It follows from Lemma 12.4 that Then $\boldsymbol{e}$ contains M entries, sampled iid from $\mathcal{N}(0,\sigma^2)$. Now Theorem 12.3 guarantees that with probability $1-O(N^{-1})$ the LASSO estimate $\hat{\boldsymbol{\alpha}}$ has the same support as $\boldsymbol{\alpha^*}$, and moreover $\|\boldsymbol{\Phi\alpha^*}-\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}\|^2 \le c_2\, k\,\sigma^2$. Finally, since both $\hat{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha^*}$ are supported on the same random k-subset of $[N]$, Theorem 12.1 guarantees that with probability not exceeding $\frac{2}{N}$, $\|\boldsymbol{\alpha^*}-\hat{\boldsymbol{\alpha}}\|^2 \le 2\|\boldsymbol{\Phi\alpha^*}-\boldsymbol{\Phi}\hat{\boldsymbol{\alpha}}\|^2$, which completes the proof. □

Theorem 12.17. *Let* m *be an odd number, and let* r *be an integer less than* $\frac{m-1}{2}$. *Let* $\boldsymbol{\Phi}$ *be an* $M\times N$ *DG frame. Suppose* $\boldsymbol{e_D}$ *be the data-domain noise vector containing* N *iid* $\mathcal{N}(0,\sigma_D^2)$ *elements,* $\boldsymbol{e_M}$ *is the measurement-domain noise vector containing* M *iid* $\mathcal{N}(0,\sigma_M^2)$ *Gaussian elements, and let* $\sigma = \sqrt{\frac{N}{M}\sigma_D^2+\sigma_M^2}$. *Then with probability at least* $1-O(N^{-1})$, *the SOST algorithm (Algorithm 9) successfully recovers the support*

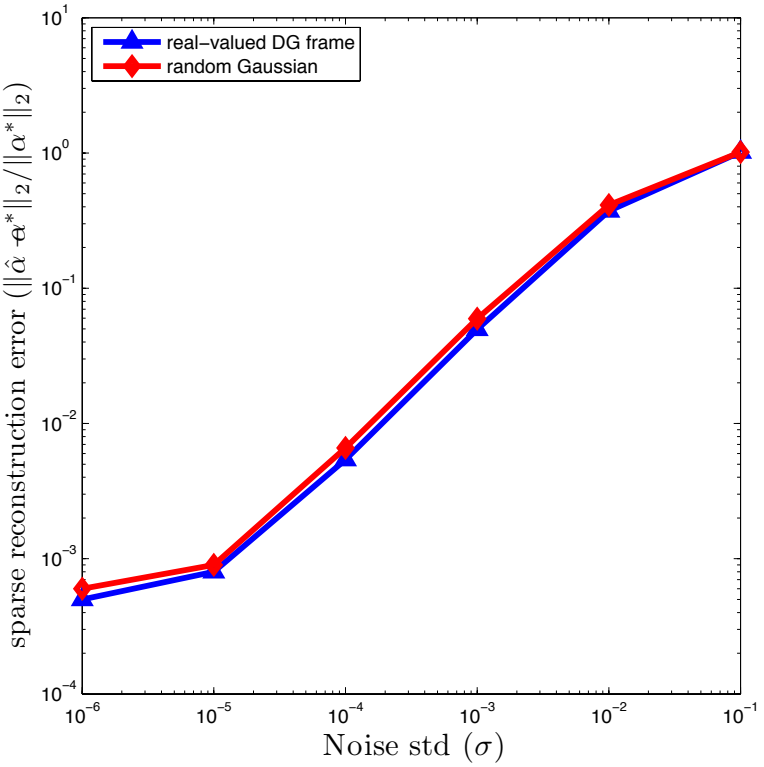


Figure 12.3: The impact of the noise in the measurement domain on the sparse approximation error $\|\boldsymbol{\alpha}^* - \hat{\boldsymbol{\alpha}}\|_2/\|\boldsymbol{\alpha}^*\|_2$ of the LASSO algorithm with real-valued DG frames (triangle), and random Gaussian matrices (square). Here the noise standard deviation ranges from 10^{-6} to 10^{-1}, and used $k = 200$, $M = 1024$, and $N = 3072$.

of $\boldsymbol{\alpha}^*$*, as long as we have that* $k \leq M/(2\log N)$ *and*

$$\text{MAR} > \max\left\{\frac{c_2 k \log N}{M \cdot \text{SNR}}, \frac{c_3' 2^{2r} k \log N}{M}\right\}. \tag{12.3.1}$$

Here, c_2 *and* c_3' *are absolute constants, and the probability of failure is with respect to the true model* $\mathcal{S}$ *and the noise vectors* $\boldsymbol{e_M}$ *and* $\boldsymbol{e_D}$*.*

Proof. Since, $\boldsymbol{\Phi}$ is a tight-frame with redundancy $\frac{N}{M}$, it follows from Lemma 12.4 that Then $\boldsymbol{e}$ contains M entries, sampled iid from $\mathcal{N}(0, \sigma^2)$. The proof of Theorem 11.4 then follows directly, from Theorem 11.3 by setting $\mu = \frac{2^r}{\sqrt{M}}$. □

Here we present numerical experiments to evaluate the performance of the LASSO program with DG frames. Here we used a complex-valued DG frame, with $m = 9$, and $r = 0$. We fixed the number of measurements to $M = 512$ and swept across

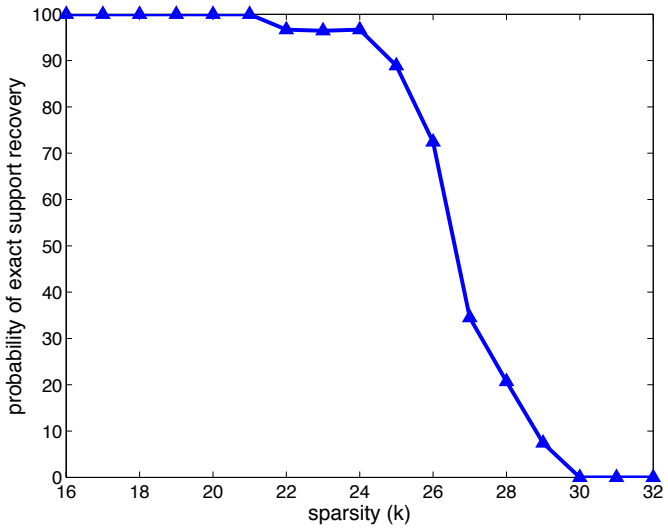


Figure 12.4: Probability of exact support recovery as a function of the sparsity level k. Here we set $M = 2^{12}$, $N = 2^{32}$ and used a real-valued DG frame. The OST algorithm is used for support recovery. Note that the OST algorithm requires less than a minute to terminate, whereas most recovery algorithms designed for random (or even expander-based) compressed sensing do not converge in a reasonable time.

the sparsity level k, and the data dimension N[1]. For each (k, N)-pair, we repeated the following 100-times: (i) generate a random sparse vector with unit norm (ii) generate compressive measurements (no noise) using the DG frame, and (iii) recover the signals using LASSO. Figure 12.1 reports the probability of exact recovery over the 100 trials.

We also performed a similar experiment in the noisy regime. Here we independently changed the standard deviations of the data-domain noise (σ_d) and the measurement noise (σ_m) from 10^{-6} to 10^{-1}. We then used the LASSO program to obtain a sparse approximation $\hat{\boldsymbol{\alpha}}$ to the k-sparse vector $\boldsymbol{\alpha}$. Figure 12.2 plots the average reconstruction error $(-10\log_{10}(\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}\|_2))$ as a function of σ_M and σ_D.

Figure 12.3 plots the sparse approximation error as a function of the noise in the measurement domain. In the measurement noise study, a $\mathcal{N}(0, \sigma_M^2)$ iid measurement noise vector is added to the sensed vector to obtain the M dimensional vector $\boldsymbol{f}$. The original k-sparse signal $\boldsymbol{\alpha}^*$ is then approximated by solving the LASSO program with $\lambda = 2\sqrt{2\log N}\sigma_M$. Figure 12.3 and many similar experiments show that real-valued DG frames and random Gaussian matrices of the same size have almost identical performance in terms of noisy signal recovery using the LASSO. However, in contrast to random Gaussian matrices, DG frames do not suffer from storage and computational limitations.

[1]To vary N, we selected the first N columns of a $DG(9, 0)$ frame (which is still an incoherent tight-frame as long as $\frac{N}{M}$ is an integer).

Finally, consider a wireless network, in which each user is identified by a unique 32 bit MAC address. Let $\mathbf{\Phi}$ be a $2^{12} \times 2^{32}$ sensing matrix obtained from selecting the first 2^{32} columns of a real-valued Delsarte-Goethals frame, based on a DG$(11, 1)$ set. Here each user is assigned to one column of $\mathbf{\Phi}$. At the transmission time, each active user i, simply submits the 2^{12} dimensional column $\boldsymbol{\varphi_i}$. Therefore, the receiver receives a superposition of the columns of all active users. However, at each time period, there are only very few $k \ll 2^{32}$ users that are active. The receiver can exploit this prior sparsity knowledge, and recover the active users. However, the number of active users is not known a priori to the receiver, and therefore the receiver must use a model-order agnostic recovery algorithm.

In Figure 12.4 we used the OST algorithm for support recovery of sparse vectors. Here we set $N = 2^{32}$, $M = 2^{12}$, and used the first 2^{32} columns of a real-valued DG$(11, 1)$ frame. Note that here the data dimension N is too large for convex optimization methods (or even most greedy methods) to converge in a reasonable time. Even expander-based methods have difficulties in this situation. Nevertheless, the OST algorithm only requires one matrix-vector multiplication using the Fast Hadamard Transform. In our implementation, this matrix-vector multiplication took less than one minute. The running time of the OST algorithm can be significantly reduced by parallelizing the algorithm.

In this experiment, the sparsity level k was changed from 16 to 32. The sparse vector was generated by selecting a random k-subset of the 2^{32} columns. To avoid the random-sign effect, the k non-zero entries of $\boldsymbol{\alpha}^*$ all had value one. We repeated each experiment 100 times independently, and recorded the average probability of exact recovery as a function of the sparsity level k. As illustrated in Figure 12.4, the algorithm can recover almost all k-sparse vectors efficiently as long as k is smaller than 25.

Part V

Model-based Compressed Sensing

Chapter 13

Fast Model-based Thresholding with Nesterov's Gradient Method

13.1 What is Model-based Compressed Sensing

In Section 3.1 we introduced the main objectives of compressed sensing as

- Designing an efficient $M \times N$ sensing matrix $\mathbf{\Phi}$, and
- Designing an efficient and robust reconstruction algorithm

We then proposed expander-based and Reed-Muller-based matrices as two families of deterministic and efficient sensing matrices. We also looked at the general problems of sparse approximation and model selection, and introduced the GAME and OST algorithms as examples of two efficient sparse recovery algorithms, with sparse approximation and model-selection guarantees. We further showed that in many practical situations our proposed matrices and recovery algorithms have similar or even better performance compared to the random sensing framework. Moreover the structures of our deterministic matrices provide several storage and computational advantages.

While such measurement rates and recovery complexities are impressive and have the potential to impact a broad set of compressive sensing applications, sparsity is merely a first-order description of signal structure; in many applications we have considerably more *a priori* information on the sparse coefficients of the state-of-the-art approaches are only recently beginning to exploit [19]. For instance, in group testing of defective elements among a collection of N items, the defective elements typically cluster across known blocks of items. In data streaming computing, large elements of $\boldsymbol{\alpha}^*$ often have a minimum distance between each other, and have positive values. In compressive imaging, the sparse coefficients of the signal cluster across the branches of tree structures, such as natural images and wavelet trees [77, 81].

In this chapter, we will show that by exploiting a priori information on coefficient structure in addition to signal sparsity, we can make the sparse linear sketching approaches to dimensionality reduction more powerful. As an indicator of what can be achieved, recent work [19, 67] leverages such structured sparsity in CS with random dense matrices to reduce, in some cases significantly, the number of measurements M required to stably recover a signal by permitting only certain configurations of the large and zero/small coefficients via dependencies on the support of the sparse coefficients–the set of indices corresponding to the nonzero entries. During signal recovery, structured sparsity also enables the recovery algorithms to better differentiate true signal information from recovery artifacts, leading to a more robust recovery.

We will derive an algorithmic framework for structured sparse recovery, which unifies combinatorial optimization with the non-smooth convex optimization framework by Nesterov [204, 203]. The algorithm proposed in this chapter can be viewed as a generalization of the OST algorithm derived in Section 11.2. In our approach, we optimally use the gradient information in the convex data error objective to navigate over the non-convex set of structured sparse signals. By optimal, we mean that our algorithms match the known convergence bounds for gradient methods for convex optimization problems [202].

Efficient combinatorial optimization is the key ingredient in this loop to calculate the best projection of a given vector onto the non-convex sparse signal set. Our combinatorial approach with expander-based and Reed-Muller-based matrices achieves the geometric, optimization-based limits for random dense matrices [98] at a fraction of their computational cost.

13.2 Problem Formulation

Model-based compressed sensing is the topic of efficient sparse recovery when some extra prior knowledge is available about the sparse vector $\boldsymbol{\alpha}^*$. Throughout this chapter, we identify any extra prior knowledge about the sparse vector $\boldsymbol{\alpha}^*$ by a model $\mathcal{M}$. We also use the notation $\Sigma_{\mathcal{M}(k)}$ to denote the set of all k-sparse vectors in that model $\mathcal{M}$.

Let $\boldsymbol{\Phi}$ be an arbitrary $M \times N$ matrix, and let $\boldsymbol{f}$ be an M dimensional vector. In the *model-based sparse-approximation in ℓ_2 norm*, we focus on the following non-convex optimization problem.

$$\text{minimize}\, \mathcal{R}(\boldsymbol{\alpha}) \quad \text{s.t. } \boldsymbol{\alpha} \in \Sigma_{\mathcal{M}(k)} \tag{13.2.1}$$

where the loss function $\mathcal{R}(\boldsymbol{\alpha}) : \Sigma_{\mathcal{M}(k)} \rightarrow \mathbb{R}^+$ is defined as the squared ℓ_2 loss

$$\mathcal{R}(\boldsymbol{\alpha}) \doteq \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2. \tag{13.2.2}$$

Here $\Sigma_{\mathcal{M}(k)}$ is any k-dimensional restricted union-of-subspace (RUS) model with a *tractable* approximation algorithm $\mathcal{M}$ that can calculate the projection of any $\boldsymbol{v} \in \mathbb{R}^N$ into $\Sigma_{\mathcal{M}(k)}$:

$$\mathcal{M}_k(\boldsymbol{v}) = \arg \min_{\boldsymbol{\alpha} \in \Sigma_{\mathcal{M}(k)}} \|\boldsymbol{v} - \boldsymbol{\alpha}\|_2. \tag{13.2.3}$$

Tractable RUS models include but are not limited to (i) k-sparse signals (Σ_k), (ii) (k, b)-sparse signals where k-sparse coefficients live in at most b unknown contiguous blocks on a chain graph, (iii) k-tree sparse where k-sparse coefficients lie on a rooted connected subtree of an N-dimensional tree, and (iv) (k, Δ)-sparse signals where k-sparse coefficients are separated by at least Δ zeros on a chain graph. For Σ_k, $\mathcal{M}_k$ is simple hard thresholding based on sorting the signal coefficients in terms of decreasing magnitude and keeping the largest k while setting the others to zero. For other models, efficient combinatorial and mixed integer model approximation algorithms exist [20].

In the rest of this chapter we propose an efficient algorithm for efficiently solving Equation (13.2.1). In Section 13.3, we set up key properties of objective function that the later sections build upon. Section 13.4 provides the proposed NIHT algorithms, and Section 13.5 illustrates the compressive sensing performance of NIHT, and compares it with the ℓ_1 minimization algorithms.

13.3 Bregman Proxies for Model-Based Sparse Approximation

The model-based sparse approximation problem is not only ill-posed (since the matrix $\boldsymbol{\Phi}$ has a nontrivial kernel), but is also known to be *NP*-hard [197]. In other words, there is no hope of being able to find an *exact* solution for Equation (13.2.1) in general. However, in Chapter 6.1 we proposed efficient algorithms for *approximately* solving the problem of sparse approximation in the ℓ_q norm. Here we use similar ideas and show that even though the model-based sparse approximation is *NP*-had, it is still possible to propose efficient algorithms for approximately solving it.

Similar to Chapter 6.1, we start by defining a proper Bregman function (see Section 7.1 for definitions). We use the loss function $\mathcal{R}(\boldsymbol{\alpha}) \doteq \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2$ as the Bregman function. The following lemma relates that the Bregman distance between any two points $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \Sigma_{\mathcal{M}(k)}$ to their Euclidean distance in the measurement domain.

Lemma 13.1. *Let $\boldsymbol{\Phi}$ be an $M \times N$ matrix, and $\boldsymbol{f}$ be an M dimensional vector. Define the Bregman function $\mathcal{R}(\boldsymbol{\alpha}) : \Sigma_{\mathcal{M}(k)} \to \mathbb{R}^+$ as*

$$\mathcal{R}(\boldsymbol{\alpha}) \doteq \|\boldsymbol{f} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2.$$

Then the Bregman distance between any two model-sparse vectors $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \Sigma_{\mathcal{M}(k)}$ is

$$\mathcal{B}_{\mathcal{R}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \|\boldsymbol{\Phi}(\boldsymbol{\alpha} - \boldsymbol{\alpha}')\|_2^2.$$

Proof. The proof of Lemma 13.1 relies on some obvious algebraic manipulation. From the definition of the Bregman distance (Definition 7.1) we have

$$\begin{aligned}\mathcal{B}_{\mathcal{R}}(\boldsymbol{\alpha},\boldsymbol{\alpha}') &= \mathcal{R}(\boldsymbol{\alpha}) - \mathcal{R}(\boldsymbol{\alpha}') - \langle(\boldsymbol{\alpha}-\boldsymbol{\alpha}'), \nabla\mathcal{R}(\boldsymbol{\alpha}')\rangle \\ &= \|\boldsymbol{f}-\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 - \|\boldsymbol{f}-\boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2^2 + 2\langle(\boldsymbol{\alpha}-\boldsymbol{\alpha}'), \boldsymbol{\Phi}^\top(\boldsymbol{f}-\boldsymbol{\Phi}\boldsymbol{\alpha}')\rangle \\ &= \|\boldsymbol{\Phi}\boldsymbol{\alpha}\|_2^2 - \|\boldsymbol{\Phi}\boldsymbol{\alpha}'\|_2^2 - 2\langle\boldsymbol{f}, \boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\rangle + 2\langle(\boldsymbol{f}-\boldsymbol{\Phi}\boldsymbol{\alpha}'), \boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\rangle \\ &= \langle\boldsymbol{\Phi}(\boldsymbol{\alpha}+\boldsymbol{\alpha}'), \boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\rangle - 2\langle\boldsymbol{\Phi}\boldsymbol{\alpha}', \boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\rangle = \|\boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\|_2^2.\end{aligned}$$

□

Throughout the rest of this chapter let L_{2k} denote the *restricted Lipschitz constant* of $\boldsymbol{\Phi}$:

$$\|\boldsymbol{\Phi}(\boldsymbol{\alpha}-\boldsymbol{\alpha}')\|_2^2 \leq \mathrm{L}_{2k}\|\boldsymbol{\alpha}-\boldsymbol{\alpha}'\|_2^2,$$

for every $\boldsymbol{\alpha},\boldsymbol{\alpha}' \in \Sigma_{\mathcal{M}(k)}$. Observe that from Lemma 13.1 we have $\mathcal{B}_{\mathcal{R}}(\boldsymbol{\alpha},\boldsymbol{\alpha}') \leq \mathrm{L}_{2k}\|\boldsymbol{\alpha}-\boldsymbol{\alpha}'\|_2^2$; therefore if we define,

$$U(\boldsymbol{\alpha},\boldsymbol{\alpha}') \doteq \boldsymbol{f}(\boldsymbol{\alpha}') + \langle(\boldsymbol{\alpha}-\boldsymbol{\alpha}'), \nabla\boldsymbol{f}(\boldsymbol{\alpha}')\rangle - \mathrm{L}_{2k}\|\boldsymbol{\alpha}-\boldsymbol{\alpha}'\|_2^2,$$

then for every $\boldsymbol{\alpha}' \in \Sigma_{\mathcal{M}(k)}$ we have

$$\min_{\boldsymbol{\alpha}\in\Sigma_{\mathcal{M}(k)}} \boldsymbol{f}(\boldsymbol{\alpha}) \leq \min_{\boldsymbol{\alpha}\in\Sigma_{\mathcal{M}(k)}} U(\boldsymbol{\alpha},\boldsymbol{\alpha}').$$

In the next section we propose an iterative approximation algorithm that iteratively selects a proper value $\boldsymbol{\alpha}'^t$ through the Nesterov scheme [204], and then solves the corresponding optimization problem

$$\text{minimize}_{\boldsymbol{\alpha}\in\Sigma_{\mathcal{M}(k)}} U(\boldsymbol{\alpha},\boldsymbol{\alpha}'^t),$$

as a proxy for minimizing the loss function $\boldsymbol{f}(\boldsymbol{\alpha})$.

13.4 Algebraic pursuits and the NIHT algorithm

In [68], Cevher proposes two algorithms, called Algebraic Pursuits (ALPS), that fuse Nesterov's optimal gradient methods with combinatorial model-based projection algorithms for sparse approximation. For instance, the fast Lipschitz iterative hard thresholding (FLIHT) scheme of ALPS has the following recursion ($a_{t+1} = 0.5\left(1+\sqrt{1+4{a_t}^2}\right)$, $a_1 = 1$, and $\theta_t = \frac{a_t-1}{a_{t+1}}$):

$$\boldsymbol{\alpha}^{t+1} \doteq \times\mathcal{M}_k\left(\boldsymbol{y}^{\boldsymbol{t}} - \frac{1}{L_{3K}}\nabla\boldsymbol{f}(\boldsymbol{y}^{\boldsymbol{t}})\right), \quad \boldsymbol{y}^{\boldsymbol{t+1}} \doteq \boldsymbol{\alpha}^t + \theta_i(\boldsymbol{\alpha}^t - \boldsymbol{\alpha}^{t-1}). \tag{13.4.1}$$

Here we propose a third algebraic pursuit algorithm, called NIHT for Nesterov Iterative Hard Thresholding. The algorithm is based upon Nesterov's proximal gradient method [204]. The pseudocode of the NIHT algorithm is summarized as Algorithm 10.

Algorithm 10 NIHT Algorithm for Model-Based Sparse Approximation in ℓ_2-norm.

Inputs: M-dimensional vector $\boldsymbol{f}$, $M \times N$ matrix $\boldsymbol{\Phi}$, and number of iterations T
Output: N-dimensional vector $\hat{\boldsymbol{\alpha}}$

0. Set $\boldsymbol{\alpha^0} \doteq \mathbf{0}_N$ and $\boldsymbol{x^0} \doteq \mathbf{0}_N$.
for $t = 1, \ldots, T$ **do**
1. Set $\boldsymbol{x^t} \doteq \boldsymbol{x^{t-1}} + \frac{2(t+1)}{\mathrm{L}_{2k}} \boldsymbol{\Phi}^\top (\boldsymbol{f} - \boldsymbol{\Phi \alpha^{t-1}})$.
2. Set $\boldsymbol{y^t} \doteq \boldsymbol{\alpha}^{t-1} + \frac{2}{\mathrm{L}_{2k}} \boldsymbol{\Phi}^\top (\boldsymbol{f} - \boldsymbol{\Phi \alpha^{t-1}})$.
3. Set $\boldsymbol{\alpha^t} \doteq \mathcal{M}_k \left(\tau_t \boldsymbol{x^t} + (1 - \tau_t) \boldsymbol{y^t} \right)$. where $\tau_t \doteq \frac{2}{t+3}$.
end for
6. Output $\hat{\boldsymbol{\alpha}} \doteq \boldsymbol{\alpha^T}$.

The proposed NIHT algorithm is a first-order gradient projection algorithm. By first-order, we mean that at each iteration, the algorithm only requires one gradient calculation and a k-sparse model approximation. If an expander-based or a Reed-Muller-based sensing matrix is used, the matrix-vector multiplication can be calculated efficiently in time $O(N \log N)$. Therefore, in contrast to convex-programming algorithms, NIHT is scalable and can handle much larger data dimensions.

Nevertheless, although the exact behavior of the NIHT algorithm is known in some special cases [68], deriving sharp estimation guarantees for the performance of the NIHT algorithm is still an interesting open problem. However, in the next section we provide several practical comparisons between the NIHT algorithm and ℓ_1-minimization methods in the compressive sensing to demonstrate its superiority.

13.5 Experimental Results

13.5.1 Phase Transition

Donoho and Tanner's combinatorial geometry based theory precisely quantifies the fundamental ℓ_1-sparsity and compression trade-off (k vs. M) that NIHT is competing with. The theory predicts the exact location in sparsity-undersampling domain where state-of-the-art algorithms exhibit phase transitions in their performance. The theory states that CS algorithms should be able to recover k-sparse signals from $M \gtrsim 2k \log\left(\frac{N}{M}\right)$ measurements; this threshold appears quite sharp for Gaussian partial Fourier, and expander-based measurement matrix ensembles [98, 29].

To see how NIHT compares to the ℓ_1 theoretical phase transitions, we performed Monte Carlo simulations amounting to a month of CPU time. We fixed the signal dimension to $N = 1000$ and sweep across k and M values (120 and 200 sample points, respectively). For each (k, M)-pair, we repeated the following 100 times: (i) generated a random sparse vector with unit norm, (ii) generated compressive measurements (no noise) using Gaussian, Fourier, and expander graph sampling matrices (incomplete), and (iii) recovered the signals using Basis Pursuit (BP), FLIHT, and NIHT. Both

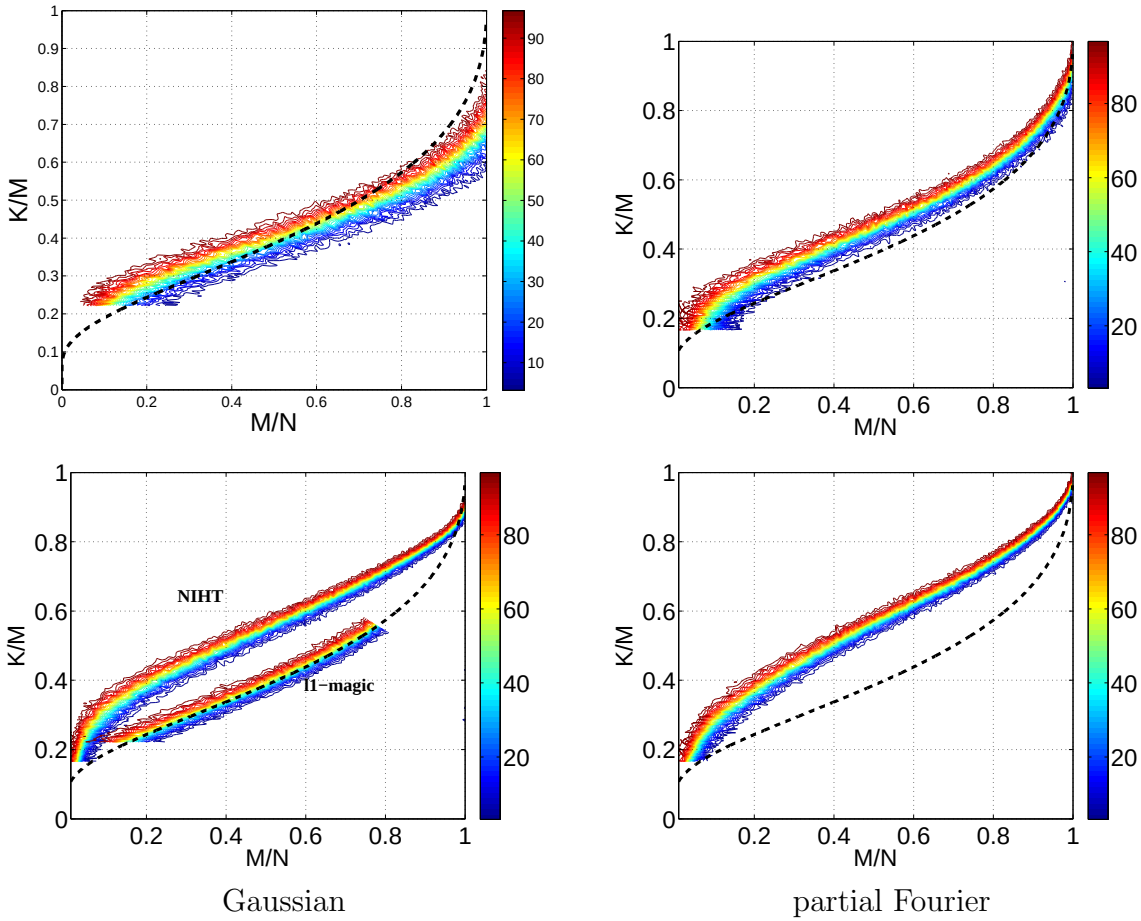
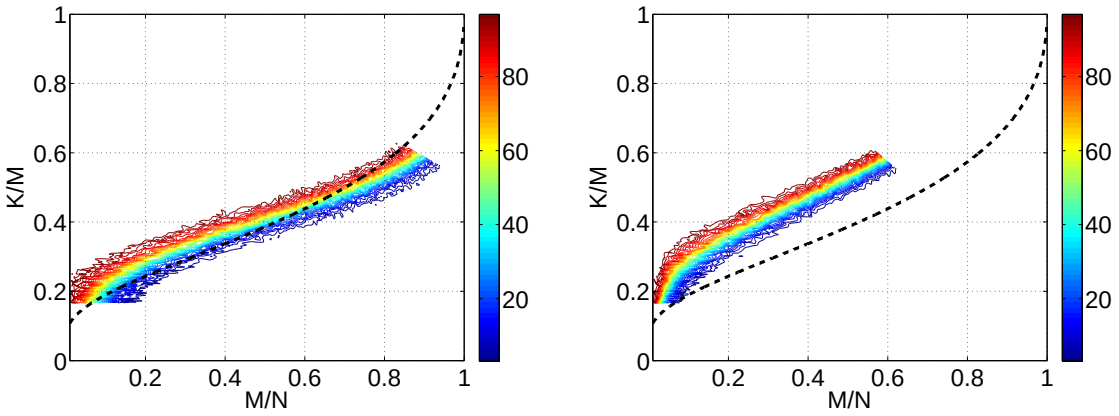


Figure 13.1: Phase transition curves FLIHT (top row) and NIHT (bottom row) are compared to Donoho-Tanner bound (dashed). Corresponding failure percentages are shown.

FLIHT and NIHT algorithms use the same number of iterations 1000. We then reported the number of recoveries that obtain this accuracy or better.

Figure 13.1 summarizes the results for Gaussian and partial Fourier matrices. The results are quite promising for NIHT. For comparison, we also provide the ℓ_1-magic basis pursuit results (the interior point method where the Newton system is solved with conjugate gradients) [52], which match the Donoho-Tanner phase transition curve (c.f., within NIHT/Gaussian). Compared to ℓ_1-magic, NIHT increases the number of sparse coefficients that can be recovered from the same measurements approximately by 25%. FLIHT performs on par with ℓ_1-magic. Both FLIHT and NIHT algorithms achieve this performance at the fraction of ℓ_1-magic's computational cost. Figure 13.2 shows the results of the same experiment with 8-regular expander graphs.



(a) Flight. (b) NIHT.

Figure 13.2: Phase transition curves of expander-based FLIHT (a) and NIHT (b) algorithms in expander-based compressed sensing. Here we fixed $N = 1000$, and $d = 8$. Corresponding failure percentages are shown.

13.5.2 Empirical Noise Robustness

Figure 13.3 illustrates the impact of the measurement noise on the sparse approximation error of the NIHT algorithm and the Basis Pursuit Denoising algorithm of Theorem 3.7. Here we set $M = 1024, N = 3072$, and $k = 100$. The measurement noise elements are sampled iid from a $\mathcal{N}(0, \sigma^2)$ Gaussian distribution. In this experiment we ranged σ from 10^{-6} to 10^{-1}. Each experiment is repeated independently 100 times in the following way. We first generated a 1024×3072 real-valued DG frame, and a random Gaussian matrix of the same size. We then generated a k-sparse vectors with random support, random sign, and unit ℓ_2 norm. The SPGL package is used for solving the Basis Pursuit Denoising problem [244, 243], and the reconstruction error is measured as $\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_2$.

Figure 13.3 compares the mean ℓ_2 approximation error of the four (matrix,algorithms) pairs as a function of the measurement noise. It turns out that NIHT outperforms Basis Pursuit Denoising in high SNR regimes, whereas Basis Pursuit Denoising is the winner when the noise level is too high. Moreover, there is almost no difference between the performance of a real-valued DG frame, and a Gaussian matrix of the same size.

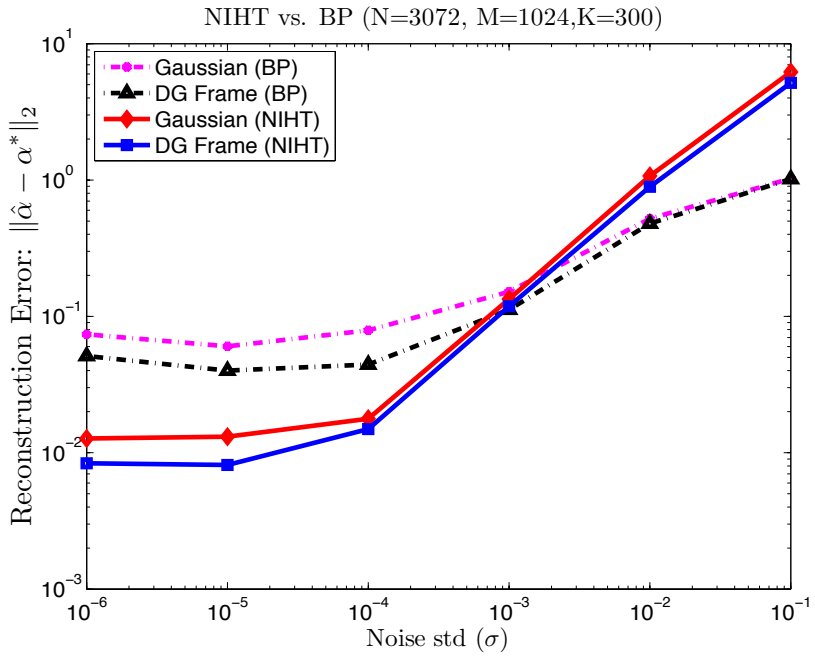


Figure 13.3: Noise tolerance of NIHT, and Basis Pursuit Denoising algorithms are illustrated for $k = 300$, $M = 1024$, and $N = 3072$. Note that there is almost no difference between the performance of a real-valued DG frame, and a random Gaussian matrix of the same size.

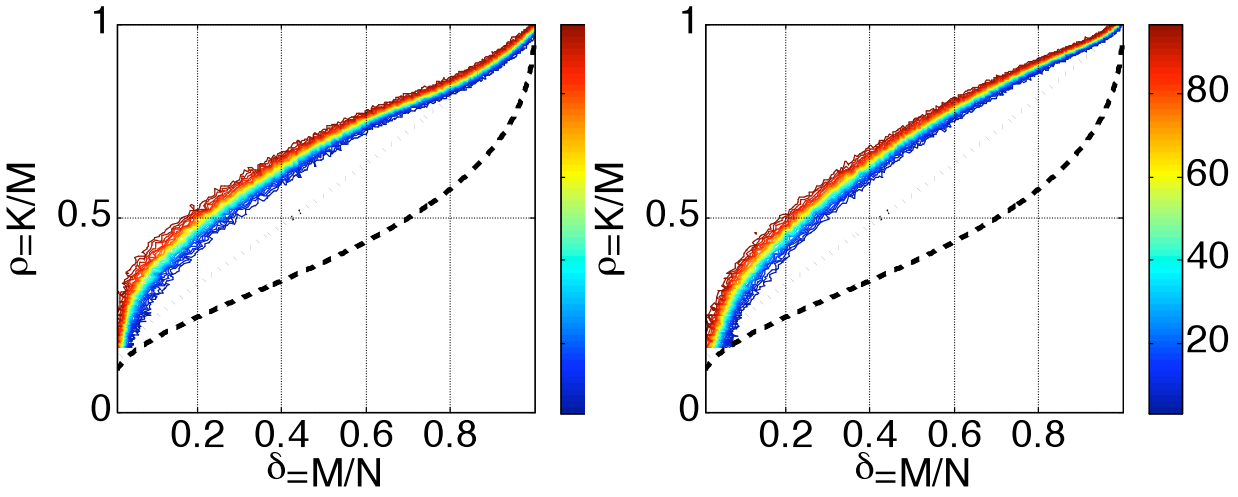
13.5.3 Model-based Recovery

All-positive Model

Figure 13.4 show the phase transition of NIHT with *positive* k-sparse signals, using Gaussian, partial Fourier, and expander-based sensing matrices. We set $N = 1000$, and each experiment was repeated independently 100 times. The ℓ_1-magic results are also provided for comparison. At the end of each iteration, the algorithm only maintains the k largest positive entries of the recovered vector. Observe that the prior positivity information, i.e. knowing that the k-sparse signal has positive values *a priori*, significantly increases the performance of the NIHT algorithm.

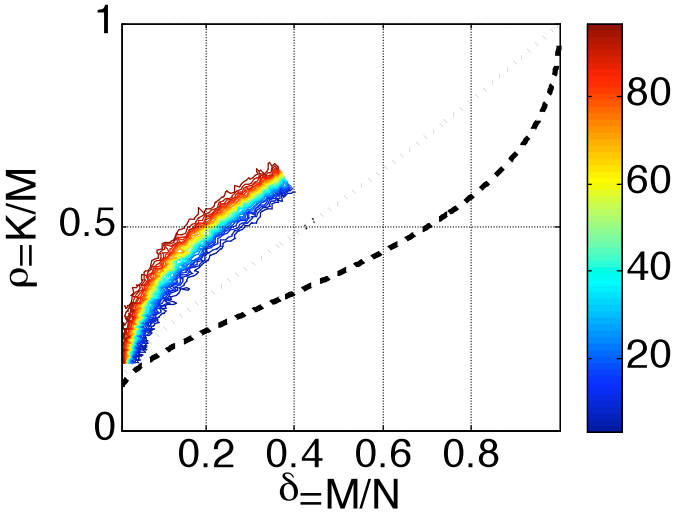
Block Sparsity Model

In this experiment, we considered a specific nested RUS model: *block sparsity*. In a block-sparse signal, the locations of the sparse coefficients cluster in blocks under a specific sorting order. Block-sparse signals have been previously studied in several



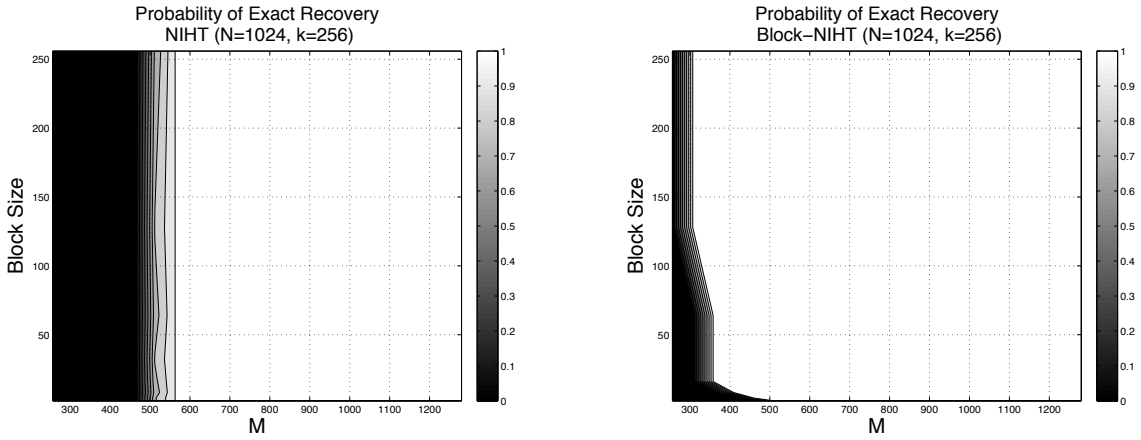
(a) Gaussian

(b) Partial Fourier.



(c) Expander.

Figure 13.4: Phase transition curve of the NIHT algorithm with positive sparse signals is compared to Donoho-Tanner bound (dashed). Not that the prior positivity knowledge significantly improves the reconstruction accuracy.



(a) Expander (b) Expander (model-based with block sparsity)

Figure 13.5: The impact of block-sparsity on the performance of the NIHT algorithm is significant. Exploiting the block structure in addition to signal sparsity, NIHT decreases the number of measurements significantly.

different applications, including DNA microarrays and magnetoencephalography. An equivalent problem arises in signal ensembles, such as array signal processing and MIMO communication [20]. It has been shown that the block-sparse structure enables signal recovery from a reduced number of CS measurements when the recovery algorithms exploit this specific structure [20, 19].

The block sparse model approximation is quite simple: if a sparse coefficient is selected within the predefined block of size b, all the coefficients must be turned within the same block. Hence, block sparse approximation is–in a way–equivalent to unstructured sparse approximation: instead of picking the top k-coefficients by their energy, we pick the top k/b blocks by summing up their ℓ_2-energy. For simplicity, we consider uniform block sizes of powers of 2 on the signal vector; hence, the signal sparsity is also restricted to be a power of 2.

Figure 13.5 investigates the advantage of incorporating the block-sparsity information on the probability of exact recovery for $N = 1024$ and $k = 256$. We vary the block sparsity level as $b = (2, 4, \ldots, 128, 256)$ and also the number of expander-based measurements from $M = 256$ to 1024. Figure 13.5(a) plots the probability of successful recovery while using the NIHT algorithm, whereas Figure 13.5(b) plots the probability of success of the NIHT algorithm with the block-sparsity projection. We observe that the block-sparsity model significantly reduces the minimum number of measurements required for exact recovery.

13.5.4 Compressive Imaging

In our first experiment, we compared the performance of the NIHT and Basis Pursuit algorithms on recovering the 128×128 phantom image. Here $N = 128^2$. We used $M = 0.33 \times N$ expander-based measurements using an expander-graph with left degree $d = 8$. We also set the number of iterations of the NIHT algorithm to 2000 with sparsity level $k = 0.5 \times M$. For sparsity basis, we chose the db2 Daubechies wavelets basis, and use the SPGL package [244, 243] for solving the Basis Pursuit optimization.

Figure 13.6 compares the NIHT algorithm with the Basis Pursuit method. Since the Lipschitz constant for this problem is not available, we chose a large constant and ran the algorithm for 2000 iterations. The reconstruction SNR is measured as $\text{SNR} \doteq -10 \log_{10} \left(\frac{\|\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\|_2}{\|\boldsymbol{\alpha}^*\|_2} \right)$, where $\boldsymbol{\alpha}^*$ is the wavelet coefficient vector and $\hat{\boldsymbol{\alpha}}$ is the output of the algorithm. As illustrated in Figure 13.6, the NIHT algorithm dominates the Basis Pursuit algorithm in terms of the reconstruction accuracy.

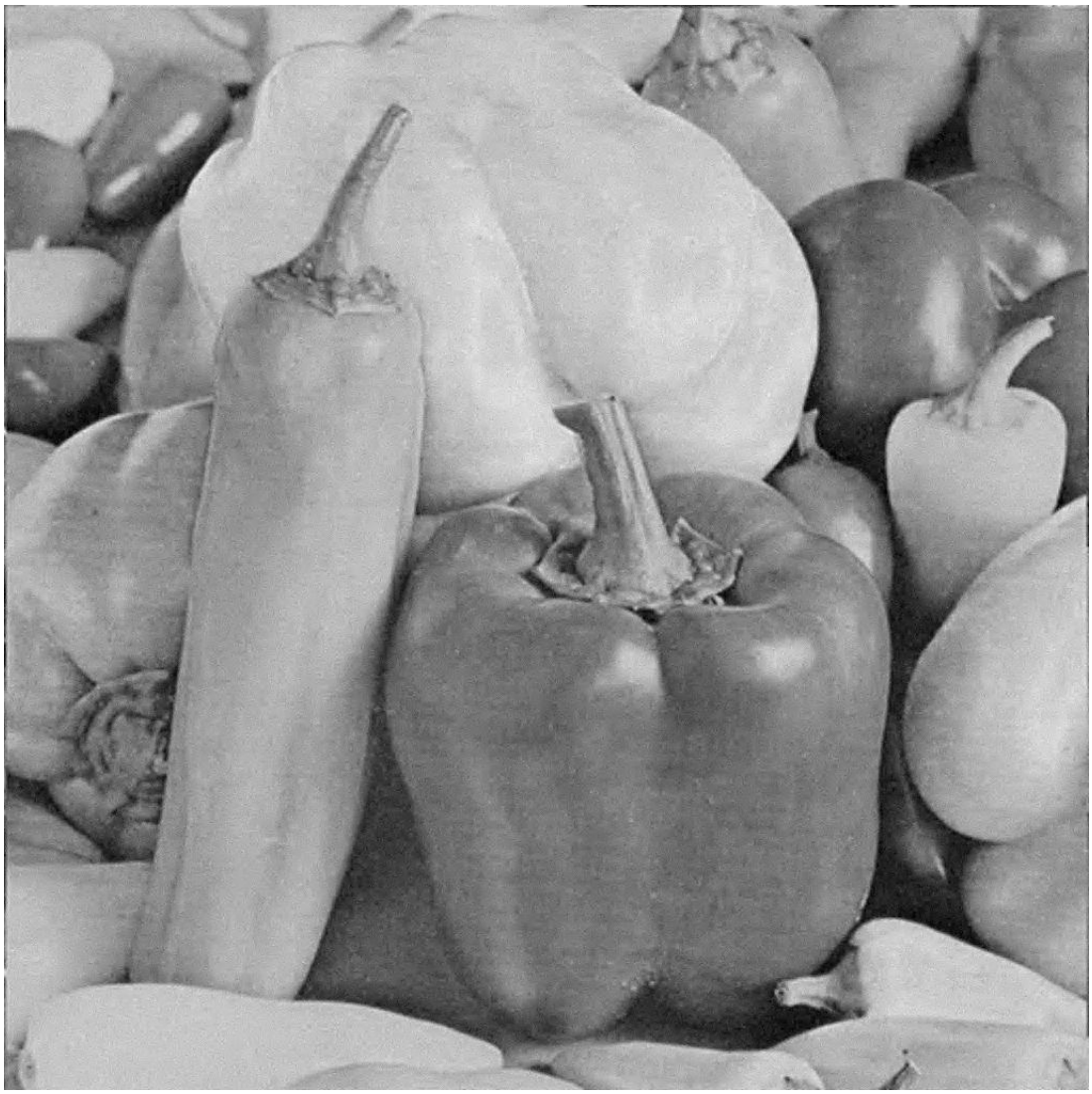
We also used a real image of size 1024×1024 and generated compressive samples using a scrambled partial Fourier sensing matrix with $M = 0.33 \times N$. For sparsity basis, we picked Daubechies db8 wavelets and judiciously chose $k = 0.15 \times N$ for sparse recovery. To recover the target image, we then ran NIIHT for 2000 iterations. Figure 13.7 illustrates the reconstructed image, as well as the difference between the reconstruction image and the true image. Here the NIHT algorithm has recovery SNR 19.81, on par with the Basis Pursuit reconstruction SNR.

(a) Original 128×128 phantom image.

(b) BP: (SNR: 12.86dB.

(c) NIHT: (SNR: 15.05dB.

Figure 13.6: Recovery of the 128×128 phantom image using the NIHT and the Basis Pursuit algorithms. The reconstruction SNR of the NIHT algorithm is about 2dB higher than that of the Basis Pursuit algorithm.



(a) NIHT (SNR: 19.81dB).

Figure 13.7: Compressive imaging with NIHT algorithm.

Part VI

Conclusion

Chapter 14

Conclusion

In this thesis, we provided two deterministic alternatives to the classical random sensing framework. The first alternative was constructed from the adjacencies of vertex expander graphs, and the second alternative was constructed from the Delsarte-Goethals codes. It has been known for a long time that both (explicit) expander graphs, and error-correcting codes, are extremely powerful pseudo-random objects. This pseudo-randomness was our first motivation for suggesting the deterministic sensing frameworks of this thesis.

Nevertheless, in contrast to truly random matrices, one can prove that neither the expander based nor the Delsarte-Goethals based sensing matrices satisfy the Restricted Isometry Property. Whereas almost every result in the classical random sensing framework relies on the RIP. Therefore, this makes the use of our suggested deterministic frameworks unintuitive. Hence, in order to show the strength of our deterministic sensing frameworks, we introduced verifiable conditions that are satisfied by our deterministic matrices, and are sufficient to guarantee successful sparse recovery. We also showed that by exploiting the *structure* of our design matrices, one can propose efficient reconstruction algorithms with similar, or in cases even better, performance than the ℓ_1 reconstruction methods.

The expander based compressive sensing framework provides efficiency in storage, computation, explicit construction, and resilience against Poisson noise. If the noise level is not too high, reconstruction is possible via a simple message-passing algorithm which requires at most $2k$ simple messages. If the noise level is high, stable reconstruction is possible using an iterative algorithm that is obtained from a game theoretic interpretation of the expander based sparse reconstruction problem.

The Delsarte-Goethals based compressed sensing framework relies on the coherence between the rows and the columns of the matrix. In this thesis we showed that these coherence properties are sufficient to guarantee successful model-selection and sparse approximation in the average-case compressed sensing framework. The Delsarte-Goethals frames have optimal coherence values, and therefore provide optimal average-

case model selection guarantees. The average-case CS is a reasonable model for applications such as multi-user detection. Our experiments suggested that in the DG sensing framework, it is possible to detect 25 active users from the set of all 2^{32} (32-bit) users by taking only 2^{12} measurements. Moreover, the reconstruction takes less than one minute. In contrast, it takes several weeks to recover the same active users in the random sensing framework.

One interesting future direction is to apply the deterministic compressed sensing framework to new applications. In applications such as speech or video recognition, or quantum computing, the sensing matrix is often provided to us by the nature. In those cases the matrix hardly satisfied the Restricted Isometry Property. However, there is a much higher chance that the matrix satisfies (some version of) the coherence property required in the deterministic sensing framework. Adopting the sparse reconstruction algorithms of this thesis to those applications is a challenging, and also extremely interesting, future plan.

Bibliography

[1] J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *the 21st Annual Conference on Learning Theory (COLT)*, pages 263–274, 2008.

[2] V. Aggarwal, L. Applebaum, A. Bennatan, A. R. Calderbank, S. D. Howard, and S. J. Searle. Enhanced CDMA communications using compressed-sensing reconstruction methods. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, pages 1211–1215, 2009.

[3] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[4] M. Akcakaya, J. Park, and V. Tarokh. Compressive sensing using low density frames, 2011. available at http://arxiv.org/abs/0903.0650.

[5] M. Akcakaya and V. Tarokh. A frame construction and a universal distortion bound for sparse representations. *IEEE Transactions on Signal Processing*, 56(6):2443–2450, 2008.

[6] W. Alltop. Complex sequences with low periodic correlations. *IEEE Transactions on Information Theory*, 26(3):350–354, 1980.

[7] N. Alon, O. Goldreich, J. Hastad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–303, 1992.

[8] N. Alon, Y. Matias, and M. Szegedy. The Space Complexity of Approximating the Frequency Moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[9] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, New York, 1992.

[10] L. Applebaum, W.U. Bajwa, M.F. Duarte, and A. R. Calderbank. Multiuser detection in asynchronous onoff random access channels using LASSO. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing*, pages 130–137, 2010.

[11] L. Applebaum, S. D. Howard, S. J. Searle, and A. R. Calderbank. Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. *Applied and Computational Harmonic Analysis*, 26(2):283–290, 2009.

[12] K. Azuma. Weighted sums of certain dependent random variables. *Tohuoku Mathematical Journal*, 19:357–367, 1967.

[13] K. D. Ba, P. Indyk, E. Price, and D. P. Woodruff. Lower bounds for sparse recovery. In *Symposium on Discrete Algorithms (SODA)*, pages 1190–1197, 2010.

[14] W. Bajwa, R. Calderbank, and S. Jafarpour. Model selection: Two fundamental measures of coherence and their algorithmic significance. In *proceedings of the International Symposium on Information Theory (ISIT)*, 2010.

[15] W. Bajwa, J. Haupt, G. Raz, S. Wright, and R. Nowak. Toeplitz-structured compressed sensing matrices. In *Proceedings of the 14th IEEE/SP Workshop on Statistical Signal Processing (SSP)*, 2007.

[16] W. U. Bajwa, R. Calderbank, and S. Jafarpour. Revisiting model selection and recovery of sparse signals using one-step thresholding. In *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, 2010.

[17] W. U. Bajwa, R. Calderbank, and S. Jafarpour. Why Gabor frames? Two fundamental measures of coherence and their role in model selection. *Journal of Communications and Networks*, 12(4):289–307, 2010.

[18] W. U. Bajwa, A. M. Sayeed, and R. Nowak. Learning sparse doubly selective channels. In *Proceedings of the 45th Annual Allerton Conference on Communication, Control, and Computing*, pages 575–582, 2008.

[19] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[20] R. G. Baraniuk, V. Cevher, and M. B. Wakin. Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of The IEEE*, 98:959–971, 2010.

[21] R. G. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[22] R.G. Baraniuk, M.A. Davenport, M.F. Duarte, and C. Hegde. An introduction to compressive sensing. In *Connexions e-textbook*, 2011.

[23] D. Baron, S. Sarvoham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Transactions on Information Thoery*, 58(1):269 – 280, 2010.

[24] L. Bassalygo and M. Pinsker. Complexity of an Optimum Nonblocking Switching Network without Reconnections. *Problem in Information Transmission*, 9(1):289–313, 1973.

[25] M. Bayati and A. Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57:765–785, 2010.

[26] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

[27] S. Becker, J. Bobin, and E. J. Candès. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2009.

[28] R. Berinde. Advances in Sparse Signal Recovery Methods. Master's thesis, MIT Department of Electrical Engineering and Computer Science, August 2009.

[29] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: a unified approach to sparse signal recovery. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 798–805, 2008.

[30] R. Berinde and P. Indyk. Sequential sparse matching pursuit. In *47th annual Allerton conference on Communication, control, and computing*, pages 36–43, 2009.

[31] R. Berinde, P. Indyk, and M. Ružić. Practical Near-optimal Sparse Recovery in the $\ell 1$ Norm. In *46th Annual Allerton Conference on Communication, Control, and Computing*, pages 198–205, 2008.

[32] J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Chichester-Wiley, 1994.

[33] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 1992.

[34] C. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, New York, 2006.

[35] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal on Fourier Analysis Applications. Special Issues on Sparsity*, 14(5):629–654, 2008.

[36] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[37] Léon Bottou. Stochastic learning. In Olivier Bousquet and Ulrike von Luxburg, editors, *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004.

[38] S. Boucheron, O. Bousquet, and G. Lugosi. Theory of classification : A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[39] J. Bourgain, S. J. Dilworth, K. Ford, S. Konyagin, and D. Kutzarova. Breaking the k^2 barrier for explicit RIP matrices. In *Proceedings of the* 43^{rd} *Symposium on Theory of Computing (STOC)*, 2011.

[40] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967.

[41] A. R. Calderbank. Reed-Muller Codes and Symplectic Geometry. In *Recent trends in Coding Theory and its Applications,* in the series *AMS / IP Studies in Advanced Mathematics, American Mathematical Society*, 2006.

[42] A. R. Calderbank, P.J. Cameron, W.M. Kantor, and J.J. Seidel. $\mathbb{Z}_4$-Kerdock Codes, orthogonal spreads and extremal euclidean line sets. In *proceedings of London Math. Society*, pages 436–480, 1997.

[43] A.R. Calderbank and G. McGuire. $\mathbb{Z}_4$–linear codes obtained as projections of Kerdock and Delsarte-Goethals Codes. *Linear Algebra and its Applications*, 226-228:647–665, 1995.

[44] R. Calderbank, S. Howard, and S. Jafarpour. Construction of a Large Class of Deterministic Sensing Matrices That Satisfy a Statistical Isometry Property. *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Compressed Sensing*, 4(2):358 – 374, 2010.

[45] R. Calderbank, S. Howard, and S. Jafarpour. Sparse reconstruction via the Reed-Muller sieve. In *proceedings of the International Symposium on Information Theory (ISIT)*, 2010.

[46] R. Calderbank, S. Howard, S. Jafarpour, and J. Kent. Sparse Approximation and Compressed Sensing Using the Reed-Muller Sieve, 2010. Technical Report, Princeton University, Department of Computer Science, TR-888-10.

[47] R. Calderbank and S. Jafarpour. Reed Muller Sensing Matrices and the LASSO. In *International Conference on Sequences and their Applications (SETA)*, page 442463, 2010.

[48] E. J. Candès. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*, volume 3, pages 1433–145, 2006.

[49] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589–592, 2008.

[50] E. J. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37(5):2145–2177, 2009.

[51] E. J. Candès and J. K. Romberg. Quantitative robust uncertainty principles and optimally sparse decompositions. *Foundations of Computational Mathematics*, 6:227–254, 2004.

[52] E. J. Candès and J. K. Romberg. ℓ_1-magic: Recovery of sparse signals via convex programming, 2005. http://www.acm.caltech.edu/l1magic.

[53] E. J. Candès and J. K. Romberg. Practical signal recovery from random projections. In *SPIN Conference on Wavelet Applications in Signal and Image Processing*, 2008.

[54] E. J. Candès, J. K. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489–509, 2006.

[55] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223., 2006.

[56] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

[57] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52:5406–5425, 2006.

[58] E. J. Candès and T. Tao. Rejoinder: the Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35:2392–2404, 2007.

[59] E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal on Fourier Analysis Applications*, 14:877–905, 2007.

[60] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[61] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness Conductors and Constant degree Expansions beyond the degree 2 Barrier. In *34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 659–668, 2002.

[62] A. Carmy, P. Gurfil, D. Kanevsky, and B. Ramabhadran. ABCS: Approximate Bayesian Compressed Sensing, 2009. Technical Report, Human Language Technologies, IBM.

[63] R. E. Carrillo, K. E. Barner, and T. C. Aysal. Robust sampling and reconstruction methods for sparse signals in the presence of impulsive noise. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):392–408, 2010.

[64] P. G. Casazza and M. Fickus. Fourier Transforms of Finite Chirps. *Eurasip Journal on Advances in Signal Processing*, pages 1–8, 2006.

[65] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, 1997.

[66] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[67] V. Cevher. Learning with compressible priors. In *Twenty-Second Annual Conference on Neural Information Processing Systems (NIPS)*, 2008.

[68] V. Cevher. An ALPS view of sparse recovery. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[69] V. Cevher and S. Jafarpour. Fast Hard Thresholding with Nesterov's Gradient Method. In *NIPS Workshop on Practical Applications of Sparse Modeling*, 2010.

[70] V. Chandar. A negative result concerning explicit matrices with the Restricted Isometry Property, 2008. preprint.

[71] V. Chandar, D. Shah, and G. Wornell. A simple message-passing algorithm for compressed sensing. In *IEEE International Symposium on Information Theory (ISIT)*, 2010.

[72] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *International Congress of Mathematicans*, pages 693–703, 2002.

[73] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[74] M. Cheraghchi. *Applications of Derandomization Theory in Coding*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2010.

[75] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *Journal of American Mathematical Society*, 22:211–231, 2009.

[76] M. Collins, R. E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1/2/3), 2002.

[77] P. L. Combettes and J. C. Pesquet. A DouglasRachford Splitting Approach to Nonsmooth Convex Variational Signal Recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1:564–574, 2007.

[78] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55:58–75, 2005.

[79] G. Cormode and S. Muthukrishnan. Combinatorial algorithms for compressed sensing. In *Colloquium on Structural Information and Communication Complexity*, pages 280–294, 2006.

[80] T. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–19, 1991.

[81] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based statistical signal processing using Hidden Markov Models. *IEEE Transactions on Signal Processing*, 46(4):886–902, 1998.

[82] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009.

[83] W. Dai, O. Milenkovic, M. A. Sheikh, and R. G. Baraniuk. Probe Design for Compressive Sensing DNA Microarrays. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 163–169, 2008.

[84] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, 1992.

[85] I. Daubechies, M. Defrise, and C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 75(11):1412–145, 2004.

[86] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. Introduction to compressed sensing. In *Compressed Sensing: Theory and Applications*, chapter 1. Cambridge University Press, 2011.

[87] M. R. de Prony. Essai experimentalle et analytique. *J. Ecole Polytech.Paris*, 1:24–76, 1795.

[88] R. A. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4-6):918–925, 2007.

[89] T. T. Do, L. Gan, N. Nguyen, and T. D. Tran. Sparsity adaptive matching pursuit algorithm for practical compressed sensing. In *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, 2008.

[90] D. Donoho. For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829, 2006.

[91] D. Donoho, A. Maleki, and A. Montanari. Message passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106:18914–18919, 2009.

[92] D. Donoho, A. Maleki, and A. Montanari. Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, 2009.

[93] D. Donoho, A. Maleki, and A. Montanari. Message Passing Algorithms for Compressed Sensing: II. Analysis and Validation. In *Proceedings of the IEEE Information Theory Workshop (ITW)*, 2009.

[94] D. Donoho, A. Maleki, and A. Montanari. The noise sensitivity phase transition in compressed sensing. *to appear in IEEE Transactions on Information Theory*, 2011.

[95] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[96] D. L. Donoho, I. Drori, Y. Tsaig, and J. L. Starck. Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit, 2006. Technical Report, Stanford University, Statistics Department.

[97] D. L. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[98] D. L. Donoho and J. Tanner. Precise Undersampling Theorems. *IEEE Transactions on Information Theory*, 98(6):913–924, 2010.

[99] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.

[100] R. Dorfman. The Detection of Defective Members of Large Populations. *Annals of Mathematical Statistics*, 14(4):436–440, 1943.

[101] D. Z. Du and F. K. Hwang. *Combinatorial group testing and its applications*. World Scientific Publishing Co. Inc., River Edge, NJ, 2010.

[102] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *EEE Signal Processing Magazine*, 25(2):83–91, 2008.

[103] J. M. Duarte-Carvajalino, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–59, 2008.

[104] J. M. Duarte-Carvajalino and G. Sapiro. Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18(7):1395–1408, 2009.

[105] J. C. Dunn. Global and asymptotic convergence rate estimates for a class of projected gradient processes. *SIAM Journal on Control and Optimization*, 19(3):368–400, 1981.

[106] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[107] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky. A wide-angle view at iterated shrinkage algorithms. In *Proceedings of SPIE (Wavelet XII)*, 2007.

[108] C. Estan and G. Varghese. New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems*, 21(3):270–313, 2003.

[109] W. Fang and L. Peterson. Inter-AS traffic patterns and their implications. In *IEEE Global Telecommunications Conference*, 1999.

[110] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. In *ACM SIGCOMM Conference*, pages 257–270, 2000.

[111] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing*, 1(4):586598, 2007.

[112] B. J. Fino and V. R. Algazi. Unified Matrix Treatment of the Fast Walsh-Hadamard Transform. *IEEE Transactions on Computers*, 25:1142–1146, 1976.

[113] A. K. Fletcher, S. Rangan, and V. K. Goyal. Necessary and sufficient conditions for sparsity pattern recovery. *IEEE Transactions on Information Theory*, 55(12):5758–5772, 2009.

[114] M. Fornasier and H. Rauhut. Iterative thresholding algorithms. *Applied and Computational Harmonic Analysis*, 25(2):187–208, 2008.

[115] D. P. Foster and E. I. George. The risk inflation criterion for multiple regression. *Annals of Statistics*, 22(4):1947–1975, 1994.

[116] Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 325–332, 1996.

[117] Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[118] J. J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004.

[119] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *International Conference on Machine Learning (ICML)*, 2009.

[120] C. Genovese, J. Jin, and L. Wasserman. Revisiting marginal regression, 2009. arXiv:0911.4080.

[121] E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339373, 1997.

[122] A. C. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. J. Strauss. Near-optimal sparse fourier representations via sampling. In *Proceedings of the thirty-forth annual ACM Symposium on Theory of Computing (STOC)*, pages 152–161, 2002.

[123] A. C. Gilbert and P. Indyk. Sparse Recovery Using Sparse Matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.

[124] A. C. Gilbert, M. A. Iwen, and M. J. Strauss. Group testing and sparse signal recovery. In *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1059 – 1063, 2008.

[125] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss. Approximate sparse recovery: optimizing time and measurements. In *42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 475–484, 2010.

[126] A. C. Gilbert, M. J. Strauss, J. Tropp, and R. Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing (STOC)*, pages 237–246, 2007.

[127] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, New York, 1996.

[128] E. D. Gluskin. On some finite-dimensional problems in the theory of widths. *Vestnik Leningrad Univ. Math*, 14:163–170, 1982.

[129] E. D. Gluskin. Norms of random matrices and widths of finite-dimensional sets. *Math. USSR Sbornik*, 48:173–182, 1984.

[130] A. A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70:709–710, 1964.

[131] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd edition, 2008.

[132] I. F. Gorodnitsky, J. George, and B.D. Rao. Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm. *Electrocephalography and Clinical Neurophysiology*, 95:231–251, 1995.

[133] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*, pages 95–110. Springer-Verlag Limited, 2008.

[134] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, January 2011.

[135] A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43:173210, 2001.

[136] S. Gurevich and R. Hadani. The statistical restricted isometry property and the Wigner semicircle distribution of incoherent dictionaries, 2009. preprint.

[137] V. Guruswami, J. Lee, and A. Razborov. Almost Euclidean subspaces of ℓ_1 via expander codes. *Combinatorica*, 30(1):47–68, 2010.

[138] V. Guruswami, J. R. Lee, and A. Wigderson. Euclidean sections of with sublinear randomness and error-correction over the reals. In *Approximation Algorithms for Combinatorial Optimization*, pages 444–454, 2008.

[139] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009.

[140] A. R. Hammons, P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Sole. The $\mathbb{Z}_4$-linearity of Kerdock Codes, Preparata, Goethals, and related codes. *IEEE Transactions on Information Theory*, 40(2):301–319, 1994.

[141] Z. Harmany, R. Marcia, and R. Willett. Sparse Poisson Intensity Reconstruction Algorithms. In *IEEE Statistical Signal Processing*, 2009.

[142] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer-Verlag, New York, 2001.

[143] E. Hazan. *Efficient Algorithms for Online Convex Optimization and Their Applications*. PhD thesis, Princeton University, 2006.

[144] E. Hazan. A survey: The convex optimization approach to regret minimization, 2011. Preprint available at http://ie.technion.ac.il/∼ehazan/papers/OCO-survey.pdf.

[145] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[146] M. A. Herman and T. Strohmer. High-resolution radar via compressed sensing. *IEEE Transactions on Signal Processing*, 57(6):2275–2284, 2009.

[147] K. K. Herrity, A. C. Gilbert, and J. A. Tropp. Sparse approximation via iterative thresholding. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.

[148] S. Hoory, N. Linial, and A. Wigderson. Expander Graphs and their Applications. *Bulletin of The American Mathematical Society*, 43:439–562, 2006.

[149] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[150] S. Howard, R. Calderbank, and S. Searle. A fast reconstruction algorithm for deterministic compressive sensing using second order Reed-Muller codes. In *Conference on Information Sciences and Systems (CISS)*, pages pp: 11 – 15, 2008.

[151] D. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

[152] P. Indyk. Explicit constructions for compressed sensing of sparse signals. In *Proceedings of the 19th annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 30–33, 2008.

[153] P. Indyk and M. Ružić. Near-optimal sparse recovery in the $\ell 1$ norm. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 199–207, 2008.

[154] M. A. Iwen. Simple Deterministically Constructible RIP Matrices with Sublinear Fourier Sampling Requirements. In *Conference on Information Sciences and Systems (CISS)*, 2008.

[155] L. Jacques, D. K. Hammond, and M. J. Fadili. Dequantizing compressed sensing with non-gaussian constraints. In *International Conference on Image Processing*, 2009.

[156] S. Jafarpour, V. Cevher, and R. E. Schapire. A game theoretic approach to expander-based compressive sensing. In *proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2011.

[157] S. Jafarpour, M. Raginsky, R. Willett, and R. Calderbank. Performance bounds for expander-based compressed sensing in the presence of poisson noise. In *44th Asilomar Conference on Signals, Systems and Computers*, 2010.

[158] S. Jafarpour, R. E. Schapire, and V. Cevher. Compressive sensing meets game theory. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[159] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank. Efficient and Robust Compressed Sensing using Optimized Expander Graphs. *IEEE Transactions on Information Theory*, 55(6):4299–4308, 2009.

[160] S. Ji, Y. Xue, and L. Carin. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, 2008.

[161] I. Johnstone. Chi-square oracle inequalities. *State of the Art in Probability and Statistic*, 37:399–418, 2001.

[162] M. Johnstone and A. Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682693, 2009.

[163] H. Jung, J. C. Ye, and E. Y. Kim. Improved k-t BLASK and k-t SENSE using FOCUSS. *Phys. Med. Biol*, 52:32013226, 2007.

[164] R. M. Kainkaryam, A. Bruex, A. C. Gilbert, J. Schiefelbein, and P. J. Woolf. poolMC: smart pooling of mRNA samples in microarray experiments. *BMC bioinformatics*, 11(1), 2010.

[165] D. Kanevsky, T. N. Sainath, B. Ramabhadran, and D. Nahamoo. An analysis of sparseness and regularization in exemplar-based methods for speech classification. In *11th Annual Conference of the International Speech Communication Association*, pages 2842–2845, 2010.

[166] B. Kashin. The widths of certain finite dimensional sets and classes of smooth functions. *Izvestia*, 41:334–35, 1977.

[167] S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice Hall, Upper Saddle River, NJ, 1998.

[168] A. Khajehnejad, W. Xu, S. Avestimehr, and B. Hassibi. Improved sparse recovery thresholds with two step reweighted ℓ_1 minimization. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, 2010.

[169] M. A. Khajehnejad, A. G. Dimakis, W. Xu, and B. Hassibi. Sparse recovery of nonnegative signals with minimal expansion. *IEEE Transactions on Signal Processing*, 59(1):196–208, 2011.

[170] K. I. Kim, K. Jung, and H. J. Kim. Face recognition using kernel principal component analysis. *IEEE Signal Processing Letters*, 9(2):40–42, 2002.

[171] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale ℓ_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(6):606–617, 2007.

[172] J. Kivinen and M. K. Warmuth. Relative loss bounds for multi-dimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

[173] T. H. Kjeldsen. John von Neumann's Conception of the Minimax Theorem: A Journey through Different Mathematical Contexts. *Archive for History of Exact Sciences*, 56:39–68, 2001.

[174] V. Kostina, M. Duarte, S. Jafarpour, and R. Calderbank. The value of redundant measurement in compressed sensing. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[175] E. Kushilevitz and Y. Mansour. Learning decision trees using the fourier sprectrum. In *Proceedings of the twenty-third annual ACM Symposium on Theory of Computing (STOC)*, pages 152–161, 1991.

[176] J. D. Lafferty, S. Della Pietra, and V. Della Pietra. Statistical learning algorithms based on Bregman distances. In *Proceedings of the Canadian Workshop on Information Theory*, 1997.

[177] J. N. Laska, M. A. Davenport, and R. G. Baraniuk. Exact signal recovery from sparsely corrupted measurements through the pursuit of justice. In *43rd Asilomar Conference on Signals, Systems and Computers*, 2009.

[178] J. Lawrence, G. E. Pfander, and D. Walnut. Gabor systems in finite dimensional vector spaces. *Journal of Fourier Analysis Applications*, 11(6):715–726, 2005.

[179] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Computational Mathematics and Mathematical Physics*, 6:1–15, 1966.

[180] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani. Counter Braids: a novel counter architecture for per-flow measurement. In *Measurement and Modeling of Computer Systems*, pages 121–132, 2008.

[181] M. Lustig, D. Donoho, and J. M. Pauly. Sparse MRI: The Application of Compressed Sensing for rapid MR Imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

[182] D. J. C. Mackay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, 2002.

[183] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. Elsevier Science Publishers, Amsterdam, 1996.

[184] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2272 – 2279, 2009.

[185] A. Maleki and D. L. Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4:330–341, 2010.

[186] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, pages 3397–3415, 1993.

[187] C. L. Mallows. Some Comments on CP. *Technometrics*, 15(1):661–675, 1973.

[188] P. Massart. A non-asymptotic theory for model selection. In *European Congress of Mathematics*, pages 309–323, 2005.

[189] P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman and Hall, London, 2nd edition, 1989.

[190] C. McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, pages 148–188, 1989. Cambridge University Press.

[191] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[192] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(4):1436–1462, 2006.

[193] A. J. Miller. *Subset selection in regression*. Chapman-Hall, New York, 1990.

[194] A. Montanari. Graphical models concepts in compressed sensing. In *Compressed Sensing: Theory and Applications*, chapter 9. Cambridge University Press, 2011.

[195] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, 1995.

[196] S. Muthukrishnan. Data Streams: Algorithms and Applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005.

[197] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal of Computing*, 24(2):227–234, 1995.

[198] R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, New York, 1996.

[199] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[200] D. Needell and R. Vershyni. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9:317–334, 2009.

[201] J. L. Nelson and V. N. Temlyakov. On the size of incoherent systems. *to appear in Journal of Approximation Theory*, 2011.

[202] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York, 1983.

[203] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004.

[204] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[205] Y. E. Nesterov and A. S. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[206] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Annalen*, 100:295–320, 1928.

[207] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, 2007.

[208] F. Parvaresh and A. Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005.

[209] D. Paul. Asymptotics of sample eigenstructure for a large-dimensional spiked covariance model. *Statistica Sinica*, 17:16171642, 2007.

[210] M. Raginsky, Z. Harmany, R. Marcia, and R. Willett. Compressed sensing performance bounds under Poisson noise. *IEEE Transactions on Signal Processing*, 58:3990–4002, 2010.

[211] M. Raginsky, S. Jafarpour, Z. Harmany, R. Marcia, R. Willett, and R. Calderbank. Performance bounds for expander-based compressed sensing in poisson noise. *to appear in IEEE Transactions on Signal Processing*, 2011.

[212] M. Raginsky, S. Jafarpour, R. Willett, and R. Calderbank. Fishing in Poisson streams: focusing on the whales, ignoring the minnows. In *Conference on Information Sciences and Systems (CISS)*, 2010.

[213] P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*, 38:1287–1319, 2010.

[214] G. Reeves and M. Gastpar. A note on optimal support recovery in compressed sensing. In *Proceedings of the 43rd Asilomar Conference on Signals, Systems and Computers*, 2009.

[215] I. Rish and G. Grabarnik. Sparse signal recovery with exponential-family noise. In *Allerton Conference on Communication, Control, and Computing*, 2009.

[216] J. K. Romberg. Imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):14–20, 2008.

[217] M. Rudelson and R. Vershynin. Geometric approach to error correcting codes and reconstruction of signals. *International Mathematical Research Notices*, 64:4019–4041, 2005.

[218] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[219] M. Rudelson and R. Vershynin. Non-asymptotic theory of random matrices: Extreme singular values. In *Proceedings of the International Congress of Mathematicians*, 2010.

[220] T. N. Sainath, B. Ramabhadran, D. Nahamoo, D. Kanevsky, and A. Sethy. Sparse representation features for speech recognition. In *11th Annual Conference of the International Speech Communication Association*, pages 2254–2257, 2010.

[221] V. Saligrama and M. Zhao. Thresholded Basis Pursuit: An LP Algorithm for Achieving Optimal Support Recovery for Sparse and Approximately Sparse Signals from Noisy Random Measurements, 2008. arXiv/0809.4883.

[222] S. Sarvotham, D. Baron, and R. G. Baraniuk. Sudocodes - Fast measurement and reconstruction of sparse signals. In *IEEE International Symposium on Information Theory (ISIT)*, 2006.

[223] K. Schnass and P. Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, 2007.

[224] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[225] A. Sethy, T. N. Sainath, B. Ramabhadran, and D. Kanevsky. Data selection for language modeling using sparse representations. In *11th Annual Conference of the International Speech Communication Association*, pages 2258–2261, 2010.

[226] A. Shabani, R. L. Kosut, M. Mohseni, H. Rabitz, M. A. Broome, M. P. Almeida, A. Fedrizzi, and A. G. White. Efficient measurement of quantum dynamics via compressive sensing. *Phys. Rev. Lett.*, 106(10):100401, 2011.

[227] A. Shokrollahi. LDPC codes: An introduction. *Digital Fountain, Inc., Technical Report*, 2003.

[228] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42:1710–1722, 1996.

[229] D. Snyder, A. Hammond, and R. White. Image recovery from data acquired with a charge-coupled-device camera. *Journal of The Optical Society of America A-optics Image Science and Vision*, 10:1014–1023, 1993.

[230] T. Strohmer and R. Heath. Grassmanian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003.

[231] M. Sudan. Decoding of Reed Solomon codes beyond the error- correction bound. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 164–172, 1996.

[232] D. Takhar, J. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. Kelly, and R. G. Baraniuk. A new compressive imaging camera architecture using optical-domain compression. In *Computational Imaging IV at SPIE Electronic Imaging*, 2006.

[233] R. Tibshirani. Regression shrinkage and selection via the LASSO. *J. Royal. Statist. Soc*, 5(1):267–288, 1996.

[234] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[235] S. K. Tjoa, M. C. Stamm, W. S. Lin, and K. J. R. Liu. Classification and clustering via dictionary learning with structured incoherence and shared features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501 – 3508, 2010.

[236] S. K. Tjoa, M. C. Stamm, W. S. Lin, and K. J. R. Liu. Harmonic variable-size dictionary learning for music source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 413–416, 2010.

[237] J. Tropp. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[238] J. Tropp. Norms of Random Submatrices and Sparse Approximation. *C. R. Acad. Sci. Paris*, 346(1):1271–1274, 2008.

[239] J. Tropp. On The Conditioning of Random Subdictionaries. *Applied and Computational Harmonic Analysis*, 25:1–24, 2008.

[240] J. Tropp. The Sparsity Gap: Uncertainty Principles Proportional to Dimension. In *proceedings of the 44th Annual IEEE Conference on Information Sciences and Systems (CISS)*, 2010.

[241] J. Tropp and A. C. Gilbert. Signal recovery from random measurements via Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[242] J. Tropp and S. Wright. Computational methods for sparse solution of linear inverse problems, 2009. Technical Report No. 2009-01, California Institute of Technology.

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

[244] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[245] S. Verdu. *Multiuser Detection*. Cambridge University Press, New York, 1998.

[246] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Yi Ma. Towards a Practical Face Recognition System: Robust Registration and Illumination via Sparse Representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[247] M. J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

[248] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers, Hanover, MA, 2008.

[249] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2003.

[250] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.

[251] L. R. Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

[252] R. Willett and M. Raginsky. Performance bounds on compressed sensing with Poisson noise. In *IEEE International Symposium on Information Theory (ISIT)*, pages 174–178, 2009.

[253] J. Wolf. Decoding of Bose-Chaudhuri-Hocquenhem Codes and Prony's method for curve fitting. *IEEE Transactions on Information Theory*, 13(4):608, 1967.

[254] J. Wright and Y. Ma. Dense Error Correction via L1-minimization. *IEEE Transactions on Information Theory*, 56(7):3540 – 3560, 2010.

[255] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and Yan. Shuicheng. Sparse Representation for Computer Vision and Pattern Recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[256] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009.

[257] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[258] W. Xu and B. Hassibi. Efficient Compressive Sensing with Deterministic Guarantees using Expander Graphs. In *IEEE Information Theory Workshop (ITW)*, 2007.

[259] W. Xu and B. Hassibi. Further Results on Performance Analysis for Compressive Sensing Using Expander Graphs. In *41st Asilomar Conference on Signals, Systems and Computers*, pages 621–625, 2007.

[260] I. Zeljkovic, P. Haffner, B. Amento, and J. Wilpon. GMM/SVM N-best speaker identification under mismatch channel conditions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[261] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

[262] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (COLT)*, 2003.