High-Dimensional Data Analysis with Low-Dimensional Models:

Principles, Computation, and Applications

JOHN WRIGHT (Columbia University)
YI MA (University of California, Berkeley)

To Mary, Isabella, and Mingshu (J.W.)

To Henry, Barron, and Diana,
and in memory of my father (Y.M.)

Foreword

I recall a moment, perhaps ten or fifteen years ago, of prodigious scientific activity. To give our reader a sense of this blessed time, consider a series of regular scientific workshops, each involving at most forty participants. Despite the small size and almost intimate nature of these workshops, they brought together an energized and enthusiastic mix of people from an array of disciplines, including mathematics, computer science, engineering, and the life sciences. What a privilege to be in a room with mathematicians such as Terence Tao and Roman Vershynin and learn about high-dimensional geometry; with applied mathematicians and engineers such as David Donoho, Joel Tropp, Thomas Ströhmer, Michael Elad, and Freddy Bruckstein and learn about the power of algorithms; with statistical physicists such as Andrea Montanari and learn about phase transitions in large stochastic systems. What a privilege to learn about fast numerical methods for large-scale optimization from computer scientists such as Stephen Wright and Stanley Osher. What a privilege to learn about compressive optical systems from David Brady and Richard Baraniuk and Kevin Kelly (of single-pixel camera fame); about compressive analog-to-digital conversion and wideband spectrum sensing from Dennis Healy, Yonina Eldar, and Azita Emami Neyestanak; about breakthroughs in computer vision from Yi Ma, John Wright, and René Vidal; and about dramatically faster scan times in magnetic resonance imaging from Michael Lustig and Leon Axel. Bringing all these people—and others I regretfully cannot name for lack of space—together, with their different perspectives and interests, sparked spirited discussions. Excitement was in the air and progress quickly followed.

Yi Ma and John Wright were frequent participants to these workshops and their book magically captures their spirit and richness. It exposes readers to (1) a variety of real-world applications including medical and scientific imaging, computer vision, wideband spectrum sensing, and so on, (2) the mathematical ideas powering algorithms in use in these fields, and (3) the algorithmic ideas needed to implement them. Let me illustrate with an example. On the one hand, this is a book in which we learn about the principles of magnetic resonance (MR) imaging. There is a chapter in which we learn how an MR scan excites the nucleus of atoms by means of a magnetic field. These nuclei have a magnetic spin, and will respond to this excitation, and it is precisely this response that gets recorded. As for other imaging modalities, such as computed tomography,

there is a mathematical transformation, which relates the object we wish to infer and the data we collect. In this case, after performing a few approximations, this mathematical transformation is given by the Fourier transform. On the other hand, this is a book in which we learn that most of the mass of a high-dimensional sphere is concentrated not just around the equator—this is already sufficiently surprising—but around any equator! Or that the intersection between two identical high-dimensional cubes, one being randomly oriented vis-à-vis the other, is essentially a sphere! These are fascinating subjects, but what is the connection? There is one, of course, and explaining it is the most wonderful strength of the book. In a nutshell, ideas and tools from probability theory, high-dimensional geometry, and convex analysis inform concrete applied problems and explain why algorithms actually work. Returning to our MR imaging problem, we learn how to leverage mathematical models of sparsity to recover exquisite images of body tissues from what appear to be far too few data points. Such a feat allows us to scan patients ten times faster today.

Through three fairly distinct parts — roughly: theory, computations, and applications — the book proposes a scientific vision concerned by the development of insightful mathematics to create models for data, to create processing algorithms, and to ultimately inspire real concrete improvements; for instance, in human health as in the example above.

The first part of the book explores data models around two main themes, namely, sparsity, and low-rankedness. Sparsity expresses the idea that most of the entries of an n-dimensional signal vanish or nearly vanish so that the information can be effectively summarized using fewer than n data bits. Low-rankedness expresses the idea that the columns of a data matrix 'live' near a linear subspace of lower dimension, thereby also suggesting the possibility of an effective summary. We then find out how to use these data models to create data processing algorithms, for instance, to find solutions of underdetermined systems of linear equations. The emphasis is on algorithms formulated as solutions to well-formulated convex optimization problems. That said, we are also introduced to nonconvex methods in Chapter 7 to learn effective empirical representations from data in which signals exhibit enhanced sparsity. All along, the authors use their rich experiences to communicate insights and to explain why some things work while others do not.

The second part reviews effective methods for solving optimization problems—convex or not—at scale; that is, involving possibly millions of decision variables and a possibly equally large number of constraints. This is an area that has seen tremendous progress in the last fifteen years and the book provides readers with a valuable point of entry to the key ideas and vast literature.

The last part is a deep dive into applications. In addition to the imaging challenges I already mentioned, we find a chapter on wireless radio communication, where we see how ideas from sparse signal processing and compressed sensing allow cognitive radios to efficiently identify the available spectrum. We also find three chapters on crucial problems in computer vision, a field in which

the authors have brought and developed formidable tools, enabling major advances and opening new perspectives. Exposition starts with a special contribution, which also exploits ideas from compressed sensing, to the crucial problem of face recognition in the presence of occlusions and other nonidealities. (I recall an exciting *Wired* article about this work when it came out.) The book then introduces methods for inferring 3D structure from a series of 2D photographs, and to identify structured textures from a single photograph; solving the latter problem is often the starting point to recover the appearance, pose, and shape of multiple objects in a scene. Finally, at the time of this writing, deep learning (DL) is all the rage. The book contains an epilogue which establishes connections between all the better understood data models reviewed in the book and DL: the one hundred million dollar question is whether they will shed significant insights on deep learning and influence or improve its practice.

Who would enjoy this book? First and foremost, students in mathematics, applied mathematics, statistics, computer science, electrical engineering, and related disciplines. Students will learn a lot from reading this book because it is so much more than a text about a tool being applied with minor variations. They will learn about mathematical reasoning, they will learn about data models and about connecting those to reality, and they will learn about algorithms. The book also contains computer scripts so that we can see ideas in action and carefully crafted exercises making it perfect for upper-level undergraduate or graduate-level instruction. The breadth and depth makes this a reference for anyone interested in the mathematical foundations of data science. I also believe that members of the applied mathematical sciences community at large would enjoy this book. They will be reminded of the power of mathematical reasoning and of the all-around positive impact it can have.

Emmanuel Candès
Stanford, California
December 2020

Preface

"*The coming century is surely the century of data. A combination of blind faith and serious purpose makes our society invest massively in the collection and processing of data of all kinds, on scales unimaginable until recently.*"

– David Donoho, *High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality, 2000*

The Era of Big Data.

In the past two decades, our world has entered the age of "Big Data." The information technology industry is now facing the challenge, and opportunity, of processing and analyzing massive amounts of data on a daily basis. The size and the dimension of the data have reached an unprecedented scale and are still increasing at an unprecedented rate.

For instance, on the technological side, the resolution of consumer digital cameras has increased nearly ten-fold in the past decade or so. Each day, over 300 million photos are uploaded to Facebook;[1] 300 hours of videos are posted on Youtube every minute; and over 20 million entertaining short videos are produced and posted to Douyin (also known as TikTok) of China.

On the business side, on a single busy day, Alibaba.com needs to take in over 800 million purchase orders for over 15 million products, handle over a billion payments, and deliver more than 30 million packages. Amazon.com also operates at a similar scale, if not even larger. Those numbers are still growing and growing fast!

On the scientific front, super-resolution microscopy imaging technologies have undergone tremendous advances in the past decades,[2] and some are now capable of producing massive quantities of images with subatomic resolution. High-throughput gene sequencing technologies are capable of sequencing hundreds of millions of DNA molecule fragments at a time,[3] and can sequence in just a few

[1] Almost all of them are passing through several processing pipelines for face detection, face recognition, and general object classification for content screening, etc.

[2] For example, in 2014, Eric Betzig, Stefan W. Hell, and William E. Moerner were awarded the Nobel Prize in Chemistry for the development of super-resolution fluorescence microscopy that bypasses the limit of 0.2 micrometers of traditional optical microscopy.

[3] In 2002, Sydney Brenner, John Sulston, and Robert Horvitz were awarded the Nobel Prize for their pioneering work and contributions to the Human Genome project.

Figure 0.1 Images of Mary & Isabella: the resolution of the image on the left is $2,500 \times 2,500$, whereas the image on the right is down-sampled to 250×250, with only 1/100th fraction of pixels of the original one.

hours an entire human genome that has a length of over 3 billion base pairs and contains 20,000 protein-encoding genes!

Paradigm Shift in Information Acquisition, Processing, and Analysis. In the past, scientists or engineers have sought to carefully control the data acquisition apparatus and process. Since the apparatus was expensive and the process time-consuming, typically only necessary data (or measurements) were collected for a specific given task. The data or signals collected were mostly informative for the task and did not contain much redundant or irrelevant information, except for some uncontrollable noise. Hence, classical signal processing or data analysis typically operated under the premise that

$$\text{Classical Premise:} \quad \textbf{Data} \approx \textbf{Information},$$

and in this classical paradigm, practitioners mostly needed to deal with problems such as removing noise or compressing the data for storage or transport.

As mentioned above, technologies such as the Internet, smart phones, high-throughput imaging, and gene sequencing have fundamentally changed the nature of data acquisition and analysis. We are moving from a "data-poor" era to a "data-rich" era. As pointed out by Jim Gray (a Turing Award winner), "increasingly, scientific breakthroughs will be powered by advanced computing capabilities that help researchers manipulate and explore massive datasets." This is now heralded as *the Fourth Paradigm* of scientific discovery [HTT09].

Nevertheless, data-rich does not necessarily imply "information-rich," at least not for free. Massive amounts of data are being collected, sometimes without any specific purpose in advance. Scientists or engineers often do not have direct control of the data acquisition process anymore, neither in the quantity nor the quality of the acquired data. Therefore, any given new task could be inundated with massive amounts of irrelevant or redundant data.

To see intuitively why this is the case, let us first consider the problem of *face recognition*. Figure 0.1 shows two images of two sisters. It is arguably the case

Figure 0.2 Detecting and recognizing faces in a large group photo, from the BIRS workshop on "*Applied Harmonic Analysis, Massive Data Sets, Machine Learning, and Signal Processing*," held at Casa Matemática Oaxaca (CMO) in Mexico, 2016.

that to human eyes, both images convey the identity of the persons equally well, even though pixels of the second image are merely 1/100th of the first one. In other words, if we view both images as vectors with their pixel values as coordinates, then the dimension of the low-resolution image vector is merely 1/100th of the original one. Clearly, the information about the identity of a person relies on statistics of much lower dimension than the original high-resolution image[4]. Hence, in such scenarios, we have a new premise:

$$\text{New Premise I:} \qquad \textbf{Data} \gg \textbf{Information}.$$

For *object detection* tasks such as face detection in images or pedestrian detection in surveillance videos, the issue is no longer with redundancy. Instead, the difficulty is to find any relevant information at all in an ocean of irrelevant data. For example, to detect and recognize familiar people from a group photo shown in Figure 0.2, image pixels associated with human faces only occupy a very tiny portion of the image pixels (10 millions in this case) whereas the mass majority of the pixels belong to completely irrelevant objects in the surroundings. In addition, the subjects of interest, say the two authors, are only two among many human faces. Now imagine scaling this problem to billions of images or millions of videos captured with mobile phones or surveillance cameras. Similar "detection"

[4] In fact, one can continue to argue that even such a low-resolution image is still highly redundant. Studies have shown that humans can recognize familiar faces from images with a resolution as low as around 7×10 pixels [SBOR06]. Recent studies in neuroscience [CT17] reveal that it is possible for the brain to encode and decode any human face using just 200 cells in the inferotemporal (IT) cortex. Modern face recognition algorithms extract merely a few hundred features for reliable face verification.

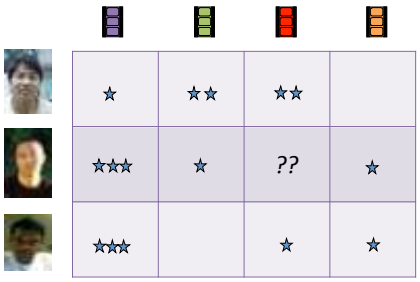


Figure 0.3 An example of collaborative filtering of user preferences: how to guess a customer's rating for a movie even if he or she has not seen it yet?

and "recognition" tasks also arise in studying genetics: out of the nearly 20,000 genes and millions of proteins they encode, scientists need to identify which one (or handful of ones) is responsible for certain genetic diseases. In scenarios like these, we have:

New Premise II: **Data = Information + Irrelevant Data**.

The explosive growth of e-commerce, online shopping, and social networks has created tremendous datasets of user preferences. Major internet companies typically have records of billions of people's preferences, across millions of commercial products, media contents, and more. By nature, such datasets of user preferences, however massive, are far from complete. For instance, in the case of a dataset of movie ratings as shown in Figure 0.3, no one could have seen all the movies and no movie would have been seen by all people. Nevertheless, companies like Netflix need to guess from such incomplete datasets a customer's preferences so that they could send the most relevant recommendations or advertisements to the customer. This problem in information retrieval literature is known as *collaborative filtering*, and most internet companies' business[5] relies on solving problems such as this one effectively and efficiently. The most fundamental reason why complete information can be derived from such a highly-incomplete dataset is that user preferences are not random and the data have structure. For instance, many people have similar tastes in movies and many movies are similar in style. Rows and columns of the user preference table would be strongly correlated, hence the intrinsic dimension (or rank) of the complete table is in fact extremely low compared to its size. Hence, for large (incomplete) datasets drawn from low-dimensional structures, we have:

New Premise III: **Incomplete Data ≈ Complete Information**.

As above examples suggest, in the modern era of big data, we often face

[5] Most internet companies make money from advertisements, including but not limited to Google, Baidu, Facebook, Bytedance, Amazon, Alibaba, Netflix, etc.

problems of recovering specific information that is buried in highly redundant, irrelevant, seemingly incomplete, or even corrupted[6] data sets. Such information without exception is encoded as certain low-dimensional structures underlying the data, and may only depends on a small (or sparse) subset of the (massive) dataset. This is very different from the classical settings and is precisely the reason why modern data science and engineering are undergoing a fundamental shift in their mathematical and computational paradigms. At its foundation, we need to develop a new mathematical framework that characterizes precise conditions under which such low-dimensional information can be correctly and effectively acquired and retained. Equally importantly, we need to develop efficient algorithms that are capable of retrieving such information from massive high-dimensional datasets, at unprecedented speed, at arbitrary scale, and with guaranteed accuracy.

Purposes of This Book.
Over the past two decades, there have been explosive developments in the study of low-dimensional structures in high-dimensional spaces. To a large extent, the geometric and statistical properties of representative low-dimensional models (such as sparse and low-rank and their variants and extensions) are now well understood. Conditions under which such models can be effectively and efficiently recovered from (minimal amount of sampled) data have been clearly characterized. Many highly efficient and scalable algorithms have been developed for recovering such low-dimensional models from high-dimensional data. The working conditions and data and computational complexities of these algorithms have also been thoroughly and precisely characterized. These new theoretical results and algorithms have revolutionized the practice of data science and signal processing, and have had significant impacts on sensing, imaging, and information processing. They have significantly advanced the state of the art for many applications in areas such as scientific imaging[7], image processing[8], computer vision[9], bioinformatics[10], information retrieval[11], and machine learning[12]. As we will see from applications featured in this book, some of these developments seem to defy conventional wisdom.

As witnesses to such historical advancements, we believe that the time is now ripe to give a comprehensive survey of this new body of knowledge and to organize these rich results under a unified theoretical and computational paradigm. There are a number of excellent existing books on this topic that already focus on the mathematical/statistical principles of compressive sensing and sparse/low-dimensional modeling [FR13, HTW15, Van16, Wai19, FLZZ20]. Nevertheless, the

[6] say due to negligence, misinformation, rumors, or malicious tampering.
[7] compressive sampling and recovery of medical and microscopic images, etc.
[8] denoising, super-resolution, inpainting of natural images, etc.
[9] regular texture synthesis, camera calibration, and 3D reconstruction, etc.
[10] microarray data analysis for gene-protein relations etc.
[11] collaborative filtering of user preferences, documents and multimedia data etc.
[12] especially for interpreting, understanding, and improving deep networks.

goal of this book is to bridge, through truly tractable and scalable computation, the gap between principles and applications of low-dimensional models for high-dimensional data analysis:

$$\text{A New Paradigm:} \quad \textbf{Principles} \xleftrightarrow{\textbf{Computation}} \textbf{Applications}.$$

Hence, not only does this book establish mathematical principles for modeling low-dimensional structures and understanding the limits on when they can be recovered, but it also shows how to systematically develop provably efficient and scalable algorithms for solving the recovery problems, leveraging both classical and recent developments in optimization.

Furthermore, through a rich collection of exemplar applications in science and technology, the book aims to further coach readers and students on how to incorporate additional domain knowledge or other non-ideal factors (e.g. nonlinearity) in order to correctly apply these new principles and methods to model real-world data and solve real-world problems successfully.

Although the applications featured in this book are inevitably biased by the authors' own expertise and experiences in practicing these general principles and methods, they are carefully chosen to convey diverse and complementary lessons we have learned (often in a hard way). We believe these lessons have value for both theoreticians and practitioners.

Intended Audience.
In many ways, the body of knowledge covered in this book has great pedagogical value to young researchers and students in the area of data science. Through rigorous mathematical development, we hope our readers are able to gain new knowledge and insights about high-dimensional geometry and statistics, far beyond what has been established in classical signal processing and data analysis. Such insights are generalizable to a wide range of useful low-dimensional structures and models, including modern deep networks, and can lead to entirely new methods and algorithms for important scientific and engineering problems.

Therefore, this book is intended to be a textbook for a course that introduces basic mathematical and computational principles for sensing, processing, analyzing and learning low-dimensional structures from high-dimensional data. The *targeted core audience* of this book are entry-level graduate students in Electrical Engineering and Computer Science (EECS), especially in the areas of

data science, signal processing, optimization, machine learning,

and applications. This book equips students with systematic and rigorous training in concepts and methods of high-dimensional geometry, statistics, and optimization. Through a very diverse and rich set of applications and (programming) exercises, the book also coaches students how to correctly use such concepts and methods to model real-world data and solve real-world engineering and scientific problems.

The book is written to be friendly to both instructors and students. It provides ample illustrations, examples, exercises, and programs from which students

may gain hands-on experience with the concepts and methods covered in the book. Materials in this book were developed from several one-semester graduate courses or summer courses offered at the University of Illinois at Urbana-Champaign, Columbia University, ShanghaiTech University, Tsinghua University, and the University of California at Berkeley in the past ten years. The main prerequisites for such a course are college-level linear algebra, optimization, and probability. To make this book accessible to a broader audience, we have tried to make the book as self-contained as possible: we give a crisp summary of facts used in this book from linear algebra, optimization, and statistics in the Appendices. For EECS students, preliminary courses on signal processing, matrix analysis, optimization or machine learning will improve their appreciation. From our experiences, besides beginning graduate students, many senior undergraduate students at these institutes were able to take the course and read the book without serious difficulty.

Organization of This Book.
The main body of this book consists of three inter-related Parts: *Principles, Computation, and Applications (PCA)*. The book also contains five *Appendices* on related background knowledge.

- *Part I: Principles (Chapters 2–7)* develops the fundamental properties and theoretical results for sparse, low-rank, and general low-dimensional models. It characterizes the conditions, in terms of sample/data complexity, under which the inverse problems of recovering such low-dimensional structures become tractable and can be solved efficiently, with guaranteed correctness or accuracy.
- *Part II: Computation (Chapters 8–9)* introduces methods from convex and nonconvex optimization to develop practical algorithms that are tailored for recovering the low-dimensional models. These methods show powerful ideas how to systematically improve algorithm efficiency and reduce overall computational complexity so that the resulting algorithms are fast and scalable to large-size and high-dimensional data.
- *Part III: Applications (Chapters 10–16)* demonstrates how principles and computational methods in the first two parts could significantly improve the solutions to a variety of real-world problems and practices. These applications also coach how the idealistic models and algorithms introduced in this book should be properly customized and extended to incorporate additional domain-specific knowledge (priors or constraints) about the applications.
- *Appendices A–E* at the end of the book are meant to make the book largely self-contained. The appendices cover basic mathematical concepts and results from Linear Algebra, Optimization, and High-Dimensional Statistics that are used in the main body of the book.

The overall organization of these chapters (and appendices) as well as their logical dependency is illustrated in Figure 0.4.

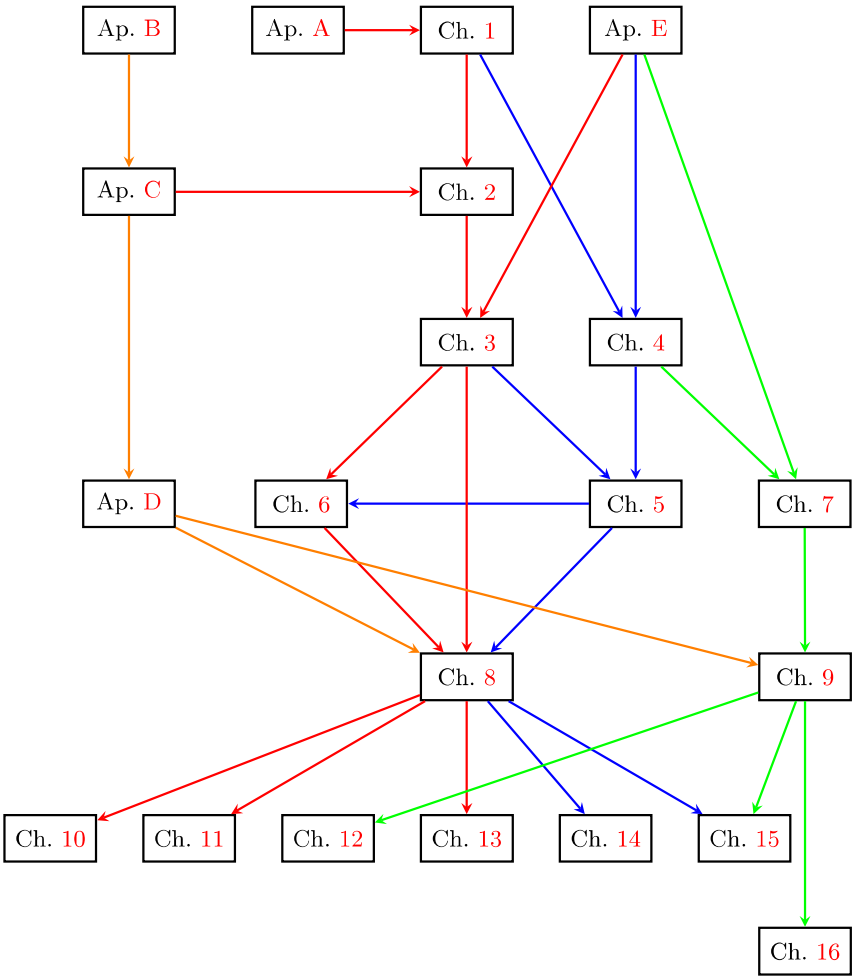


Figure 0.4 Organization Chart of the Book: dependency among chapters and appendices. Red route: sparse recovery via convex optimization; Blue route: low-rank recovery via convex optimization; Green route: nonconvex approach to low-dimensional models; Orange route: development of optimization algorithms.

How to Use This Book to Teach or to Learn.
The book contains enough material for a two-semester course series. We have purposely organized the material in the book in a modular fashion so that the chapters and even sections can be easily selected and organized to support different types of courses. Here are some examples:

- *A One-Quarter Course on Sparse Models and Methods* for Graduate or Upper Division Undergraduate Students: the introduction Chapter 1 and two theoretical Chapters 2 and 3; the convex optimization Chapter 8, and two to three applications from Chapters 10, 11, and 13, plus some appendices will be ideal

for an eight to ten-week summer or quarter course for senior undergraduate students and early year graduate students. That is essentially the red route highlighted in Figure 0.4.

- *A One-Semester Course on Low-Dimensional Models* for early year Graduate Students: the introduction Chapter 1 and the four theoretical Chapters 2–5; the convex optimization Chapter 8, and the several application Chapters 10, 11, 13–15, plus the appendices will be adequate for a one-semester course on low-dimensional models for graduate students. That is essentially both the red and the blue routes highlighted in Figure 0.4.
- *An Advanced-Topic Course on High-Dimensional Data Analysis* for senior Graduate Students who conduct research in related areas: with the previous course as prerequisite, a more in-depth exposition of the mathematical principles including Chapter 6 on convex methods for general low-dimensional models and Chapter 7 on nonconvex methods. One then can give a more in-depth account of the associated convex and nonconvex optimization methods in Chapters 8 and 9, and several application Chapters 12, 15, and 16 for nonlinear and nonconvex problems. Those are essentially the green and the orange routes highlighted in Figure 0.4. In addition, the instructor may choose to cover new developments in the latest literature, such as broader families of low-dimensional models, more advanced optimization methods, and extensions to deep networks (for low-dimensional submanifolds), say along open directions suggested in the epilogue of Chapter 16.

Certainly, this book can be used as a supplementary textbook for existing (graduate-level) courses on *Signal Processing* or *Image Processing*, since it offers more advanced new models, methods, and applications. It can also be used as a complementary textbook for more traditional courses on *Optimization* as Chapters 8 and 9 give a rather complete and modern coverage of the first-order (hence more scalable) methods. For a conventional *Machine Learning* or *Statistical Data Analysis* course, this book may serve as an additional reference for deeper and broader extensions to classic regression analysis, principal component analysis, and deep learning. For a more theoretical course on *High-dimensional Statistics and Probability*, this book can be used as a secondary text and provides ample motivating and practical examples.

In the future, we would very much like to hear from experienced instructors and seasoned researchers about other good ways to teach or study material in this book. We will share those experiences, suggestions, and even new contributions (examples, exercises, illustrations etc.) at the book's website:

https://book-wright-ma.github.io.

Yi Ma, Berkeley, California
John Wright, New York, New York
December 2020

Acknowledgements

Yi was first introduced to the subject of sparse representation by professor David Donoho of Stanford when David visited the University of Illinois in 2005. During the dinner, David and Yi discussed about Yi's research interests at that time: in particular *Generalized Principal Component Analysis* (GPCA) [VMS16], a subject that aims to learn an arbitrary mixture of low-dimensional subspaces from high-dimensional mixed data. David commented that GPCA in its most general setting would be an extremely challenging problem. He suggested why not starting with the simpler sparse model for which data are assumed to lie on a special family of subspaces.[13] Soon after that, Yi studied the subject of sparse representation systematically, especially during his sabbatical leave at Microsoft Research Asia in Dr. Harry Shum's group in 2006 and later at Berkeley in Professor Shankar Sastry's group in 2007. He was profoundly influenced and inspired by a series of seminal work at the time from Emmanuel Candès and Terence Tao on compressive sensing, error correction, and low-rank matrix recovery.

Since then, we have had the greatest fortune to work closely with many wonderful colleagues in this exciting new field. They are: Emmanuel Candès, Michael Elad, Guillermo Sapiro, Mario Figueiredo, René Vidal, Robert Fossum, Harm Derksen, Thomas Huang, Xiaodong Li, Shankar Sastry, Jitendra Malik, Carlos Fernandez, Julien Mairal, Yuxin Chen, Zhihui Zhu, Daniel Spielman, Peter Kinget, Abhay Pasupathy, Daniel Esposito, Szabolcs Marka, and Zsuzsa Marka. We would also like to thank many of our former colleagues when we visited or worked at Microsoft Research and other places: Harry Shum, Baining Guo, Weiying Ma, Zhouchen Lin, Yasuyuki Matsushita, Zuowen Tu, David Wipf, Jian Sun, Kaiming He, Shuicheng Yan, Lei Zhang, Liangshen Zhuang, Weisheng Dong, Xiaojie Guo, Xiaoqin Zhang, Kui Jia, Tsung-Han Chan, Zinan Zeng, Guangcan Liu, Jingyi Yu, Shenghua Gao, and Xiaojun Yuan. These collaborations have broadened our knowledge and enriched our experience in this field. Many results featured in this book are conveniently borrowed from these years of fruitful collaborations.

We would like to send special thanks to our former students Allen Yang,

[13] In the last chapter of this book, Chapter 16, we will see a rather unexpected connection between sparse models and GPCA, through an unexpected third party: *deep learning*. Concepts developed for GPCA such as lossy coding rates for clustering subspaces, in Chapter 6 of [VMS16], will play a crucial role in understanding deep networks.

Chaobing Song, Qing Qu and Yuqian Zhang for directly helping with content in some of the chapters. Allen has been of great help during early germination of the book project, back to early 2013. He has helped draft early versions of the application chapters on MRI and robust face recognition. Chaobing has helped transform the optimization chapters with a unified parsimonious approach to optimization algorithm design and brought this classic topic to the modern context of scalable computation. We would also like to thank some of our colleagues who have generously shared some material for this book: Bruno Olshausen, Michael Lustig, Julien Mairal, Yuxin Chen, Sam Buchanan, and Tingran Wang.

We would like to thank many of our former and current students. Their research has contributed to many of the results featured in this book. Many of them have also kindly helped proofreading drafts of the book during different stages or helped developing exercises as they were taking or teaching assistants for the courses based on early drafts of this book. They are Allen Yang, Arvind Ganesh, Andrew Wagner, Shankar Rao, Zihan Zhou, Hossein Mobahi, Jianchao Yang, Kerui Min, Zhengdong Zhang, Yigang Peng, Xiao Liang, Xin Zhang, Yuexiang Zhai, Haozhi Qi, Yaodong Yu, Christina Baek, Zhengyuan Zhou, Chaobing Song, Chong You, Yuqian Zhang, Qing Qu, Han-Wen Kuo, Yenson Lau, Robert Colgan, Dar Gilboa, Sam Buchanan, Tingran Wang, Jingkai Yan, and Mariam Avagyan.

Last but not the least, we would like to thank generous financial support through all these years from the National Science Foundation, Office of Naval Research, Tsinghua Berkeley Shenzhen Institute, Simons Foundation, Sony Research, HTC and VIA Technologies Inc.

Yi Ma, Berkeley, California
John Wright, New York, New York
December 2020

Contents

1 Introduction

"*Entities should not be multiplied without necessity.*"
– William of Ockham, *Law of Parsimony*

1.1 A Universal Task: Pursuit of Low-Dimensional Structure

The problem of identifying low-dimensional structure of signals or data in high-dimensional spaces is one of the most fundamental problems that, through a long history, interweaves many engineering and mathematical fields such as system theory, pattern recognition, signal processing, machine learning, and statistics.

1.1.1 Identifying Dynamical Systems and Serial Data

The low-dimensionality of real-world signals or data often arises from the intrinsic physical mechanisms from which the data are generated. Many real-world signals or data are observations of physical processes governed by certain generative mechanisms. For instance, magnetic resonance (MR) images[1] are generated by manipulating magnetic fields that obey Maxwell's equations; dynamics of any mechanical systems such as cars and legged robots follow Newton's laws of motion.

Mathematically such dynamics can often be modeled by a set of differential equations,[2] also known as a *state-space model* in system theory [CD91, Sas99]:

$$\begin{cases} \dot{\boldsymbol{x}}(t) &= f(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) &= g(\boldsymbol{x}(t), \boldsymbol{u}(t)), \end{cases} \tag{1.1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state, $\boldsymbol{u} \in \mathbb{R}^{n_i}$ is the input, and $\boldsymbol{y} \in \mathbb{R}^{n_o}$ is the (observed) output. Governed by such dynamical models, the output $\boldsymbol{y}(t)$ and state $\boldsymbol{x}(t)$ as functions in time t cannot be free and they are restricted to certain *low-dimensional submanifold* in their respective functional space.

To see this more clearly, we consider the simplified case when the dynamical

[1] that we will study in detail in Chapter 10.
[2] Here for simplicity, we only consider ordinary differential equations. But the same argument carries over to data or signals associated with partial differential equations.

model is (discrete) linear time invariant [CD91, OSB99][3]:

$$\begin{cases} \boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t). \end{cases} \tag{1.1.2}$$

According to the theory of system identification [VdM96], the observed output $\{\boldsymbol{y}(t)\}_{t=1}^{\infty}$ are correlated with the input $\{\boldsymbol{u}(t)\}_{t=1}^{\infty}$ through a subspace of dimension no more than $n = \dim(\boldsymbol{x})$. To be more precise, let us define two *Hankel* type matrices:

$$\boldsymbol{Y} \doteq \begin{bmatrix} \boldsymbol{y}(1) & \boldsymbol{y}(2) & \cdots & \boldsymbol{y}(N) \\ \boldsymbol{y}(2) & \boldsymbol{y}(3) & \cdots & \boldsymbol{y}(N+1) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}(N) & \boldsymbol{y}(N+1) & \cdots & \boldsymbol{y}(2N-1) \end{bmatrix} \in \mathbb{R}^{n_o N \times N}, \quad \boldsymbol{U} \doteq \begin{bmatrix} \boldsymbol{u}(1) & \boldsymbol{u}(2) & \cdots & \boldsymbol{u}(N) \\ \boldsymbol{u}(2) & \boldsymbol{u}(3) & \cdots & \boldsymbol{u}(N+1) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{u}(N) & \boldsymbol{u}(N+1) & \cdots & \boldsymbol{u}(2N-1) \end{bmatrix} \in \mathbb{R}^{n_i N \times N}.$$

Then from (1.1.2), the two matrices $\boldsymbol{Y}$ and $\boldsymbol{U}$ are related as:

$$\boldsymbol{Y} = \boldsymbol{G}\boldsymbol{X} + \boldsymbol{H}\boldsymbol{U}, \tag{1.1.3}$$

where $\boldsymbol{G}$ and $\boldsymbol{H}$ are matrices with blocks of the form $\boldsymbol{C}\boldsymbol{A}^i$ and $\boldsymbol{C}\boldsymbol{A}^i\boldsymbol{B}$ respectively, and

$$\boldsymbol{X} = [\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(N)] \quad \in \mathbb{R}^{n \times N}.$$

Let $\boldsymbol{U}^\perp$ be the orthogonal complement to $\boldsymbol{U}$.[4] We have:

$$\boldsymbol{Y}\boldsymbol{U}^\perp = \boldsymbol{G}\boldsymbol{X}\boldsymbol{U}^\perp. \tag{1.1.4}$$

Hence we have:

Fact 1.1 (Linear System Identification). *Regardless of the measurement sequence length N, the so-defined input-output matrix $\boldsymbol{Y}\boldsymbol{U}^\perp$ is always of rank less than or equal to the dimension n of the state space:*

$$\mathit{rank}(\boldsymbol{Y}\boldsymbol{U}^\perp) \leq n. \tag{1.1.5}$$

In other words, the column vectors of the matrix $\boldsymbol{Y}\boldsymbol{U}^\perp$ span an n-dimensional subspace in an ambient space of $\mathbb{R}^{n_o N}$. From the theory of system identification [VdM96, LV09, LV10], recovering this n-dimensional subspace associated with the input and output is the key to identifying the (unknown) parameters of the system $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ as they can subsequently be computed from the singular value decomposition[5] of the matrix $\boldsymbol{Y}\boldsymbol{U}^\perp$. In fact, system identification is one of the first problems that have inspired the convex approach for low-rank models [FHB01], which we will thoroughly study in Chapter 4.

[3] In many applications, linear time invariant models can be viewed as a good approximation to real dynamical systems that could be mildly nonlinear or slowly time-varying. Or for many classes of nonlinear systems, they can be converted, either via feedback linearization [Sas99] or via a smooth nonlinear Koopman operator [Koo31, LKB18], to linear dynamical systems.

[4] That is, columns of $\boldsymbol{U}^\perp$ span the null space of $\boldsymbol{U}$. See Appendix A.

[5] For details on singular value decomposition, please see Section A.8 of Appendix A.

Figure 1.1 From Left to Right: a texture image of regular pattern, a binary image of a Chinese character which is nearly symmetric, and an image of the Tiantan Temple of Beijing, which has a cylindrical body with its surface decorated with regular structural patterns.

EXAMPLE 1.2 (Recurrent Neural Network). *Notice that, in modern practice of* deep neural networks *(DNNs), variants to such state-space models*[6] *have been widely adopted, also known as* recurrent neural networks *(RNNs). A typical RNN model is of the so-called Jordan form [Jor97]:*

$$\begin{cases} \boldsymbol{x}(t+1) &= \sigma_{\boldsymbol{x}}\big(\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{b}\big), \\ \boldsymbol{y}(t) &= \sigma_{\boldsymbol{y}}\big(\boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{d}\big), \end{cases} \tag{1.1.6}$$

where $\sigma_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{y}}$ are certain nonlinear activation functions[7]*. RNNs and its many variants have empirically proven to be very effective for modeling serial data such as speech signals, videos, and natural languages. The intrinsic low-dimensionality of such models is the key to capturing structure or order in such serial data. Fundamental concepts, principles, and methods developed in this book will lead to a principled understanding of such deep models, as we will see in Chapter 16.*

1.1.2 Patterns and Orders in Man-Made World

Of course, many other factors may attribute to the ubiquitous presence of low-dimensional structures in real world data that do not necessarily involve natural dynamics or serial order. Another ample source of low-dimensional structures is due to human influence: almost all man-made objects are built by following simple code, rules, and procedures, both for economy and beauty. Those structures often visually manifest as repeated patterns in textures and decorations; symmetry in letters and characters; parallel, orthogonal, and regular shapes in man-made objects and architectures etc, as the few examples shown in Figure 1.1 and many more to be given in Chapter 15.

If we are to model such structures mathematically, low-dimensional models become the natural choices. For example, consider the leftmost image of a regular texture in Figure 1.1. We may view pixels of the 2D image array as the entries of

[6] usually with additional nonlinear activations introduced to places in the state space model.

[7] Popular choices of activation functions include the sigmoid function $\sigma(x) = \frac{e^x}{e^x+1}$ or the rectified linear unit (ReLU) function $\sigma(x) = \max\{0, x\}$.

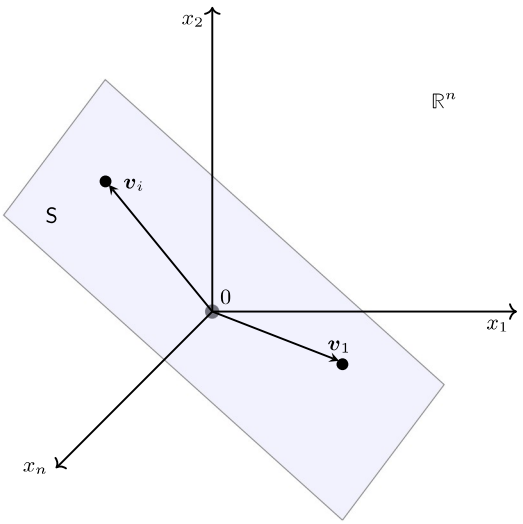


Figure 1.2 Column vectors $\boldsymbol{v}_i \in \mathbb{R}^n$ of a low-rank $n \times n$ matrix span a low-dimensional subspace $\mathsf{S} \subset \mathbb{R}^n$.

a matrix $\boldsymbol{M}$, say a matrix of $n \times n$ pixels. Obviously the column (or row) vectors of this matrix, viewed as vectors $\boldsymbol{v}_i$ in $\mathbb{R}^n$, are highly linearly dependent. They actually span only a very low-dimensional subspace S whose dimension, say d, is much less than n, as illustrated in Figure 1.2. That is

$$\operatorname{rank}(\boldsymbol{M}) = d \ll n. \tag{1.1.7}$$

Notice that this is the same type of low-rank condition that we have seen in the system identification problem (1.1.4). In the application Chapter 15, we will see how such natural low-rank regular textures would allow us to efficiently, accurately and robustly recover geometric information encoded in such images – revealing the reason why we are able to accurately perceive 3D geometry of the Tiantan Temple and recover the rectified 2D texture from only a single image, shown on the right of Figure 1.1.

As a matter of fact, even for any generic 3D scene, when taken photos from multiple poses, the multiple 2D images of the same point, line, plane or (symmetric) object in 3D are all related in such a way that certain measurement matrix, known as the *multiple-view matrix* $\boldsymbol{M}$, becomes low-rank [MSKS04]. In fact, somewhat remarkably, the rank of such matrices will always be

$$\operatorname{rank}(\boldsymbol{M}) = 1 \text{ or } 2, \tag{1.1.8}$$

regardless of the number of views or the size of the matrix. A similar low-rank condition applies to multiple images of the same scene taken under different lighting conditions: $\operatorname{rank}(\boldsymbol{M}) = 3$, as we will study thoroughly in Chapter 14.

In general, we do not expect all data in human society to be equally regular

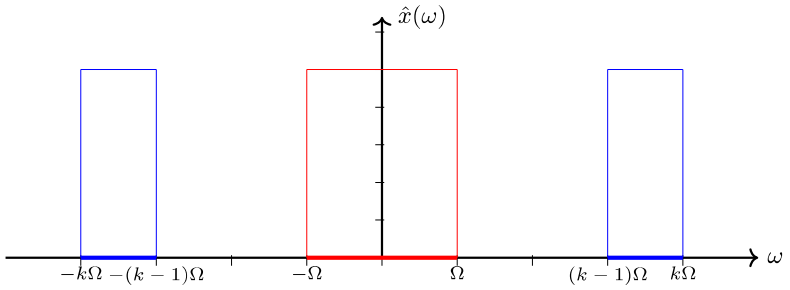


Figure 1.3 Functions with spectrum supported in the red region are known as band-limited function. They have the same size of spectral support as functions with spectrum supported in the two blue regions.

and orderly. Nevertheless, many data that arise from societal, commercial, and financial activities or from social networks do exhibit very good patterns that can be well approximated by low-dimensional models, as we will see from plenty of examples in Chapters 4 and 5 and in the application Chapters 14–16. In this book we will establish the fundamental principles and algorithms that would allow us to exploit such low-dimensional structures in real data for correct and efficient recovering information from minimal (incomplete or imperfect) observations.

1.1.3 Efficient Data Acquisition and Processing

In classical signal processing, the intrinsic low-dimensionality of data is mostly exploited for purposes of efficient sampling, storage, and transport [OSB99, PV08]. In applications such as communication, it is often reasonable to assume the signals of interest mainly consist of limited frequency components[8]. To be more precise, consider a signal $x(t)$ as a function of time t and its Fourier transform:[9]

$$\hat{x}(\omega) \doteq \int_{-\infty}^{\infty} x(t) \exp(-\mathrm{i}\omega t) dt. \tag{1.1.9}$$

Typically $\hat{x}(\omega)$ will be zero when $|\omega| \geq \Omega$ for some $\Omega > 0$. Let $\mathcal{B}_1(\Omega)$ be the set of *band-limited functions* whose Fourier transform vanishes outside of the spectrum $[-\Omega, \Omega]$:

$$\mathcal{B}_1(\Omega) \doteq \left\{ x \in L^1(\mathbb{R}) \mid \hat{x}(\omega) = 0 \ \ \forall\, |\omega| > \Omega \right\}, \tag{1.1.10}$$

as illustrated in Figure 1.3.

In other words, all functions in $\mathcal{B}_1$ has a maximal cut-off frequency $f_{\max} =$

[8] as analog and digital information is often physically carried by modulating periodic signals generated by resonant circuits, as we will elaborate more in Chapter 11.

[9] One may see Appendix A for a discretized version of the Fourier transform, equation (A.7.13), that can be applied to discretized signals or vectors.

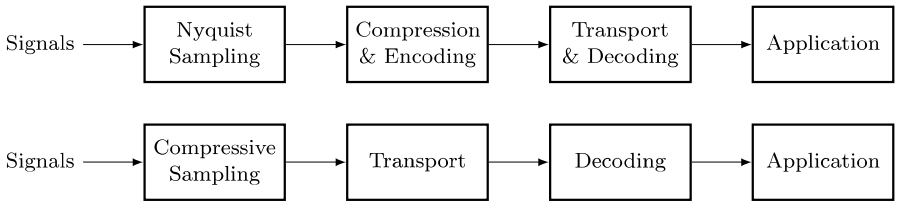


Figure 1.4 Comparison of classical signal acquisition and processing pipeline (top) and the compressive sensing paradigm to be introduced in this book (bottom).

$\Omega/2\pi$. Notice that $\mathcal{B}_1$ forms *a subspace* in the space of all functions, just like the range of low-rank matrix is a subspace in a vector space. This structure allows us to represent such functions rather efficiently with their discrete samples. To see this, given $\hat{x}(\omega)$ the signal $x(t)$ can be expressed by the inverse Fourier transform:

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{x}(\omega)\exp(\mathrm{i}\omega t)d\omega = \frac{1}{2\pi}\int_{-\Omega}^{\Omega} \hat{x}(\omega)\exp(\mathrm{i}\omega t)d\omega. \tag{1.1.11}$$

So if we view $\hat{x}(\omega)$ as a periodic function in the spectral domain with a period 2Ω, it is fully determined by all its Fourier coefficients:

$$x\Big(\frac{n\pi}{\Omega}\Big) \doteq \frac{1}{2\pi}\int_{-\Omega}^{\Omega} \hat{x}(\omega)\exp\Big(\mathrm{i}\omega\frac{n\pi}{\Omega}\Big)d\omega, \quad n = 0, \pm 1, \pm 2, \ldots. \tag{1.1.12}$$

Notice that the left hand side is precisely the values of the function $x(t)$ sampled with a period $T = \frac{\pi}{\Omega}$, or equivalently at a frequency

$$f = \frac{1}{T} = 2\cdot\frac{\Omega}{2\pi}. \tag{1.1.13}$$

Hence we have:

Fact 1.3 (Nyquist-Shannon Sampling). *To perfectly recover a band-limited signal $x(t)$, we need to sample it at a rate that is twice its maximal frequency $f_{\max} = \Omega/2\pi$.*

This is known as the classical *Nyquist-Shannon* sampling theorem [OSB99]. The sampled (hence discrete) signal can then be digitized and compressed based on its additional statistics. For images, such sampling and subsequent compression are done by the popular schemes such as JPEG or MPEG for videos. The compressed data are then used for storage, transport, and to be decoded later for various applications. Figure 1.4 (top) illustrates a traditional pipeline for data acquisition and processing.

However, for signals that contain both low-frequency and high-frequency components, sampling at the Nyquist rate sometimes can be rather costly. For instance, as shown in Figure 1.3, for signals with their spectrum supported only in the red area, their maximum cut-off frequency is $\Omega/2\pi$; yet for signals with spectrum supported only in the blue areas, the maximum frequency is $k\cdot\Omega/2\pi$.

So when k is very large (which is the situation in modern wide-band wireless communication, see Chapter 11), the Nyquist sampling scheme would be rather expensive to realize. As an important example, in order to capture sharp edges or boundaries in natural images,[10] the number of pixels of imaging sensors in digital cameras has increased dramatically in recent years. Such a *brute force* sensing scheme is obviously rather wasteful since sharp edges occupy only a very tiny fraction of the image and yet all the relatively smooth regions are sampled at the same rate! In medical imaging, such brute force increasing of sampling density is not even allowed due to patient comfort and safety [LDP07].

As we will see in this book, the number of samples truly needed to recover a signal should be proportional to the total width of its spectral support regardless of the location! For the examples shown in Figure 1.3, both types of signals would have the same effective bandwidth of 2Ω and in principle can be correctly recovered with effectively the same sampling rate. As a result, to acquire signals with spectrum supported in the blue regions, the sampling rate can be significantly lower than the Nyquist sampling rate [Tro10, ME10], hence the notion of "*compressed sensing*" or "*compressive sensing*", coined by [Don06a, Can06]. We will see in Chapter 11 precisely how such a new sampling scheme is realized in the context of modern wide-band wireless communications.

In this book, we will systematically study the theoretical foundation for designing such compressive sampling schemes in a principled manner and develop algorithms for recovering the full signal from such samples correctly and efficiently. In general, such compressive samples of the signals are already compact enough for storage and transport, and the original signals can be fully recovered later when they are eventually being used. Figure 1.4 (bottom) illustrates this new data acquisition and processing paradigm. In additional to wide-band communications, we will also see a few striking applications of this paradigm at work. For instance, this new paradigm has revolutionized the field of medical imaging [LDP07], as we will elaborate more in Chapter 2 and further in Chapter 10.

1.1.4 Interpretation of Data with Graphical Models

In the practice of modern data science, we often deal with data that are not necessarily generated from any clear physical processes or artificial protocols. Their generative mechanisms can be hidden from us or are difficult to derive from first principles. Data such as customer ratings, web documents, natural languages, and gene expression data are such examples. Nevertheless, such data are by no means structureless, and there are usually strong and rich statistical correlation, dependency/independency, and causal relationships among the data.

To model such structure, one may view the observed data as samples of a set of random variables $\boldsymbol{x}_o \in \mathbb{R}^{n_o}$, which are generated through certain conditional

[10] A sharp edge can be represented by a step function which is not band-limited!

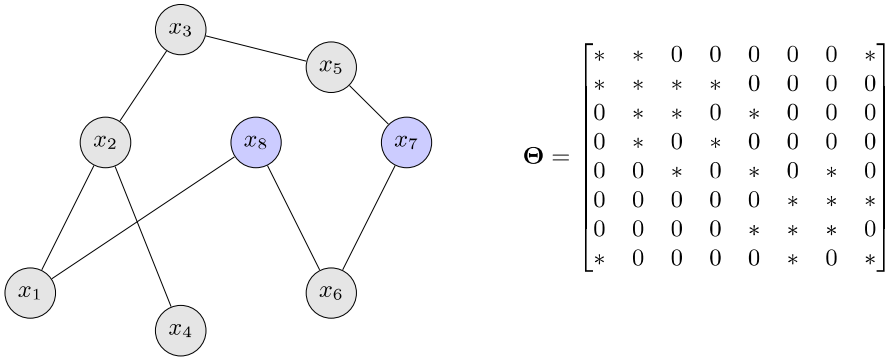


Figure 1.5 Graphical model for a set of jointly Gaussian random variables. The inverse covariance matrix $\boldsymbol{\Theta}$ is often sparse if the dependency graph is sparsely connected. Suppose that gray nodes represent observed variables $\boldsymbol{x}_o = [x_1, x_2, \ldots, x_6]^*$ and blue ones $\boldsymbol{x}_h = [x_7, x_8]^*$ are hidden.

probability distribution given another set of hidden or *latent* variables $\boldsymbol{x}_h \in \mathbb{R}^{n_h}$. The structure of the data is fully described by the joint distribution of the random vector $\boldsymbol{x} = (\boldsymbol{x}_o, \boldsymbol{x}_h) \in \mathbb{R}^n$ with $n = n_o + n_h$. Now consider the n random variables $\{x_i\}_{i=1}^n$ in $\boldsymbol{x}$. For simplicity, let us assume that $\{x_i\}_{i=1}^n$ are jointly zero-mean Gaussian[11], i.e., $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ with a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$. Let

$$\boldsymbol{\Theta} \equiv \boldsymbol{\Sigma}^{-1} \quad \in \mathbb{R}^{n \times n}$$

be the inverse of its covariance matrix. From statistics, we have the following well-known fact:

FACT 1.4 (Conditional Independence in Graphical Model). *Any two variables x_i and x_j are conditionally independent given all other variables $\{x_k \mid k \neq i, j\}$ if and only if the (i, j)-th entry of $\boldsymbol{\Theta}$ satisfies $\theta_{ij} = 0$.*

In machine learning, such dependencies among random variables in $\boldsymbol{x} = \{x_i\}_{i=1}^n$ is often described with a *graphical model* [Pea00, Jor03, WJ08], denoted as $\mathcal{G} = (\mathsf{V}, \mathsf{E})$: The set of vertices V consists of all the random variables $\mathsf{V} = \{x_i\}_{i=1}^n$, and the set of edges $\mathsf{E} = \{e_{ij}\}$ indicate dependency among pairs of random variables (x_i, x_j) – there is an edge between x_i and x_j if and only if they are conditionally dependent. Figure 1.5 shows one such example. In fact, the state-space model (1.1.1) in Section 1.1.1 can be viewed as a special case of such latent variable graphical models[12].

A fundamental and challenging problem in statistical learning is how to infer the joint distribution of $\boldsymbol{x}$ from marginal statistics of the observed variables $\boldsymbol{x}_o$

[11] In practice, Gaussian can be used to approximate any distribution up to its second-order statistics.

[12] the input $\boldsymbol{u}$ and output $\boldsymbol{y}$ would be the observations and the (randomly initialized) state $\boldsymbol{x}$ would be the hidden latent variables.

even if the number of latent variables and their relationships with the observed ones are unknown. In the most basic case when all the variables are jointly Gaussian, we may partition the covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{x} = (\boldsymbol{x}_o, \boldsymbol{x}_h)$ as:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_o & \boldsymbol{\Sigma}_{o,h} \\ \boldsymbol{\Sigma}_{o,h}^* & \boldsymbol{\Sigma}_h \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{\Theta}_o & \boldsymbol{\Theta}_{o,h} \\ \boldsymbol{\Theta}_{o,h}^* & \boldsymbol{\Theta}_h \end{bmatrix}^{-1} \in \mathbb{R}^{n \times n}. \tag{1.1.14}$$

Notice that in the above covariance matrix, only the covariance associated with the observed data $\boldsymbol{\Sigma}_o$ can be obtained from (statistics of) the data. Using facts from linear algebra, one can show that $\boldsymbol{\Sigma}_o$ is of the form:

$$\boldsymbol{\Sigma}_o^{-1} = \boldsymbol{\Theta}_o - \boldsymbol{\Theta}_{o,h}\boldsymbol{\Theta}_h^{-1}\boldsymbol{\Theta}_{o,h}^* \quad \in \mathbb{R}^{n_o \times n_o}. \tag{1.1.15}$$

In the above expression, the first term $\boldsymbol{\Theta}_o$ will be sparse if the graph $\mathcal{G}$ is and the second term $\boldsymbol{\Theta}_{o,h}\boldsymbol{\Theta}_h^{-1}\boldsymbol{\Theta}_{o,h}^*$ has a rank less than the number of latent variables, which is often relatively small. For the example shown in Figure 1.5, there are only two hidden nodes; hence the rank of the second term would be at most 2 and the first term $\boldsymbol{\Sigma}_o$ would have the same pattern as the upper-left 6×6 submatrix of $\boldsymbol{\Theta}$ shown on the right of the figure. It has been shown that, in general, a graphical model is identifiable via tractable means *only if* the graphical model $\mathcal{G}$ is sufficiently sparse [CPW12]. Popular models such as trees and multi-layer deep networks are representative examples of such graphical models.

Under such conditions, the covariance matrix $\boldsymbol{\Sigma}_o$ of the observed variables $\boldsymbol{x}_o$ always has the following *decomposable structure*:

$$\boldsymbol{\Sigma}_o^{-1} = \boldsymbol{S} + \boldsymbol{L} \quad \in \mathbb{R}^{n_o \times n_o}, \tag{1.1.16}$$

where $\boldsymbol{S}$ is a sparse matrix and $\boldsymbol{L}$ is a low-rank matrix. The rank of $\boldsymbol{L}$ is associated with the number of (independent) latent variables in the graph: $\operatorname{rank}(\boldsymbol{L}) = \dim(\boldsymbol{x}_h)$; the sparse matrix $\boldsymbol{S}$ is associated with the conditional dependency of the observed variables – an entry s_{ij} of $\boldsymbol{S}$ is zero if the two observed variables x_i and x_j are *conditionally independent* given the others.

So to a large extend, the problem of inferring the full graphical model $\mathcal{G}$, or the covariance matrix $\boldsymbol{\Sigma}$ in the Gaussian case, reduces to a problem of decomposing a matrix $\boldsymbol{\Sigma}_o^{-1}$ into a low-rank matrix $\boldsymbol{L}$ and a sparse matrix $\boldsymbol{S}$. Although this decomposition problem (1.1.16) is generally *NP-hard*,[13] we will see in Chapter 5, when both $\boldsymbol{L}$ and $\boldsymbol{S}$ are sufficiently low-dimensional, this problem actually becomes *tractable* and can be solved correctly and efficiently by methods introduced in this book.

[13] The well studied "planted clique" problem [GZ19, BB20] in complexity theory is a special case of this problem, as we will discuss in Chapter 5.

1.2 A Brief History

Due to the ubiquity and importance of low-dimensional structures, there has been a long and rich history of studying, understanding, and exploiting them in Science, Engineering, Statistics, and Computation.

1.2.1 Neural Science: Sparse Coding

Through millions of years of evolution, the brains of humans and other animals, in particular the visual cortex, has adapted well to its living environment. The natural vision systems of primates are able to exploit statistics of natural images and achieves highly accurate visual perception with extreme efficiency in time and energy. This phenomenon has long been observed and studied extensively in neural science. Back in 1972, visual neuroscientist Horace Barlow proposed the following *dogma for natural vision* [Bar72]:

"*... the overall direction or aim of information processing in higher sensory centres is to represent the input as completely as possible by activity in as few neurons as possible.*"

In 1987, David Field provided the first scientific evidence in support of this conjecture by showing that the oriented receptive fields of simple cells in the visual cortex are well suited to encode natural images with a small fraction of active units [Fie87]. His results support Barlow's dogma that *the goal of natural vision is to represent the information in the natural environment with minimal redundancy.*

Later in 1996, Bruno Olshausen and David Field had further hypothesized in their seminal work [OF97] that in biological vision systems, visual sensory input data, say $\boldsymbol{y} \in \mathbb{R}^m$, are represented in terms of linear combination of a set of elementary patterns (or features) $\boldsymbol{a}_i \in \mathbb{R}^m$:

$$\boldsymbol{y} = \sum_{i=1}^{n} x_i \boldsymbol{a}_i + \boldsymbol{\varepsilon} \quad \in \mathbb{R}^m, \tag{1.2.1}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^* \in \mathbb{R}^n$ are sparse coefficients[14] and $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ is some small modeling errors. The collection of all patterns $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$ is called *a dictionary*, which is learned from statistics of the input. When adapted to a large collection of image patches extracted from natural images, the dictionary converges to a set of localized, oriented bandpass functions at different scales (or spatial-frequencies) strikingly similar to the receptive fields found in visual cortex (see Figure 1.6). Such a learned dictionary enables the vision system to reformat sensory information into a sparse code $\boldsymbol{x}$ during the early stages of visual processing. Subsequent studies of a wide range of animals (e.g. mouse, rat, rabbit, cat, monkey) and human brain have provided further evidences for sparse coding of sensory input in natural vision [OF04]. More recent studies of

[14] That is, most x_i's are zeros.

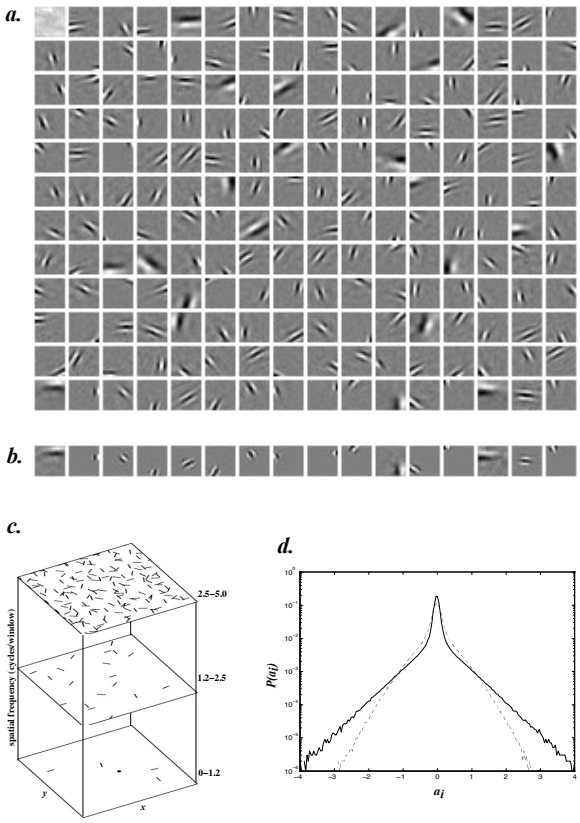


Figure 1.6 ***a***. Results from training a system of 192 basis functions on 16×16-pixel image patches extracted form natural scenes [OF96b]. ***b***. The receptive fields corresponding to the last row of basis functions in ***a***. ***c***. The distribution of the learned basis functions in space, orientation and scale. ***d***. Activity histograms averaged over all coefficients for the learned basis functions (solid line) and for random initial conditions (broken line). Image reprinted with permission from Bruno Olshausen.

neurons in the monkey cerebellum by Reza Shadmehr's group at Johns Hopkins [HKSS15, HKSS18] further suggest that the same sparse coding dictionary organizes sensory motor control output and prediction errors which, in turn, organizes the entire closed-loop learning network for natural vision.

The fact that sparse coding becomes a central principle for natural vision sends two encouraging messages to engineers: first, seemingly complex real data, such as natural images, do have good intrinsic structures that can be exploited for compact and efficient representations [OF96a]; second, such structures and representations are already learned effectively and efficiently by nature [OF97, GS12, LLT18]. To mathematicians and computer scientists, the second message might seem a little surprising. It contradicts a known fact that finding the sparse

code $\boldsymbol{x} \in \mathbb{R}^n$ for a given signal

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \quad \in \mathbb{R}^m \tag{1.2.2}$$

is in general an *NP-hard* problem even when the dictionary $\boldsymbol{A}$ is known but over-complete, i.e., $m < n$ (see Theorem 2.8). Hence sparse coding can be computationally prohibitive and yet nature seems to learn to do it effortlessly. To a large extent, studies in this book reconcile this contradiction by characterizing conditions under which the sparse coding problem can be solved efficiently and effectively (Chapter 3). Furthermore, we will see in later part of this book (Chapter 7) that, even when the dictionary $\boldsymbol{A}$ is not known in advance and needs to be learned (as in natural vision), given sufficient observations $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$:

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} \quad \in \mathbb{R}^{m \times N}, \tag{1.2.3}$$

both the correct dictionary $\boldsymbol{A}$ and associated sparse codes $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ can be learned correctly and efficiently, under fairly broad conditions! Eventually, towards the end of the last Chapter 16, we will see how mathematical and computational principles developed in this book might provide compelling mathematical justification for the need of sparse coding (even in nature), as well as other computational mechanisms that resonate more deeply with phenomena observed in neural science or cognitive science.

1.2.2 Signal Processing: Sparse Error Correction

The properties of sparse signals and data have long been studied by mathematicians and statisticians. Throughout history many have explored and proposed computationally efficient ways to exploit such properties. A classical problem in data analysis is to model an observation, say $y \in \mathbb{R}$, as a linear function of a set of known variables $\boldsymbol{a}^* = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^n$:

$$y = f(\boldsymbol{a}) = \boldsymbol{a}^*\boldsymbol{x} = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n, \tag{1.2.4}$$

where the $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^* \in \mathbb{R}^n$ are some unknown parameters to be determined. Given multiple, say m, observations of the form:

$$y_i = \boldsymbol{a}_i^*\boldsymbol{x} + \varepsilon_i, \quad i = 1, 2, \ldots, m, \tag{1.2.5}$$

where ε_i is possible measurement noise or error, we may stack y_i as entries of a vector $\boldsymbol{y} \in \mathbb{R}^m$ and $\boldsymbol{a}_i^* \in \mathbb{R}^n$ as rows of a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The goal is then to find a set of parameters $\boldsymbol{x} \in \mathbb{R}^n$ such that $\boldsymbol{A}\boldsymbol{x}$ fits well with the given observation $\boldsymbol{y} \in \mathbb{R}^m$. In the classical setting, we usually have the number of measurements larger than the unknowns, i.e., $m \geq n$. Hence there may be no solution $\boldsymbol{x}$ that satisfies the equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ precisely due to measurement errors.

Least Absolute Deviations versus Least Squares.
As early as in 1750, French mathematician Roger Joseph Boscovich had proposed to solve for $\boldsymbol{x}$ that minimizes the absolute deviations between $\boldsymbol{y}$ and $\boldsymbol{A}\boldsymbol{x}$ [Bos50],

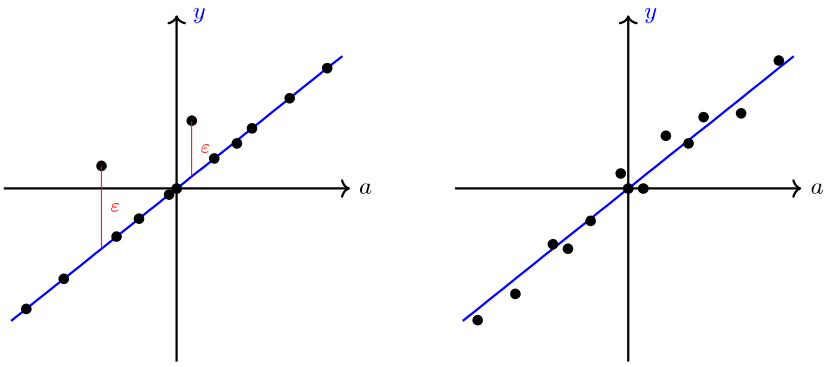


Figure 1.7 Data fitting with few but large errors versus small noises on almost every data points. The least absolute deviations (minimizing ℓ^1 norm of ε) is more suitable for the situation on the left whereas the least squares is for the right.

namely:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1 = \sum_{i=1}^{m} |y_i - \boldsymbol{a}_i^* \boldsymbol{x}|, \tag{1.2.6}$$

where $\|\cdot\|_1$ is the ℓ^1 norm of a vector which is the sum of absolute values of all its entries. This is also known as the *method of least absolute deviations*. According to historical account [Pla72], this work has made significant influence on Laplace's conception of Laplace distribution [Lap74], see Exercise 1.5. During the period which followed Boscovich and Laplace, mainly in early 1800's, the *method of least squares* was proposed independently by Legendre in 1805 [Leg05] and Gauss in 1809 [Gau09]:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 = \sum_{i=1}^{m} (y_i - \boldsymbol{a}_i^* \boldsymbol{x})^2. \tag{1.2.7}$$

The method of least squares (or minimizing the ℓ^2 norm of errors) is known to be statistically optimal when the measurement errors ε_i's are i.i.d. Gaussian noise[15]. In addition, the optimal minimizer $\boldsymbol{x}_\star$ admits a *closed-form* solution (which we leave as an exercise to the reader), hence is very appealing to practitioners before the age of computers.

At the time of Boscovich and Gauss, people intuitively knew that the least absolute deviations method (1.2.6) is more robust if the measurements contain *large but few* errors, as illustrated in Figure 1.7. However, the precise working conditions of ℓ^1 minimization were mostly not know or clarified, and unlike least

[15] To Gauss' credit, in his work [Gau09], he went beyond Legendre and established the connection between least squares and statistics, and showed its optimality for errors with Gaussian, also known as the normal, distribution. See Exercise 1.5.

squares, there is no closed-form solution to ℓ^1 minimization[16]. As a result, the method of least squares had dominated data analysis for the next nearly three centuries! Nevertheless, as we will see in this book, the lack of closed-form solution for ℓ^1 minimization is very much alleviated by modern efficient optimization methods. With computers, solving ℓ^1 minimization is no longer a bottleneck even when the scale is very large (see Chapter 8). Advance in computation has paved the way for a strong return of methods based on numerical solutions such as ℓ^1 minimization. The remaining questions are when ℓ^1 minimization works and why.

Logan's Phenomenon.
The theoretical analysis of ℓ^1 minimization for error correction has its earliest roots in work by Benjamin Logan[17] in the 1960's. His PhD thesis, completed at the Electrical Engineering Department of the Columbia University, featured the following intriguing result:

"*Suppose we observe a signal y which consists of a* band-limited *signal x_o, superimposed with an error e_o which is* sparse *in the time domain. If the product of the bandwidth of x_o and the size of the support of e_o is less than $\pi/2$, the true band-limited signal can be recovered by ℓ^1 minimization, no matter how large the error is in magnitude, or where its support is located.*"

This observation is known as *Logan's phenomenon*. To state this result slightly more formally, let $\mathcal{B}_1(\Omega)$ be the set of *band-limited functions* whose Fourier transform vanishes outside of $[-\Omega, \Omega]$, as previously defined in (1.1.10). A formal statement of Logan's theorem is as follows:

FACT 1.5 (Logan's Theorem). *Suppose that $y = x_o + e_o$, with $x_o \in \mathcal{B}_1(\Omega)$, $\|e_o\|_1 = \int_t |e_o(t)|dt < +\infty$ and $\mathrm{supp}(e_o) \subseteq T$. If*

$$|T| \times \Omega < \frac{\pi}{2}, \tag{1.2.8}$$

then x is the unique solution to the (conceptual) optimization problem

$$\begin{array}{ll} \min & \|x - y\|_1 \\ \text{subject to} & x \in \mathcal{B}_1(\Omega). \end{array} \tag{1.2.9}$$

Here, $|T|$ should be interpreted as the length of T (if T is an interval) or the Lebesgue measure of T (if T is a more general set). This result says that no matter how large the error e_o is in magnitude, as long as it is sparse enough, it can be exactly corrected by ℓ^1 minimization. Figure 1.8 illustrates the implication of this result. It highlights three different areas (red, blue, and green) of the same size in the spectrum-time space for x_o and e_o, respectively. If the area size is less than $\pi/2$, then x_o and e_o can be separated by ℓ^1 minimization.

[16] nor were there computers at the time!
[17] Harmonic analyst and signal processor at Bell Labs, and also a renowned bluegrass fiddler.

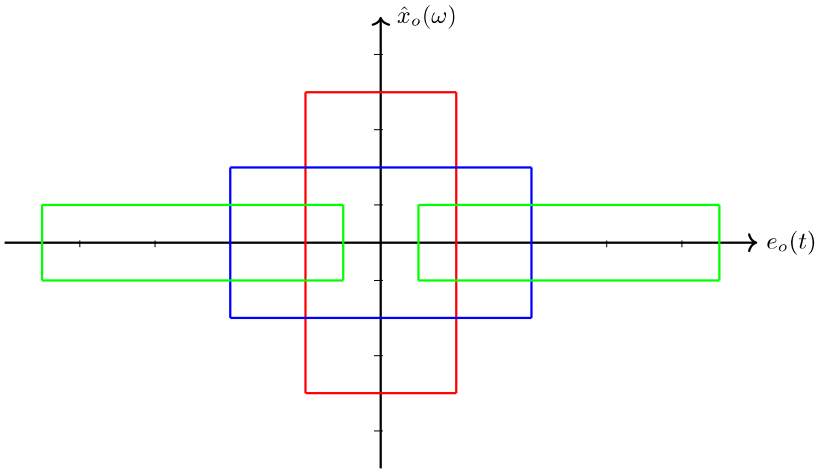


Figure 1.8 Illustration of Logan's Phenomenon: horizontal axis indicates support of e_o in time t, and vertical axis indicates support of the Fourier transform $\hat{x}_o$ of x_o in spectrum ω. All three colored areas have the same separability by ℓ^1 minimization according to Logan's statement.

Logan was working with an eye toward applications in audio signal processing, in which a band-limited signal is the target of interest, and the corruption e_o is to be removed. Although Logan's result is stated for continuous-time signals, we will give a concrete example that shows how it works for discretized digital signals in Section 2.3.4 of Chapter 2. At this point, acute readers may have recognized strong conceptual similarity between Logan's problem and the decomposition problem (1.1.16) that we have encountered in learning graphical models.

Logan obtained his result in the mid-1960's. It would be several decades before the modern theory of ℓ^1 minimization began taking form. However, practitioners in many applied computational disciplines were very actively practicing ℓ^1 minimization and related techniques for robust statistical inference with erroneous data, notably practice in the geosciences since the 1970's [CM73, SS86] as well as the work in robust statistics in the 1980's [Hub81, HRRS86]. In many cases, they observed intriguing phenomena, which seemed to parallel Logan's result: ℓ^1 minimization often exactly recovered sparse-enough solutions, and exactly corrected sparse-enough errors. Beginning in the early 2000's, a sequence of theoretical breakthroughs led to increasingly sharper and broader characterizations of the conditions under which ℓ^1 minimization succeeds in error correction (e.g., [CT05, WM10]). These are the conditions which we will develop thoroughly in this book.

1.2.3 Classical Statistics: Sparse Regression Analysis

A classical problem in statistical data modeling is to study how a given random variable, say $y \in \mathbb{R}$, depends on a set of predictive random variables (also known as predictors or features), say $\boldsymbol{a}^* = [a_1, a_2, \ldots, a_n] \in \mathbb{R}^n$. This is known as *regression analysis* [HTF09]. The most popular form is the linear regression in which we try to represent y as a linear superposition of (some or all of) the variables:

$$y = \boldsymbol{a}^* \boldsymbol{x} + \varepsilon = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + \varepsilon, \tag{1.2.10}$$

where ε is an error term whose variance is to be minimized:

$$\min \mathbb{E}\big[(y - \boldsymbol{a}^* \boldsymbol{x})^2\big]. \tag{1.2.11}$$

In practice, the problem becomes to find the coefficients $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^* \in \mathbb{R}^n$ from multiple, say m, samples $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^*$:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon} \quad \in \mathbb{R}^m, \tag{1.2.12}$$

where rows of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are corresponding samples of the predictors. The method of least squares discussed earlier by Legendre and Gauss:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \tag{1.2.13}$$

is arguably the earliest, and the most popular, form of regression in which all the variables $a_1, a_2, \ldots, a_n$ are used to predict y. See Figure 1.9 left for an example. This is often a reasonable thing to do if the number of variables n is small and they are already chosen to be somewhat independent of one another. One may refer to the recent book [BV18, FLZZ20] for a more extensive exposition of this topic.

Best Subset Selection.
In many settings of data analysis, the number of variables n can be very large. Many variables can be irrelevant for the prediction or there could be tremendous redundancy among the relevant ones[18]. Very often the number of predictors could even be larger than the number of available samples, i.e., $n > m$.[19] Hence, in addition to fitting the prediction $\boldsymbol{y}$ with $\boldsymbol{A}\boldsymbol{x}$, one often prefers to find a much smaller subset of the most relevant variables that can best fit $\boldsymbol{y}$ – the so called *variable selection*. In other words, the coefficient vector $\boldsymbol{x}$ is desired to be a sparse

[18] This is certainly the case with natural vision: to detect or identify an object in an image, the possible predictors can be in the same magnitude as the number of pixels. Hence dictionary learning and sparse coding becomes crucial in order to identify the most informative features that help with the detection.

[19] In the over-determined case, the least square problem (1.2.13) no longer has a unique solution. A classical way to fix this is through introducing an additional Tikhonov-type regularization term $\lambda\|\boldsymbol{x}\|_2^2$, resulting in the so called *ridge regression* $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2$. We leave this as an exercise for the reader, see Exercise 1.8.

vector with only a few, say $k \leq \min\{m, n\}$, of its entries being nonzero. A natural proposal to select $\boldsymbol{x}$ is to use the least squares metric:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_0 \leq k, \tag{1.2.14}$$

where $\|\boldsymbol{x}\|_0$ indicates the ℓ^0 norm – the number of nonzero entries of a vector. This is called *the best subset selection problem* in regression analysis and had originally proposed by Hocking and Leslie [HL67] and Beale et. al. [BKM67] in 1967. This notion of choosing minimal subset of relevant variables is related to the more general *principle of minimum description length* proposed by Rissanen in 1978 [Ris78], which argues that in choosing between various models, we should prefer models which can be encoded most efficiently [HY01].

Although this seems a sensible thing to hope for, directly solving the above subset selection problem is computationally intractable: when k and m become very large, the number of possible supports $\binom{m}{k}$ grows exponentially in k and m. In fact, we will soon see in the next chapter this problem is in general NP-hard. Hence, through the history, several other approaches have been proposed to address the variable selection problem via computationally tractable means.

Stepwise Regression.
In 1966, Efroymson [Efr66] proposed a greedy forward (or backward) *stepwise regression* scheme for variable selection: starting from an empty index set $\mathsf{I}_0 = \varnothing$, then at each step add to the index set I_k the index of a variable which gives the lowest squared error among all the remaining variables. To be more precise, let $\mathcal{P}_{\mathsf{I}}$ be the orthogonal projection on the range of the submatrix $\boldsymbol{A}_{\mathsf{I}}$ that consists of columns of $\boldsymbol{A}$ indexed by I. The greedy selection at each step is given by:

$$i_k = \arg\min_{i \notin \mathsf{I}_k} \|\boldsymbol{y} - \mathcal{P}_{\mathsf{I}_k \cup \{i\}}(\boldsymbol{y})\|_2^2, \tag{1.2.15}$$

and the index set is updated accordingly:

$$\mathsf{I}_{k+1} = \mathsf{I}_k \cup \{i_k\}. \tag{1.2.16}$$

This forward stepwise selection scheme is very much similar to more recent greedy algorithms proposed to solve the sparse coding problem, such as the *orthogonal matching pursuit* method that we will see in Chapter 8. Tools introduced in this book will allow us to clarify conditions under which such a greedy scheme succeeds in finding the optimal subset.

Lasso Regression.
Notice that the main difficulty in solving the subset selection problem (1.2.14) is the ℓ^0 norm constraint: $\|\boldsymbol{x}\|_0 \leq k$. It makes the problem combinatorial hence challenging to optimize via conventional optimization methods.[20] In 1996, Tibishirani

[20] Recently there has been some exciting progress in improving computation efficiency of the variable selection problem (1.2.14) via mixed-integer programming [BKM16].

proposed to relax this constraint with the ℓ^1 norm: $\|\boldsymbol{x}\|_1 \leq k$. This leads to the so called *lasso regression* [Tib96]

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{x}\|_1 \leq k. \tag{1.2.17}$$

A similar formulation, known as *basis pursuit*, was proposed in 1998 by [CDS98] which solves the following program:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{1.2.18}$$

Via convex duality, these problems are equivalent to an unconstrained *convex* optimization:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{1.2.19}$$

with $\lambda > 0$ a tuning parameter.[21] Compared to the greedy stepwise regression (1.2.15), the global nature of lasso and basis pursuit leads to many favorable properties, and arguably, they have become the most popular regression methods since the method of least squares. In this book (Chapter 3), we will develop theoretical tools that allow us to fully understand the role of ℓ^1 norm minimization. These tools will help characterize the precise conditions when the above programs, or their variants, succeed in recovering the correct sparse coefficients. In Chapter 8 we further develop efficient algorithms that can solve these optimization problems in very large scale.

1.2.4 Data Analysis: Principal Component Analysis

In many applications, the observations can be modeled as samples from a multivariate random vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^* \in \mathbb{R}^m$. As the dimension m can be very high and there is often redundancy among these variables $y_1, y_2, \ldots, y_m$, a central problem in statistics or data analysis is to identify possible strong correlation among these variables and remove the redundancy.

Statistical Perspective.
Principal component analysis (PCA) is a classical tool for this purpose. It was first proposed by Pearson in 1901 [Pea01] and later independently by Hotelling in 1933 [Hot33]. The main idea is to project the high-dimensional random vector $\boldsymbol{y}$ onto much fewer directions, represented by a sequence of mutually orthonormal vectors $\{\boldsymbol{u}_i \in \mathbb{R}^m\}_{i=1}^d$, such that the variances are maximized:

$$\boldsymbol{u}_i = \arg\max_{\boldsymbol{u} \in \mathbb{R}^m} \mathrm{Var}(\boldsymbol{u}^*\boldsymbol{y}) \quad \text{subject to} \quad \boldsymbol{u}^*\boldsymbol{u} = 1,\ \boldsymbol{u} \perp \boldsymbol{u}_j\ \forall j < i. \tag{1.2.20}$$

The vectors $\boldsymbol{u}_i \in \mathbb{R}^m, i = 1, \ldots, d$ are called *principal directions* of $\boldsymbol{y}$ and the projections $w_i = \boldsymbol{u}_i^*\boldsymbol{y}$ are called *principal components* of $\boldsymbol{y}$. By construction w_i

[21] In contrast, the classical *ridge regression* considers an ℓ^2 norm regularization on $\boldsymbol{x}$: $\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2$, see Exercise 1.8.

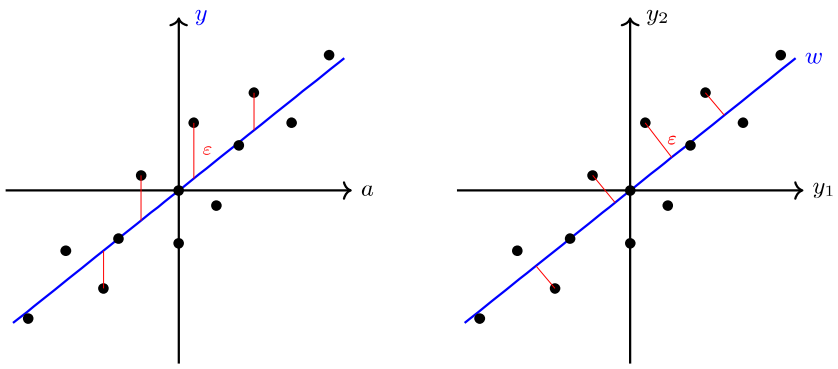


Figure 1.9 Illustration of linear regression on the left versus principal component analysis on the right. Linear regression minimizes the least squares of ε, error in predicting the (one) variable y; Principal component analysis (PCA) minimizes the least squares of ε, distance to the estimated low-dimensional principal component w.

will be uncorrelated and they represent directions in which variables in $\boldsymbol{y}$ are most correlated.

Or equivalently, for a properly chosen d, the original high-dimensional random vector is best-approximated by the $d < m$ principal components as:

$$\boldsymbol{y} = \boldsymbol{u}_1 w_1 + \boldsymbol{u}_2 w_2 + \cdots + \boldsymbol{u}_d w_d + \boldsymbol{\varepsilon} \doteq \boldsymbol{U}\boldsymbol{w} + \boldsymbol{\varepsilon}, \tag{1.2.21}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d] \in \mathbb{R}^{m\times d}$, $\boldsymbol{w} = [w_1, w_2, \ldots, w_d]^* \in \mathbb{R}^d$, and the variance of the residual $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ is minimized:

$$\min \mathbb{E}\big[\|\boldsymbol{y} - \boldsymbol{U}\boldsymbol{w}\|_2^2\big]. \tag{1.2.22}$$

Notice that both linear regression (1.2.10) and PCA minimize least squares of the fitting errors by a low-dimensional linear model. Nevertheless, in regression, one dimension of the data y is preferred and all other variables $a_1, a_2, \ldots, a_n$ are used to predict it, whereas in PCA, all dimensions $y_1, y_2, \ldots, y_n$ are treated equally and the principal components reveal their joint (low-dimensional) structure.[22] Figure 1.9 illustrates the relationship and difference between regression analysis and principal component analysis.

A classical result in statistics states a solution to PCA:

Fact 1.6 (Principal Component Analysis). *For a zero-mean random vector $\boldsymbol{y} \in \mathbb{R}^m$, its first d principal directions $\{\boldsymbol{u}_i \in \mathbb{R}^m\}_{i=1}^d$ are the d orthonormal eigenvectors of the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^*] \in \mathbb{R}^{m\times m}$ associated with the largest d eigenvalues $\{\lambda_i\}_{i=1}^d$. Moreover, $\lambda_i = Var(\boldsymbol{u}_i^*\boldsymbol{y}), i = 1, 2, \ldots, d$.*

To estimate the principal directions $\boldsymbol{U}$ from samples of $\boldsymbol{y}$, we may stack the

[22] In terms of machine learning language, one may say that (linear) regression analysis is a *supervised learning* problem whereas principal component analysis is *unsupervised learning*.

samples as columns of a matrix $\boldsymbol{Y} \doteq [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{m\times n}$. The covariance of $\boldsymbol{y}$ can be estimated by the sample covariance $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{y}} \doteq \frac{1}{n}\boldsymbol{Y}\boldsymbol{Y}^* \in \mathbb{R}^{m\times m}$. So if

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \tag{1.2.23}$$

is the singular value decomposition (SVD) of $\boldsymbol{Y}$, the estimated principal directions of $\boldsymbol{y}$ will be precisely the leading d singular vectors – the first d columns of $\boldsymbol{U}$. For a more detailed characterization of SVD, one may refer to Appendix A.

Low-rank Approximation Perspective.
Singular value decomposition of a matrix was initially developed in the numerical linear algebra literature by Eckart and Young in 1936 [EY36], independent of PCA.[23] The basic idea of singular value decomposition is to approximate a matrix with a superposition of a few rank-1 matrices (usually expressed in a bilinear outer product form):

$$\boldsymbol{Y} = \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^* + \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^* + \cdots + \sigma_d\boldsymbol{u}_d\boldsymbol{v}_d^* + \boldsymbol{E}, \tag{1.2.24}$$

where $\boldsymbol{E}$ is a matrix of small errors or residuals. In fact, the origin of matrix approximation by bilinear forms can be traced back as early as in the work of Beltrami [Bel73] and Jordan [Jor74] in early 1870's.

To see the connection between SVD and PCA, let us consider the problem of approximating a given (sampled data) matrix $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$ by a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ of rank less than d in the least squares sense:

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_2^2 \quad \text{subject to} \quad \operatorname{rank}(\boldsymbol{X}) \leq d. \tag{1.2.25}$$

Fact 1.7 (Low-rank Approximation). *Let $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ be the SVD of the matrix $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$. The optimal solution to the above low-rank matrix approximation problem (1.2.25) is given by*

$$\boldsymbol{X}_\star = \boldsymbol{U}_d\boldsymbol{\Sigma}_d\boldsymbol{V}_d^*, \tag{1.2.26}$$

where $\boldsymbol{U}_d \in \mathbb{R}^{m\times d}$, $\boldsymbol{\Sigma}_d \in \mathbb{R}^{d\times d}$, and $\boldsymbol{V}_d \in \mathbb{R}^{n\times d}$ are submatrices associated to the top d singular vectors and singular values in $\boldsymbol{U}, \boldsymbol{\Sigma}$, and $\boldsymbol{V}$, respectively.

While principal components were initially defined exclusively in a statistical sense [Pea01, Hot33], one can show that the above SVD-based solution gives asymptotically unbiased estimates of the true parameters in the case of Gaussian noise, according to the work of Householder and Young in 1938 [HY38] and then Gabriel in 1978 [Gab78]. A systematic and complete account of statistical properties of PCA can be found in the classical book by Jolliffe in 1986 [Jol86]. Generalization of PCA to models of *multiple* low-dimensional subspaces can be found in a more recent book by Vidal, Ma, and Sastry [VMS16].

Low-rank approximation by least squares fitting (1.2.25) is a special case for which we have a simple tractable solution as stated in the Fact 1.7. This is in general not the case as rank minimization is typically NP-hard. In Chapters 4

[23] So SVD is also known as the Eckart and Young decomposition [HMH00].

and 5 we will study a much broader family of rank minimization problems and characterize conditions under which they can be solved efficiently.

1.3 The Modern Era

As we have seen in previous sections, low-dimensional structures arise ubiquitously in scientific, mathematical, and engineering problems. Many important instances have been long studied in various fields at different times of the history. Many good ideas have been proposed and many effective computational methods have been developed for identifying and exploiting such structures.

1.3.1 From Curses to Blessings of High-Dimensionality

In the classical era, due to limited computing resources, studies[24] had typically focused on formulations which allow closed-form solutions or on methods that are amenable to "hand computation," at least when the dimension is moderate (such as PCA, according to Pearson in 1901 [Pea01]). As a result, methods that rely on heavy numerical methods but conceptually superior formulations have been severely under studied and often ignored or forgotten. For instance, as we have seen in the previous section, for both sparse error correction or sparse regression, ℓ^1 minimization is conceptually the preferred formulation. However, its significant advantages have never been fully brought to light until very recently, thanks to efficient optimization methods and powerful computers. They have helped reveal striking properties and phenomena of ℓ^1 minimization, especially *when the dimension becomes high enough*. Such empirical observations have motivated subsequent theoretical analysis and led to a rather complete and comprehensive theory featured in this book. This renewed understanding of many beneficial geometrical and statistical properties of sparse and many other low-dimensional models in high-dimensional space was celebrated as the "*blessings of dimensionality*" for data science, by Donoho in 2000 [Don00].

Speaking more broadly, in the classical settings, statistical methods and optimization methods were typically applied to data of relatively low dimension or to problems of relatively small scale. Although many profound (and useful) geometric and statistical properties of low-dimensional structure in high-dimensional space were long developed and known to mathematicians [Mat02], such properties had been completely out of reach for computation hence oblivious to the practice of data analysis till very recently. Around the turn of this century, data science had entered into *a new era*, due to the rise of the Internet and social networks (and many other technological advancements mentioned in the Preface). There has been an explosively growing demand to solve ever larger scale problems and compute with ever higher dimensional data. To address such demand, powerful computing platforms and software tools have been developed to

[24] especially studies that aim to reach at implementable algorithms or practical schemes.

solve large-scale optimization problems. Nowadays data scientists and engineers are fully exposed to both good and bad traits of high-dimensional data. Understanding such traits is hence crucial for practitioners and researchers to develop more efficient and reliable algorithms and systems in the future.

As we are entering the new era of *big data computation*, many classical results and methods have become increasingly inadequate for modern data science in one crucial aspect:

lack of precise account of data complexity and computational complexity.

As our previous survey of the fields and history has shown, many theoretic results have provided profound understanding and correct guidelines for approaching the problems of interest. However, many of the classical results do not directly translate to computationally tractable algorithms or solutions. Many of the statistical and information-theoretic concepts and analyses rely on conditions such as the distributions of interest are generic. These concepts[25] often become *ill-defined* when the distributions become degenerate (low-dimensional) or *intractable* to compute when the ambient space is high. Most theoretical guarantees for correctness are *asymptotic* in nature. Straightforward implementation of such methods often leads to algorithms whose worst sample complexity or computational complexity grows exponentially in space or time, hence impractical for high-dimensional problems. Practitioners often find existing models and theory ineffective or even irrelevant to their real-world data and problems, hence resort to brute force, heuristic, and sometimes even *ad hoc* methods instead.[26]

Therefore, to provide practitioners in modern data science truly pertinent engineering principles and methodologies, we need to develop a new theoretical platform that can rigorously characterize the precise working conditions of a proposed method for low-dimensional structures in high-dimensional spaces:

- The theory would reveal the fundamental reasons why many seemingly intractable high-dimensional problems can be solved efficiently without suffering the curses of dimensionality: *because the intrinsic dimension of the data hence solution is very low relative to the dimension of the ambient state space.*
- The platform should also lead to tractable and scalable solutions and algorithms that work in the non-asymptotic regime: *giving precise characterization of the required data complexity*[27] *and computational complexity*[28] *for certain guaranteed accuracy or probability of success.*

Only through the lens of computation can we truly bridge the gap between theory and practice for high-dimensional data analysis and learning, which is the main purpose of this book. To a large extent, the main task of Part I of the book is

[25] including some of the most basic quantities such as likelihood, entropy, and mutual information [CT91].

[26] In recent years, the gap between theory and practice has been significantly enlarged by the empirical success and popularity of deep learning, as we will try to address and resolve in Chapter 16.

[27] say in the number of samples or measurements, random or designed.

[28] say in the number of evaluations of gradients.

to characterize precisely the data complexity; that of Part II is to characterize precisely the computational complexity; and that of Part III is to deal with other non-ideal factors in real data and applications, such as nonlinearity.

1.3.2 Compressive Sensing, Error Correction, and Deep Learning

Compressive Sensing.
In late 1990's, regression methods such as lasso or basis pursuit:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{1.3.1}$$

have been extensively experimented and practiced in statistics for sparse variable selection. Despite the fact that solving the sparsest solution to an underdetermined linear system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ $(m < n)$ is known to be NP-hard in general, overwhelming empirical evidences show that the correct solution can be recovered effectively and efficiently under fairly broad conditions: for randomly chosen matrix $\boldsymbol{A}$, the above ℓ^1 minimization is able to recover a sparse vector $\boldsymbol{x}$ with support up to a constant fraction of n! This was eventually proven to be the case in 2006 by David Donoho [Don06b], Emmanuel Candès, Justin Romberg, and Terence Tao [CRT06b].

In a nutshell, these results suggest that for a k sparse signal $\boldsymbol{x}$ in an n dimensional space $\mathbb{R}^n$, we only need to take approximately $O(k)$ general linear measurements in order to have all its information. In addition, the signal can be correctly and efficiently recovered by minimizing the ℓ^1 norm of $\boldsymbol{x}$ (see Chapter 3). One implication of this result is that if $\boldsymbol{x}$ is a signal that has a high bandwidth but nevertheless sparse in its spectral domain (as shown in Figure 1.3), then one can sample and recover it at a rate much lower than the Nyquist sampling rate [Tro10, ME10], hence the notion of "*compressed sensing*" [Don06a] or "*compressive sampling*" [Can06]. We will give a real application of this new revelation to wide-band wireless communication in Chapter 11.

Error Correction.
As we have seen in the previous section, historically ℓ^1 minimization:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1, \tag{1.3.2}$$

was proposed to correct (sparse) errors $\boldsymbol{e}$ in signal $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$ by Boscovich and later by Logan. The connection between sparse signal recovery and sparse error correction reappeared in the seminal paper "*Decoding by Linear Programming*" by Candès and Tao in 2005 [CT05], in which more general conditions for the sparse error correction problem were derived. Their work has inspired many highly striking applications such as robust face recognition [WYG+09] by the authors, which we will soon see in the next chapter and Chapter 13.

Ever since, the conditions under which ℓ^1 minimization recovers sparse signals or corrects sparse errors were quickly improved and extended to broader family of settings and structures. For instance, both the compressive sensing and

error correction results for sparse vectors were soon generalized to low-rank matrices [RFP10, CLMW11] (which will be studied in Chapters 4–5) and broader families of low-dimensional structures (see Chapter 6). Collectively, these results have started to reshape the foundation of modern data science, especially high-dimensional data analysis, which we will study systematically in this book.

Deep Learning.
The above models are somewhat idealistic in the sense that the relationships between the measurements (output) $\boldsymbol{y}$ and the structured data $\boldsymbol{x}$ are linear and known. In many real-world problems and data, the mapping from $\boldsymbol{x}$ to $\boldsymbol{y}$ can be nonlinear or *unknown* and even the low-dimensional structures of the data $\boldsymbol{x}$ can be *nonlinear*. In this case, one may choose to *compose a sequence of simple maps* to incrementally approximate such a nonlinear and unknown mapping:

$$\begin{cases} \boldsymbol{z}_{\ell+1} & = & \phi(\boldsymbol{A}^\ell \boldsymbol{z}_\ell), \quad \boldsymbol{z}_0 = \boldsymbol{x}, \quad \ell = 0, 1, \ldots, L-1, \\ \boldsymbol{y} & = & \phi(\boldsymbol{C}\boldsymbol{z}_L), \end{cases} \tag{1.3.3}$$

where $\boldsymbol{A}^\ell, \boldsymbol{C}$ are (unknown) matrices, representing linear mappings, and $\phi(\cdot)$ is some basic, typically *sparsity-promoting*, nonlinear activation. The RNN in (1.1.6) is one such example. This type of models are also widely known as *deep networks*. Artificial (deep) neural networks have been proposed since 1940-50s [MP43, Ros58] and extensively studied in the following decades for a variety of problems in pattern recognition, functional approximation, and statistical inference etc. (see [AB99] for a systematic introduction to this classic topic).

Due to the availability of big data and advancement in high-performance computation in the past decade, it has been shown in the seminal work of Krizhevsky, Sutskever, and Hinton [KSH12] in 2012 that this class of models can be learned efficiently and effectively and give useful representations for large-scale real world (visual) data. This has led to tremendous empirical successes of deep networks in a wide variety of applications such as computer vision, speech recognition, and natural languages [LBH15, GBC16]. Despite explosive technological advancements, the practice of deep networks has constantly been haunted by the lack of interpretability and understanding of the so-learned "black box" models, hence lack of rigorous performance guarantees.

Towards the end of the book in Chapter 16, we will see that the role of deep networks, together with their design principles and crucial properties, can be clearly explained, rigorously justified, and even derived as a "white box" from the perspective of learning discriminative low-dimensional representations for high-dimensional data. Therefore, concepts, principles, and methods covered in this book also serve as the foundation for a rigorous and deeper understanding of deep learning, or machine learning in general, in the future.

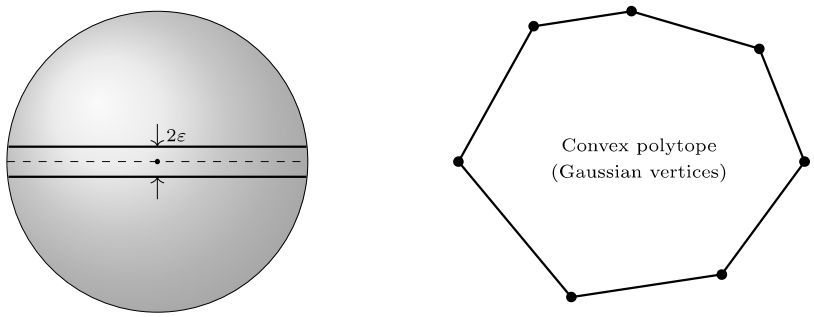


Figure 1.10 Two examples of rather counterintuitive high-dimensional phenomena. **Left:** almost all area of a high-dimensional sphere is concentrated in an ε-strip around its equator, and actually around any great circle! **Right:** random samples of a high-dimensional Gaussian span a highly neighborly convex polytope, which is, however, impossible to illustrate with any 2D polytope.

1.3.3 High-Dimensional Geometry and Non-Asymptotic Statistics

To fully understand the reason why information about low-dimensional structure can be encoded by a nearly minimal number of (linear or nonlinear) measurements, and why it can be accurately and efficiently recovered by tractable methods such as convex and nonconvex optimization, we must resort to fundamental mathematical concepts and tools from high-dimensional geometry and non-asymptotic statistics. These are the tools that have enabled people to characterize the precise conditions under which the proposed methods are expected to work.

High-dimensional geometry and statistics are full of phenomena that are diabolically *counterintuitive*. Our geometric intuition developed in the familiar low (two or three) dimensional space is completely useless for understanding what normally takes place in a high-dimensional space.[29] Actually our intuition may often be exactly opposite to the truth! Although many seemingly paradoxical properties of high-dimensional spaces have been long known to mathematicians and theoretical physicists in certain fields, they have stayed mostly alien to engineers and practitioners till not so long ago. This book aims to introduce some of the properties that are most pertinent to modern data science and engineering.[30] Here as a prelude, we give two examples of high-dimensional phenomena that, as we will see later, have a lot to do with explaining the magic of ℓ^1 minimization.

[29] While most people are rather presumptuous about their geometric intuition, be reminded that it took an Einstein to think correctly about the four-dimensional space and time!

[30] For mathematically oriented readers, we recommend the excellent recent books by Wainwright [Wai19] or Vershynin [Ver18] for a systematic exposition of non-asymptotic high-dimensional statistics and probability.

Measure Concentration on a Sphere [Mat02].
Figure 1.10 left shows an ε-strip around a great circle of a sphere $\mathbb{S}^{n-1}$ in $\mathbb{R}^n$. Here the great circle is the equator with $x_n = 0$. If we want the strip to cover majority, say 99%, of the area of the sphere:

$$\text{Area}\{\boldsymbol{x} \in \mathbb{S}^{n-1} : -\varepsilon \le x_n \le \varepsilon\} = 0.99 \cdot \text{Area}(\mathbb{S}^{n-1}), \tag{1.3.4}$$

our experience with low-dimensional spheres suggests that ε should be large (close to 1). However, simple calculation shows that, as dimension n increases, ε decreases in the order of $n^{-1/2}$. That is, the width of the strip 2ε can be arbitrarily small as n becomes large. Hence almost all area of the sphere concentrates around the equator, as shown in Figure 1.10 left. If this is not strange enough, the area also concentrates on the ε-strip around *any* great circle! A rigorous statement will be given in Theorem 3.6 of Chapter 3. There are many bizarre implications of this fact and we encourage the readers do some brain exercises of their own. We here point out one such implication which has something to do with our later study: if we randomly sample a point on the high-dimensional sphere, say $\boldsymbol{v} \in \mathbb{S}^{n-1}$, then with high probability, this vector will be very close to any of the equators. That is, the inner product of $\boldsymbol{v}$ with each of the standard base vectors (the poles) $\boldsymbol{e}_i \in \mathbb{R}^n$ will be:

$$\langle \boldsymbol{v}, \boldsymbol{e}_i \rangle \approx 0, \quad i = 1, 2, \ldots, n. \tag{1.3.5}$$

In other words, $\boldsymbol{v}$ will be simultaneously nearly orthogonal to all the base vectors $\boldsymbol{e}_i$, or highly *incoherent* to them, in terminology to be used in this book.

Neighborly Polytopes from Gaussian Samples [DT09, DT10].
Consider an m-dimensional Gaussian random vector $\boldsymbol{a} \in \mathbb{R}^m$ whose entries are i.i.d. Gaussian $\mathcal{N}(0, 1/m)$. Now take say $n = 5 \times m$ i.i.d. samples of this random vector and collect them into a matrix: $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{m \times n}$. This gives us a set of n random sample points in $\mathbb{R}^m$. When m is large, say $m = 1,000$, then we have $n = 5,000$ points. Our experience with low (two or three) dimensional Gaussian distributions suggests that many of the samples would be "close to the center" as the probability density is the highest there. However, as we will see later, with high probability, these 5,000 random points span a convex polytope with every point being one of its vertices, as illustrated in Figure 1.10 right. No points would be inside the interior of the polytope at all! If this is not strange enough, try connecting every pair of the vertices with a line segment. Then none of the segments will be in the interior either and each is an edge of the convex polytope! Actually this is also true for any k vertices for k up to certain large number. These vertices will span a k-face of the polytope. Such a polytope is called a *k-neighborly polytope* [DT09]. Neighborly polytopes are a rare breed in low-dimensional spaces[31] but are rather abundant and common in high-dimensional spaces. They are also very easy to construct (say by random

[31] Only the triangle in $\mathbb{R}^2$ and the tetrahedron in $\mathbb{R}^3$.

sampling). As we will see later in Chapter 3 and Chapter 6, it is precisely such properties of high-dimensional polytopes that allow ℓ^1 minimization (1.3.1) to recover any k-sparse vector $\boldsymbol{x}$ from m random measurements $\boldsymbol{Ax}$, with m not so much larger than k.

1.3.4 Scalable Optimization: Convex and Nonconvex

The theoretic developments since early 2000's mentioned above have offered exciting new prospect for practitioners of modern data science. They have provided theoretical guarantees that a very important family of problems, previously deemed as computationally prohibitive (NP-hard) to solve, can become *tractable* under fairly broad conditions. The studies also provide the mathematical tools needed to characterize the precise conditions under which this takes place, hence provide practitioners very pertinent guidelines when such methods are expected to work.

There is one last hurdle though: just because a problem has become tractable, say being reduced to a tractable convex program, it does not mean the existing solutions or algorithms are already *practical* – meaning efficient enough for high-dimensional data and large-scale problems in the real world.

Return of First-order Methods.
Convex optimization is a classic topic and has been well developed in the literature, e.g., see the textbook by Boyd and Vandenberghe [BV04]. For small to medium size problems, algorithms such as *the interior point methods* developed in late 1980's [Wri87, Meg89, MA89a, MA89b] have proven to be extremely efficient and very much become the gold standard for convex programs. However, such algorithms rely on second-order information of the objective function, like the classic Newton's method. The computational and memory cost of computing the second-order derivatives, i.e., the Hessian matrix, can quickly become impractical when the dimension of the problems becomes very large – say the number of variables is in the millions or billions.[32]

This has compelled people to use instead *first-order* optimization methods primarily for high-dimensional large-scale problems. The strive for ever growing scalability has shifted the study of optimization to more careful characterization of the computational complexity of the proposed algorithms, even within the family of first-order methods [Nes03, Nem07]. As a result, the acceleration techniques developed by Nesterov in 1983 [Nes83] have drawn significantly new attention. In fact, in recent years, almost all ideas that could have helped improve the convergence rate and reduce computational cost are carefully reexamined and

[32] In addition to solving sparse coding problems, this is also the case for modern optimization methods for training deep neural networks which normally have millions or billions of parameters to tune. For an example, the latest GPT-3 model from OpenAI for natural language processing has a total of *175 billion parameters* to optimize [B+20] and the latest Switch Transformers model from Google has 1.6 trillion parameters [FZS21].

further refined, leaving almost no stone unturned. Because of this, we feel it is necessary to give a renewed account of optimization methods within the new context of supporting scalable computation: Chapter 8 is for the convex case and Chapter 9 for the nonconvex case.

Return of Nonconvex Formulation and Optimization.
When we face a new class of challenging problems, the most natural approach is trying to reduce them to problems for which we already know a good solution. This is the case with the sparse and low-rank recovery problems. We are fortunate that in many cases they can indeed be reduced to convex programs which admit efficient solutions.

However, first of all, convexification has its theoretical limitations (as we will elaborate on in Section 6.3 of Chapter 6), and many problems we encounter in high-dimensional data analysis do not admit meaningful convex relaxation (as we will study in Chapter 7).

Secondly, models considered in this book (e.g. sparse or low-rank) are idealistic for developing the fundamental concepts and core principles. They typically assume the low-dimensional data structures are *piecewise linear*. As we will see in the application Chapters 12, 15, and 16, real-world data often have *nonlinear* low-dimensional structures instead. Part of the data modeling and analysis process hence entails to learn and undo such nonlinear transforms if we want to apply principles from this book correctly and successfully.

Finally, very often in practice, we can be forced to adopt a nonconvex formulation due to computational constraints or implementation limitations. Let us consider the example of recovering a low-rank matrix, say $\boldsymbol{X} \in \mathbb{R}^{n \times n}$. When the dimension n becomes extremely high, it could become impossible to store the matrix as it is. We may have to represent the matrix as the product of two unknown low-rank factors:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^*, \quad \boldsymbol{U} \in \mathbb{R}^{n \times r}, \boldsymbol{V} \in \mathbb{R}^{n \times r}, \tag{1.3.6}$$

with $r \ll n$, in order to push for better scalability of the implementation. In such cases, we are forced to deal with the nonlinear nature of the representation or nonconvex nature of the program head on [CLC19].

Interestingly enough, such somewhat forced choices lead to very nice surprises [SQW15]. It has been well known that unlike convex optimization, it is very difficult to ensure global optimality or algorithm efficiency for general nonconvex problems. Nevertheless, as we will see in Chapter 7, for many families of problems that we encounter in high-dimensional data analysis, the problems have natural *symmetric* structure. For example, to represent the low-rank matrix $\boldsymbol{X}$ by two factors as in (1.3.6), there is an equivalent class of solutions: $\boldsymbol{U}\boldsymbol{V}^* = \boldsymbol{U}\boldsymbol{R}\boldsymbol{R}^*\boldsymbol{V}^*$ for any orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{r \times r}$ in the orthogonal group $\mathsf{O}(r)$. As a result, the associated nonconvex objective functions have extremely good local and global geometric properties. These properties make them amenable to extremely *simple and efficient* algorithms, such as gradient descent

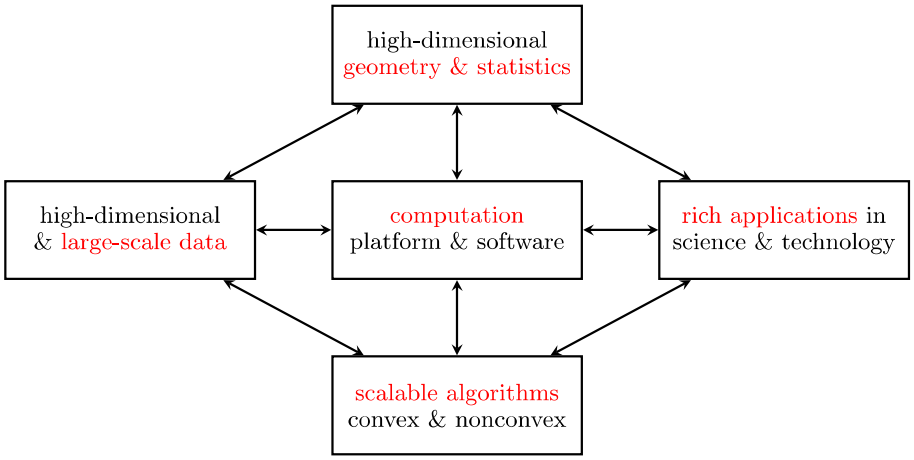


Figure 1.11 A Perfect Storm for revolutionary knowledge and technology advancement: confluence of the availability of massive data, powerful computational platforms, high-dimensional geometry and statistics, scalable optimization algorithms, and rich applications in science and technology.

and its variants, detailed in Chapter 9. Under very benign conditions, these algorithms actually can converge to *the globally optimal solution* with high efficiency and accuracy [SQW15, MWCC18], quite atypical of nonconvex problems!

Although this is still a rather active research area, scalable nonconvex optimization algorithms used to solve such problems have been well developed for long and their computational complexities have been precisely characterized recently. So in Chapter 9 we give a rather complete and coherent survey of scalable nonconvex optimization methods as well as guarantees they offer in terms of the type of critical points converged to and the associated computational complexity. These algorithms are not only useful in the context of recovering low-dimensional structures but also essential to many modern large-scale machine learning problems such as constructing and training deep neural networks, which we will elaborate on more in the final Chapter 16.

1.3.5 A Perfect Storm

According to Wikipedia, "*a perfect storm is an event in which a rare combination of circumstances drastically aggravates the event.*" Then what have taken place in data science and technology in the last couple of decades can be precisely characterized as a "perfect storm," a good one that is. An unexpected combination of several factors has almost simultaneously advanced and contributed to a *revolution* in data science and technology: the massive high-dimensional data, rich scientific or technological applications, and powerful computational and data platforms (such as the cloud technology) have set an ideal stage for fundamental

knowledge in high-dimensional geometry and statistics to be efficiently realized and exploited through scalable optimization algorithms. The confluence of these factors, as illustrated in Figure 1.11, has truly brought us into a new era of scientific discovery and engineering marvel.

1.4 Exercises

1.1 (Nyquist-Shannon Sampling Theorem). *Prove the Fact 1.3.*

1.2 (Conditional Independence of Gaussian Variables). *Prove the Fact 1.4 for the case of a joint Gaussian vector with three variables* $\boldsymbol{x} = [x_1, x_2, x_3]^*$ *in which* x_1 *and* x_2 *are conditionally independent given* x_3.

1.3. *Given a jointly Gaussian random vector* $\boldsymbol{x} = (\boldsymbol{x}_o, \boldsymbol{x}_h)$, *prove that the structure of the covariance matrix of the observable part* $\boldsymbol{x}_o$ *has the structure given in* (1.1.15).

1.4. *Derive a closed-form solution to the method of least squares* (1.2.7).

1.5 (Maximum Likelihood Estimate with Laplace or Gaussian Noise). *Recall that the probability density function a Laplace distribution* $\mathcal{L}(\mu, b)$ *is*

$$p(x) = \frac{1}{2b} \exp\Big(- \frac{|x - \mu|}{b} \Big),$$

and the Gaussian, or normal, distribution, $\mathcal{N}(\mu, \sigma)$ *is*

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{(x - \mu)^2}{2\sigma^2} \Big).$$

Given a measurement model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon}$, *consider the following two types of noise:*

1 Entries of $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m]^*$ *are i.i.d. zero-mean Laplace.*
2 Entries of $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_m]^*$ *are i.i.d. zero-mean Gaussian.*

Derive the log maximum likelihood function for estimating $\boldsymbol{x}$ *under these two noise models. Discuss their relationships to the* ℓ^1 *minimization and* ℓ^2 *minimization, respectively.*

1.6. *Prove the Fact 1.6 for the case* $d = 1$. *That is, the principal direction of a random vector* $\boldsymbol{y}$ *is the eigenvector associated with the largest eigenvalue of its covariance matrix* $\boldsymbol{\Sigma}_{\boldsymbol{y}}$. *Furthermore, prove the Theorem A.29 in Appendix A.*

1.7. *Prove the Fact 1.7.*

1.8 (Ridge Regression). *To solve a system of linear equations* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, *especially when the system is ill-posed (say under-determined) or with (Gaussian) noise*

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\varepsilon}$, *one popular way to estimate* $\boldsymbol{x}$ *is to consider the so-called* ridge regression*:*

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2, \tag{1.4.1}$$

for some $\lambda > 0$.[33] *This is also known as* Tikhonov regularization.[34]

1 *Show that the optimal solution* $\boldsymbol{x}_\star$ *to the above optimization is given by:*

$$\boldsymbol{x}_\star = (\boldsymbol{A}^*\boldsymbol{A} + \lambda \boldsymbol{I})^{-1}\boldsymbol{A}^*\boldsymbol{y}, \tag{1.4.2}$$

given that the matrix $\boldsymbol{A}^*\boldsymbol{A} + \lambda \boldsymbol{I}$ *is invertible.*

2 *Discuss the conditions on the matrix* $\boldsymbol{A}$ *and* λ *so that the matrix* $\boldsymbol{A}^*\boldsymbol{A} + \lambda \boldsymbol{I}$ *is guaranteed to be invertible.*

Ridge regression is arguably the most widely studied and used form of regression in the classic statistical literature [HTF09]. There are many good properties of this type of regressions, related to important methods such as the Wiener filter in signal processing. The reader may refer to the recent book [FLZZ20] for a more detailed study of ridge regression and many variants.

[33] This can be viewed as a Lagrangian formulation of the constrained optimization considered by Theorem A.25 in Appendix A.

[34] Strictly speaking, Tikhonov regularization may consider a more general class of regularization of the form $\|\boldsymbol{\Lambda}\boldsymbol{x}\|_2^2$ for some properly chosen positive definite matrix $\boldsymbol{\Lambda}$.

Part I

Principles of Low-Dimensional Models

2 Sparse Signal Models

"It is quite probable that our mathematical insights and understandings are often used to achieve things that could in principle also be achieved computationally – but where blind computation without much insight may turn out to be so inefficient that it is unworkable."

– Roger Penrose, *Shadows of the Mind*

This book is about modeling and exploiting *simple structure* in signals, images, and data. In this chapter, we take our first steps in this direction. We study a class of models known as *sparse models*, in which the signal of interest is a superposition of a few basic signals (called "atoms") selected from a large "dictionary." This basic model arises in a surprisingly large number of applications. It also illustrates fundamental tradeoffs in modeling and computation that will recur throughout the book.

2.1 Applications of Sparse Signal Modeling

Why do we need signal models at all? We give a pragmatic answer. Many problems arising in modern signal processing and data analysis are intrinsically *ill-posed*. Often, the number of unknowns vastly exceeds the number of observations. In this situation, prior knowledge is absolutely essential to solving the problem correctly.

To describe this phenomenon mathematically, consider the simple equation

$$\underset{\text{observation}}{\boldsymbol{y}} = \boldsymbol{A} \underset{\text{unknown}}{\boldsymbol{x}}. \tag{2.1.1}$$

Here, $\boldsymbol{y} \in \mathbb{R}^m$ is our observation, while $\boldsymbol{x} \in \mathbb{R}^n$ is unknown. The matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ represents the data generation process: the observed data $\boldsymbol{y}$ is a linear function of the unknown (or hidden) signal $\boldsymbol{x}$. This is a simple model; however, we will see that it is rich enough to bear on a vast array of practical applications.

Recovering the unknown $\boldsymbol{x}$ from observation $\boldsymbol{y}$ may appear trivial: we simply have to solve a linear system of equations! However, many practical applications raise a substantial challenge: the number of observations, m, can be significantly smaller than the number of elements n in the signal to be recovered. From linear

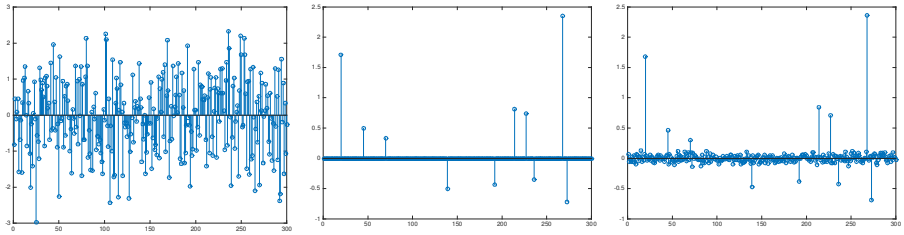


Figure 2.1 Dense vs. Sparse Vectors. Left: a generic *dense* vector $\boldsymbol{x} \in \mathbb{R}^n$, with entries being independent standard normal random variables. **Center:** a *sparse* vector, with only a few nonzero entries. **Right:** a *compressible* vector, with only a few significant entries.

algebra,[1] we know that when $m < n$, the system of equations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ does not necessarily have any solution, but if it has any solution at all, then the solution space has at least dimension $n - m$. Hence, either there is no solution, or there are infinitely many solutions. Only one of them is the one we wish to recover! To make progress, we need to leverage some additional properties of the target solution.

Sparsity is one such property, which has strong implications on our ability to solve underdetermined systems. A vector $\boldsymbol{x} \in \mathbb{R}^n$ is considered *sparse* if only a few of its elements are nonzero. Figure 2.1 (center) shows an example of such a vector. Some form of sparsity arises naturally in almost every type of high-dimensional signal or data that we encounter in practical applications. Below, we illustrate with a few representative examples.

2.1.1 An Example from Medical Imaging

Figure 2.2 shows a *magnetic resonance* (MR) image of the brain. This is a digital image $I \in \mathbb{R}^{N \times N}$. Each entry $I(\boldsymbol{v})$ (here, $\boldsymbol{v} \in \mathbb{R}^2$) corresponds to the density of protons at a given spatial location inside the brain. This essentially indicates where water is in the brain, and can reveal many biological structures that are important for disease diagnosis and monitoring. To caricature the MRI problem a bit, our goal is to estimate I, without opening up the brain! This is possible, if we subject the patient to a large, spatially and temporally varying magnetic field. The magnetic field causes the protons to oscillate at a frequency that depends on their locations and energy states. Each proton essentially acts as its own radio transmitter, and in aggregate they create a signal we can measure.

As we will see from a more detailed derivation of the physical model for MRI

[1] Appendix A provides a detailed review of linear algebra and matrix analysis. In particular, Appendix A.6 reviews the existence and uniqueness of solutions to linear systems, which we use here to motivate our study of sparse approximation.

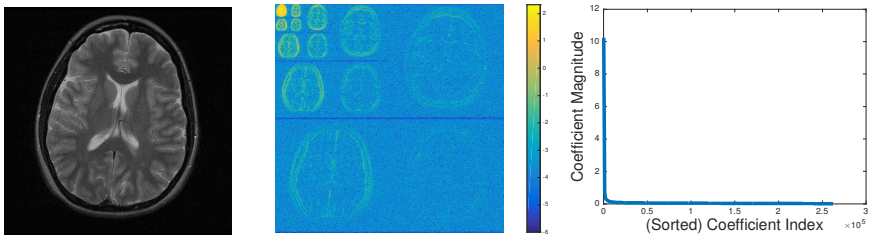


image $I(\boldsymbol{v})$ **wavelet coefficients $\boldsymbol{x}$: $I = \boldsymbol{\Psi}[\boldsymbol{x}]$.**

Figure 2.2 A Magnetic Resonance Image. Left: target image of a human brain. **Right:** coefficients in the wavelet decomposition $I = \sum_i \boldsymbol{\psi}_i x_i$, and their magnitudes, sorted in descending order. The large wavelet coefficients concentrate around sharp edges in the image; wavelet coefficients corresponding to smooth regions are much smaller. The wavelet coefficients are highly compressible: their magnitude decays rapidly. Image reprinted with permission from Michael Lustig [Lus13].

in Chapter 10, it turns out that the signal we observe is simply a sample of the two-dimensional Fourier transform of I:

$$y = \int_{\boldsymbol{v}} I(\boldsymbol{v}) \exp(-\mathsf{i}\, 2\pi\, \boldsymbol{u}^* \boldsymbol{v})\, d\boldsymbol{v}. \tag{2.1.2}$$

Here, $\mathsf{i} = \sqrt{-1}$ is the imaginary unit, and $(\cdot)^*$ denotes the (complex conjugate) transpose of a vector. The two-dimensional frequency vector $\boldsymbol{u}^* = [u_1, u_2] \in \mathbb{R}^2$ depends on how the magnetic field we applied varies over space. Here, letting $\mathcal{F}$ denote the 2D Fourier transform, the above expression is

$$y = \mathcal{F}[I](\boldsymbol{u}). \tag{2.1.3}$$

By changing the applied magnetic field, we can vary $\boldsymbol{u}$, and collect m samples of the Fourier transform, corresponding to different applied magnetic fields, parameterized by $\mathsf{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\}$. We can concatenate all of our observations into a vector $\boldsymbol{y} \in \mathbb{C}^m$, given by

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \mathcal{F}[I](\boldsymbol{u}_1) \\ \vdots \\ \mathcal{F}[I](\boldsymbol{u}_m) \end{bmatrix} \doteq \mathcal{F}_{\mathsf{U}}[I]. \tag{2.1.4}$$

Here, $\mathcal{F}_{\mathsf{U}}$ is simply the operator that obtains the Fourier samples of I, indexed by U. If you imagine the Fourier transform as acting by matrix multiplication, $\mathcal{F}_{\mathsf{U}}$ is simply the matrix we get if we discard all the rows of $\mathcal{F}$ that are not indexed by U.

One very basic property of the integral (2.1.2), and hence of the operator $\mathcal{F}_{\mathsf{U}}$, is that it is *linear* in its input I. This means that for any pair of inputs I and J and complex scalars α, β,

$$\mathcal{F}_{\mathsf{U}}[\alpha I + \beta J] = \alpha \mathcal{F}_{\mathsf{U}}[I] + \beta \mathcal{F}_{\mathsf{U}}[J]. \tag{2.1.5}$$

Because $\mathcal{F}_{\mathsf{U}}$ is a linear operator, the problem of finding I from $\boldsymbol{y}$ using the

observation equation (2.1.4) "just" consists of solving a large linear system of equations.

There is a substantial catch though. In this system of equations, there are typically far more unknowns (here $n = N^2$) than observations m. This is necessary: it is generally too time and energy intensive to simply measure all N^2 Fourier coefficients. This is even more pressing of a concern in *dynamic MRI*, where the object being imaged is changing over time, and so acquisition needs to be time-efficient. So, in general, we need m to be as small as is just necessary to guarantee accurate reconstruction – and certainly significantly smaller than n.

This leaves us with a seemingly impossible situation: we have n unknowns and $m \ll n$ equations. Unless we can make some additional assumptions on the structure of I, the problem is ill-posed. Fortunately, real signals are not completely unstructured.[2] Figure 2.2 (right) shows a *wavelet transform* of I. The wavelet transform expresses I as a superposition of a collection of basis functions $\boldsymbol{\Psi} = \{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{N^2}\}$:

$$\underset{\text{image}}{I} = \sum_{i=1}^{N^2} \underset{i\text{-th basis signal}}{\boldsymbol{\psi}_i} \times \underset{i\text{-th coefficient}}{x_i}. \tag{2.1.6}$$

Here, $x_1, \ldots, x_{N^2} \in \mathbb{R}$ are coefficients of the image I with respect to the basis $\boldsymbol{\Psi}$. The entries in Figure 2.2 (right) are the magnitudes $|x_i|$ for the N^2 coefficients x_i. The important point is that many of these coefficients are extremely small. If let $\mathsf{J} = \{i_1, \ldots, i_k\}$ denote the k largest coefficients, we can approximate I as

$$\underset{\text{target image}}{I} \approx \underset{\text{superposition of } k \text{ basis functions}}{\tilde{I}_k = \sum_{i \in \mathsf{J}} \boldsymbol{\psi}_i x_i}. \tag{2.1.7}$$

Figure 2.3 visualizes the reconstruction and reconstruction error $I - \tilde{I}_k$. It seems that even if we retain only a relatively small fraction of the coefficients, we still obtain an accurate approximation, and most of what remains is noise. This suggests that the sequence $\boldsymbol{x}$ is *compressible* – it is very close to a sparse vector.

In order to recover I, we can first try to reconstruct the sparse vector $\boldsymbol{x}$, using

[2] Indeed, we can construct a "generic" element I_{generic} of $\mathbb{R}^{N \times N}$, by choosing its entries at random – say from a standard Gaussian distribution $\mathcal{N}(0, 1)$. With very high probability, I_{generic} will simply look like noise. The target magnetic resonance image in Figure 2.2 certainly does not look like noise!

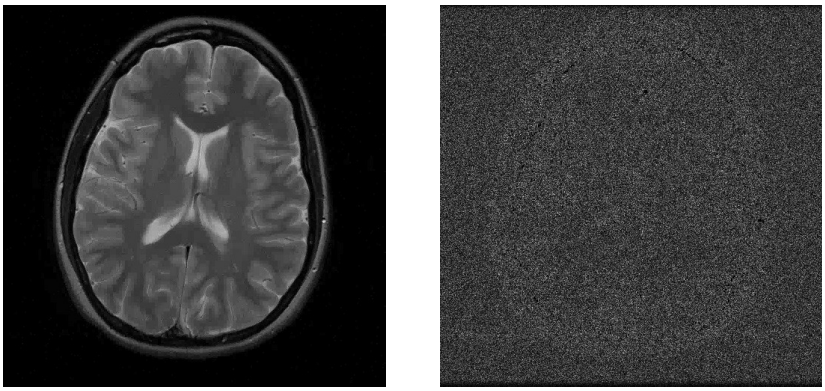


Figure 2.3 Wavelet Approximation $\tilde{I}$ to I and Approximation Error. Left: approximation to the image in Figure 2.2 using the most significant 7% of the wavelet coefficients. **Right:** approximation error $|I - \tilde{I}|$. The error contains mostly noise, suggesting that most of the important structure of the image is captured in the wavelet approximation $\tilde{I}$.

the observation equation

$$\begin{aligned}
\underset{\text{observed Fourier coefficients}}{\boldsymbol{y}} &= \mathcal{F}_{\mathsf{U}}[I], \\
&= \mathcal{F}_{\mathsf{U}}\Big[\, \boldsymbol{\psi}_1 x_1 + \cdots + \boldsymbol{\psi}_{N^2} x_{N^2} \,\Big], \\
&= \mathcal{F}_{\mathsf{U}}[\boldsymbol{\psi}_1]\, x_1 \;+\; \cdots \;+\; \mathcal{F}_{\mathsf{U}}[\boldsymbol{\psi}_{N^2}]\, x_{N^2}, \\
&= \Big[\mathcal{F}_{\mathsf{U}}[\boldsymbol{\psi}_1] \mid \cdots \mid \mathcal{F}_{\mathsf{U}}[\boldsymbol{\psi}_{N^2}]\Big]\, \boldsymbol{x}, \\
&\quad \text{matrix } \boldsymbol{A} \in \mathbb{R}^{m \times N^2},\ m \ll N^2. \\
&= \boldsymbol{A}\boldsymbol{x}. \qquad (2.1.8)
\end{aligned}$$

After these manipulations, we end up with a system of equations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. The vector $\boldsymbol{x}$ contains the coefficients of the target image I in the wavelet basis. The i-th column of the matrix $\boldsymbol{A}$ contains a subset U of the Fourier coefficients of the i-th basis signal $\boldsymbol{\psi}_i$. To reconstruct I, we can look for a solution $\hat{\boldsymbol{x}}$ to this system, and then set

$$\hat{I} \;=\; \sum_{i=1}^{N^2} \boldsymbol{\psi}_i \hat{x}_i. \qquad (2.1.9)$$

Because $\boldsymbol{x}$ has N^2 entries, but we only have $m \ll N^2$ observations, the system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is underdetermined. Nevertheless, because the wavelet coefficients of I are (nearly) sparse – say, only its k largest coefficients are significant and others are negligible, the desired solution $\boldsymbol{x}$ to this system is sparse. To reconstruct I we need to find a sparse solution to an underdetermined system! In Chapter 10, we will illustrate how to actually apply such a "compressive sampling" scheme to real MRI images under more realistic conditions.

2.1.2 An Example from Image Processing

In the previous example, we used the fact that the image I had a good sparse approximation in terms of a "dictionary" of basic elements $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{N^2}$:

$$I \approx \sum_{i \in \mathsf{J}} \boldsymbol{\psi}_i x_i \quad = \underbrace{\boldsymbol{\Psi}}_{N^2 \times N^2 \text{ matrix}} \underbrace{\boldsymbol{x}}_{\text{sparse vector}}, \tag{2.1.10}$$

where $x_i = 0$ for $i \notin \mathsf{J}$, and $k = |\mathsf{J}| \ll N^2$. Expressions of this form play a central role in lossy data compression. Image compression standards such as JPEG [Wal91] and JPEG 2000 [TM01] leverage sparse approximations (in the discrete cosine transform (DCT) [ANR74] and wavelet bases [VK95], respectively). Generally speaking, the sparser the representation is, the more an input image can be compressed. However, sparse representations of images are not *just* useful for compression: they can be used for solving inverse problems, in which we try to reconstruct I from noisy, corrupted or incomplete observations. We already saw an example of this in the previous section, in which we used sparsity in the wavelet domain to reconstruct MR images. To facilitate all of these tasks, we can seek representations of I that are as sparse as possible, by replacing $\boldsymbol{\Psi}$ with more general dictionaries $\boldsymbol{A}$. For example, we might consider *overcomplete dictionaries* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $n > m$, which consist of several orthonormal bases (e.g., DCT and wavelets together). The idea is that each individual representation may capture a particular type of signal well – say, DCT for smooth variations and wavelets for signals with sharp edges. Together, they can represent a broader class of signals.

An even more aggressive idea is to simply *learn* $\boldsymbol{A}$ from data, rather than designing it by hand. Conceptually this leads to an even more challenging problem, known as *dictionary learning*, which we will study later in Chapter 7. This approach tends to produce better sparsity-accuracy tradeoffs for representing images I, and is also useful for a wealth of other problems, including denoising, inpainting, and super-resolution that involve reconstructing I from incomplete or corrupted observations. Each of these problems leads to an underdetermined linear system of equations; the goal is to use the prior knowledge that the target signal I has a compact representation in some dictionary $\boldsymbol{A}$ to make the problem well-posed. Figure 2.4 shows an example of this for the problem of color image denoising, from [MES08]. We observe a noisy image

$$I_{\text{noisy}} = \underbrace{I_{\text{clean}}}_{\text{target image}} + \underbrace{\boldsymbol{z}}_{\text{noise}}. \tag{2.1.11}$$

We assume[3] that patches of the clean image have an accurate sparse approximation in some dictionary $\boldsymbol{A}$: if we break I_{clean} into patches $\boldsymbol{y}_{1\,\text{clean}}, \ldots, \boldsymbol{y}_{p\,\text{clean}}$

$$\boldsymbol{y}_{i\,\text{clean}} \approx \underbrace{\boldsymbol{A}}_{\text{patch dictionary}} \times \underbrace{\boldsymbol{x}_i}_{\text{sparse coefficient vector}}. \tag{2.1.12}$$

[3] Of course, this assumption needs to be justified! See Exercise 2.16 and the notes and references to this chapter. We will also have ample examples in later chapters when we introduce methods to learn sparsifying dictionaries for real images.

Figure 2.4 Image Denoising by Sparse Approximation. Left: A noisy input image. The image is broken into patches $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p$. A dictionary $\boldsymbol{A} = [\boldsymbol{a}_1 \mid \cdots \mid \boldsymbol{a}_n]$ is learned such that each input patch can be approximated as $\boldsymbol{y}_i \approx \boldsymbol{A}\boldsymbol{x}_i$, with $\boldsymbol{x}_i$ sparse. **Right:** dictionary patches $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. **Center:** denoised image, reconstructed from the approximations $\hat{\boldsymbol{y}}_i = \boldsymbol{A}\boldsymbol{x}_i$. Figures from [MES08, WMM+10]. Image reprinted with permission from Julien Mairal.

In denoising, we do not actually observe $\boldsymbol{y}_{i\text{clean}}$. Rather, we observe *noisy* patches

$$\boldsymbol{y}_i = \boldsymbol{y}_{i\text{clean}} + \boldsymbol{z}_i = \boldsymbol{A} \times \boldsymbol{x}_i + \boldsymbol{z}_i, \quad i = 1, \ldots, p.$$

Based on these patches $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_p$, we learn a dictionary $\hat{\boldsymbol{A}}$ such that

$$\underset{i\text{-th image patch}}{\boldsymbol{y}_i} \approx \underset{\text{learned dictionary}}{\hat{\boldsymbol{A}}} \times \underset{\text{sparse coefficient vector}}{\hat{\boldsymbol{x}}_i} = \underset{\text{denoised patch}}{\hat{\boldsymbol{y}}_i}.$$

The dictionary $\hat{\boldsymbol{A}}$ and sparse coefficients $\hat{\boldsymbol{x}}_i$ can be learned by solving a nonconvex optimization problem, which attempts to strike an optimal balance between the sparsity of the coefficients $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_p$ and the accuracy of the approximation $\boldsymbol{y}_i \approx \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_i$. More detail will be given in Chapter 7. We take $\hat{\boldsymbol{y}}_i = \hat{\boldsymbol{A}}\hat{\boldsymbol{x}}_i$ as an estimate of $\boldsymbol{y}_{i\text{clean}}$.

Figure 2.4 (left) shows the noisy input image; Figure 2.4 (center) shows a denoised image constructed from $\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_p$. Figure 2.4 (right) shows the dictionary $\hat{\boldsymbol{A}}$ learned from the noisy patches. Although the sparse dictionary prior is relatively simple, and does not capture all of the global geometric structure of the image, it leads to surprisingly good performance on many low-level image processing tasks including image super-resolution [YWHM10] or restoration [MES08]. We discuss modeling and computational aspects of dictionary learning in detail in Chapters 7 and 9. For now, the key point is that the problem of reconstructing the clean image from noisy patches again leads us to an underdetermined linear system of equations, $\boldsymbol{y}_i \approx \boldsymbol{A}\boldsymbol{x}_i$.

2.1.3 An Example from Face Recognition

Sparsity also arises naturally in problems in which we wish to perform reliable inference from unreliable measurements. For example, due to sensor errors or

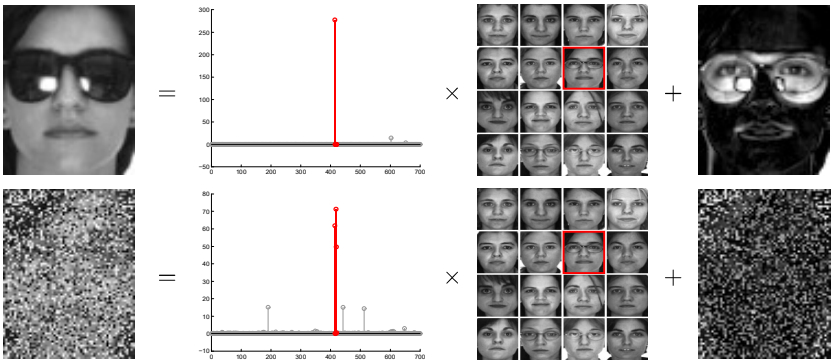


Figure 2.5 Face Recognition via Sparse Representation. Top: input face image $\boldsymbol{y}$ is wearing sunglasses; **Bottom:** input face image $\boldsymbol{y}$ is with 50% pixels arbitrarily corrupted. Each test image $\boldsymbol{y}$ is approximated as a sparse combination $\boldsymbol{Bx}$ of the training images, plus a sparse error $\boldsymbol{e}$ due to occlusion. In this example, red coefficients correspond to images of the correct subject. Results and Figures from [WYG+09].

malicious tampering, a vector-valued observation $\boldsymbol{y} \in \mathbb{R}^m$ might be grossly corrupted in a few of its entries:

$$\underset{\text{observation}}{\boldsymbol{y}} = \underset{\text{clean data}}{\boldsymbol{y}_o} + \underset{\text{sparse error}}{\boldsymbol{e}}. \tag{2.1.13}$$

We illustrate this more concretely using an example from automatic face recognition. Imagine that we have a database consisting of a number of subjects. For each subject i, we collect grayscale training images $I_{i,1}, \ldots, I_{i,n_i} \in \mathbb{R}^{W \times H}$, and vectorize them to form a base matrix $\boldsymbol{B}_i \in \mathbb{R}^{m \times n_i}$, with $m = W \times H$. We can further concatenate these matrices to form a large training "dictionary"

$$\boldsymbol{B} = \underset{\text{all training images}}{[\boldsymbol{B}_1 \mid \boldsymbol{B}_2 \mid \cdots \mid \boldsymbol{B}_n]} \in \mathbb{R}^{m \times n}, \quad n = \sum_i n_i. \tag{2.1.14}$$

Suppose our system is confronted with a new image $\boldsymbol{y} \in \mathbb{R}^m$, taken under some new lighting condition, and possibly occluded – see Figure 2.5. For now, we can assume that the input $\boldsymbol{y}$ is well-aligned to the training images (i.e., the faces occur at the same position in the training and test images).[4] There is a beautiful physical argument [BJ03] that shows that in an average case sense, images of "nice" objects taken under varying lighting conditions lie very close to low-dimensional linear subspaces of the high-dimensional image space $\mathbb{R}^m$.[5] This suggests that if we have seen enough training examples, we can approximate the input sample $\boldsymbol{y}$ as a linear combination of the training samples from the same

[4] Relaxing this assumption is essential to building systems that work with unconstrained input images. We will talk about how to relax this assumption in Chapter 13.

[5] We will give a more detailed justification for this fact in Chapter 14 based on a simplified physical model.

class:

$$\underset{\text{observed image}}{\boldsymbol{y}} \approx \underset{\text{linear combination of training images from } i_\star\text{-th class}}{\boldsymbol{B}_{i_\star}\boldsymbol{x}_{i_\star}}. \tag{2.1.15}$$

Unfortunately, in practice, this equation is violated in at least two ways: first, we don't know the true identity $i_\star$ ahead of time. Second, nuisance factors such as occlusion cause the equation to be badly violated on a portion of the image pixels (those that are occluded). For the first problem, we note that we can *still* write down an expression for $\boldsymbol{y}$ as a linear combination of elements of the database $\boldsymbol{B}$ as a whole: $\boldsymbol{y} \approx \boldsymbol{B}\boldsymbol{x}$. To deal with occlusion, we need to introduce an additional term $\boldsymbol{e}$, giving

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{e}. \tag{2.1.16}$$

Because the errors caused by occlusion are large in magnitude, this error $\boldsymbol{e}$ cannot simply be ignored or treated with techniques designed for small noise. Unfortunately, this means that the system is underdetermined: we have m equations, but $m+n$ unknowns $\bar{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{e})$. Writing $\boldsymbol{A} = [\boldsymbol{B} \mid \boldsymbol{I}]$, we again have a very large underdetermined system

$$\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}}. \tag{2.1.17}$$

If we did not have prior information about $\bar{\boldsymbol{x}}$, there would be no hope of recovering it from this observation. Fortunately, both $\boldsymbol{x}$ and $\boldsymbol{e}$ are very structured. The nonzero values of $\boldsymbol{x}$ should be concentrated only on those images of the true subject, $i_\star$, and so it should be a *sparse vector*. The nonzero values of the error $\boldsymbol{e}$ should be concentrated only on those pixels that are occluded or corrupted, and so it should also be sparse.[6]

Figure 2.5 shows two examples of a sparse solution to this system of equations for a given input image $\boldsymbol{y}$. Notice that the coefficients in the estimated $\hat{\boldsymbol{x}}$ are concentrated on images of the correct subject (red) and that the error indeed corresponds to the physical occlusion. The setting we have described so far is somewhat idealized – we will discuss both the modeling and system building aspects of this problem in the application section of this book, see Chapter 13. For our purposes here, it is enough to note that *if* we can somehow obtain a sparse $(\boldsymbol{x}, \boldsymbol{e})$, it should suffice to identify the subject, despite nuisances such as illumination, occlusion, and corruption.

[6] Of course, the goal is to correct as many errors as possible. One of the surprises of high dimensions is it is indeed possible to correct large fractions of errors using simple, efficient algorithms. Understanding precisely how many errors we can correct (and how dense the vector $\bar{\boldsymbol{x}}$ can be before our methods break down) will be a major theoretical thrust of this book. In Chapter 13, we will give a more precise characterization about how large a fraction of errors can be corrected for a system of linear equations, similar to those that arise in the robust face recognition setting.

2.2 Recovering a Sparse Solution

Suppose, as in the above examples, that we know the ground truth signal $\boldsymbol{x}_o$ is sparse. How powerful is this knowledge? Can it render ill-posed problems such as MR image acquisition or occluded face recognition well-posed? To answer these questions, we need a formal notion of sparsity. In the next two subsections, we begin by introducing the concept of a norm of a vector, which generalizes the concept of *length*. We then introduce an "ℓ^0 norm", which counts the number of nonzero entries in a vector, a basic measure of how dense (or sparse) that vector is.

2.2.1 Norms on Vector Spaces

A *vector space* $\mathbb{V}$ consists of a collection of elements (vectors), field such as the real numbers $\mathbb{R}$ or complex numbers $\mathbb{C}$ (scalars) and operations (adding vectors and multiplying vectors with scalars) that work in ways that conform to our intuitions from $\mathbb{R}^3$. Appendix A reviews the formal definition of a vector space, and gives examples. In the above application examples, our signals of interest consisted of collections of real or complex numbers – e.g., in MR imaging, the target image I was an element of $\mathbb{R}^{N\times N}$. We can view $\mathbb{R}^{N\times N}$ as a vector space, with scalar field $\mathbb{R}$ (written $\mathbb{V} = (\mathbb{R}^{N\times N}, \mathbb{R})$). In the other examples as well, the signals of interest reside in vector spaces.

A *norm* on a vector space $\mathbb{V}$ gives *a way of measuring lengths of vectors, that conforms in important ways to our intuition* from lengths in $\mathbb{R}^3$. Formally:

DEFINITION 2.1 (Norm). *A* norm *on a vector space* $\mathbb{V}$ *over* $\mathbb{R}$ *is a function* $\|\cdot\| : \mathbb{V} \to \mathbb{R}$ *that is*

1 *nonnegatively homogeneous:* $\|\alpha\boldsymbol{x}\| = |\alpha|\|\boldsymbol{x}\|$ *for all vectors* $\boldsymbol{x} \in \mathbb{V}$*, scalars* $\alpha \in \mathbb{R}$*,*
2 *positive definite:* $\|\boldsymbol{x}\| \geq 0$*, and* $\|\boldsymbol{x}\| = 0$ *if and only if* $\boldsymbol{x} = \boldsymbol{0}$*,*
3 *subadditive:* $\|\cdot\|$ *satisfies the triangle inequality* $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ *for all* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{V}$*.*

For our purposes, the most important family of norms are the ℓ^p norms (read "ell p norm"). We will use norms from this family to derive practical algorithms for finding sparse solutions to linear systems of equations, and for studying their properties. If we take $\mathbb{V} = (\mathbb{R}^n, \mathbb{R})$, and $p \in (0, \infty)$, we can write

$$\|\boldsymbol{x}\|_p \doteq \left(\sum_i |x_i|^p\right)^{1/p}. \tag{2.2.1}$$

The function $\|\boldsymbol{x}\|_p$ is a norm for any $p \geq 1$.[7] The most familiar example is the

[7] We leave as an exercise for the reader to show that for $0 < p < 1$, $\|\boldsymbol{x}\|_p$ is not a norm in the strict sense of Definition 2.1.

ℓ^2 norm or "Euclidean norm"

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_i |x_i|^2} = \sqrt{\boldsymbol{x}^*\boldsymbol{x}},$$

which coincides with our usual way of measuring length. Two other cases are of almost equal importance: $p = 1$, and $p \to \infty$. Setting $p = 1$ in (2.2.1), we obtain

$$\|\boldsymbol{x}\|_1 = \sum_i |x_i|, \tag{2.2.2}$$

which will play a very large role in this book.[8] Finally, as p becomes larger, the expression in (2.2.1) accentuates large $|x_i|$. As $p \to \infty$, $\|\boldsymbol{x}\|_p \to \max_i |x_i|$. We extend the definition of the ℓ^p norm to $p = \infty$ by defining

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i|. \tag{2.2.3}$$

To appreciate the distinction between the various ℓ^p norms, we can visualize their unit balls B_p, which consist of all vectors $\boldsymbol{x}$ whose norm is at most one:[9]

$$\mathsf{B}_p \doteq \left\{ \boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq 1 \right\}. \tag{2.2.4}$$

The ℓ^2 ball is a (solid) sphere, the ℓ^∞ ball is a cube, and the ℓ^1 ball is a kind of diamond shape, also known as a *cross polytope* – see Figure 2.6. [10]

Notice that for $p \leq p'$, $\mathsf{B}_p \subseteq \mathsf{B}_{p'}$. This is because when $p \leq p'$, $\|\boldsymbol{x}\|_p \geq \|\boldsymbol{x}\|_{p'}$ for all $\boldsymbol{x}$.

REMARK 2.2. *This containment becomes even more striking in higher dimensions: in* $\mathbb{R}^n$, $\mathrm{vol}(\mathsf{B}_\infty) = 2^n$, *while* $\mathrm{vol}(\mathsf{B}_1) = 2^n/n!$ *(see e.g., [Mat02]). So, in* $n = 2$ *dimensions* $\mathrm{vol}(\mathsf{B}_1) = (1/2) \times \mathrm{vol}(\mathsf{B}_\infty)$, *while in* $n = 1,000$ *dimensions* $\mathrm{vol}(\mathsf{B}_1) \approx 10^{-2,568} \times \mathrm{vol}(\mathsf{B}_\infty)$ – *a truly negligible fraction!*

REMARK 2.3. *This may seem to be in contrast to the mathematical fact that "in finite dimensions, all norms are equivalent" in the sense that they define the same topology for the space (see, e.g., Appendix A). Formally, this statement means that in a finite dimensional vector space* $\mathbb{V}$, *such as* $\mathbb{R}^n$, *for any pair of norms* $\|\cdot\|_\diamond$ *and* $\|\cdot\|_\square$ *there exist numbers* $0 < \alpha, \beta < \infty$ *such that for every* $\boldsymbol{x} \in \mathbb{V}$,

$$\alpha \|\boldsymbol{x}\|_\square \;\leq\; \|\boldsymbol{x}\|_\diamond \;\leq\; \beta \|\boldsymbol{x}\|_\square . \tag{2.2.5}$$

So, the norms $\|\cdot\|_\square$ *and* $\|\cdot\|_\diamond$ *can be compared in size. However, as the example in*

[8] Anyone who has traveled in Manhattan should have good appreciation for the distinction between ℓ^1 and ℓ^2 – in fact, the ℓ^1 norm is sometimes called the Manhattan norm! This example illustrates a simple, but important point – the proper choice of norm depends quite a bit on the properties of the problem and design goals. Unless you can leap tall buildings in a single bound, measuring distance using the ℓ^2 norm would underestimate how much travel you need to reach your destination.

[9] For a ball of radius ε in terms of ℓ^p norm, we denote it as $\mathsf{B}_p(\varepsilon)$ or $\varepsilon \cdot \mathsf{B}_p = \left\{ \boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq \varepsilon \right\}$.

[10] To see this in action, you can run `Chapter_2_Illustrate_Lp_Balls.m`.

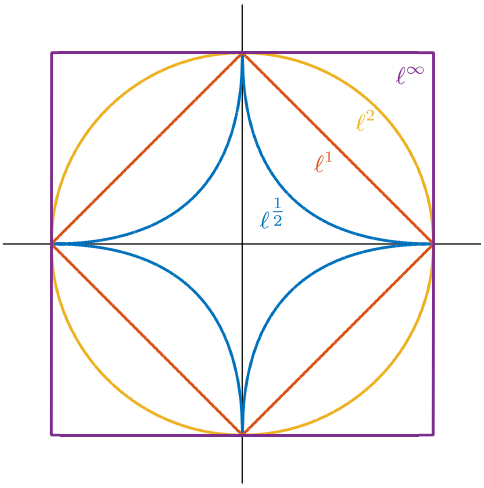


Figure 2.6 The ℓ^p Balls $\mathsf{B}_p = \left\{ \boldsymbol{x} \mid \|\boldsymbol{x}\|_p \leq 1 \right\}$ for $0 < p \leq \infty$. For $p \geq 1$, B_p is a convex set, and $\|\cdot\|_p$ is a norm. For $p < 1$, $\|\cdot\|_p$ is not a norm, in the formal sense.

Remark 2.2 shows, in high dimensions, the unit balls of the various ℓ^p norms can be very different – hence α and β can be very far apart. In applications involving high-dimensional signals, different choices in norm can lead to radically different solutions.

2.2.2 The ℓ^0 Norm

With the notion of a norm in hand, we are prepared to define a formal notion of sparsity. For this, we introduce a function, called the "ℓ^0 norm" (read "ell zero norm"), which is simply the number of nonzero entries in a vector $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_0 = \#\left\{ i \mid \boldsymbol{x}(i) \neq 0 \right\}. \tag{2.2.6}$$

Loosely speaking, $\boldsymbol{x}$ is sparse whenever $\|\boldsymbol{x}\|_0$ is small.

The ℓ^0 norm $\|\cdot\|_0$ is *not* a norm, in the formal sense of Definition 2.1: since for $\alpha \neq 0$, $\|\alpha\boldsymbol{x}\|_0 = \|\boldsymbol{x}\|_0$, it does not have the property of nonnegative homogeneity. It *does* have the other two properties, however. In particular, $\|\cdot\|_0$ is subadditive:

$$\forall\, \boldsymbol{x},\ \boldsymbol{x}', \qquad \|\boldsymbol{x} + \boldsymbol{x}'\|_0 \ \leq\ \|\boldsymbol{x}\|_0 + \|\boldsymbol{x}'\|_0 . \tag{2.2.7}$$

This is easily checked by noting that the set of nonzero entries for $\boldsymbol{x} + \boldsymbol{x}'$ is contained in the union of the set of nonzero entries of $\boldsymbol{x}$ and the set of nonzero entries of $\boldsymbol{x}'$.

Although the ℓ^0 norm is not a norm in the strict sense of Definition 2.1, it is related to the ℓ^p norm and can be viewed as a "continuation" of p from large to

small. To understand this, note that for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$\lim_{p \searrow 0} \|\boldsymbol{x}\|_p^p = \sum_{i=1}^{n} \lim_{p \searrow 0} |\boldsymbol{x}(i)|^p = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}(i) \neq 0} = \|\boldsymbol{x}\|_0. \tag{2.2.8}$$

In this sense, the ℓ^0 norm can be considered to be generated from the ℓ^p norms, by taking p (infinitesimally) small. In the context of Figure 2.6, this can be understood as follows: in $\mathbb{R}^2$, the sparse vectors correspond to the coordinate axes. As p drops towards zero, the unit ball of the ℓ^p norm becomes more concentrated around the coordinate axes, i.e., around the sparse vectors.

The geometric relationship between the ℓ^0 and ℓ^p norms is useful for deriving algorithms, and for understanding why small p tends to favor sparse solutions. With this said, the formal notation $\|\boldsymbol{x}\|_0$ has a very simple meaning: *it counts the number of nonzero entries in* $\boldsymbol{x}$. In all of the applications discussed above, our goal is to recover a vector $\boldsymbol{x}_{\text{true}}$ with $\|\boldsymbol{x}_{\text{true}}\|_0$ small. In this book, we often use $\boldsymbol{x}_o$ as a shorthand for $\boldsymbol{x}_{\text{true}}$.

2.2.3 The Sparsest Solution: Minimizing the ℓ^0 Norm

Suppose we observe $\boldsymbol{y} \in \mathbb{R}^m$, with $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, and that our goal is to recover $\boldsymbol{x}_o$. If we know that $\boldsymbol{x}_o$ is sparse, it seems reasonable to form an estimate $\hat{\boldsymbol{x}}$ by choosing the *sparsest* vector $\boldsymbol{x}$ that satisfies the equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. That is, we choose the sparsest $\boldsymbol{x}$ that could have generated our observation. We can write this as an optimization problem

$$\begin{array}{ll} \text{min} & \|\boldsymbol{x}\|_0 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{2.2.9}$$

How might we solve this problem numerically? Call

$$\text{supp}\,(\boldsymbol{x}) = \{i \mid \boldsymbol{x}(i) \neq 0\} \subset \{1, \ldots, n\} \tag{2.2.10}$$

the *support* of the vector $\boldsymbol{x}$ – this set contains the indices of the nonzero entries. The ℓ^0 minimization problem (2.2.9) asks us to find a vector $\boldsymbol{x}$ of smallest support that agrees with the observation $\boldsymbol{y}$. One approach to finding such an $\boldsymbol{x}$ is to simply try every possible subset of indices $\mathsf{I} \subseteq \{1, \ldots, n\}$ as a candidate support. For each such set I, we can form a system of equations

$$\boldsymbol{A}_{\mathsf{I}} \boldsymbol{x}_{\mathsf{I}} = \boldsymbol{y}, \tag{2.2.11}$$

where $\boldsymbol{A}_{\mathsf{I}} \in \mathbb{R}^{m \times |\mathsf{I}|}$ is the column submatrix of $\boldsymbol{A}$ formed by keeping only those columns indexed by I, and similarly for $\boldsymbol{x}_{\mathsf{I}} \in \mathbb{R}^{|\mathsf{I}|}$. We can attempt to solve (2.2.11) for $\boldsymbol{x}_{\mathsf{I}}$. If such an $\boldsymbol{x}_{\mathsf{I}}$ exists, we can obtain a solution $\boldsymbol{x}$ to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ by filling in the remaining entries of $\boldsymbol{x}$ with zeros. This exhaustive search procedure is spelled out formally as Algorithm 2.1.

Algorithm 2.1: ℓ^0**-Minimization by Exhaustive Search**

Input: a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{y} \in \mathbb{R}^m$.
for $k = 0, 1, 2, \ldots, n$,
  for each $\mathsf{I} \subseteq \{1, \ldots, n\}$ of size k,
    if the system of equations $\boldsymbol{A}_{\mathsf{I}} \boldsymbol{z} = \boldsymbol{y}$ has a solution $\boldsymbol{z}$,
      set $\boldsymbol{x}_{\mathsf{I}} = \boldsymbol{z}$, $\boldsymbol{x}_{\mathsf{I}^c} = \boldsymbol{0}$.
      return $\boldsymbol{x}$.
    end if
  end for
end for

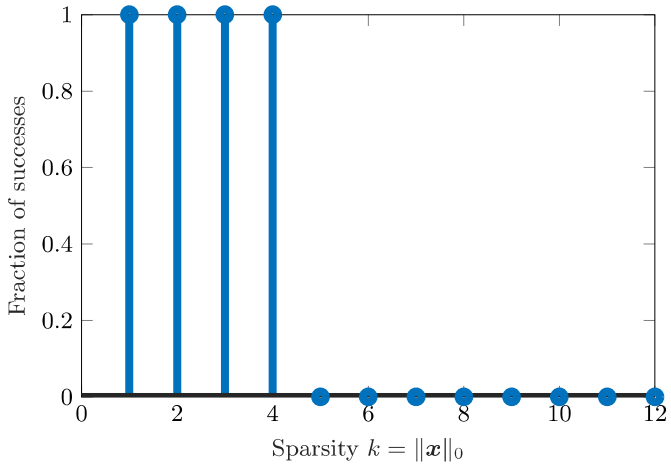


Figure 2.7 Transitions in ℓ^0 Recovery. Fraction of correct recoveries across 100 trials, as a function of the sparsity of the target solution $\boldsymbol{x}_o$. The system is of size 5×12. In this experiment, ℓ^0 minimization successfully recovers all $\boldsymbol{x}_o$ with $k \leq 4$ nonzeros.

EXAMPLE 2.4. *Let us examine how the algorithm behaves numerically, using the code* `Chapter_2_L0_recovery.m` *and* `Chapter_2_L0_transition.m` *from the book website. These examples generate random underdetermined linear systems* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$*, with* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$*, and* $\boldsymbol{x}_o$ *sparse. Apply Algorithm 2.1 (*`minimize_L0.m`*) to recover a vector* $\hat{\boldsymbol{x}}$*, and ask whether* $\hat{\boldsymbol{x}}$ *is equal to* $\boldsymbol{x}_o$ *up to machine precision. Fixing the system parameters* (m, n)*, varying the sparsity* $k = 0, 1, \ldots,$ *and performing many random trials, we produce Figure 2.7, which shows that as long as* k *is not too large, the algorithm almost always succeeds.*

Is there any mathematical explanation for this phenomenon? To understand why ℓ^0 minimization succeeds, it is worth first thinking about when it would fail. Suppose that there is a non-zero k-sparse vector $\boldsymbol{x}_o \in \operatorname{null}(\boldsymbol{A})$. Then

$$\boldsymbol{A}\boldsymbol{x}_o \;=\; \boldsymbol{0} \;=\; \boldsymbol{A}\boldsymbol{0}. \tag{2.2.12}$$

Hence, for this $\boldsymbol{x}_o \neq \mathbf{0}$, when solving $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o = \mathbf{0}$, the ℓ^0 minimizer is simply $\hat{\boldsymbol{x}} = \mathbf{0}$, and the true $\boldsymbol{x}_o$ is not recovered. Put simply: if the null space of $\boldsymbol{A}$ contains sparse vectors (aside from $\mathbf{0}$), ℓ^0 minimization may fail to recover the desired sparse vector $\boldsymbol{x}_o$.

In fact, the converse statement is also true: when the null space of $\boldsymbol{A}$ *does not* contain sparse vectors (aside from $\mathbf{0}$), ℓ^0 minimization *does* recover any sufficiently sparse vector $\boldsymbol{x}_o$. To state the argument simply, let us suppose that $\|\boldsymbol{x}_o\|_0 \leq k$, and assume:

($\star$) *the only* $\boldsymbol{\delta} \in \operatorname{null}(\boldsymbol{A})$ *with* $\|\boldsymbol{\delta}\|_0 \leq 2k$ *is* $\boldsymbol{\delta} = \mathbf{0}$.

Let $\hat{\boldsymbol{x}}$ denote the solution to the ℓ^0 minimization problem, so $\|\hat{\boldsymbol{x}}\|_0 \leq \|\boldsymbol{x}_o\|_0 \leq k$. If we define the *estimation error*

$$\boldsymbol{\delta} = \hat{\boldsymbol{x}} - \boldsymbol{x}_o, \tag{2.2.13}$$

then

$$\|\boldsymbol{\delta}\|_0 = \|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_0 \leq \|\hat{\boldsymbol{x}}\|_0 + \|\boldsymbol{x}_o\|_0 \leq 2k. \tag{2.2.14}$$

So, $\boldsymbol{\delta}$ is a sparse vector. Moreover,

$$\boldsymbol{A}\boldsymbol{\delta} = \boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o) = \boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x}_o = \boldsymbol{y} - \boldsymbol{y} \;=\; \mathbf{0}. \tag{2.2.15}$$

So, $\boldsymbol{\delta}$ is a sparse vector *in the null space of* $\boldsymbol{A}$. Property ($\star$) states that the only sparse vector in $\operatorname{null}(\boldsymbol{A})$ is $\mathbf{0}$. So, if ($\star$) holds, $\boldsymbol{\delta} = \mathbf{0}$, and so $\hat{\boldsymbol{x}} = \boldsymbol{x}_o$: ℓ^0 minimization indeed recovers $\boldsymbol{x}_o$.

Property ($\star$) is a property of the matrix $\boldsymbol{A}$. The above reasoning suggests a slogan: *the "good" $\boldsymbol{A}$ for recovering sparse vectors $\boldsymbol{x}_o$ are those $\boldsymbol{A}$ that have no sparse vectors in their null space.* We can restate property ($\star$) more conveniently in terms of the columns of $\boldsymbol{A}$: property ($\star$) holds if and only if every set of $2k$ columns of $\boldsymbol{A}$ is linearly independent.

Definition 2.5 (Kruskal Rank [Kru77]). *The* Kruskal rank *of a matrix $\boldsymbol{A}$, written as* $\operatorname{krank}(\boldsymbol{A})$, *is the largest number r such that every subset of r columns of $\boldsymbol{A}$ is linearly independent.*

From the above reasoning, if $\|\boldsymbol{x}_o\|_0$ is at most half of $\operatorname{krank}(\boldsymbol{A})$, ℓ^0 minimization will recover $\boldsymbol{x}_o$:

Theorem 2.6 (ℓ^0 Recovery). *Suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with*

$$\|\boldsymbol{x}_o\|_0 \;\leq\; \tfrac{1}{2}\operatorname{krank}(\boldsymbol{A}). \tag{2.2.16}$$

Then $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^0 minimization problem

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_0 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{2.2.17}$$

Notice that Theorem 2.6 agrees with the behavior in Figure 2.7.[11] Theorem 2.6 predicts that as long as $\boldsymbol{x}_o$ is *sufficiently sparse*, it will be recovered by ℓ^0 minimization. The level of allowable sparsity depends on the Kruskal rank of the matrix $\boldsymbol{A}$. It is not hard to see that in general,

$$0 \leq \operatorname{krank}(\boldsymbol{A}) \leq \operatorname{rank}(\boldsymbol{A}). \tag{2.2.18}$$

For "generic" $\boldsymbol{A}$, the Kruskal rank is quite large:

PROPOSITION 2.7. *Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $n \geq m$, with A_{ij} independent identically distributed $\mathcal{N}(0,1)$ random variables. Then, with probability one,* $\operatorname{krank}(\boldsymbol{A}) = m$.

Proof Exercise 2.7 guides the interested reader through the proof. □

The intuition is that to have $\operatorname{krank}(\boldsymbol{A}) < m$, there must be some subset of m columns of $\boldsymbol{A}$ which are linearly dependent, i.e., there is some subset $\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \ldots, \boldsymbol{a}_{i_m}$ which lies on a linear subspace of dimension $m-1$. For a Gaussian random matrix $\boldsymbol{A}$, the probability that this happens is zero. This is true of many other random matrices.[12] We can interpret this as saying that under generic circumstances, knowing that the target $\boldsymbol{x}_o$ is sparse turns an ill-posed problem into a well-posed one. The ℓ^0 minimization problem recovers vectors $\boldsymbol{x}_o$ whose number of nonzeros is as large as $\frac{m}{2}$. This level of sparsity is well beyond what is needed for most applications.

2.2.4 Computational Complexity of ℓ^0 Minimization

The theoretical results in the previous section show the power of sparsity: knowing that the target solution $\boldsymbol{x}_o$ is even moderately sparse can render the problem of recovering $\boldsymbol{x}_o$ well-posed. Unfortunately, Algorithm 2.1 is not very useful in practice. Its worst-case running time is on the order of n^k, where $k = \|\boldsymbol{x}_o\|_0$ is the number of nonzero entries we wish to recover. For example, at the time of writing this book, to solve a problem with $m = 50$, $n = 200$, and $k = 10$, on a standard laptop, Algorithm 2.1 would require ≈ 140 *centuries*. This is still a very small problem by the standard of most modern-day applications!

Exhaustively searching all possible supports I may not seem like a particularly intelligent strategy for solving the ℓ^0-minimization problem (2.2.9). However, no significantly better algorithm is currently known that can solve this class of problems efficiently. Is this because we are not clever enough and have not found the correct (efficient) algorithm yet? Or is it the nature of this class of problems such that an efficient algorithm simply does not exist? To answer this question

[11] Actually, the behavior in Figure 2.7 is slightly better than what Theorem 2.6 predicts – with probability one the Kruskal rank of $\boldsymbol{A}$ is m, and so the theorem shows that ℓ^0 minimization succeeds when $k \leq \frac{m}{2} = 2$. However, in the experiment, success always occurs when $k \leq 4$. Exercise 2.8 asks you to explain this discrepancy, by proving a modified version of Theorem 2.6.

[12] For example, $\operatorname{krank}(\boldsymbol{A}) = m$ with probability one whenever $\boldsymbol{A}$ is distributed according to any absolutely continuous measure, i.e., there is a probability density function.

more rigorously, we need to borrow some formal tools and results from complexity theory.

Complexity Classes and NP*-Hardness.*
If you don't have any background in complexity theory, you can loosely think of the situation as follows. The problem class **P** consists of problems that we can solve in time polynomial in the size of the problem. The problem class **NP** consists of those problems for which, if we are given a "certificate" describing the optimal solution, we can check that it is correct in polynomial time. That is, **P** contains problems for which *finding* the right answer is "easy," while **NP** contains problems for which *checking* the right answer is easy. Anyone who has ever struggled with a problem for days, only to have a colleague or teacher easily demonstrate an obviously correct solution can appreciate the difference between finding the right answer and checking the right answer!

It turns out that amongst the **NP** problems, there are certain "NP*-complete*" problems to which *every* problem in **NP** can be reduced, in polynomial time, to each other. So, solving one of these problems efficiently would enable you to solve every problem in **NP** efficiently! It is remarkable that this class of problems exists, and that it is quite large. It includes famous examples such as the Traveling Salesman Problem and the Multiway Cut Problem.

To understand the phrase "NP*-hard*," we have to appreciate one technicality regarding the above definitions of **P** and **NP**: they pertain only to *decision* problems, in which the goal is to produce a YES/NO answer. For example, the decision version of the Traveling Salesman Problem asks: "Is it possible to visit all of the nodes of a given graph (cities) while traveling a distance at most $d_\star$?" The decision version of the ℓ^0 problem asks: "Does the system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ have a solution with at most k nonzero entries?"

Often in practice we care much more about *optimization problems* than *decision problems* – we do not just want to know whether a solution exists, we want to know the way to find it! Strictly speaking, optimization problems cannot be "NP-complete" – in the formal definition of **NP**, we only include decision problems. Nevertheless, we may call an optimization problem NP*-hard* if an efficient solution to that problem can be used to efficiently solve NP-complete problems. For example, the optimization version of the Traveling Salesman Problem asks: "Find the shortest path that visits all of the nodes in a given graph." If one can solve this problem efficiently, one can clearly also solve the decision version efficiently, just by checking whether the optimal path has length at most $d_\star$.

NP-complete problems are considered highly unlikely to be efficiently solvable (i.e., solvable on standard (model) computers polynomial in time and the size of the problem).[13] This class of problems includes notoriously difficult examples, such as the Traveling Salesman Problem. Fully appreciating the mathematical

[13] This is known as the "**P** versus **NP**" problem, one of the most famous open problems in mathematics and theoretical computing. The Clay Mathematics Institute is offering a reward of 1 million dollars to anyone who has a formal proof that **P**=**NP** or that **P**≠**NP**.

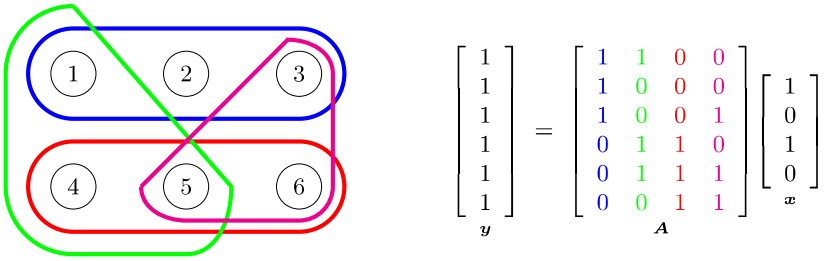


Figure 2.8 Exact 3-Set Cover as a Sparse Representation Problem. Left: a universe $\mathsf{S} = \{1, \ldots, 6\}$ and four subsets $\mathsf{U}_1, \ldots, \mathsf{U}_4 \subseteq \mathsf{S}$. $\{\mathsf{U}_1, \mathsf{U}_3\}$ is an exact 3-set cover. **Right:** the same problem as a linear system of equations. The columns of $\boldsymbol{A}$ are the incidence vectors for the sets $\mathsf{U}_1, \mathsf{U}_2, \mathsf{U}_3, \mathsf{U}_4$. The Exact 3-Cover $\{\mathsf{U}_1, \mathsf{U}_3\}$ corresponds to a solution $\boldsymbol{x}$ to the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ with only $m/3 = 2$ nonzero entries.

content of complexity theory requires formal modeling of computation (Turing machines, complexity theory for different problem classes, etc.) that is beyond the scope of this book. For interested readers, we refer to the book [GJ90] for a formal introduction to this important subject.

NP-*Hardness of* ℓ^0*-Minimization.*
For our purposes here, we are interested whether the ℓ^0-minimization problem (2.2.9) is equivalent (in its complexity) to certain known NP-hard problems. Indeed, we can show that:

THEOREM 2.8 (Hardness of ℓ^0 Minimization). *The ℓ^0-minimization problem (2.2.9) is* NP*-hard.*

Proof of Theorem 2.8: Hardness results are typically proved by reduction: we show that if we can solve the problem of interest efficiently, this would allow us to also efficiently solve some other problem, which is already known to be hard. For the ℓ^0 minimization problem, we do this by showing that ℓ^0 minimization can be used to solve certain (hard) set covering problems.

Consider the following problem:

Exact 3-Set Cover (E3C): Given a set $\mathsf{S} = \{1, \ldots, m\}$ and a collection $\mathcal{C} = \{\mathsf{U}_1, \ldots, \mathsf{U}_n\}$ of subsets $\mathsf{U}_j \subseteq \mathsf{S}$ each of which has size $|\mathsf{U}_j| = 3$, does there exist a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ that exactly covers S, i.e., $\forall i \in \mathsf{S}$ there is exactly one $\mathsf{U} \in \mathcal{C}'$ with $i \in \mathsf{U}$?

This problem is known to be NP-complete [Kar72, GJ79]. To reduce it to ℓ^0 minimization, suppose that we are given an instance of E3C: Form an $m \times n$ matrix $\boldsymbol{A} \in \{0,1\}^{m \times n}$ by letting $A_{ij} = 1$ if $i \in \mathsf{U}_j$, and $A_{ij} = 0$ otherwise. Set $\boldsymbol{y} = \mathbf{1} \in \mathbb{R}^m$ (i.e., an m-dimensional vector of ones). Figure 2.8 illustrates this construction. We show:

Claim: The system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ has a solution $\boldsymbol{x}_o$ with $\|\boldsymbol{x}_o\|_0 \leq m/3$ if and only if there exists an exact 3-set cover.

($\Leftarrow$) Suppose there exists an exact 3-set cover $\mathcal{C}'$. Clearly, $|\mathcal{C}'| = m/3$. Set

$$x_j = \begin{cases} 1 & \mathsf{U}_j \in \mathcal{C}' \\ 0 & \text{else} \end{cases} .$$

Then $\|\boldsymbol{x}\|_0 = m/3$, and $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$.

($\Rightarrow$) Let $\boldsymbol{x}_o$ be a solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ with at most $m/3$ nonzero entries. Set $\mathcal{C}' = \{\mathsf{U}_j \mid \boldsymbol{x}_o(j) \neq 0\}$. We claim $\mathcal{C}'$ is the desired cover. Let $\mathsf{I} = \operatorname{supp}(\boldsymbol{x}_o)$. Since each column of $\boldsymbol{A}$ has exactly 3 nonzero entries, and $\boldsymbol{A}_\mathsf{I}$ has at most $m/3$ columns, the matrix $\boldsymbol{A}_\mathsf{I}$ has at most m nonzero entries. Since $\boldsymbol{A}_\mathsf{I}\boldsymbol{x}_{o\mathsf{I}} = \boldsymbol{y}$, each row of $\boldsymbol{A}_\mathsf{I}$ has at least one nonzero entry. Hence, each row of $\boldsymbol{A}_\mathsf{I}$ has *exactly* one nonzero entry, and the set $\mathcal{C}'$ gives an exact cover. □

In fact, the truth is even worse than Theorem 2.8 suggests: The ℓ^0 minimization problem remains NP-hard even if we only demand that $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{y}$, in an appropriate sense. It is also NP-hard to find an $\boldsymbol{x}$ whose number of nonzero entries is within a constant factor of the smallest possible! See more discussions in the Notes Section 2.5. Based on our current understanding of complexity theory, it is extraordinarily unlikely that anyone will ever discover an efficient algorithm that solves any interesting variant of the ℓ^0 minimization problem for all possible inputs $(\boldsymbol{A}, \boldsymbol{y})$.

2.3 Relaxing the Sparse Recovery Problem

The rather bleak worst-case picture for ℓ^0-minimization has not stopped engineers from searching for efficient heuristics for finding sparse solutions to linear systems.[14] There is always some possibility for optimism:

"Although the *worst* sparse recovery problem may be impossible to solve efficiently, perhaps my *particular* instance (or a subclass of instances) of interest is not so hard."

This optimism is occasionally rewarded in a rather striking fashion. In the next few chapters, we will see that many sparse recovery problems that matter for engineering practice *are* solvable efficiently. Our first step is to find a proper surrogate for the ℓ^0 norm which still encourages sparsity, but can be optimized efficiently.

2.3.1 Convex Functions

If our goal is efficient optimization, perhaps the most natural class of objective functions to consider is the *convex* functions. Smooth convex functions often appear "bowl shaped" – as in Figure 2.9 (left). Indeed, a necessary and sufficient

[14] as it has never stopped nature from learning and exploiting sparse coding.

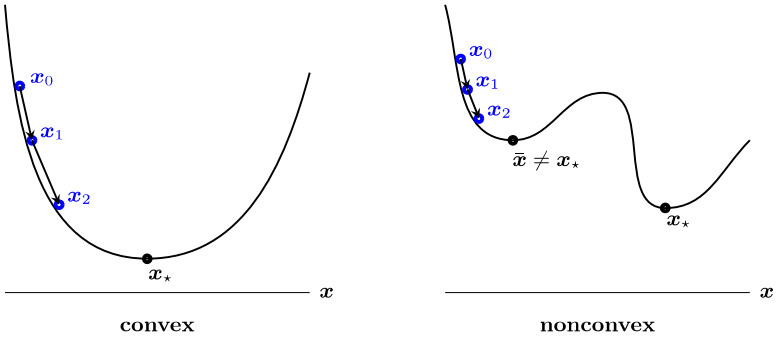


Figure 2.9 Convex and Nonconvex Functions. Left: a convex function. Local descent methods such as gradient descent produce a sequence of points $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$ which approach the global minimizer $\boldsymbol{x}_\star$. **Right:** A nonconvex function. For this particular function, depending on the initial point $\boldsymbol{x}_0$, local descent methods may produce the suboptimal local minimum $\bar{\boldsymbol{x}}$. Motivated by their good properties for optimization, in the first part of this book, we will seek convex formulations for recovering sparse (and otherwise structured) signals.

condition for a smooth function $f(x) : \mathbb{R} \to \mathbb{R}$ to be convex is that it exhibits nonnegative curvature – its second derivative $\frac{d^2 f}{dx^2}(x) \geq 0$ at every point x.[15]

Iterative methods for optimization seek a minimizer of an objective function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$, by starting from some initial point $\boldsymbol{x}_0$,[16] and then generating a new point $\boldsymbol{x}_1$ based on the local shape of the objective function in the vicinity of $\boldsymbol{x}_0$. For a smooth function $f(\boldsymbol{x})$, the negative gradient $-\nabla f(\boldsymbol{x})$ defines the direction in which the objective function decreases most rapidly. A natural strategy for choosing $\boldsymbol{x}_1$ is to move in this descending direction

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - t\nabla f(\boldsymbol{x}_0), \tag{2.3.1}$$

where t is a step size. Continuing in this manner to produce points $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$, we obtain the *gradient descent* method,[17] a natural and intuitive algorithm for minimizing a smooth function $f(\boldsymbol{x})$. For the function f in Figure 2.9 (left), assuming we choose the step size t appropriately, the iterates $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots$ will converge to the global minimizer $\boldsymbol{x}_\star$. For the nonconvex function to the right, this strategy only guarantees a local minimizer.[18]

Convex functions such as Figure 2.9 (left) have the property that every local

[15] For a multi-variate function $f(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$, we need the Hessian of the function to be positive semi-definite: $\nabla^2 f(\boldsymbol{x}) \succeq \boldsymbol{0}$.

[16] In this book, we will use $\boldsymbol{x}_0$ to indicate the initial point of an iterative algorithm, which is not to be confused with $\boldsymbol{x}_o$, the desired ground truth.

[17] Gradient descent, also known as steepest descent, was first introduced by Cauchy in 1847 [Cau47]. Appendix C gives a more detailed account of optimization algorithms, including gradient descent.

[18] More precise conditions for convergence and complexity will be given in Chapters 8 and 9 for convex and nonconvex problems, respectively.

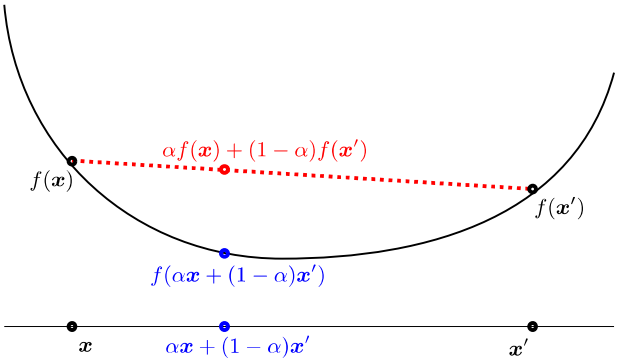


Figure 2.10 Definition of Convexity. A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is one which satisfies the inequality $f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}') \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}')$ for all $\alpha \in [0,1]$ and $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$. Geometrically, this means that if we take the points $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{x}', f(\boldsymbol{x}'))$ on the graph of f, and then draw a **line** joining them, the graph of the function falls below this line segment.

minimizer is a global minimizer.[19] Moreover, many convex functions arising in practice can be optimized efficiently using variants of gradient descent. Indeed, in Chapter 8, we will see that the particular convex functions that we encounter in computing with sparse signals (and their generalizations) *can* be efficiently optimized, even on a large scale and in high dimensions.

We review the properties of convex functions more formally in Appendix C. Here, we briefly remind the reader of the general definition of convex functions:[20]

Definition 2.9 (Convex Function on $\mathbb{R}^n$). *A continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for every pair of points $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$ and $\alpha \in [0,1]$,*

$$f\Big(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{x}'\Big) \;\leq\; \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}'). \tag{2.3.2}$$

This inequality can be visualized as follows. Consider two points $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{x}', f(\boldsymbol{x}'))$ on the graph of f. If we form the line segment joining these two points, this line segment lies above the graph of f. Figure 2.10 visualizes this inequality with an example.

A *convex combination* of a collection of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ is an expression of

[19] It is worth noting that for many of the problems we will later discuss (e.g., MRI, spectrum sensing, face recognition), global optimality is very important – there is a *true* signal that we are trying to recover, and it is important to build algorithms that can do this reliably. In our simulated example of ℓ^0 minimization, we declared the solution $\hat{\boldsymbol{x}}$ correct, because it coincided with the true $\boldsymbol{x}_o$ that generated the observation $\boldsymbol{y}$. This is in contrast to some applications of optimization (e.g., in finance) where the objective function measures the goodness of the solution (say the expected rate of return on an investment), and locally improving the solution is meaningful, or even desirable, if the objective corresponds to dollars earned/lost!

[20] On the surface, this definition appears much more complicated than simply asking the second derivative to be positive. The reason for this complication is that we will need to

the form $\sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i$, where the weights λ_i are nonnegative and $\sum_{i=1}^{k} \lambda_i = 1$. For example, for $\alpha \in [0,1]$, the expression $\boldsymbol{z} = \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{x}'$ is a convex combination of the points $\boldsymbol{x}$ and $\boldsymbol{x}'$. The definition (2.3.2) states that at the point $\boldsymbol{z}$, the function f is no larger than the corresponding combination $\alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{x}')$ of the function values at the points $\boldsymbol{x}$ and $\boldsymbol{x}'$.

This property of convex functions generalizes and gives the important Jensen's inequality, which states that the value of a convex function f at a convex combination of points is no greater than the corresponding convex combination of the function values:

PROPOSITION 2.10 (Jensen's Inequality). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then for any k, any collection of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathbb{R}^n$ and any nonnegative scalars $\lambda_1, \ldots, \lambda_k$ satisfying $\sum_{i=1}^{k} \lambda_i = 1$,*

$$f\left(\sum_{i=1}^{k} \lambda_i \boldsymbol{x}_i\right) \leq \sum_{i=1}^{k} \lambda_i f\left(\boldsymbol{x}_i\right). \tag{2.3.3}$$

2.3.2 A Convex Surrogate for the ℓ^0 Norm: the ℓ^1 Norm

With the good properties of convex functions in mind, let us try to find a convex "surrogate" for the ℓ^0 norm. In one dimension, x is a scalar, and $\|x\|_0 = \mathbb{1}_{x \neq 0}$ is simply the indicator function for nonzero x. From Figure 2.11, it is clear that if we restrict our attention to the interval $x \in [-1, 1]$, the largest convex function which does not exceed $\|\cdot\|_0$ on this interval is simply the absolute value $|x|$. In the language of convex analysis, $|x|$ is the *convex envelope* of the function $\|x\|_0$ over the set $[-1, 1]$. This means that $|x|$ is the largest convex function f which satisfies $f(x) \leq \|x\|_0$ for every $x \in [-1, 1]$, i.e., it is the largest convex underestimator of $\|x\|_0$ over this set. Thus, in one dimension, we might consider the absolute value of x as a plausible replacement for $\|x\|_0$.

For higher-dimensional $\boldsymbol{x}$ (i.e., $\boldsymbol{x} \in \mathbb{R}^n$), the ℓ^0 norm is[21]

$$\|\boldsymbol{x}\|_0 = \sum_{i=1}^{n} \mathbb{1}_{\boldsymbol{x}(i) \neq 0}. \tag{2.3.4}$$

Applying the above reasoning to each of the coordinates $\boldsymbol{x}(i)$, we obtain the ℓ^1 norm

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |\boldsymbol{x}(i)|. \tag{2.3.5}$$

As in the scalar case, this function is the tightest convex underestimator of $\|\cdot\|_0$, over an appropriate set of vectors $\boldsymbol{x}$:

work with convex functions that are not smooth; the general condition given in Definition 2.9 handles this situation as well.

[21] In this book, we use $\boldsymbol{x}(i)$ to indicate the i-th entry of a vector $\boldsymbol{x}$. Also we often use the shorthand $x_i = \boldsymbol{x}(i) \in \mathbb{R}$.

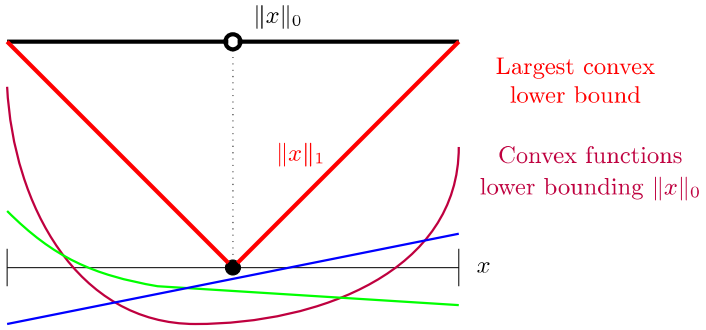


Figure 2.11 A Convex Surrogate for the ℓ^0 Norm. In **black**, we plot the graph of the ℓ^0 norm of a scalar x, over the interval $x \in [-1, 1]$. This function takes on the value 0 at $x = 0$, and $+1$ everywhere else. In **purple**, **green** and **blue**, we plot various convex function examples $f(x)$ which underestimate $\|x\|_0$ on $[-1, 1]$, in the sense that $f(x) \leq \|x\|_0$ for all $x \in [-1, 1]$. In **red**, we plot the function $f(x) = |x|$. This is the largest convex function which underestimates $\|x\|_0$ on $[-1, 1]$. We call $|x|$ the *convex envelope* of $\|x\|_0$ on $[-1, 1]$.

THEOREM 2.11. *The function $\|\cdot\|_1$ is the convex envelope of $\|\cdot\|_0$, over the set $\mathsf{B}_\infty = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_\infty \leq 1\}$ of vectors whose elements all have magnitude at most one.*

Proof Let f be a convex function satisfying $f(\cdot) \leq \|\cdot\|_0$ on B_∞. We prove that $f(\cdot) \leq \|\cdot\|_1$ on B_∞ as well. Consider the cube $\mathsf{C} = [0, 1]^n$. Its vertices are the vectors $\boldsymbol{\sigma} \in \{0, 1\}^n$. Any $\boldsymbol{x} \in \mathsf{C}$ can be written as a convex combination of these vertices:

$$\boldsymbol{x} = \sum_i \lambda_i \boldsymbol{\sigma}_i. \tag{2.3.6}$$

Because $f(\cdot) \leq \|\cdot\|_0$, $f(\boldsymbol{\sigma}_i) \leq \|\boldsymbol{\sigma}_i\|_0 = \|\boldsymbol{\sigma}_i\|_1$. Because f is convex,

$$\begin{aligned} f(\boldsymbol{x}) = f\left(\textstyle\sum_i \lambda_i \boldsymbol{\sigma}_i\right) &\leq \sum_i \lambda_i f(\boldsymbol{\sigma}_i) && \text{[Jensen's inequality]} \\ &\leq \sum_i \lambda_i \|\boldsymbol{\sigma}_i\|_0 = \sum_i \lambda_i \|\boldsymbol{\sigma}_i\|_1 && [\boldsymbol{\sigma}_i \text{ are binary}] \\ &= \|\boldsymbol{x}\|_1 . \end{aligned} \tag{2.3.7}$$

Hence, $f(\cdot) \leq \|\cdot\|_1$ on the intersection of B_∞ with the nonnegative orthant. Repeating the argument for each of the orthants, we obtain that $f(\cdot) \leq \|\cdot\|_1$ on B_∞, and hence $\|\cdot\|_1$ is the convex envelope of $\|\cdot\|_0$ over B_∞. □

So, at least in the sense of convex envelopes, the ℓ^1 norm provides a good replacement for the ℓ^0 norm. Replacing the ℓ^0 norm in (2.2.9) with the ℓ^1 norm, we obtain a convex ℓ^1 minimization problem,

$$\begin{aligned} &\min && \|\boldsymbol{x}\|_1 \\ &\text{subject to} && \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{aligned} \tag{2.3.8}$$

In contrast to the ℓ^0 problem, this problem *can* be solved efficiently.

2.3.3 A Simple Test of ℓ^1 Minimization

Theorem 2.11 is a strong initial motivation for considering ℓ^1 minimization (2.3.8) for recovering a sparse solution – it says that in a certain sense, the ℓ^1 norm is the canonical convex surrogate for the ℓ^0 norm. Some care is in order, though. Theorem 2.11 does not say anything at all about the *correctness* of (2.3.8) – whether the solution to (2.3.8) is actually the desired sparse vector $\boldsymbol{x}_o$.

The easiest way to get some insight into this question is to do an experiment! For this, we will need to solve the problem (2.3.8) computationally and see how well it works. How do we solve the optimization problem (2.3.8)? Appendix D gives a quick introduction to some general optimization techniques that may help us solve problems of this kind. More specifically, since the objective function is convex, the geometry of a convex function in Figure 2.12 (left) suggests that we should do quite well just using local information about the slope of the objective function. Indeed, if our objective function were differentiable, this would very naturally suggest the classical *gradient descent* method for solving problems of the form

$$\min \quad f(\boldsymbol{x}). \tag{2.3.9}$$

This algorithm starts at some initial point $\boldsymbol{x}_0$, and then generates a sequence of points $(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \ldots)$ by iteratively moving in the direction of greatest decrease of $f(\cdot)$:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k). \tag{2.3.10}$$

Here, $t_k \geq 0$ is a properly chosen step size.

There are two main difficulties that prevent us from directly applying the gradient descent iteration (2.3.10) to the ℓ^1 minimization problem (2.3.8):

- **Nontrivial constraints:** Unlike the general unconstrained problem (2.3.9), in the problem (2.3.8) we are only interested in $\boldsymbol{x}$ that satisfy $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.
- **Nondifferentiable objective:** The objective function in (2.3.8) is not differentiable, and so at certain points the gradient $\nabla f(\boldsymbol{x})$ does not exist. Figure 2.12 (right) shows this: the function is pointed at zero! Since zero is sparse, this is precisely one of the points we are most interested in.

Constraints.
One approach to handle the first problem is to replace the gradient descent iteration with *projected gradient descent*. This algorithm aims at general problems of the form

$$\begin{array}{ll} \min & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \mathsf{C}, \end{array} \tag{2.3.11}$$

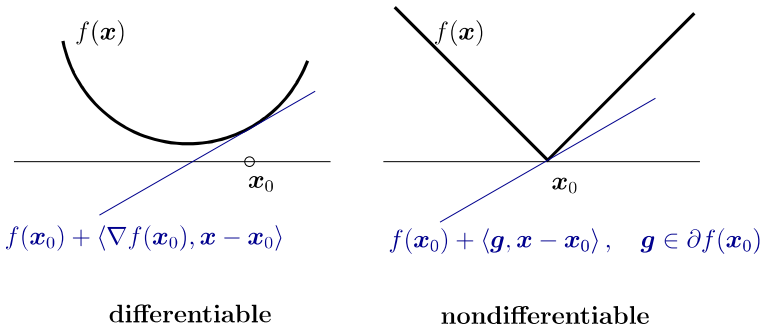


Figure 2.12 Subgradients of Convex Functions. Left: for a differentiable convex function, the best linear approximation at any point $\boldsymbol{x}_0$ is a *global* lower bound on the function. **Right**: for a nondifferentiable function, we say that $\boldsymbol{g}$ is a subgradient of f at $\boldsymbol{x}_0$ (and write $\boldsymbol{g} \in \partial f(\boldsymbol{x}_0)$ if $\boldsymbol{g}$ is the slope of a linear function that takes on the value $f(\boldsymbol{x}_0)$ at $\boldsymbol{x}_0$, and globally lower bounds f.

where C is some constraint set. This algorithm is exactly the same as gradient descent, except that at each iteration it *projects* the result $\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)$ onto the set C. The projection of a point $\boldsymbol{z}$ onto the set C is simply the nearest point to $\boldsymbol{z}$ in C:

$$\mathcal{P}_{\mathsf{C}}[\boldsymbol{z}] = \arg\min_{\boldsymbol{x} \in \mathsf{C}} \ \tfrac{1}{2} \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \equiv h(\boldsymbol{x}). \tag{2.3.12}$$

For general C, the projection may not exist, or may not be unique (think about how this could happen). However, for closed, convex sets, the projection is well-defined, and satisfies a wealth of useful properties. If $\boldsymbol{A}$ has full row rank, the projection onto the convex set $\mathsf{C} = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$ has an especially simple form:

$$\mathcal{P}_{\{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}}[\boldsymbol{z}] = \boldsymbol{z} - \boldsymbol{A}^* \left(\boldsymbol{A}\boldsymbol{A}^*\right)^{-1} \left[\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\right]. \tag{2.3.13}$$

Figure 2.13 visualizes the projection onto this particular C. This formula can be derived by noting two properties of the projection $\hat{\boldsymbol{x}} = \mathcal{P}_{\mathsf{C}}[\boldsymbol{z}]$:

1 **Feasibility**: $\hat{\boldsymbol{x}} \in \mathsf{C}$, i.e., $\boldsymbol{A}\hat{\boldsymbol{x}} = \boldsymbol{y}$.
2 **Residual is orthogonal**: $\boldsymbol{z} - \hat{\boldsymbol{x}} \perp \mathrm{null}(\boldsymbol{A})$. Since $\boldsymbol{z} - \hat{\boldsymbol{x}} = -\nabla h\left(\hat{\boldsymbol{x}}\right)$, this condition can be stated as

$$-\nabla h(\hat{\boldsymbol{x}}) \text{ is orthogonal to } \mathsf{C} \text{ at } \hat{\boldsymbol{x}}.$$

Exercise 2.11 guides the interested reader through the derivation of this expression. For the general problem (2.3.11), with differentiable objective f, the *projected gradient algorithm* simply repeats the iteration

$$\boldsymbol{x}_{k+1} = \mathcal{P}_{\mathsf{C}}\left[\,\boldsymbol{x}_k - t_k \nabla f(\boldsymbol{x}_k)\,\right]. \tag{2.3.14}$$

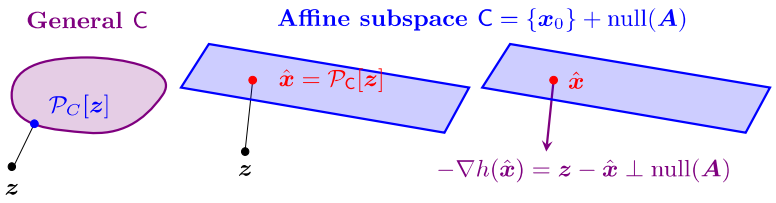


Figure 2.13 Projection onto Convex Sets. Left: projection onto a general convex set. **Middle:** projection onto an affine subspace. **Right:** projection onto the affine subspace can be characterized as the point $\hat{\boldsymbol{x}}$ at which the gradient $\nabla h(\hat{\boldsymbol{x}})$ is orthogonal to null($\boldsymbol{A}$).

Nondifferentiability.
The problem of nondifferentiability is slightly trickier. To handle it properly, we need to generalize the notion of derivative to include functions that are not differentiable. For this, we draw inspiration from geometry. Consider Figure 2.12 (left). It displays a convex, differentiable function $f(\boldsymbol{x})$, as well as a linear approximation $\hat{f}(\boldsymbol{x})$, taken at a point $\boldsymbol{x}_0$:

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle \,. \tag{2.3.15}$$

The salient point here is that the graph of f lies entirely above the graph of the approximation $\hat{f}$:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{x} - \boldsymbol{x}_0 \rangle \,, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n. \tag{2.3.16}$$

It is not too difficult to prove that this property holds for *every* convex differentiable function and every point $\boldsymbol{x}_0$, simply by using calculus and the definition of convexity.

This geometry opens the door for generalizing the notion of the gradient to nonsmooth functions. For nonsmooth functions such as $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, at a point of nonsmoothness $\boldsymbol{x}_0$, the gradient does not exist, but we can still make a linear under-estimator

$$\hat{f}(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \,, \tag{2.3.17}$$

as in Figure 2.12 (right). Here, $\boldsymbol{u}$ replaces ∇f in the previous expression, and plays the role of the "slope" of the approximation. We say that $\boldsymbol{u}$ is a *subgradient* of f at $\boldsymbol{x}_0$ if the linear approximation defined by $\boldsymbol{u}$ is indeed an under-estimator of f (i.e., it lower bounds $f(\boldsymbol{x})$ at all points $\boldsymbol{x}$):

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle \,, \quad \forall\, \boldsymbol{x}. \tag{2.3.18}$$

Let us consider our function of interest – the ℓ^1 norm. For $\boldsymbol{x} \in \mathbb{R}$ (one dimension), $\|\boldsymbol{x}\|_1 = |x|$ is simply the absolute value. For $x < 0$, the slope of the graph of $|x|$ is -1, while for $|x| > 0$, it is $+1$. Convince yourself that if we take $\boldsymbol{x}_0 \neq 0$, then the only u satisfying the above definition is $u = \text{sign}(x)$.

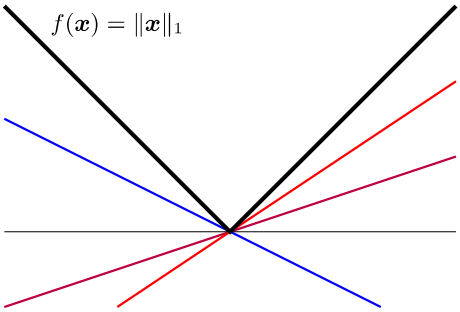


Figure 2.14 Subdifferential of the ℓ^1 Norm. In **black**, $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$. In **blue**, **purple**, and **red**, three linear lower bounds of the form $g(\boldsymbol{x}) = f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle$, taken at $\boldsymbol{x}_0 = \mathbf{0}$, with slope $\boldsymbol{u} = -\frac{1}{2}$, $\frac{1}{3}$, and $\frac{2}{3}$, respectively. It should be clear that any slope $\boldsymbol{u} \in [-1, 1]$ defines a linear lower bound on $f(\boldsymbol{x})$ around $\boldsymbol{x}_0 = \mathbf{0}$. So, $\partial|\cdot|(0) = [-1, 1]$. For $\boldsymbol{x}_0 > 0$, the only linear lower bound has slope $\boldsymbol{u} = 1$; for $\boldsymbol{x}_0 < 0$, the only linear lower bound has slope $\boldsymbol{u} = -1$. So, $\partial|\cdot|(\boldsymbol{x}) = \{-1\}$ for $\boldsymbol{x} < 0$ and $\partial|\cdot|(\boldsymbol{x}) = \{1\}$ for $\boldsymbol{x} > 0$. Lemma 2.13 proves this formally, and extends to higher-dimensional $\boldsymbol{x} \in \mathbb{R}^n$.

However, at 0 the function $|x|$ is "pointy," namely, nondifferentiable, and something different happens: at $x_0 = 0$, every $u \in [-1, 1]$ defines a linear approximation that underestimates f. So, in fact, every $u \in [-1, 1]$ is a subgradient. Thus, at points of nondifferentiability there may exist multiple subgradients. We call the collection of all subgradients of f at a point $\boldsymbol{x}_0$ the *subdifferential* of f at $\boldsymbol{x}_0$, and denote it by $\partial f(\boldsymbol{x}_0)$. Formally:

Definition 2.12 (Subgradient and Subdifferential). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. A* subgradient *of f at $\boldsymbol{x}_0$ is any $\boldsymbol{u}$ satisfying*

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle, \quad \forall\, \boldsymbol{x}. \tag{2.3.19}$$

The subdifferential *of f at $\boldsymbol{x}_0$ is the set of all subgradients of f at $\boldsymbol{x}_0$:*

$$\partial f(\boldsymbol{x}_0) = \{\boldsymbol{u} \mid \forall\, \boldsymbol{x} \in \mathbb{R}^n,\ f(\boldsymbol{x}) \geq f(\boldsymbol{x}_0) + \langle \boldsymbol{u}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle\}. \tag{2.3.20}$$

With these definitions in mind, we might imagine that in the nonsmooth case, a suitable replacement for the gradient algorithm might be the *subgradient method*, which chooses (somehow) $\boldsymbol{g}_k \in \partial f(\boldsymbol{x}_k)$, and then proceeds in the direction of $-\boldsymbol{g}_k$: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - t_k \boldsymbol{g}_k$. Incorporating projection onto the feasible set C, we arrive at the following *projected subgradient algorithm*[22]:

$$\boldsymbol{x}_{k+1} = \mathcal{P}_{\mathsf{C}}[\boldsymbol{x}_k - t_k \boldsymbol{g}_k], \quad \boldsymbol{g}_k \in \partial f(\boldsymbol{x}_k). \tag{2.3.21}$$

[22] Projected subgradient methods were first developed by Naum Shor [Sho85] and Boris Polyak etc. in 1960's.

To apply the projected subgradient method, we need an expression for the subdifferential of the ℓ^1 norm. Figure 2.14 visualizes this. In one dimension, $\|\boldsymbol{x}\|_1 = |x|$; this function is differentiable away from $x = 0$. For $x > 0$, $\partial|\cdot|(x) = \{1\}$, while for $x < 0$, $\partial|\cdot|(x) = \{-1\}$. At $x = 0$, $|x|$ is not differentiable, and there are multiple possible linear lower bounds. Figure 2.14 visualizes three of these lower bounds. It is not difficult to see that lower bounds at $x = 0$ can have any slope from -1 to 1; hence, the subdifferential is

$$\partial|\cdot|(x) = [-1, 1], \quad \text{at } x = 0.$$

The following lemma extends this observation to higher-dimensional $\boldsymbol{x} \in \mathbb{R}^n$:

LEMMA 2.13 (Subdifferential of $\|\cdot\|_1$). *Let $\boldsymbol{x} \in \mathbb{R}^n$, with $\mathsf{I} = \operatorname{supp}(\boldsymbol{x})$,*

$$\partial \|\cdot\|_1 (\boldsymbol{x}) = \{\boldsymbol{v} \in \mathbb{R}^n \mid \boldsymbol{P}_{\mathsf{I}}\boldsymbol{v} = \operatorname{sign}(\boldsymbol{x}),\ \|\boldsymbol{v}\|_\infty \leq 1\}. \tag{2.3.22}$$

Here, $\boldsymbol{P}_{\mathsf{I}} \in \mathbb{R}^{n\times n}$ is the orthoprojector onto coordinates I:

$$[\boldsymbol{P}_{\mathsf{I}}\boldsymbol{v}](j) = \begin{cases} \boldsymbol{v}(j) & j \in \mathsf{I} \\ 0 & j \notin \mathsf{I} \end{cases}. \tag{2.3.23}$$

Proof The subdifferential $\partial \|\cdot\|_1 (\boldsymbol{x})$ consists of all vectors $\boldsymbol{v}$ that satisfy

$$\sum_{i=1}^{n} |\boldsymbol{x}'(i)| \ \geq \ \sum_{i=1}^{n} |\boldsymbol{x}(i)| + \boldsymbol{v}(i)\,(\boldsymbol{x}'(i) - \boldsymbol{x}(i)) \tag{2.3.24}$$

for every $\boldsymbol{x}$ and $\boldsymbol{x}'$. A sufficient condition is that for every index i and every scalar z,

$$|z| \ \geq \ |\boldsymbol{x}(i)| + \boldsymbol{v}(i)(z - \boldsymbol{x}(i)). \tag{2.3.25}$$

Taking $\boldsymbol{x}' = \boldsymbol{x} + (z - \boldsymbol{x}(i))\boldsymbol{e}_i$ in (2.3.24) shows that (2.3.25) is also necessary. If $\boldsymbol{x}(i) = 0$, (2.3.25) becomes $|z| \geq \boldsymbol{v}(i)z$, which holds for all z if and only if $|\boldsymbol{v}(i)| \leq 1$. If $\boldsymbol{x}(i) \neq 0$, the inequality is satisfied if and only if $\boldsymbol{v}(i) = \operatorname{sign}(\boldsymbol{x}(i))$. Hence, $\boldsymbol{v} \in \partial \|\cdot\|_1$ if and only if for all $i \in \mathsf{I}$, $\boldsymbol{v}(i) = \operatorname{sign}(\boldsymbol{x}(i))$, and for all i, $|\boldsymbol{v}(i)| \leq 1$. This conclusion is summarized as (2.3.22). □

The projected subgradient method alternates between subgradient steps, which move in the direction of $-\operatorname{sign}(\boldsymbol{x})$, and orthogonal projections onto the feasible set $\{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}$ according to equation (2.3.13). We obtain a very simple algorithm that solves (2.3.8), which we spell out in detail as Algorithm 2.2.

REMARK 2.14 (Projected Subgradient and Better Alternatives). *In many respects, this is a bad method for solving the ℓ^1 problem. It is correct, but it converges very slowly compared to methods that exploit a certain piece of problem-specific structure, which we will describe in later chapters. The main virtue of Algorithm 2.2 is that it is simple and intuitive, and also serves our exposition*

Algorithm 2.2: ℓ^1**-Minimization by Projected Subgradient**

Input: a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and a vector $\boldsymbol{y} \in \mathbb{R}^m$.
Compute $\boldsymbol{\Gamma} \leftarrow \boldsymbol{I} - \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}\boldsymbol{A}$, and $\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{A}^\dagger \boldsymbol{y} = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}\boldsymbol{y}$.
$\boldsymbol{x}_0 \leftarrow \boldsymbol{0}$.
$t \leftarrow 0$.
repeat many times
    $t \leftarrow t+1$;
    $\boldsymbol{x}_t \leftarrow \tilde{\boldsymbol{x}} + \boldsymbol{\Gamma}\left(\, \boldsymbol{x}_{t-1} - \frac{1}{t}\operatorname{sign}(\boldsymbol{x}_{t-1}) \,\right)$;
end while

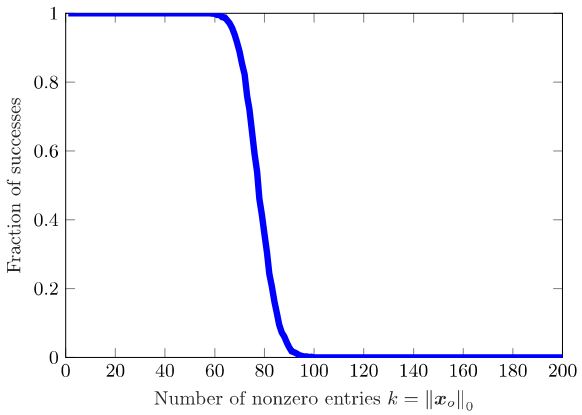


Figure 2.15 Phase Transition in ℓ^1 Minimization. We consider the problem of recovering a sparse vector $\boldsymbol{x}_o$ from measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, where $\boldsymbol{A} \in \mathbb{R}^{100\times 200}$ is a Gaussian matrix. We vary the number of nonzero entries $k = \|\boldsymbol{x}_o\|_0$ across $k = 0, 1, \ldots, 200$, and plot the fraction of instances where ℓ^1 minimization successfully recovers $\boldsymbol{x}_o$, over 50 independent experiments for each value of k. Notice that this probability of success exhibits a (rather sharp) transition from 1 (guaranteed success) to 0 (guaranteed failure) as k increases. Notice moreover, that *for sufficiently well-structured problems (k small), ℓ^1 minimization always succeeds.*

by introducing or reminding us of subgradients and projection operators.[23] *The projected subgradient method for ℓ^1 minimization can be implemented in just a few lines of Matlab code. In Chapter 8, we will systematically develop a number of more advanced optimization methods that can fully utilize the structures in this problem for better efficiency and scalability.*

To see how well does ℓ^1 minimization (as implemented through the projected subgradient method) perform, run `Chapter_2_L1_recovery.m` from the book

[23] Also, we would like you to have a feel for at least one *very* simple way for implementing ℓ^1 minimization in code and to play with it. Our experience is that this helps to think more concretely about the optimization problem and its applications, rather than leaving it as a mathematical abstraction.

website. You may see an interesting phenomenon! Although the method does not *always* succeed, it *does* succeed whenever the target solution $\boldsymbol{x}_o$ is *sufficiently sparse*! Figure 2.15 illustrates this in a more systematic way. In the figure, we generate random matrices $\boldsymbol{A}$ of size 200×400 and random vectors $\boldsymbol{x}_o$ with k nonzero entries. We vary k from 1 to 200. For each k, we run 50 experiments and plot the fraction of trials in which ℓ^1 minimization correctly recovers $\boldsymbol{x}_o$, up to numerical error. Notice that indeed, ℓ^1 minimization succeeds whenever $\boldsymbol{x}_o$ is sufficiently sparse.

2.3.4 Sparse Error Correction via Logan's Phenomenon

In Section 1.2.2 of the introduction chapter, we have discussed the work of Benjamin Logan, who has shown that ℓ^1 minimization can be used to remove sparse errors in band-limited signals. To connect its content more closely to our setting here, let us consider a discretized analogue of the result, in which we consider a finite dimensional signal $\boldsymbol{y} \in \mathbb{C}^n$. Let $\boldsymbol{F} \in \mathbb{C}^{n \times n}$ be the Discrete Fourier Transform (DFT) basis for $\mathbb{C}^n$ (see equation (A.7.13) of Appendix A). That is, we have:

$$F_{kl} = \frac{1}{\sqrt{n}} \exp\left(2\pi \mathfrak{i} \frac{kl}{n}\right), \quad k = 0, \ldots, n-1,\ l = 0, \ldots, (n-1). \tag{2.3.26}$$

Let $\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{(n-1)}$ denote the columns of the DFT matrix:

$$\boldsymbol{F} = \left[\boldsymbol{f}_0 \mid \cdots \mid \boldsymbol{f}_{(n-1)}\right] \quad \in \mathbb{C}^{n \times n}. \tag{2.3.27}$$

Form a submatrix $\boldsymbol{B} \in \mathbb{C}^{n \times (d+1)}$, corresponding to the d lowest-frequency elements of this basis and their conjugates[24]:

$$\boldsymbol{B} = \left[\boldsymbol{f}_{-\frac{d-1}{2}} \mid \cdots \mid \boldsymbol{f}_{\frac{d-1}{2}}\right] \quad \in \mathbb{C}^{n \times (d+1)}, \tag{2.3.28}$$

where we use $\boldsymbol{f}_{-i}$ to indicate the conjugate of $\boldsymbol{f}_i$. Let us imagine that $\boldsymbol{x}_o = \boldsymbol{B}\boldsymbol{w}_o \in \mathrm{col}(\boldsymbol{B})$, and

$$\boldsymbol{y} = \boldsymbol{x}_o + \boldsymbol{e}_o, \tag{2.3.29}$$

where $\|\boldsymbol{e}_o\|_0 \leq k$. Our task is to recover $\boldsymbol{x}_o$ (which is equivalent to removing $\boldsymbol{e}_o$). A discrete analogue of the program suggested in Logan's theorem would be to solve[25]

$$\begin{array}{ll} \min & \|\boldsymbol{y} - \boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{x} \in \mathrm{col}(\boldsymbol{B}). \end{array} \tag{2.3.30}$$

[24] We use pairs of conjugate bases to represent real signals. One may view the range of $\boldsymbol{B}$ as the discretized version of the band-limited functions $\mathcal{B}_1(\Omega)$ introduced earlier in Logan's Theorem 1.5.

[25] For complex vectors, the ℓ^1 norm is simply the sum of absolute values of the real and imaginary parts. Or equivalently, we identify a complex vector in $\mathbb{C}^n$ as a real vector in $\mathbb{R}^{2n}$.

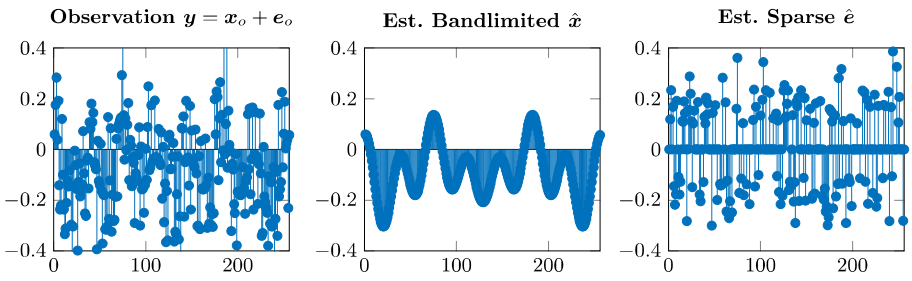


Figure 2.16 Logan's Phenomenon. Left: the superposition $\boldsymbol{y} = \boldsymbol{x}_o + \boldsymbol{e}_o$ of a band-limited signal $\boldsymbol{x}_o$ and a sparse error $\boldsymbol{e}_o$. **Middle:** estimate $\hat{\boldsymbol{x}}$ by ℓ^1 minimization. **Right:** estimate $\hat{\boldsymbol{e}}$ by ℓ^1 minimization. Both estimates are accurate to within relative error 10^{-6}.

This problem is actually very much equivalent to the sparse signal recovery problem discussed so far. To see this, let $\boldsymbol{A}$ be a matrix whose rows span the left null space of $\boldsymbol{B}$ – i.e., $\operatorname{rank}(\boldsymbol{A}) = n - d$, and $\boldsymbol{AB} = \boldsymbol{0}$. Then $\boldsymbol{Ax}_o = \boldsymbol{0}$, and our observation equation (2.3.29) is equivalent to

$$\bar{\boldsymbol{y}} = \boldsymbol{Ae}_o, \tag{2.3.31}$$

where $\bar{\boldsymbol{y}} = \boldsymbol{Ay}$. From this, it is not difficult to argue that the optimization problem (2.3.30) is equivalent to

$$\begin{array}{ll} \min & \|\boldsymbol{e}\|_1 \\ \text{subject to} & \boldsymbol{Ae} = \bar{\boldsymbol{y}}, \end{array} \tag{2.3.32}$$

in the sense that $\boldsymbol{e}_\star$ is an optimal solution to (2.3.32) if and only if $\boldsymbol{y} - \boldsymbol{e}_\star \in \operatorname{col}(\boldsymbol{B})$ is an optimal solution to (2.3.30). Figure 2.16 shows an example of this discrete analogue of Logan's phenomenon. You can reproduce this result by running `E6886_Lecture2_Demo_Logan.m` from the book webpage.

Given the examples we have seen thus far of how sparsity arises in application problems, the phenomenon associated with ℓ^1 minimization is certainly intriguing. In the coming chapters, we will study it first from a mathematical perspective, to understand *why* it occurs and what its limitations are; we will then investigate its implications for practical applications in later chapters.

2.4 Summary

Let us briefly recap what we have learned in this chapter. In many modern data analysis and signal processing applications, we need to solve very large, underdetermined systems of linear equations:

$$\boldsymbol{y} = \boldsymbol{Ax}, \quad \boldsymbol{A} \in \mathbb{R}^{m \times n}, \; m < n.$$

Such problems are inherently ill-posed: they admit infinitely many solutions.

Uniqueness of the Sparse Solution.
To make such problems well-posed, or to make the solution unique, we need to leverage additional properties of the solution that we wish to recover. One important property, which arises in many practical applications, is sparsity (or compressibility). This is a powerful piece of information: although the signals themselves reside in a very high-dimensional space, they have only a few intrinsic degrees of freedom – they can be represented as a linear superposition of just a few atoms from a properly chosen dictionary. As Theorem 2.6 shows, under fairly general conditions, imposing sparsity on $\boldsymbol{x}$ can indeed make the problem of solving

$$\min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

well conditioned: As long as the target solution $\boldsymbol{x}_o$ is sufficiently sparse w.r.t. *the Kruskal rank* of $\boldsymbol{A}$, the sparsest solution to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is unique and is the correct solution.

Tractability of the Sparse Solution via Convex Relaxation.
Computationally, however, finding the sparsest solution to a linear system is in general intractable (i.e., NP-hard, Theorem 2.8). To alleviate the computational difficulty, we relax the ℓ^0 minimization problem and replace the ℓ^0 norm of $\boldsymbol{x}$ with its convex envelope, the ℓ^1 norm:

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

Projected Subgradient Descent.
We have introduced a very basic subgradient descent algorithm (Algorithm 2.2) that solves the convex ℓ^1 minimization problem. From the results of the algorithm, we observe a striking phenomenon that ℓ^1 minimization can effectively recover the sparse solution under fairly broad conditions. We will explain why this is the case in the next chapter after we carefully characterize exact conditions under which ℓ^1 minimization gives the correct sparse solution.

2.5 Notes

Application Vignettes.
Some of the early applications of sparse representation are in signal processing, such as medical imaging [LDP07], seismic signals [HH08], and image processing [YWHM08, MES08]. The three applications described in this chapter illustrate various aspects of sparse modeling and sparse recovery. The medical imaging application is described in the work of Lustig et. al. [LDP07, LDSP08]. The denoising results shown in Section 2.1.2 are due to Mairal et. al. [MES08]. The face recognition formulation in Section 2.1.3 is described in [WYG+09]. The

discussion in this chapter only touches the surface of these problems; we will revisit medical imaging in Chapter 10 and face recognition in Chapter 13. Please see these chapters and their references for broader context and related work on each of these problems. These are just a few of the vast array of applications of sparse methods; a few of these are highlighted in Part III of the book, such as Chapters 11–16.

NP Hardness of ℓ^0 Minimization and Related Problems.
The hardness result for ℓ^0 minimization, Theorem 2.8, is due to Natarajan [Nat95]; see also Davis, Mallat, and Avellaneda [DMA97]. Results of Amaldi and Kann [AK95, AK98] and Arora, Babai, Stern, and Sweedyk [ABSS93] show that ℓ^0 minimization problems are also NP-hard to approximate. Delineating the boundaries between tractable and intractable instances of sparse approximation remains an active topic of research: see, e.g., Zhang, Wainwright, and Jordan [ZWJ14] or Foster, Karloff, and Thaler [FKT15] for more recent developments. There are hardness results for a number of problems that relate closely to sparse approximation. These results also have implications for sparse error correction. There are also hardness results around the problem of *matrix sparsification* in numerical analysis, which seeks to replace a given matrix $\boldsymbol{A}$ with a sparse matrix $\hat{\boldsymbol{A}}$ such that $\text{range}(\boldsymbol{A}) \approx \text{range}(\hat{\boldsymbol{A}})$: see McCormick [McC83], Coleman and Pothen [CP86], and Gottlieb and Neylon [GN16] for discussions of the hardness of this and related problems. Based on reduction techniques similar to that classical complexity theory, the most recent work of Brennan and Bresler [BB20] has systematically studied the gaps between statistical and computational complexity for a broad family of related problems such as sparse linear regression and sparse PCA, as well as many problems related to matrices and tensors that we will study in later chapters.

2.6 Exercises

2.1 (Convexity of ℓ^p Norms). *Show that*

$$\|\boldsymbol{x}\|_p = \Big(\sum_i |x_i|^p \Big)^{1/p} \tag{2.6.1}$$

is convex for $p \geq 1$, and nonconvex for $0 < p < 1$.

2.2. *Show that for $0 < p < 1$, $\|\boldsymbol{x}\|_p$ is not a norm in the sense of Definition 2.1.*

2.3 (Relationship between ℓ^p Norms). *Show that for $p < q$,*

$$\|\boldsymbol{x}\|_p \geq \|\boldsymbol{x}\|_q \tag{2.6.2}$$

for every $\boldsymbol{x}$. For what $\boldsymbol{x}$ is equality obtained (i.e., $\|\boldsymbol{x}\|_p = \|\boldsymbol{x}\|_q$)?

2.4 (Computing the Kruskal Rank). *Write a Matlab function that takes as an input a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and outputs the Kruskal rank* $\operatorname{krank}(\boldsymbol{A})$. *There is no known way to efficiently compute the Kruskal rank. It is fine if your code takes time exponential in n. Corroborate the conclusion of Theorem 2.6, by generating a 4×8 Gaussian matrix $\boldsymbol{A}$, via* `A = randn(4,8)`, *and computing its Kruskal rank.*

2.5 (A Structured Matrix with Small Kruskal Rank). *Consider a 4×8 dimensional complex matrix generated as*

$$\boldsymbol{A} = [\, \boldsymbol{I} \mid \boldsymbol{F} \,], \tag{2.6.3}$$

where $\boldsymbol{I}$ is the 4×4 identity matrix, and $\boldsymbol{F}$ is a 4×4 Discrete Fourier Transform (DFT) matrix: in Matlab, `A = [ eye(4), dftmtx(4) ]`. *Either using your code from Exercise 2.4, or hand calculations, determine the Kruskal rank of $\boldsymbol{A}$. You should find that it is smaller than 4! A general version of this phenomenon can be observed with the* Dirac comb, *which is sparse in both time and frequency.*

2.6 (The Spark). *Results on ℓ^0 uniqueness are sometimes described in terms of the* spark *of a matrix, which is the number of nonzero entries in the sparsest nonzero element of the null space of $\boldsymbol{A}$:*

$$\operatorname{spark}(\boldsymbol{A}) \quad = \quad \min_{\boldsymbol{d} \neq \boldsymbol{0},\ \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}} \|\boldsymbol{d}\|_0.$$

What is the relationship between $\operatorname{spark}(\boldsymbol{A})$ *and* $\operatorname{krank}(\boldsymbol{A})$*?*

2.7 (Kruskal Rank of Random Matrices). *In this exercise we prove that for a generic $m \times n$ matrix $\boldsymbol{A}$ with entries $\sim_{\text{iid}} \mathcal{N}(0,1)$,* $\operatorname{krank}(\boldsymbol{A}) = m$ *with probability one.*

1 *Argue that for* any *$m \times n$ matrix $\boldsymbol{A}$,* $\operatorname{krank}(\boldsymbol{A}) \leq m$.

2 *Let $\boldsymbol{A} = [\boldsymbol{a}_1 \mid \cdots \mid \boldsymbol{a}_n]$ with $\boldsymbol{a}_i \in \mathbb{R}^m$ as column vectors. Let* span *denote the linear span of a collection of vectors. Argue that*

$$\mathbb{P}\left[\boldsymbol{a}_m \in \operatorname{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{m-1})\right] = 0. \tag{2.6.4}$$

3 *Argue that* $\operatorname{krank}(\boldsymbol{A}) < m$ *if and only if there exist some indices $i_1, \ldots, i_m$ such that*

$$\boldsymbol{a}_{i_m} \in \operatorname{span}(\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_m - 1}) \tag{2.6.5}$$

4 *Conclude that* $\operatorname{krank}(\boldsymbol{A}) = m$ *with probability one, by noting that*

$$\begin{aligned}
&\mathbb{P}\left[\exists i_1, \ldots, i_m \ : \ \boldsymbol{a}_{i_m} \in \operatorname{span}(\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{m-1}})\right] \\
&\leq \sum_{i_1, \ldots, i_m} \mathbb{P}\left[\boldsymbol{a}_{i_m} \in \operatorname{span}(\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{m-1}})\right] \\
&\leq m^n \times \underbrace{\mathbb{P}\left[\boldsymbol{a}_m \in \operatorname{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{m-1})\right]}_{=\,0} \\
&= 0.
\end{aligned}$$

2.8 (ℓ^0 Minimization and Typical Examples). *We showed that there is a* worst case *phase transition in ℓ^0 minimization at $\frac{\text{krank}(\boldsymbol{A})}{2}$. This means that ℓ^0 minimization recovers* every *$\boldsymbol{x}_o$ satisfying $\|\boldsymbol{x}_o\|_0 < \frac{\text{krank}(\boldsymbol{A})}{2}$. We also know that for a Gaussian matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\text{krank}(\boldsymbol{A}) = m$, with probability one.*

Using code for ℓ^0 minimization provided (or write your own!), please do the following: generate a 5×12 Gaussian matrix `A = randn(5,12)`. *What is* $\text{rank}(\boldsymbol{A})$*? Generate a sparse vector $\boldsymbol{x}_o$, with four nonzero entries, via* `xo = zeros(12,1); xo(1:4) = randn(4,1)`. *Now, set* `y = A xo`. *Solve the ℓ^0 minimization problem, to find the sparsest vector $\boldsymbol{x}$ satisfying $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$. Is it the same as $\boldsymbol{x}_o$? Check whether* `norm(x - xo)` *is small, where* `x` *is the solution produced by your code.*

Notice that the worst case *theory for ℓ^0 predicts that we can only recover vectors with at most* 2 *nonzero entries. But we have observed ℓ^0 succeeding with* 4 *nonzero entries! This is an example of a* typical case *performance which is better than the worst case.*

Please explain this! Argue that if $\boldsymbol{x}_o$ is a fixed vector supported on some set I of size $< m$, then the probability that there exists a subset $\mathsf{I}' \neq \mathsf{I}$ of size $< m$ satisfying $\boldsymbol{A}\boldsymbol{x}_o \in \text{range}(\boldsymbol{A}_{\mathsf{I}'})$ is zero.

Does your argument imply that the worst case theory based on rank *can be improved? Why or why not?*

2.9 (Subdifferentials). *Compute the subdifferentials for the following functions:*

1 *The subdifferential for $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_\infty$ with $\boldsymbol{x} \in \mathbb{R}^n$.*
2 *The subdifferential for $f(\boldsymbol{X}) = \sum_{j=1}^{n} \|\boldsymbol{X}\boldsymbol{e}_j\|_2$ with $\boldsymbol{X}$ a matrix in $\mathbb{R}^{n\times n}$.*
3 *The subdifferential for $f(\boldsymbol{x}) = \|\boldsymbol{X}\|_*$ with $\boldsymbol{X}$ a matrix in $\mathbb{R}^{n\times n}$.*

2.10 (Implicit Bias of Gradient Descent). *Consider the problem of solving an under-determined system of linear equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m < n$. Of course the solution is not unique. Nevertheless, let us solve it by minimizing the least square error*

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \doteq \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2,$$

say using the simplest gradient descent algorithm:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha \nabla f(\boldsymbol{x}_k).$$

Show that if we initialize $\boldsymbol{x}_0$ as the origin $\boldsymbol{0}$, then when the above gradient descent algorithm converges, it must converge to the solution $\boldsymbol{x}_\star$ of the minimal 2-norm. That is, it converges to the optimal solution of the following problem

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_2^2 \quad \textit{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

This is a phenomenon widely exploited in the practice of learning deep neural networks. Although due to over-parameterized, parameters that minimize the cost function might not be unique, the choice of optimization algorithms with proper initialization (here gradient descent starting from the origin) introduces implicit bias for the optimization path and converges to a desirable solution.

2.11 (Projection onto an Affine Subspace). *In deriving the projected subgradient method for ℓ^1 minimization, we used the fact that for an affine subspace*

$$\mathsf{C} = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\}, \tag{2.6.6}$$

where $\boldsymbol{A}$ is a matrix with full row rank, and $\boldsymbol{y} \in \operatorname{range}(\boldsymbol{A})$, the Euclidean projection on C is given by

$$\mathcal{P}_{\mathsf{C}}[\boldsymbol{z}] = \arg\min_{\boldsymbol{A}\boldsymbol{x}=\boldsymbol{y}} \|\boldsymbol{x} - \boldsymbol{z}\|_2^2 \tag{2.6.7}$$

$$= \boldsymbol{z} - \boldsymbol{A}^* \left(\boldsymbol{A}\boldsymbol{A}^*\right)^{-1} \left[\boldsymbol{A}\boldsymbol{z} - \boldsymbol{y}\right]. \tag{2.6.8}$$

Prove that this formula is correct. You may use the following geometric characterization of $\mathcal{P}_{\mathsf{C}}[\boldsymbol{z}]$: $\boldsymbol{x} = \mathcal{P}_{\mathsf{C}}[\boldsymbol{z}]$ if and only if (i) $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ and (ii) for any $\tilde{\boldsymbol{x}}$ satisfying $\boldsymbol{A}\tilde{\boldsymbol{x}} = \boldsymbol{y}$, we have

$$\langle \boldsymbol{z} - \boldsymbol{x}, \tilde{\boldsymbol{x}} - \boldsymbol{x} \rangle \leq 0. \tag{2.6.9}$$

2.12. *Projected gradient descent aims to:*

$$\min f(\boldsymbol{x}) \quad \textit{subject to} \quad \boldsymbol{x} \in \mathsf{C}.$$

Show an example of when the projection onto set C:

1 does not exist;
2 is not unique.

(Tips: This problem does not have a unique solution, *you can either answer this question by drawing pictures or giving mathematical formula, so use your creativity!)*

2.13 (Sparse Error Correction). *In coding theory and statistics, we often encounter the following situation: we have an observation $\boldsymbol{z}$, which* should *be expressible as $\boldsymbol{B}\boldsymbol{x}$,* except that *some of the entries are corrupted. We can express our corrupted observation as*

$$\underset{\text{observation}}{\boldsymbol{z}} = \underset{\text{encoded message}}{\boldsymbol{B}\boldsymbol{x}} + \underset{\text{sparse corruption}}{\boldsymbol{e}}. \tag{2.6.10}$$

Here $\boldsymbol{z} \in \mathbb{R}^n$ is the observation $\boldsymbol{x} \in \mathbb{R}^r$ is a message of interest; $\boldsymbol{B} \in \mathbb{R}^{n \times r}$ $(n > r)$ is a tall matrix with full column rank r, and $\boldsymbol{e} \in \mathbb{R}^n$ represents any corruption of the message. In many applications, the observation may be subject to corruption which is large in magnitude, but affects only a few of the observations, i.e., $\boldsymbol{e}$ is sparse vector. Let $\boldsymbol{A} \in \mathbb{R}^{(n-r)\times n}$ be a matrix whose rows span the left null space of $\boldsymbol{B}$, i.e., $\operatorname{rank}(\boldsymbol{A}) = n - r$, and $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{0}$. Prove that for any k, (2.6.10) has a solution $(\boldsymbol{x}, \boldsymbol{e})$ with $\|\boldsymbol{e}\|_0 = k$ if and only if the underdetermined system

$$\boldsymbol{A}\boldsymbol{e} = \boldsymbol{A}\boldsymbol{z} \tag{2.6.11}$$

has a solution $\boldsymbol{e}$ with $\|\boldsymbol{e}\|_0 = k$. Argue that that the optimization problems

$$\min_{\boldsymbol{x}} \|\boldsymbol{B}\boldsymbol{x} - \boldsymbol{z}\|_1 \tag{2.6.12}$$

and

$$\min_{\boldsymbol{e}} \|\boldsymbol{e}\|_1 \quad \textit{subject to} \quad \boldsymbol{A}\boldsymbol{e} = \boldsymbol{A}\boldsymbol{z} \tag{2.6.13}$$

are equivalent, in the sense that for every solution $\hat{\boldsymbol{x}}$ *of* (2.6.12), $\hat{\boldsymbol{e}} = \boldsymbol{B}\hat{\boldsymbol{x}} - \boldsymbol{z}$ *is a solution to* (2.6.13)*; and for every solution* $\hat{\boldsymbol{e}}$ *of* (2.6.13)*, there is a solution* $\hat{\boldsymbol{x}}$ *of* (2.6.12) *such that* $\hat{\boldsymbol{e}} = \boldsymbol{B}\hat{\boldsymbol{x}} - \boldsymbol{z}$*.*

It is sometimes observed that "sparse representation and sparse error correction are equivalent." In what sense is this true?

2.14 (ℓ^1 vs. ℓ^∞ minimization). *We have studied the* ℓ^1 *minimization problem*

$$\min \|\boldsymbol{x}\|_1 \quad \textit{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \tag{2.6.14}$$

for recovering sparse $\boldsymbol{x}_o$*. We can obtain other convex optimization problems by replace* $\|\cdot\|_1$ *with* $\|\cdot\|_p$ *for* $p \in (1, \infty]$*. For what kind of* $\boldsymbol{x}_o$ *would you expect* ℓ^∞ *minimization to outperform* ℓ^1 *minimization (in the sense of recovering* $\boldsymbol{x}_o$ *more accurately)?*

2.15 (Faces and Linear Subspaces). *Download* `face_intro_demo.zip` *from the book website. Run* `load_eyb_recognition` *to load a collection of images under varying illumination into memory. The training images (under different lighting) will be stored in* `A_train`*, the identities of the subjects in* `label_train`*. Form a matrix* `B` *by selecting those columns of* `A_train` *that correspond to Subject 1. We will use the* singular value decomposition *to investigate how well-approximated the columns of* `B` *are by a linear subspace.*

Compute the singular values *of* `B` *using* `sigma = svd(B)`*. How many singular values* r *are needed to capture 95% of the energy of* $\boldsymbol{B}$*? That is, to ensure that*

$$\sum_{i=1}^{r} \sigma_i^2 > .95 \times \sum_{i=1}^{n} \sigma_i^2 \; ? \tag{2.6.15}$$

What about 99% of the energy? Repeat this calculation for several subjects.

2.16 (Sparsity of MR Images). *In this exercise, we study the wavelet-domain sparsity of anatomical MRI data from a real dataset, the BOLD5000 fMRI dataset. As we saw in the vignette presented in lecture, the signal acquired in MRI settings is the 2D Fourier transform of the relevant spatial slice of the object being imaged; the specific mathematical details of a modeling and analysis of this acquisition process are presented in Chapter 10.*

The focus in this exercise is on understanding the data, and in particular the relationships between its representations in several transform domains (spatial, 2D Fourier frequency, and 2D discrete wavelet). Since, in this setting, the MR image is sparse in the wavelet domain but acquired in the frequency domain, there is a question of whether the composite acquisition map will have the properties necessary for us to perform recovery from underdetermined measurement maps. We will study such questions in details in later chapters and exercises.

3 Convex Methods for Sparse Signal Recovery

"Algebra is but written geometry; geometry is but drawn algebra."
– Sophie Germain

In the previous chapter, we saw many problems for which the goal is to find a sparse solution to an underdetermined linear system of equations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. This problem is NP-hard in general. However, we also observed that certain well-structured instances *can* be solved efficiently: in experiments, when $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ and $\boldsymbol{x}_o$ was *sufficiently sparse*, tractable ℓ^1 minimization

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \end{array} \tag{3.0.1}$$

exactly recovered $\boldsymbol{x}_o$: $\boldsymbol{x}_o$ was the unique optimal solution to this optimization problem.

The experiments in the previous chapter are inspiring, and perhaps surprising. In this chapter, we will study this phenomenon mathematically, and try to precisely characterize the behavior of (3.0.1). The engineering motivation is simple: we would like to know whether the behavior in the previous chapter is some lucky instances or should be expected in general, and if it is the latter case, whether we can use it to build reliable systems.

3.1 Why Does ℓ^1 Minimization Succeed? Geometric Intuitions

Before diving into a formal proof that the ℓ^1 minimization (3.0.1) correctly recovers sparse signals, we describe two intuitive, geometric pictures of why this is the case.

Coefficient Space Picture.
We first visualize the problem in the space $\mathbb{R}^n$ of coefficient vectors $\boldsymbol{x}$. The set of vectors $\boldsymbol{x}$ that satisfy the constraint $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$ in (3.0.1) is an *affine subspace*[1]

$$\mathsf{S} = \{\boldsymbol{x} \mid \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}\} = \{\boldsymbol{x}_o\} + \operatorname{null}(\boldsymbol{A}). \tag{3.1.1}$$

[1] In (3.1.1), the set addition $\{\boldsymbol{x}_o\} + \operatorname{null}(\boldsymbol{A})$ is in the sense of Minkowski, i.e., for sets S and T, $\mathsf{S} + \mathsf{T} = \{\boldsymbol{s} + \boldsymbol{t} \mid \boldsymbol{s} \in \mathsf{S}, \boldsymbol{t} \in \mathsf{T}\}$.

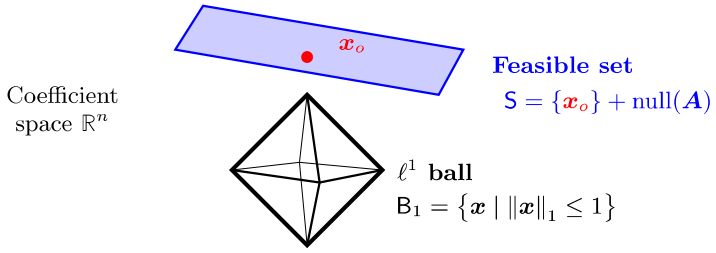


Figure 3.1 Coefficient-Space Picture. The set of all solutions x to the equation $Ax = y$ is an affine subspace S of the coefficient space $\mathbb{R}^n$. The ℓ^1 ball B_1 consists of all coefficient vectors x whose objective function is at most one.

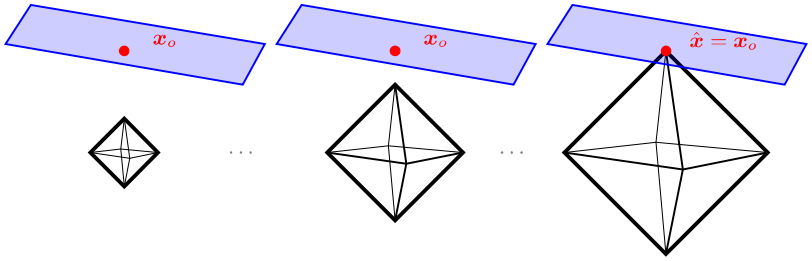


Figure 3.2 ℓ^1 Minimization in the Coefficient-Space Picture. ℓ^1 minimization can be visualized geometrically as follows: we squeeze the ℓ^1 ball down to zero, and then slowly expand it until it first touches the feasible set S. The point (or points) at which it first touches S is the ℓ^1 minimizer $\hat{x}$.

Figure 3.1 visualizes this set. The ℓ^1 minimization problem (3.0.1) picks, out of all of the points in the set S, the one (or ones) with smallest ℓ^1 norm. This can be visualized as follows. Consider the ℓ^1 ball of radius one

$$\mathsf{B}_1 = \{x \mid \|x\|_1 \le 1\} \quad \subset \mathbb{R}^n. \tag{3.1.2}$$

This contains all the vectors x with objective function at most one. Scaling this object by $t \ge 0$ produces the set of vectors x with objective function at most t:

$$t \cdot \mathsf{B}_1 = \{x \mid \|x\|_1 \le t\} \quad \subset \mathbb{R}^n. \tag{3.1.3}$$

If we first scale B_1 down to zero, by setting $t = 0$, and then slowly expand it, by increasing t, the ℓ^1 minimizer is obtained when $t \cdot \mathsf{B}_1$ first touches the affine subspace S. This contact point is the solution to (3.0.1) – see Figure 3.2. From the geometry of the ball, it seems that these contact points will tend to be the vertices or edges of B_1, which precisely correspond to the sparse vectors!

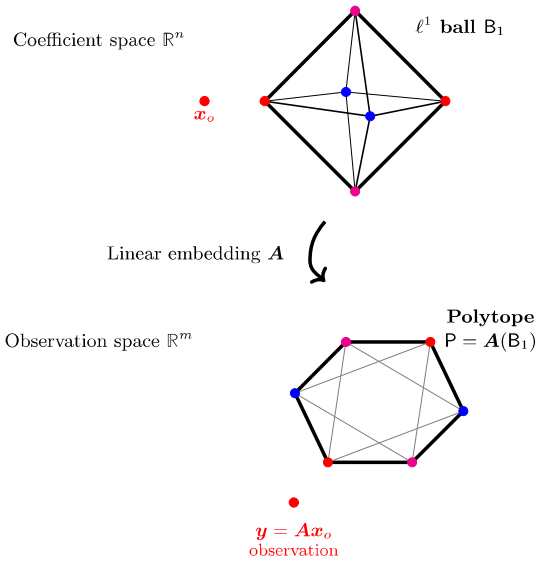


Figure 3.3 Observation-Space Picture. The ℓ^1 ball is a convex polytope B_1 in the coefficient space $\mathbb{R}^n$. The linear map $\boldsymbol{A}$ projects this down to a lower-dimensional set $\mathsf{P} = \boldsymbol{A}(\mathsf{B}_1)$ in the observation space $\mathbb{R}^m$. The vertices $\boldsymbol{v}_i$ of P are subsets of the projections $\boldsymbol{A}\boldsymbol{\nu}_j$ of B_1.

Observation Space Picture

We can also visualize ℓ^1 minimization in the space $\mathbb{R}^m$ of observation vectors $\boldsymbol{y}$. This picture is slightly more complicated, but turns out to be very useful. The $m \times n$ matrix $\boldsymbol{A}$ maps n-dimensional vectors $\boldsymbol{x}$ to $m \ll n$ dimensional vectors $\boldsymbol{y}$. Let us consider how the matrix $\boldsymbol{A}$ acts on the ℓ^1 ball $\mathsf{B}_1 \subset \mathbb{R}^n$. Applying $\boldsymbol{A}$ to each of the vectors $\boldsymbol{x} \in \mathsf{B}_1$, we obtain a lower-dimensional object $\mathsf{P} = \boldsymbol{A}(\mathsf{B}_1)$, which we visualize in Figure 3.3 (right). The lower-dimensional set P is a *convex polytope*. Every vertex $\boldsymbol{v}$ of P is the image $\boldsymbol{A}\boldsymbol{\nu}$ of some vertex $\boldsymbol{\nu} = \pm\boldsymbol{e}_i$ of B_1. More generally, every k-dimensional face of P is the image of some face of B_1.

The polytope P consists of all points $\boldsymbol{y}'$ of the form $\boldsymbol{A}\boldsymbol{x}'$ for some $\boldsymbol{x}'$ with objective function $\|\boldsymbol{x}'\|_1 \leq 1$. ℓ^1 minimization corresponds to squeezing B_1 down to the origin, and then slowly expanding it until it first touches $\boldsymbol{y}$. The touching point is the image $\boldsymbol{A}\hat{\boldsymbol{x}}$ of the ℓ^1 minimizer – see Figure 3.4.

So, ℓ^1 will correctly recover $\boldsymbol{x}_o$ whenever $\boldsymbol{A}\boldsymbol{x}_o$ is on the outside of $\mathsf{P} = \boldsymbol{A}(\mathsf{B}_1)$. For example, in Figure 3.3, all of the vertices of B_1 map to the outside of $\boldsymbol{A}(\mathsf{B}_1)$, and so ℓ^1 recovers any 1-sparse $\boldsymbol{x}_o$. However, certain edges (one-dimensional faces) of B_1 map to the inside of $\boldsymbol{A}(\mathsf{B}_1)$. ℓ^1 minimization will not recover these $\boldsymbol{x}_o$.

From this picture, it may be very surprising that ℓ^1 works as well as it does. However, as we will see in the remainder of this chapter, the high-dimensional picture differs significantly from the low-dimensional picture (and our intuition!)

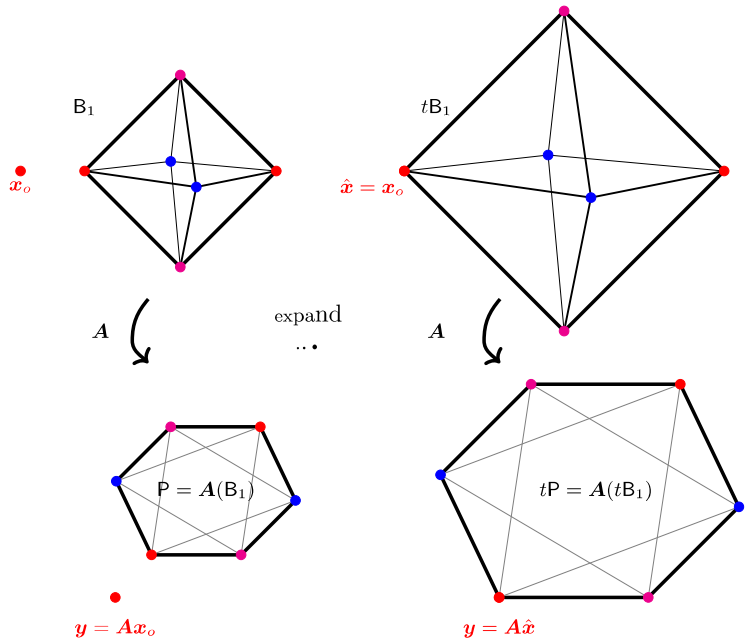


Figure 3.4 ℓ^1 **Minimization in the Observation-Space Picture.** ℓ^1 minimization corresponds to scaling B_1 down to zero, and then slowly expanding it. As B_1 expands, so does $\mathsf{P} = \boldsymbol{A}(\mathsf{B}_1)$. The optimal value for the ℓ^1 minimization problem is the first scalar t such that $t\mathsf{P} = \boldsymbol{A}(t\mathsf{B}_1)$ touches the observation vector $\boldsymbol{y}$. The first point that touches $\boldsymbol{y}$ is the image $\boldsymbol{A}\hat{\boldsymbol{x}}$ of the ℓ^1 minimizer $\hat{\boldsymbol{x}}$. This means that ℓ^1 *minimization recovers point* $\boldsymbol{x}_o$ *if and only if* $\boldsymbol{A}\frac{\boldsymbol{x}_o}{\|\boldsymbol{x}_o\|_1}$ *lies on the boundary of* P.

in ways that are very useful – a "blessing of dimensionality." In particular, if we are in m dimensions and n is proportional to m, not only do all of the vertices of B_1 map to the outside of $\boldsymbol{A}(\mathsf{B}_1)$, so do all the one-dimensional faces, and all of the two-dimensional faces, and so on, all the way up to k-dimensional faces with k proportional to m!

3.2 A First Correctness Result for Incoherent Matrices

With solid empirical evidence and a bit of geometric intuition at hand, our next task is to develop some rigorous understanding of this phenomenon.

3.2.1 Coherence of a Matrix

What determines whether ℓ^1 minimization can recover a target sparse solution $\boldsymbol{x}_o$? Our discussion on ℓ^0 minimization isolated two key factors: how structured the target $\boldsymbol{x}_o$ is (i.e., how many nonzero entries) and how nice the map $\boldsymbol{A}$ is (mea-

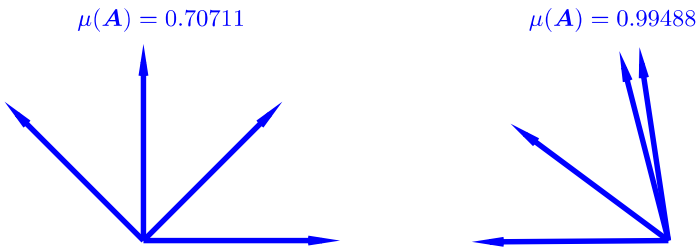


Figure 3.5 Mutual Coherence for Two Configurations of Columns of $\boldsymbol{A}$. Left: well-spread vectors in $\mathbb{S}^2$: $\mu(\boldsymbol{A}) \approx 0.707$. This is the smallest achievable μ for four vectors in two dimensions. In higher dimensions, the mutual coherence can be *much* smaller: for example, a random $m \times 2m$ dimensional matrix has coherence on the order of $\sqrt{\log(m)/m}$, which diminishes to zero as m increases. **Right:** $\mu(\boldsymbol{A}) \approx 0.995$. Mutual coherence depends on the closest pair $\boldsymbol{a}_i, \boldsymbol{a}_j$, and so in this example it is very large.

sured there through the Kruskal rank). Moreover, there was a tradeoff between the two factors: *the nicer $\boldsymbol{A}$ is, the denser $\boldsymbol{x}_o$ we can recover.*

In fact, this qualitative tradeoff carries over to tractable algorithms such as the ℓ^1 relaxation as well. However, we need a slightly stronger notion of the "niceness" of $\boldsymbol{A}$ to guarantee that the tractable relaxation succeeds. Our first notion measures how "spread out" the columns of $\boldsymbol{A}$ are in the high dimensional space $\mathbb{R}^m$:

Definition 3.1 (Mutual Coherence). *For a matrix*

$$\boldsymbol{A} = \left[\boldsymbol{a}_1 \mid \boldsymbol{a}_2 \mid \cdots \mid \boldsymbol{a}_n \right] \quad \in \mathbb{R}^{m \times n}$$

with nonzero columns, the mutual coherence $\mu(\boldsymbol{A})$ *is the largest normalized inner product between two distinct columns:*

$$\mu(\boldsymbol{A}) = \max_{i \neq j} \left| \left\langle \frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|_2}, \frac{\boldsymbol{a}_j}{\|\boldsymbol{a}_j\|_2} \right\rangle \right|. \tag{3.2.1}$$

As the mutual coherence only depends on the direction of the column vectors, for simplicity, we typically assume the columns are normalized to be of unit length.

The mutual coherence takes values in $[0, 1]$. If the columns of $\boldsymbol{A}$ are orthogonal, $\mu(\boldsymbol{A})$ is zero. If $n > m$, the columns of $\boldsymbol{A}$ cannot be orthogonal. The quantity $\mu(\boldsymbol{A})$ captures how close they are to orthogonal, in the worst case sense. Matrices with small $\mu(\boldsymbol{A})$ have columns that are more spread out; we will see that such matrices tend to be better for sparse recovery, in the sense that ℓ^1 succeeds in recovering denser $\boldsymbol{x}_o$. Figure 3.5 visualizes the columns $\boldsymbol{A}$ and displays the coherence, for two examples of $\boldsymbol{A} \in \mathbb{R}^{2 \times n}$.

One intuition for why small $\mu(\boldsymbol{A})$ is helpful is the following: suppose that

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with $\boldsymbol{x}_o$ sparse, and I the support of $\boldsymbol{x}_o$. Then $\boldsymbol{y} = \sum_{i\in\mathsf{I}} \boldsymbol{a}_i \boldsymbol{x}_o(i)$. Intuitively speaking, it should be easier to "guess" which columns $\boldsymbol{a}_i$ participate in this linear combination if distinct columns are not too similar to each other.

To connect the mutual coherence more formally to sparse recovery, we will show that whenever $\mu(\boldsymbol{A})$ is small, the Kruskal rank $\operatorname{krank}(\boldsymbol{A})$ is large. Recall that $\operatorname{krank}(\boldsymbol{A}) \geq k$ if and only if every subset of k columns of $\boldsymbol{A}$ is linearly independent, i.e., every k-column submatrix $\boldsymbol{A}_\mathsf{I}$ has full column rank. In fact, if the coherence $\mu(\boldsymbol{A})$ is small, then column submatrices of $\boldsymbol{A}$ not only have full column rank – they are even *well-conditioned*, in the sense that their smallest singular value $\sigma_{\min}$ is not far from their largest singular value $\sigma_{\max}$. To see this, let $\mathsf{I} \subset [n]$ with $k = |\mathsf{I}|$. Write diagonal and off diagonal entries as:

$$\boldsymbol{A}_\mathsf{I}^* \boldsymbol{A}_\mathsf{I} = \boldsymbol{I} + \boldsymbol{\Delta}. \tag{3.2.2}$$

Because $\|\boldsymbol{\Delta}\| \leq \|\boldsymbol{\Delta}\|_F < k \|\boldsymbol{\Delta}\|_\infty \leq k\mu(\boldsymbol{A})$,[2] we have

$$1 - k\mu(\boldsymbol{A}) \;<\; \sigma_{\min}(\boldsymbol{A}_\mathsf{I}^* \boldsymbol{A}_\mathsf{I}) \;\leq\; \sigma_{\max}(\boldsymbol{A}_\mathsf{I}^* \boldsymbol{A}_\mathsf{I}) \;<\; 1 + k\mu(\boldsymbol{A}). \tag{3.2.3}$$

In particular, if $k\mu(\boldsymbol{A}) \leq 1$, $\boldsymbol{A}_\mathsf{I}$ has full column rank. Combining this observation with our previous discussion of the Kruskal rank, we obtain:

PROPOSITION 3.2 (Coherence Controls Kruskal Rank). *For any* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$,

$$\operatorname{krank}(\boldsymbol{A}) \;\geq\; \frac{1}{\mu(\boldsymbol{A})}. \tag{3.2.4}$$

In particular, if $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ *and*

$$\|\boldsymbol{x}_o\|_0 \;\leq\; \frac{1}{2\mu(\boldsymbol{A})}, \tag{3.2.5}$$

then $\boldsymbol{x}_o$ *is the unique optimal solution to the* ℓ^0 *minimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_0 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.2.6}$$

Thus, provided $\mu(\boldsymbol{A})$ is small enough, ℓ^0 minimization will uniquely recover $\boldsymbol{x}_o$.

3.2.2 Correctness of ℓ^1 Minimization

The previous result showed that if $\mu(\boldsymbol{A})$ is small, then ℓ^0 minimization recovers sufficiently sparse $\boldsymbol{x}_o$. The next result shows that under the same hypotheses, if $\mu(\boldsymbol{A})$ is small, the *tractable* ℓ^1 minimization heuristic also recovers $\boldsymbol{x}_o$. This implies that sparse solutions can be reliably obtained using efficient algorithms! The result is as follows:

[2] The first inequality comes because the operator norm is always bounded by the Frobenius norm: $\|\boldsymbol{\Delta}\| = \max_i \sigma_i(\boldsymbol{\Delta})$ and $\|\boldsymbol{\Delta}\|_F = \sqrt{\sum_i \sigma_i^2(\boldsymbol{\Delta})}$. The second inequality arises because $\|\boldsymbol{\Delta}\|_F^2 = \sum_{ij} |\boldsymbol{\Delta}_{ij}|^2$. The diagonal entries of $\boldsymbol{\Delta}$ are zero, and so in this case, $\|\boldsymbol{\Delta}\|_F^2 = \sum_{i\neq j} |\boldsymbol{\Delta}_{ij}|^2 \leq k(k-1)\|\boldsymbol{\Delta}\|_\infty^2$.

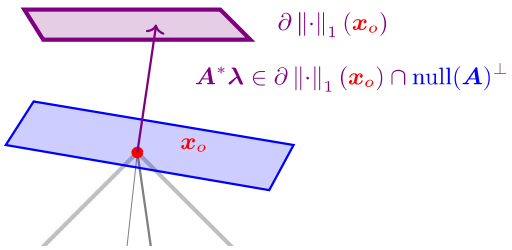


Figure 3.6 Geometry of the Proof of ℓ^1 Recovery. We prove that $\boldsymbol{x}_o$ is an optimal solution to the ℓ^1 minimization problem, by demonstrating that there exists $\boldsymbol{\lambda}$ such that $\boldsymbol{A}^*\boldsymbol{\lambda}$ is in the subdifferential of $\partial \|\cdot\|_1 (\boldsymbol{x}_o)$. In this picture, there is a *subgradient* of the objective which is orthogonal to $\text{null}(\boldsymbol{A})$. This generalizes the condition for projecting onto an affine subspace (Figure 2.13), in which the gradient of the approximation error is orthogonal to $\text{null}(\boldsymbol{A})$.

THEOREM 3.3 (ℓ^1 Succeeds under Incoherence). *Let $\boldsymbol{A}$ be a matrix whose columns have unit ℓ^2 norm, and let $\mu(\boldsymbol{A})$ denote its mutual coherence. Suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with*

$$\|\boldsymbol{x}_o\|_0 \leq \frac{1}{2\mu(\boldsymbol{A})}. \tag{3.2.7}$$

Then $\boldsymbol{x}_o$ is the unique optimal solution to the problem

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \end{array} \tag{3.2.8}$$

REMARK 3.4. *It is possible to improve the condition of Theorem 3.3 slightly, to allow recovery of $\boldsymbol{x}_o$ satisfying*

$$\|\boldsymbol{x}_o\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right). \tag{3.2.9}$$

This is the best possible statement of this form: there exist examples of $\boldsymbol{A}$ and $\boldsymbol{x}_o$ with $\|\boldsymbol{x}_o\|_0 > \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$ for which ℓ^1 minimization does not recover $\boldsymbol{x}_o$. Nevertheless, we will see later in this chapter that for certain classes of $\boldsymbol{A}$ of practical importance, far better guarantees are possible, and that this has important implications for sensing, error correction, and a number of related problems.

Proof Ideas for ℓ^1 Recovery.
Before embarking on a rigorous proof of Theorem 3.3, we sketch our approach. Recall from the previous chapter that for any $\boldsymbol{v} \in \partial \|\cdot\|_1 (\boldsymbol{x}_o)$ and $\boldsymbol{x}' \in \mathbb{R}^n$, the subgradient inequality,

$$\|\boldsymbol{x}'\|_1 \geq \|\boldsymbol{x}_o\|_1 + \langle \boldsymbol{v}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle \tag{3.2.10}$$

lower bounds the ℓ^1 norm of $\boldsymbol{x}'$. Notice that if $\boldsymbol{x}'$ is feasible for (3.2.8), then $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}'$ and so $\boldsymbol{A}(\boldsymbol{x}' - \boldsymbol{x}_o) = \boldsymbol{0}$. Hence, for any $\boldsymbol{\lambda} \in \mathbb{R}^m$,

$$\langle \boldsymbol{A}^*\boldsymbol{\lambda}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle = \langle \boldsymbol{\lambda}, \boldsymbol{A}(\boldsymbol{x}' - \boldsymbol{x}_o) \rangle = 0. \tag{3.2.11}$$

So *if* we can produce a $\boldsymbol{\lambda}$ such that $\boldsymbol{A}^*\boldsymbol{\lambda} \in \partial \left\| \cdot \right\|_1 (\boldsymbol{x}_o)$, plugging into (3.2.10) we necessarily have

$$\left\| \boldsymbol{x}' \right\|_1 \geq \left\| \boldsymbol{x}_o \right\|_1 \tag{3.2.12}$$

for every $\boldsymbol{x}' \in \mathbb{R}^n$. This implies that $\boldsymbol{x}_o$ is *an* optimal solution. Figure 3.6 visualizes this construction geometrically.

Let I denote the support of $\boldsymbol{x}_o$, and $\boldsymbol{\sigma} = \operatorname{sign}(\boldsymbol{x}_{o\mathsf{I}}) \in \{\pm 1\}^k$. Recall that the subdifferential $\partial \left\| \cdot \right\|_1 (\boldsymbol{x}_o)$ consists of those vectors $\boldsymbol{v}$ such that

$$\boldsymbol{v}_{\mathsf{I}} = \boldsymbol{\sigma}, \tag{3.2.13}$$

$$\left\| \boldsymbol{v}_{\mathsf{I}^c} \right\|_\infty \leq 1. \tag{3.2.14}$$

Hence, the condition $\boldsymbol{A}^*\boldsymbol{\lambda} \in \partial \left\| \cdot \right\|_1 (\boldsymbol{x}_o)$ places two conditions on the vector $\boldsymbol{A}^*\boldsymbol{\lambda}$:

$$\boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{\lambda} = \boldsymbol{\sigma}, \tag{3.2.15}$$

$$\left\| \boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{\lambda} \right\|_\infty \leq 1. \tag{3.2.16}$$

The first condition is a linear system of k equations, in m unknowns $\boldsymbol{\lambda}$. The second is a system of $n - k$ inequality constraints. The system of equations (3.2.15) is underdetermined. Our approach will be to look at the simplest possible solution to this underdetermined system,

$$\hat{\boldsymbol{\lambda}}_{\ell^2} = \boldsymbol{A}_{\mathsf{I}}(\boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{A}_{\mathsf{I}})^{-1}\boldsymbol{\sigma}. \tag{3.2.17}$$

This putative solution automatically satisfies the equality constraints (3.2.15). Moreover, $\hat{\boldsymbol{\lambda}}_{\ell^2}$ is a superposition of the columns of $\boldsymbol{A}_{\mathsf{I}}$. Because $\mu(\boldsymbol{A})$ is small, the columns of $\boldsymbol{A}_{\mathsf{I}^c}$ are almost orthogonal to the columns of $\boldsymbol{A}_{\mathsf{I}}$, and so $\left\| \boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{\lambda} \right\|_\infty$ is also small.

Below, we make the above discussion rigorous. The details are slightly more complicated than the above sketch, because we wish to prove that $\boldsymbol{x}_o$ is not just *an* optimal solution, but actually *the unique* optimal solution. We will see that if we can ensure that $\boldsymbol{A}_{\mathsf{I}}$ has full column rank and $\left\| \boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{\lambda} \right\|_\infty$ is strictly smaller than one, this follows.

Proof of Theorem 3.3 Let $\mathsf{I} = \operatorname{supp}(\boldsymbol{x}_o)$ and $\boldsymbol{\sigma} = \operatorname{sign}(\boldsymbol{x}_{o\mathsf{I}}) \in \{\pm 1\}^k$. Notice that $\sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{A}_{\mathsf{I}}) > 1 - k\mu(\boldsymbol{A})$, and so under our assumption $\boldsymbol{A}_{\mathsf{I}}$ has full column rank. Suppose that there exists $\boldsymbol{\lambda}$ such that

$$\boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{\lambda} = \boldsymbol{\sigma}, \tag{3.2.18}$$

$$\left\| \boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{\lambda} \right\|_\infty \leq 1. \tag{3.2.19}$$

Consider any $\boldsymbol{x}'$ which is feasible, i.e., satisfies $\boldsymbol{A}\boldsymbol{x}' = \boldsymbol{y}$. Let $\boldsymbol{v} \in \mathbb{R}^n$ be a vector

such that $\boldsymbol{v}_{\mathsf{I}} = \boldsymbol{\sigma}$, and $\boldsymbol{v}_{\mathsf{I}^c} = \operatorname{sign}([\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c})$. Notice that $\boldsymbol{v} \in \partial \|\cdot\|_1 (\boldsymbol{x}_o)$, and so by the subgradient inequality,

$$\|\boldsymbol{x}'\|_1 \geq \|\boldsymbol{x}_o\|_1 + \langle \boldsymbol{v}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle . \tag{3.2.20}$$

Since $\boldsymbol{x}' - \boldsymbol{x}_o \in \operatorname{null}(\boldsymbol{A})$, $\langle \boldsymbol{A}^* \boldsymbol{\lambda}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle = 0$, and the above equation implies that

$$\begin{aligned}
\|\boldsymbol{x}'\|_1 &\geq \|\boldsymbol{x}_o\|_1 + \langle \boldsymbol{v}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle \\
&= \|\boldsymbol{x}_o\|_1 + \langle \boldsymbol{v} - \boldsymbol{A}^* \boldsymbol{\lambda}, \boldsymbol{x}' - \boldsymbol{x}_o \rangle \\
&= \|\boldsymbol{x}_o\|_1 + \langle \boldsymbol{v}_{\mathsf{I}^c} - \boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{\lambda}, [\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c} \rangle \\
&\geq \|\boldsymbol{x}_o\|_1 + \|[\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c}\|_1 - \|\boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{\lambda}\|_\infty \|[\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c}\|_1 \\
&= \|\boldsymbol{x}_o\|_1 + (1 - \|\boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{\lambda}\|_\infty) \|[\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c}\|_1 .
\end{aligned} \tag{3.2.21}$$

Since $\|\boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{\lambda}\|_\infty < 1$, either $\|\boldsymbol{x}'\|_1 > \|\boldsymbol{x}_o\|_1$, or $\|[\boldsymbol{x}' - \boldsymbol{x}_o]_{\mathsf{I}^c}\|_1 = 0$. In the latter case, this means that $\operatorname{supp}(\boldsymbol{x}') \subseteq \mathsf{I}$, and $\boldsymbol{x}'_{\mathsf{I}} - \boldsymbol{x}_{o\mathsf{I}} \in \operatorname{null}(\boldsymbol{A}_{\mathsf{I}})$. Since $\boldsymbol{A}_{\mathsf{I}}$ has full column rank, this implies that $\boldsymbol{x}'_{\mathsf{I}} = \boldsymbol{x}_{o\mathsf{I}}$, and so $\boldsymbol{x}' = \boldsymbol{x}$.

Hence, if we can construct a $\boldsymbol{\lambda}$ satisfying (3.2.18)-(3.2.19), then any alternative feasible solution $\boldsymbol{x}'$ has larger ℓ^1-norm than $\boldsymbol{x}_o$. Let us try to produce such a $\boldsymbol{\lambda}$. The first equation (3.2.18) above is an underdetermined linear system of equations, with k equations and $m > k$ unknowns $\boldsymbol{\lambda}$. Let us write down one particular solution to this system of equations:

$$\hat{\boldsymbol{\lambda}}_{\ell^2} = \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \boldsymbol{\sigma}. \tag{3.2.22}$$

By construction, $\boldsymbol{A}_{\mathsf{I}}^* \hat{\boldsymbol{\lambda}}_{\ell^2} = \boldsymbol{\sigma}$. We are just left to verify (3.2.19), by calculating

$$\left\| \boldsymbol{A}_{\mathsf{I}^c}^* \hat{\boldsymbol{\lambda}}_{\ell^2} \right\|_\infty = \left\| \boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \boldsymbol{\sigma} \right\|_\infty . \tag{3.2.23}$$

Consider a single element of this vector, which has the form (for some $j \in \mathsf{I}^c$) of

$$|\boldsymbol{a}_j^* \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \boldsymbol{\sigma}| \leq \underbrace{\|\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{a}_j\|_2}_{\leq \sqrt{k}\mu} \underbrace{\left\| (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \right\|_{2,2}}_{< \frac{1}{1 - k\mu(\boldsymbol{A})}} \underbrace{\|\boldsymbol{\sigma}\|_2}_{=\sqrt{k}} \tag{3.2.24}$$

$$< \frac{k\mu(\boldsymbol{A})}{1 - k\mu(\boldsymbol{A})} \tag{3.2.25}$$

$$\leq \underbrace{1}_{\text{Provided } k\mu(\boldsymbol{A}) \leq 1/2.} . \tag{3.2.26}$$

In (3.2.25), we have used that for any invertible $\boldsymbol{M}$, $\left\| \boldsymbol{M}^{-1} \right\| = 1/\sigma_{\min}(\boldsymbol{M})$ and our previous calculation that $\sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \geq 1 - k\mu(\boldsymbol{A})$ to bound $\left\| (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \right\|_{2,2}$. This calculation shows that under our assumptions, condition (3.2.19) is verified. □

3.2.3 Constructing an Incoherent Matrix

In Theorem 3.3, we have shown that if $\|\boldsymbol{x}_o\|_0 \leq 1/2\mu(\boldsymbol{A})$, $\boldsymbol{x}_o$ is correctly recovered by ℓ_1 minimization. Many extensions and variants of this result are known. According to this result, matrices with smaller coherence admit better bounds.

Historically, results of this nature were first proved for special $\boldsymbol{A}$, which consisted of a concatenation of two orthonormal bases:

$$\boldsymbol{A} = [\boldsymbol{\Phi} \mid \boldsymbol{\Psi}], \tag{3.2.27}$$

with $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \mid \cdots \mid \boldsymbol{\phi}_n] \in \mathsf{O}(n)$, $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1 \mid \cdots \mid \boldsymbol{\psi}_n] \in \mathsf{O}(n)$. For instance, $\boldsymbol{\Phi}$ can be the classic Fourier transform bases and $\boldsymbol{\Psi}$ certain wavelet transform bases. In this case, it is possible to prove a sharper bound based on the cross-coherence:

$$\max_{ij} \left| \langle \boldsymbol{\phi}_i, \boldsymbol{\psi}_j \rangle \right|. \tag{3.2.28}$$

Another case which is of great interest is when the matrix $\boldsymbol{A}$ has the form $\boldsymbol{A} = \boldsymbol{\Phi}_{\mathsf{I}}^* \boldsymbol{\Psi}$, where $\mathsf{I} \subset [n]$, and $\boldsymbol{\Phi}_{\mathsf{I}} \in \mathbb{R}^{n \times |\mathsf{I}|}$ is a submatrix of an orthogonal base. For example, in the MRI problem in the previous chapter, $\boldsymbol{\Phi}$ would correspond to the Fourier transform, while $\boldsymbol{\Psi}$ was the basis of sparsity (e.g., wavelets).

As it turns out, incoherence is a generic property for almost all matrices. So the easiest way to build a matrix $\boldsymbol{A}$ with small $\mu(\boldsymbol{A})$ is simply to choose the matrix at random. The following theorem makes this precise:

THEOREM 3.5. *Let* $\boldsymbol{A} = [\boldsymbol{a}_1 \mid \cdots \mid \boldsymbol{a}_n]$ *with columns* $\boldsymbol{a}_i \sim \mathrm{uni}(\mathbb{S}^{m-1})$ *chosen independently according to the uniform distribution on the sphere. Then with probability at least* $3/4$,

$$\mu(\boldsymbol{A}) \;\leq\; C\sqrt{\frac{\log n}{m}}, \tag{3.2.29}$$

where $C > 0$ *is a numerical constant.*

This result is essentially just a calculation. The main tool needed is the following result, which observes that a Lipschitz function on the sphere concentrates sharply about its median:

THEOREM 3.6 (Spherical Measure Concentration). *Let* $\boldsymbol{u} \sim \mathrm{uni}(\mathbb{S}^{m-1})$ *be distributed according to the uniform distribution on the sphere. Let* $f : \mathbb{S}^{m-1} \to \mathbb{R}$ *be an* 1*-Lipschitz function:*

$$\forall\, \boldsymbol{u}, \boldsymbol{u}', \quad |f(\boldsymbol{u}) - f(\boldsymbol{u}')| \;\leq\; 1 \cdot \|\boldsymbol{u} - \boldsymbol{u}'\|_2, \tag{3.2.30}$$

and let $\mathrm{med}(f)$ *denote any median of the random variable* $Z = f(\boldsymbol{u})$. *Then*

$$\mathbb{P}\left[f(\boldsymbol{u}) > \mathrm{med}(f) + t \right] \;\leq\; 2\exp\left(-\frac{mt^2}{2}\right), \tag{3.2.31}$$

$$\mathbb{P}\left[f(\boldsymbol{u}) < \mathrm{med}(f) - t \right] \;\leq\; 2\exp\left(-\frac{mt^2}{2}\right). \tag{3.2.32}$$

This result is the precise reason behind the counterintuitive example about the sphere shown in Figure 1.10 of the Introduction chapter. We have laid out some basic facts in measure concentration and their proofs in the Appendix E. For a more detailed introduction to measure concentration, the reader may refer to [Led01, Mat02]. For now, we will take this result for granted and use it to prove our Theorem 3.5.

Proof of Theorem 3.5: For any fixed $\boldsymbol{v} \in \mathbb{S}^{m-1}$, we have

$$||\boldsymbol{v}^*\boldsymbol{a}| - |\boldsymbol{v}^*\boldsymbol{a}'|| \;\leq\; |\boldsymbol{v}^*(\boldsymbol{a}-\boldsymbol{a}')| \;\leq\; \|\boldsymbol{a}-\boldsymbol{a}'\|_2 . \tag{3.2.33}$$

So, the function $f(\boldsymbol{a}) = |\boldsymbol{v}^*\boldsymbol{a}|$ is 1-Lipschitz. A quick calculation shows that for $\boldsymbol{a} \sim \mathrm{uni}(\mathbb{S}^{m-1})$, we have

$$\mathbb{E}[(\boldsymbol{v}^*\boldsymbol{a})^2] = \frac{1}{m}. \tag{3.2.34}$$

As x^2 is convex, $\mathbb{E}\left[|\boldsymbol{v}^*\boldsymbol{a}|\right]^2 \leq \mathbb{E}\left[(\boldsymbol{v}^*\boldsymbol{a})^2\right]$. So, we have $\mathbb{E}\left[|\boldsymbol{v}^*\boldsymbol{a}|\right] \leq \frac{1}{\sqrt{m}}$.

Applying the Markov inequality $\mathbb{P}\left[X \geq a\right] \leq \frac{\mathbb{E}[X]}{a}$ to f with $a = \mathrm{med}(f)$, then any median of f satisfies

$$\mathrm{med}(f) \;\leq\; 2\mathbb{E}[f] \;\leq\; \frac{2}{\sqrt{m}}. \tag{3.2.35}$$

Finally applying the measure concentration fact from Theorem 3.6, we have

$$\mathbb{P}\left[|\boldsymbol{v}^*\boldsymbol{a}| > \frac{2+t}{\sqrt{m}}\right] \leq 2\exp\left(-\frac{t^2}{2}\right). \tag{3.2.36}$$

Since this holds for every fixed $\boldsymbol{v}$, it also holds if $\boldsymbol{v}$ is an independent random vector uniformly distributed on $\mathbb{S}^{m-1}$. So,

$$\mathbb{P}\left[|\boldsymbol{a}_i^*\boldsymbol{a}_j| > \frac{2+t}{\sqrt{m}}\right] \;\leq\; 2\exp\left(-\frac{t^2}{2}\right). \tag{3.2.37}$$

Summing the failure probability over all $n(n-1)/2$ pairs of distinct $(\boldsymbol{a}_i, \boldsymbol{a}_j)$, we have an upper (union) bound on the probability of all failure events:

$$\mathbb{P}\left[\exists\,(i,j) \;:\; |\boldsymbol{a}_i^*\boldsymbol{a}_j| > \frac{2+t}{\sqrt{m}}\right] \;\leq\; n(n-1)\exp\left(-\frac{t^2}{2}\right). \tag{3.2.38}$$

Setting $t = 2\sqrt{\log 2n}$, the above probability is less than $1/4$ and we obtain the result. □

There are several points about Theorem 3.5 that are worth remarking on here. First, there is nothing particularly special about the success probability 3/4. By a slightly different choice of t (which affects the constant C), one can make the success probability arbitrarily close to 1. Second, there is nothing particularly special about the uniform distribution on $\mathbb{S}^{m-1}$ – many distributions will produce similar results, although this one is especially convenient to analyze.

Figure 3.7 plots the average mutual coherence of matrices sampled according to Theorem 3.5, for various values of n and $m = n/8$. The observations seem to agree with the predictions of the theorem: the average observed mutual coherence is very close to $1.75\sqrt{\frac{\log n}{m}}$.

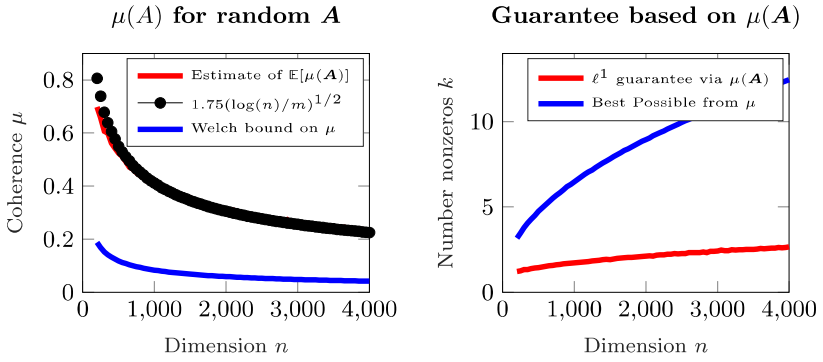


Figure 3.7 How Does Coherence Decay with Dimension? Left: Average mutual coherence across 50 trials, for $\boldsymbol{A}$ with columns $\boldsymbol{a}_i \sim_{iid} \text{uniform}(\mathbb{S}^{m-1})$, for various values of n and $m = n/8$. The black curve, given for reference, is $1.75\sqrt{\frac{\log n}{m}}$. The blue curve is the Welch lower bound $\mu_{\min}$ on the smallest achievable mutual coherence for an $m \times n$ matrix (see Theorem 3.7). **Right:** Average number of nonzeros k which can we can guarantee to reconstruct using the observe $\mu(\boldsymbol{A})$ and Theorem 3.3 (red). The blue curve bounds the best possible number of nonzero entries using Theorem 3.3, for *any* matrix $\boldsymbol{A}$ of size $m \times n$, using the Welch bound.

3.2.4 Limitations of Incoherence

Theorem 3.3 gives a quantitative tradeoff between niceness of $\boldsymbol{A}$ and sparsity of $\boldsymbol{x}_o$, which asserts that when $\boldsymbol{x}_o$ is sparse enough: $\|\boldsymbol{x}_o\|_0 \leq 1/2\mu(\boldsymbol{A})$, then $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 minimization problem. This gives a sufficient condition for the ℓ^1 minimization to be correct.

But how sharp is this result? According to Theorem 3.5, a random matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with high probability has its coherence bounded from above as $\mu(\boldsymbol{A}) \leq C\sqrt{\frac{\log n}{m}}$. So, for a "generic" $\boldsymbol{A}$, the above recovery guarantee implies correct recovery of $\boldsymbol{x}_o$ with $O(\sqrt{m/\log n})$ nonzeros. If we turn this around, and think of the matrix multiplication $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ as a sampling procedure, then for appropriately distributed random $\boldsymbol{A}$, we can recover k-sparse $\boldsymbol{x}_o$ from

$$m \geq C'k^2 \log n \tag{3.2.39}$$

observations. When k is small, this is substantially better than simply sampling all n entries of $\boldsymbol{x}$. On the other hand, the measurement burden $m = \Omega(k^2)$ seems a little too high – to specify a k-sparse $\boldsymbol{x}$, we only need to specify its k nonzero entries, ... and yet the theory demands k^2 samples!

One might naturally guess that the choice of $\boldsymbol{A}$ as a random matrix was a poor one – perhaps some delicate deterministic construction can yield a better performance guarantee, by making $\mu(\boldsymbol{A})$ smaller. How small can the coherence $\mu(\boldsymbol{A})$ be? We already noted that if $\boldsymbol{A}$ is a square matrix with orthogonal columns, $\mu(\boldsymbol{A}) = 0$. However, if we fix m and allow the number of columns, n, to grow,

we are forced to pack more and more vectors $\boldsymbol{a}_j$ into a compact set $\mathbb{S}^{m-1}$. As we increase n, the minimum achievable coherence μ increases.

As it turns out in this case, no matter what we do, we cannot construct a matrix whose coherence is significantly smaller than a randomly chosen one: the coherence of the random matrix $\boldsymbol{A}$ is within $C \log n$ of optimal. The following theorem makes this precise:

THEOREM 3.7 (Welch Bound). *For any matrix* $\boldsymbol{A} = [\boldsymbol{a}_1 \mid \cdots \mid \boldsymbol{a}_n] \in \mathbb{R}^{m\times n}$, $m \leq n$, *and suppose that the columns* $\boldsymbol{a}_i$ *have unit* ℓ^2 *norm. Then*

$$\mu(\boldsymbol{A}) \;=\; \max_{i\neq j} |\langle \boldsymbol{a}_i, \boldsymbol{a}_j\rangle| \;\geq\; \sqrt{\frac{n-m}{m(n-1)}}. \tag{3.2.40}$$

Proof Let $\boldsymbol{G} = \boldsymbol{A}^*\boldsymbol{A} \in \mathbb{R}^{n\times n}$, and let $\lambda_1 \geq \cdots \geq \lambda_m \geq 0$ denote its nonzero eigenvalues.[3] Notice that

$$\sum_{i=1}^{m} \lambda_i(\boldsymbol{G}) = \operatorname{trace}(\boldsymbol{G}) = \sum_{i=1}^{n} \|\boldsymbol{a}_i\|_2^2 = n. \tag{3.2.41}$$

Using this fact, we obtain that

$$\frac{n^2}{m} \leq \frac{n^2}{m} + \sum_{i=1}^{m} \left(\lambda_i(\boldsymbol{G}) - \frac{n}{m}\right)^2 \tag{3.2.42}$$

$$= \frac{n^2}{m} + \sum_{i=1}^{m} \left\{\lambda_i^2(\boldsymbol{G}) + \frac{n^2}{m^2} - 2\frac{n}{m}\lambda_i(\boldsymbol{G})\right\} \tag{3.2.43}$$

$$= \sum_{i=1}^{m} \lambda_i^2(\boldsymbol{G}) \;=\; \|\boldsymbol{G}\|_F^2 \tag{3.2.44}$$

$$= \sum_{i,j} |\boldsymbol{a}_i^*\boldsymbol{a}_j|^2 \;=\; n + \sum_{i\neq j} |\boldsymbol{a}_i^*\boldsymbol{a}_j|^2 \tag{3.2.45}$$

$$\leq n + n(n-1)\left(\max_{i\neq j} |\boldsymbol{a}_i^*\boldsymbol{a}_j|\right)^2. \tag{3.2.46}$$

Simplifying, we obtain the desired result.

In the above sequence of inequalities, we have used in (3.2.44) the fact that for any symmetric matrix $\boldsymbol{G}$, $\|\boldsymbol{G}\|_F^2 = \sum_i \lambda_i(\boldsymbol{G})^2$, which follows from the eigenvector decomposition $\boldsymbol{G} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^*$ and the fact that for any matrix $\boldsymbol{M}$ and orthogonal matrices $\boldsymbol{P}, \boldsymbol{Q}$ of appropriate size, $\|\boldsymbol{M}\|_F = \|\boldsymbol{P}\boldsymbol{M}\boldsymbol{Q}\|_F$. □

The important thing to notice here is that if we take n proportional to m, i.e., $n = \beta m$ for some $\beta > 1$, then the bound says that for *any* $m \times n$ matrix $\boldsymbol{A}$,

$$\mu(\boldsymbol{A}) \;\geq\; \Omega\left(\frac{1}{\sqrt{m}}\right). \tag{3.2.47}$$

Hence, in the best possible case, Theorem 3.3 guarantees we can recover $\boldsymbol{x}_o$ with

[3] Because $\operatorname{rank}(\boldsymbol{G}) \leq m$, it has at most m nonzero eigenvalues.

about $\sqrt{m}$ nonzero entries. Or equivalently, no matter how well we choose $\boldsymbol{A}$, to guarantee success Theorem 3.3 would demand

$$m \geq C'' k^2 \tag{3.2.48}$$

samples to reconstruct a k-sparse vector, which is only $\log n$ factor better than the previous bound (3.2.39) for a randomly chosen $\boldsymbol{A}$.

Does this behavior reflect a fundamental limitation of the ℓ^1 relaxation? Or is our analysis loose? It turns out that for generic matrices, the situation is much better than the bounds (3.2.39)-(3.2.48) seem to suggest. Again, the easiest way to see this is to do an experiment! We can try solving problems with constant aspect ratio (say, $m = n/2$), and n growing. Try to set $k = \|\boldsymbol{x}_o\|_0$ proportional to m – say, $k = m/4$ (a much better scaling than $k \sim \sqrt{m}$!). Now, try different aspect ratios $m = \alpha n$ and sparsity ratios $k = \beta m$. We leave this as an exercise to the reader. You may notice something intriguing:

In a proportional growth setting $m \propto n$, $k \propto m$, ℓ^1 minimization succeeds with very high probability whenever the constants of proportionality n/m and k/m are small enough.

This is a very important observation, since it implies that

- **more error correction***: we can correct constant fractions of errors, using an efficient algorithm.*
- **better compressive sampling***: we can sense sparse vectors using a number of measurements that is proportional to the intrinsic "information content" of the signal – the number of nonzero entries.*

However, to have a theory that can explain such observation, we will need a more refined measure of the goodness of $\boldsymbol{A}$ than the (rather crude) coherence or incoherence. In addition, we are going to need to sharpen our theoretical tools too.

3.3 Towards Stronger Correctness Results

3.3.1 The Restricted Isometry Property (RIP)

In the previous section, we saw that the ℓ^1 minimization problem

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \end{array} \tag{3.3.1}$$

correctly recovers a sparse $\boldsymbol{x}_o$ from observation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, provided two conditions are in force:

- $\boldsymbol{x}_o$ **is structured**: $k = \|\boldsymbol{x}_o\|_0 \ll n$.
- $\boldsymbol{A}$ **is "nice"**: its coherence $\mu(\boldsymbol{A})$ is small.

The intuition provided by incoherence is qualitatively very suggestive, but it does not provide a quantitative explanation for the good behavior we have seen in our experiments so far. How can we strengthen the condition? Suppose that $\boldsymbol{A}$ has unit norm columns. Then it is easy to calculate that for every two-column submatrix $\boldsymbol{A}_{\mathsf{I}} = [\boldsymbol{a}_i \mid \boldsymbol{a}_j] \in \mathbb{R}^{m\times 2}$,

$$\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}} = \begin{bmatrix} 1 & \boldsymbol{a}_i^* \boldsymbol{a}_j \\ \boldsymbol{a}_j^* \boldsymbol{a}_i & 1 \end{bmatrix}. \tag{3.3.2}$$

Exercise 3.6 asks you to show that since $|\boldsymbol{a}_i^* \boldsymbol{a}_j| \leq \mu(\boldsymbol{A})$, this matrix is well conditioned:

$$1 - \mu(\boldsymbol{A}) \;\leq\; \sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; \sigma_{\max}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; 1 + \mu(\boldsymbol{A}). \tag{3.3.3}$$

This property holds simultaneously for every two-column submatrix $\boldsymbol{A}_{\mathsf{I}}$. So, the property that the columns of $\boldsymbol{A}$ are well-spread implies that *the column submatrices of $\boldsymbol{A}$ are well-conditioned.*

We can generalize both properties by taking the set I to be larger than 2. Indeed, we can demand that all k-column submatrices of $\boldsymbol{A}$ are well-conditioned: For every $\mathsf{I} \subset \{1, \ldots, n\}$ of size k, we have

$$1 - k\mu(\boldsymbol{A}) \;\leq\; \sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; \sigma_{\max}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; 1 + k\mu(\boldsymbol{A}), \quad \forall\, \mathsf{I} \text{ of size} \leq k. \tag{3.3.4}$$

This controls the Kruskal rank: if $1 - k\mu(\boldsymbol{A}) > 0$, then $\operatorname{krank}(\boldsymbol{A}) \geq k$. This implies that an incoherent matrix with small μ tends to have large Kruskal rank. Hence according to Theorem 2.6, any sufficiently sparse $\boldsymbol{x}_o$ is *the sparsest* solution to the observation equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$.

In (3.3.4), we saw that the coherence $\mu(\boldsymbol{A})$ controls the conditioning of the column submatrices $\boldsymbol{A}_{\mathsf{I}}$ – if $\mu(\boldsymbol{A})$ is small, every submatrix spanned by just a few columns of $\boldsymbol{A}$ is well-conditioned:

$$1 - \delta \;\leq\; \sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; \sigma_{\max}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \;\leq\; 1 + \delta, \tag{3.3.5}$$

with δ small. This turned out to be critical in our proof of Theorem 3.3. In fact, we will see that for certain well-structured matrices $\boldsymbol{A}$, including random matrices, the bounds in (3.3.5) hold with δ far smaller than would be predicted by (3.3.4) using only the coherence.[4] They also hold for far larger $k = |\mathsf{I}|$ than might have been predicted from coherence alone. We will see that this leads (via different and slightly more complicated arguments), to substantially tighter guarantees for the performance of both ℓ^0 and ℓ^1 minimization.

The bounds in (3.3.5) hold uniformly over sets I of size k if and only if

$$\forall\, \boldsymbol{x} \;\; k\text{-sparse}, \quad (1-\delta)\,\|\boldsymbol{x}\|_2^2 \;\leq\; \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \;\leq\; (1+\delta)\,\|\boldsymbol{x}\|_2^2\,. \tag{3.3.6}$$

[4] For example, if $\boldsymbol{A}_{\mathsf{I}}$ is a large $m \times k$ ($k < m$) matrix with entries independent $\mathcal{N}(0, 1/m)$, $\sigma_{min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \approx (\sqrt{1} - \sqrt{k/m})^2 \geq 1 - 2\sqrt{k/m}$, and $\sigma_{max}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}) \approx (\sqrt{1} + \sqrt{k/m})^2 \leq 1 + 3\sqrt{k/m}$. You can check these values numerically; the aforementioned bounds can be made into rigorous statements using tools for Gaussian processes.

That is to say, the mapping $\boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ approximately preserves the norm of sparse vectors $\boldsymbol{x}$. Informally, we call such a mapping a *restricted isometry*: it is (nearly) an isometry[5], *if* we restrict our attention to the sparse vectors $\boldsymbol{x}$.

Definition 3.8 (Restricted Isometry Property [CT05]). *The matrix* $\boldsymbol{A}$ *satisfies the* restricted isometry property (RIP) *of* order k, *with constant* $\delta \in [0, 1)$, *if*

$$\forall\, \boldsymbol{x} \;\; k\text{-sparse}, \quad (1-\delta)\,\|\boldsymbol{x}\|_2^2 \;\le\; \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \;\le\; (1+\delta)\,\|\boldsymbol{x}\|_2^2 . \tag{3.3.7}$$

The order-k restricted isometry constant $\delta_k(\boldsymbol{A})$ *is the smallest number* δ *such that the above inequality holds.*

Whenever $\delta_k(\boldsymbol{A}) < 1$, every k-column submatrix has full column rank k. This implies that ℓ^0 recovery succeeds under RIP:

Theorem 3.9 (ℓ^0 Recovery under RIP [CRT06a, Can08]). *Suppose that* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, *with* $k = \|\boldsymbol{x}_o\|_0$. *If* $\delta_{2k}(\boldsymbol{A}) < 1$, *then* $\boldsymbol{x}_o$ *is the unique optimal solution to*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_0 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.3.8}$$

Proof Suppose on the contrary that there exists $\boldsymbol{x}' \neq \boldsymbol{x}_o$ with $\|\boldsymbol{x}'\|_0 \le k$. Then $\boldsymbol{x}_o - \boldsymbol{x}' \in \mathrm{null}(\boldsymbol{A})$, and $\|\boldsymbol{x}_o - \boldsymbol{x}'\|_0 \le 2k$. This implies that $\delta_{2k}(\boldsymbol{A}) \ge 1$, contradicting our assumption. □

So, provided the RIP constant of order $2k$ is bounded away from one, ℓ^0 minimization successfully recovers $\boldsymbol{x}_o$. If we tighten our demand to $\delta_{2k}(\boldsymbol{A}) < \sqrt{2}-1$, ℓ^1 minimization succeeds as well:

Theorem 3.10 (ℓ^1 Recovery under RIP). *Suppose that* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, *with* $k = \|\boldsymbol{x}_o\|_0$. *If* $\delta_{2k}(\boldsymbol{A}) < \sqrt{2}-1$, *then* $\boldsymbol{x}_o$ *is the unique optimal solution to*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.3.9}$$

The significance of this result comes from the fact that for "generic" matrices, the condition $\delta_{2k}(\boldsymbol{A}) < \sqrt{2}-1$ holds even when k is nearly proportional to m:

Theorem 3.11 (RIP of Gaussian Matrices [CRT06a, BDDW08]). *There exists a numerical constant* $C > 0$ *such that if* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *is a random matrix with entries independent* $\mathcal{N}\left(0, \frac{1}{m}\right)$ *random variables, with high probability,* $\delta_k(\boldsymbol{A}) < \delta$, *provided*

$$m \;\ge\; Ck\log(n/k)/\delta^2. \tag{3.3.10}$$

This implies that recovery of k-sparse $\boldsymbol{x}$ is possible from about $m \ge Ck\log(n/k)$ random measurements. This is a substantial improvement over our previous estimate of $m \sim k^2$. In particular, it allows (k, m, n) to scale proportionally [Don06b, CT05]. This improvement has stimulated a lot of work on efficient sensing and sampling schemes in various application domains.

[5] An isometry is a mapping that preserves the norm of every vector.

3.3.2 Restricted Strong Convexity Condition

We have stated the above two theorems without proof. We will prove Theorem 3.10 in several stages. In this section, we introduce two intermediate properties of the sensing matrix $\boldsymbol{A}$, which turn out to be very useful in their own right. In the next section, we prove Theorem 3.10 by proving that when $\delta_{2k}(\boldsymbol{A}) < \sqrt{2}-1$, these intermediate properties are satisfied, and hence ℓ^1 minimization succeeds.

As above, suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, for some $\|\boldsymbol{x}_o\|_0 \leq k$. We hope that under certain conditions, $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 minimization program

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.3.11}$$

Let $\boldsymbol{x}'$ be any feasible point, i.e., any point satisfying $\boldsymbol{A}\boldsymbol{x}' = \boldsymbol{y}$. Because $\boldsymbol{A}\boldsymbol{x}_o = \boldsymbol{y}$ as well, *the difference* $\boldsymbol{h} = \boldsymbol{x}' - \boldsymbol{x}_o$ *belongs to the null space* $\mathrm{null}(\boldsymbol{A})$.

Let I denote the support of $\boldsymbol{x}_o$, and I^c its complement. Then

$$\|\boldsymbol{x}'\|_1 = \|\boldsymbol{x}_o + \boldsymbol{h}\|_1 \tag{3.3.12}$$

$$\geq \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 . \tag{3.3.13}$$

Hence, if $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 > \|\boldsymbol{h}_{\mathsf{I}}\|_1$, $\boldsymbol{x}'$ has strictly larger objective function than $\boldsymbol{x}_o$ and so $\boldsymbol{x}'$ is not optimal. Conversely, if the nulls pace of $\boldsymbol{A}$ contains no vectors $\boldsymbol{h} \neq \boldsymbol{0}$ for which $\|\boldsymbol{h}_{\mathsf{I}}\|_1 \geq \|\boldsymbol{h}_{\mathsf{I}^c}\|_1$, then $\boldsymbol{x}_o$ must be the unique optimal solution to (3.3.11).

It is helpful to ask what if this were not true? What happens if the optimal solution to the above program, say $\hat{\boldsymbol{x}}_{\ell^1}$, was not $\boldsymbol{x}_o$. Under what conditions could their difference $\boldsymbol{h} \doteq \hat{\boldsymbol{x}}_{\ell^1} - \boldsymbol{x}_o$ be nonzero? Recall that I is the support of $\boldsymbol{x}_o$ and I^c its complement.

Since $\hat{\boldsymbol{x}}_{\ell^1}$ is the optimal solution to the above program, we must have

$$\begin{aligned} 0 &\geq \|\hat{\boldsymbol{x}}_{\ell^1}\|_1 - \|\boldsymbol{x}_o\|_1 \\ &= \|\boldsymbol{x}_o + \boldsymbol{h}\|_1 - \|\boldsymbol{x}_o\|_1 \\ &\geq \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 - \|\boldsymbol{x}_o\|_1 \\ &= -\|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}^c}\|_1. \end{aligned} \tag{3.3.14}$$

That is, we have

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \|\boldsymbol{h}_{\mathsf{I}}\|_1. \tag{3.3.15}$$

Meanwhile, since $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o = \boldsymbol{A}\hat{\boldsymbol{x}}_{\ell^1}$, we also have

$$\boldsymbol{A}\boldsymbol{h} = \boldsymbol{0}. \tag{3.3.16}$$

In other words, in order for the ℓ^1 program to admit a better solution $\hat{\boldsymbol{x}}_{\ell^1}$ than the original sparse solution $\boldsymbol{x}_o$, we must have the above two conditions (3.3.15) and (3.3.16) hold simultaneously. Therefore, in order to show that $\boldsymbol{x}_o$ is the unique optimal solution for the ℓ^1 program, we only have to show that these conditions cannot all be true for any such $\boldsymbol{h}$.

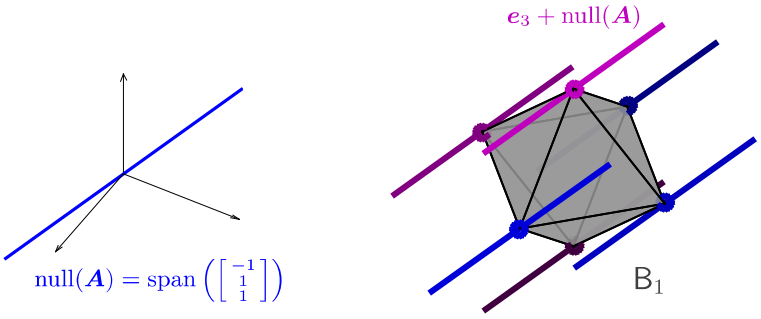


Figure 3.8 Visualizing the Nullspace Property in three dimensions. **Left:** the sensing matrix $\left[\begin{smallmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \end{smallmatrix}\right]$ has nullspace spanned by $[-1, 1, 1]^*$. This matrix satisfies the nullspace property of order $k = 1$. **Right:** Geometrically this implies that any translate $\pm\boldsymbol{e}_j + \text{null}(\boldsymbol{A})$ to a vertex of the ℓ^1 ball B_1 intersects B_1 only at the vertex $\pm\boldsymbol{e}_j$.

Null Space Property.
The above discussion suggests that the null space of $\boldsymbol{A}$ is very important for understanding when we can recover $\boldsymbol{x}_o$. Previous ℓ^0 recovery results all come by showing that the null space does not contain sparse vectors. The condition that for every nonzero $\boldsymbol{h} \in \text{null}(\boldsymbol{A})$, $\|\boldsymbol{h}_{\mathsf{I}}\|_1 < \|\boldsymbol{h}_{\mathsf{I}^c}\|_1$, can be interpreted as saying that the null space does not contain any vector that is concentrated on the (small) set of coordinates I. This is sufficient for ℓ^1 minimization to recover $\boldsymbol{x}_o$ with support I. If we want to guarantee recovery of *any* k-sparse $\boldsymbol{x}_o$, we can ask that for every set I of k coordinates and every nonzero null vector $\boldsymbol{h}$, $\|\boldsymbol{h}_{\mathsf{I}}\|_1 < \|\boldsymbol{h}_{\mathsf{I}^c}\|_1$:

Definition 3.12 (Null Space Property). *The matrix $\boldsymbol{A}$ satisfies the* null space property *of order k if for every $\boldsymbol{h} \in \text{null}(\boldsymbol{A}) \setminus \{\boldsymbol{0}\}$ and every I of size at most k,*

$$\|\boldsymbol{h}_{\mathsf{I}}\|_1 < \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 . \tag{3.3.17}$$

This can be interpreted as saying that the null space does not contain any near-sparse vectors, where sparsity is measured via the ℓ^1 norm. If $\boldsymbol{A}$ satisfies the null space property, then ℓ^1 succeeds in recovering any k-sparse $\boldsymbol{x}_o$:

Lemma 3.13. *Suppose that $\boldsymbol{A}$ satisfies the null space property of order k. Then for any $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with $\|\boldsymbol{x}_o\|_0 \leq k$, $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.3.18}$$

Proof Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with $\|\boldsymbol{x}_o\|_0 \leq k$, and let $\mathsf{I} = \text{supp}(\boldsymbol{x}_o)$. Let $\hat{\boldsymbol{x}}_{\ell^1}$ be the optimal solution, so $\boldsymbol{h} = \hat{\boldsymbol{x}}_{\ell^1} - \boldsymbol{x}_o \in \text{null}(\boldsymbol{A})$. If $\boldsymbol{h} \neq \boldsymbol{0}$, then $\|\hat{\boldsymbol{x}}_{\ell^1}\|_1 = \|\boldsymbol{x}_o + \boldsymbol{h}\|_1 \geq \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 > \|\boldsymbol{x}_o\|_1$, contradicting the optimality of $\hat{\boldsymbol{x}}_{\ell^1}$. □

In the viewpoint of the coefficient space picture for ℓ^1 minimization introduced in Section 3.2.2, the nullspace condition asserts that when null($\boldsymbol{A}$) is translated to any k-sparse point $\boldsymbol{x}_o$ on the boundary of the ℓ^1 ball B_1, the translate $\boldsymbol{x}_o$+null($\boldsymbol{A}$) does not intersect the interior of B_1. Figure 3.8 visualizes this condition for the special case in which $n = 3$, and null($\boldsymbol{A}$) is one-dimensional. In the literature, the null space property has been used to establish various sufficient conditions for the success of ℓ^1 minimization for sparse recovery. In fact, Theorem 3.10 can be proved by showing that the RIP condition on the matrix $\boldsymbol{A}$ implies the null space property.

Restricted Strong Convexity Condition.
Alternatively and equivalently, we can study the success of ℓ^1 minimization by considering possible perturbations $\boldsymbol{h}$ that could reduce the value of the objective function. According to condition (3.3.15), they must satisfy

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \|\boldsymbol{h}_{\mathsf{I}}\|_1 . \tag{3.3.19}$$

To ensure the original k-sparse $\boldsymbol{x}_o$ is the unique optimal solution, we can require that for any nonzero perturbation $\boldsymbol{h}$ satisfying (3.3.19), $\boldsymbol{A}\boldsymbol{h} \neq \boldsymbol{0}$:

$$\|\boldsymbol{A}\boldsymbol{h}\|_2^2 > 0. \tag{3.3.20}$$

Since the set $\mathsf{S} = \cup_{\mathsf{I}}\{\boldsymbol{h} : \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \|\boldsymbol{h}_{\mathsf{I}}\|_1, \|\boldsymbol{h}\|_2^2 = 1\}$ is compact, $\|\boldsymbol{A}\boldsymbol{h}\|_2^2$ must attain its minimum $\mu > 0$. The above condition is therefore equivalent to:

$$\|\boldsymbol{A}\boldsymbol{h}\|_2^2 \geq \mu\|\boldsymbol{h}\|_2^2, \quad \forall \boldsymbol{h} \;\; \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \|\boldsymbol{h}_{\mathsf{I}}\|_1 \tag{3.3.21}$$

for some $\mu > 0$.

If we consider the quadratic loss, $L(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, the second derivative in the $\boldsymbol{h}$ direction is $\boldsymbol{h}^*\nabla^2 L(\boldsymbol{x})\boldsymbol{h} = \|\boldsymbol{A}\boldsymbol{h}\|_2^2 > 0$. The above condition can be interpreted as saying that the function $L(\boldsymbol{x})$ is *strongly convex* when restricted to directions $\boldsymbol{h}$ satisfying (3.3.19) – see Figure 3.9 for a visualization of this interpretation. We term this *(uniform) restricted strong convexity*:

Definition 3.14 (Restricted Strong Convexity). *The matrix $\boldsymbol{A}$ satisfies the* restricted strong convexity *(RSC) condition of order k, with parameters $\mu > 0$, $\alpha \geq 1$, if for every I of size at most k and for all nonzero $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \alpha\|\boldsymbol{h}_{\mathsf{I}}\|_1$,*

$$\|\boldsymbol{A}\boldsymbol{h}\|_2^2 \geq \mu\|\boldsymbol{h}\|_2^2. \tag{3.3.22}$$

In this definition, we have generalized the condition (3.3.19) to consider instead $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \alpha \|\boldsymbol{h}_{\mathsf{I}}\|_1$. This generalization will be used in an essential way later when we study sparse recovery from noisy measurements. For now, we note that for noiseless measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, restricted strong convexity indeed implies that ℓ^1 minimization succeeds:

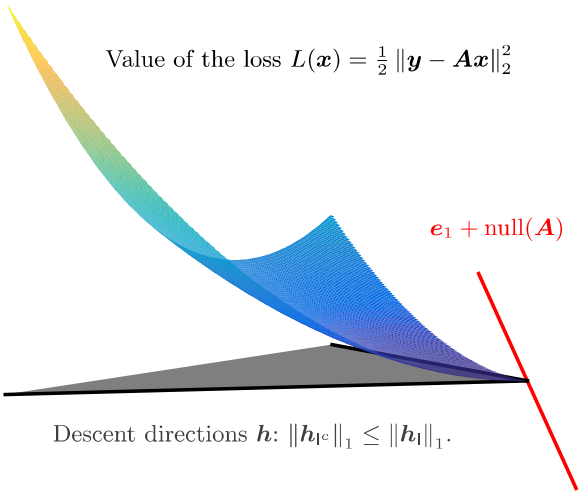


Figure 3.9 Restricted Strong Convexity implies that the loss $L(\boldsymbol{x})$ exhibits positive curvature along the potential **descent directions** $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \|\boldsymbol{h}_{\mathsf{I}}\|_1$. Here, $\boldsymbol{x}_o = \boldsymbol{e}_1$. Red: the **feasible set** of $\boldsymbol{x}$ that satisfy $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}$. Under RSC, the loss is strictly positive at any point $\boldsymbol{x}$ whose ℓ^1 norm is smaller than $\|\boldsymbol{x}_o\|_1$.

LEMMA 3.15. *Suppose that $\boldsymbol{A}$ satisfies the restricted strong convexity condition of order k with constant $\alpha \geq 1$, for some $\mu > 0$. Then for any $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, with $\|\boldsymbol{x}_o\|_0 \leq k$, $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \end{array} \tag{3.3.23}$$

Proof We leave it as an exercise for the reader to prove this result by verifying that Restricted Strong Convexity implies the nullspace property. □

3.3.3 Success of ℓ^1 Minimization under RIP

In this section we prove Theorem 3.10. Earlier, in Section 3.2.2, we followed a fairly simple path to prove Theorem 3.3: write down an optimality condition, and then construct a dual certificate using a bit of cleverness. This approach can be used to prove a variant of Theorem 3.10 [CT05]. However, the argument is more delicate than before.

So here, to prove Theorem 3.10, we will take a slightly different path, which utilizes properties of "good" sensing matrices $\boldsymbol{A}$ that we have introduced in the previous section. As we have discussed there, to prove that RIP implies correct recovery, it suffices to show that RIP implies the restricted strong convexity

(RSC) condition. Our proof here follows close to that of [CRT06b, Can08].[6] In doing so, we will use the following property of the restricted isometry constants:

LEMMA 3.16. *If $\boldsymbol{x}$, $\boldsymbol{z}$ are vectors with disjoint support, and $|\mathrm{supp}(\boldsymbol{x})|+|\mathrm{supp}(\boldsymbol{z})| \leq k$, then*

$$|\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{z}\rangle| \;\leq\; \delta_k(\boldsymbol{A})\, \|\boldsymbol{x}\|_2\, \|\boldsymbol{z}\|_2\,. \tag{3.3.24}$$

Proof Because the expression is invariant to scaling $\boldsymbol{x}$ and $\boldsymbol{z}$, we lose no generality in assuming that $\|\boldsymbol{x}\|_2 = \|\boldsymbol{z}\|_2 = 1$. Notice that

$$\|\boldsymbol{p}+\boldsymbol{q}\|_2^2 = \|\boldsymbol{p}\|_2^2 + \|\boldsymbol{q}\|_2^2 + 2\,\langle \boldsymbol{p}, \boldsymbol{q}\rangle\,, \tag{3.3.25}$$

$$\|\boldsymbol{p}-\boldsymbol{q}\|_2^2 = \|\boldsymbol{p}\|_2^2 + \|\boldsymbol{q}\|_2^2 - 2\,\langle \boldsymbol{p}, \boldsymbol{q}\rangle\,. \tag{3.3.26}$$

Hence,

$$|\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{z}\rangle| \leq \frac{1}{4}\left|\|\boldsymbol{A}\boldsymbol{x}+\boldsymbol{A}\boldsymbol{z}\|_2^2 - \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{A}\boldsymbol{z}\|_2^2\right| \tag{3.3.27}$$

$$\leq \frac{1}{4}\left|(1+\delta_k)\,\|\boldsymbol{x}+\boldsymbol{z}\|_2^2 - (1-\delta_k)\,\|\boldsymbol{x}-\boldsymbol{z}\|_2^2\right|. \tag{3.3.28}$$

Because $\boldsymbol{x}$ and $\boldsymbol{z}$ have disjoint support, $\|\boldsymbol{x}+\boldsymbol{z}\|_2^2 = \|\boldsymbol{x}-\boldsymbol{z}\|_2^2 = 2$, and the result follows. □

We are now ready to prove the following theorem.

THEOREM 3.17 (RIP Implies RSC). *If a matrix $\boldsymbol{A}$ satisfies RIP with $\delta_{2k}(\boldsymbol{A}) < \frac{1}{1+\alpha\sqrt{2}}$, then $\boldsymbol{A}$ satisfies the RSC condition of order k with constant α.*

Proof Let I be any set of size k and let $\boldsymbol{h} \in \mathbb{R}^n$ any vector that satisfies the restriction

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \alpha \cdot \|\boldsymbol{h}_{\mathsf{I}}\|_1. \tag{3.3.29}$$

Form disjoint subsets $\mathsf{J}_1, \mathsf{J}_2, \mathsf{J}_3, \cdots \subseteq \mathsf{I}^c$ as follows:

J_1 indexes the k largest (in magnitude) elements of $\boldsymbol{h}_{\mathsf{I}^c}$,
J_2 indexes the k largest (in magnitude) elements of $\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}$,
J_3 indexes the k largest (in magnitude) elements of $\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1\cup\mathsf{J}_2)^c}$,
$\vdots$

Notice that because every entry of J_i is at least as large as every entry of J_{i+1}, the average magnitude of an entry in J_i is at least as large as the largest entry in J_{i+1}:

$$\forall\, i \geq 1, \qquad \left\|\boldsymbol{h}_{\mathsf{J}_{i+1}}\right\|_\infty \;\leq\; \frac{\|\boldsymbol{h}_{\mathsf{J}_i}\|_1}{k}. \tag{3.3.30}$$

We also note that for any vector $\boldsymbol{z}$ with $\|\boldsymbol{z}\|_0 \leq k$, $\|\boldsymbol{z}\|_1 \leq \sqrt{k}\,\|\boldsymbol{z}\|_2$ and $\|\boldsymbol{z}\|_2 \leq \sqrt{k}\,\|\boldsymbol{z}\|_\infty$.

[6] We have modified the original proof that shows RIP implies the null space property to RSC.

Using the RIP with the $2k$-sparse vector $\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}$ and the fact

$$\boldsymbol{A}\boldsymbol{h}_{\mathsf{I}} + \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_1} = \boldsymbol{A}\boldsymbol{h} - \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_2} - \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_3} - \cdots, \tag{3.3.31}$$

we have that

$$\begin{aligned}
&(1-\delta_{2k})\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2^2 \;\le\; \|\boldsymbol{A}\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2^2 \\
&= \langle \boldsymbol{A}\boldsymbol{h}_{\mathsf{I}} + \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_1}, -\boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_2} - \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_3} - \cdots\rangle + \langle \boldsymbol{A}\boldsymbol{h}_{\mathsf{I}} + \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_1}, \boldsymbol{A}\boldsymbol{h}\rangle \\
&\le \sum_{j=2}^{\infty} \left(\left|\left\langle \boldsymbol{A}\boldsymbol{h}_{\mathsf{I}}, \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_j}\right\rangle\right| + \left|\left\langle \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_1}, \boldsymbol{A}\boldsymbol{h}_{\mathsf{J}_j}\right\rangle\right| \right) + \|\boldsymbol{A}\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2 \\
&\le \delta_{2k}(\|\boldsymbol{h}_{\mathsf{I}}\|_2 + \|\boldsymbol{h}_{\mathsf{J}_1}\|_2) \sum_{j=2}^{\infty} \left\|\boldsymbol{h}_{\mathsf{J}_j}\right\|_2 + (1+\delta_{2k})^{1/2}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2 \\
&\le \delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \sum_{j=2}^{\infty} \left\|\boldsymbol{h}_{\mathsf{J}_j}\right\|_2 + (1+\delta_{2k})^{1/2}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2 \\
&\le \delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \sum_{j=2}^{\infty} \left\|\boldsymbol{h}_{\mathsf{J}_j}\right\|_\infty \sqrt{k} + (1+\delta_{2k})^{1/2}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2 \\
&\le \delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \sum_{j=1}^{\infty} \left\|\boldsymbol{h}_{\mathsf{J}_j}\right\|_1 / \sqrt{k} + (1+\delta_{2k})^{1/2}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2 \\
&= \delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \,\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 / \sqrt{k} + (1+\delta_{2k})^{1/2}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\|\boldsymbol{A}\boldsymbol{h}\|_2.
\end{aligned} \tag{3.3.32}$$

After dividing through by $\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2$, we have

$$(1-\delta_{2k})\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \le \delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 / \sqrt{k} + (1+\delta_{2k})^{1/2}\,\|\boldsymbol{A}\boldsymbol{h}\|_2\,. \tag{3.3.33}$$

Since $\boldsymbol{h}$ satisfies the restricted cone condition, we have

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \le \alpha\|\boldsymbol{h}_{\mathsf{I}}\|_1 \le \alpha\sqrt{k}\|\boldsymbol{h}_{\mathsf{I}}\|_2 \le \alpha\sqrt{k}\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2. \tag{3.3.34}$$

Substituting this into the previous inequality, we obtain:

$$(1-\delta_{2k})\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \le \alpha\delta_{2k}\sqrt{2}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + (1+\delta_{2k})^{1/2}\,\|\boldsymbol{A}\boldsymbol{h}\|_2\,. \tag{3.3.35}$$

This gives

$$\|\boldsymbol{A}\boldsymbol{h}\|_2 \ge \frac{1-\delta_{2k}(1+\alpha\sqrt{2})}{(1+\delta_{2k})^{1/2}}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2\,. \tag{3.3.36}$$

Since the i-th element of $\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}$ is no larger than the mean of the first i elements of $\boldsymbol{h}_{\mathsf{I}^c}$, we have

$$|\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}|_{(i)} \le \|\boldsymbol{h}_{\mathsf{I}^c}\|_1/i. \tag{3.3.37}$$

Combining with the restriction (3.3.29), we have

$$\|\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}\|_2^2 \le \|\boldsymbol{h}_{\mathsf{I}^c}\|_1^2 \sum_{i=k+1}^{\infty} \frac{1}{i^2} \tag{3.3.38}$$

$$\le \frac{\|\boldsymbol{h}_{\mathsf{I}^c}\|_1^2}{k} \le \frac{\alpha^2\|\boldsymbol{h}_{\mathsf{I}}\|_1^2}{k} \tag{3.3.39}$$

$$\le \alpha^2\|\boldsymbol{h}_{\mathsf{I}}\|_2^2 \le \alpha^2\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2^2. \tag{3.3.40}$$

So we have

$$\|\boldsymbol{h}\|_2^2 \le (1+\alpha^2)\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2^2. \tag{3.3.41}$$

Combining this with the previous condition on $\|\boldsymbol{A}\boldsymbol{h}\|_2$, we get

$$\|\boldsymbol{A}\boldsymbol{h}\|_2 \ge \frac{1-\delta_{2k}(1+\alpha\sqrt{2})}{(1+\delta_{2k})^{1/2}\sqrt{1+\alpha^2}}\,\|\boldsymbol{h}\|_2\,. \tag{3.3.42}$$

So as long as $\delta_{2k} < \frac{1}{1+\alpha\sqrt{2}}$, $\boldsymbol{A}$ satisfies the RSC condition of order k with the constant

$$\mu = \frac{\big(1-\delta_{2k}(1+\alpha\sqrt{2})\big)^2}{(1+\delta_{2k})(1+\alpha^2)}, \tag{3.3.43}$$

as claimed. □

Theorem 3.10 then becomes a corollary to this theorem for the case $\alpha = 1$ since the restriction set we need to consider is $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \le \|\boldsymbol{h}_{\mathsf{I}}\|_1$ for the ℓ^1 minimization in Theorem 3.10 and that gives the RIP constant $\delta_{2k} = \frac{1}{1+\sqrt{2}} = \sqrt{2}-1$.

3.4 Matrices with Restricted Isometry Property

The RIP gives a useful tool for analyzing the performance of sparse recovery with random matrices $\boldsymbol{A}$. Below, we will prove the probabilistic result, Theorem 3.11, which asserts that Gaussian random matrix $\boldsymbol{A}$ has RIP when $m > Ck\log(n/k)$. We will make heavy use of the following simple inequality:

LEMMA 3.18. *Let $\boldsymbol{g} = [g_1, \ldots, g_m]^* \in \mathbb{R}^m$ be an m-dimensional random vector whose entries are iid $\mathcal{N}(0, 1/m)$. Then for any $t \in [0, 1]$,*

$$\mathbb{P}\left[\left|\|\boldsymbol{g}\|_2^2 - 1\right| > t\right] \;\le\; 2\exp\left(-\frac{t^2 m}{8}\right). \tag{3.4.1}$$

This result can be obtained via the Cramer-Chernoff exponential moment method (in a similar fashion to the Hoeffdings inequality). See Appendix E for more information.

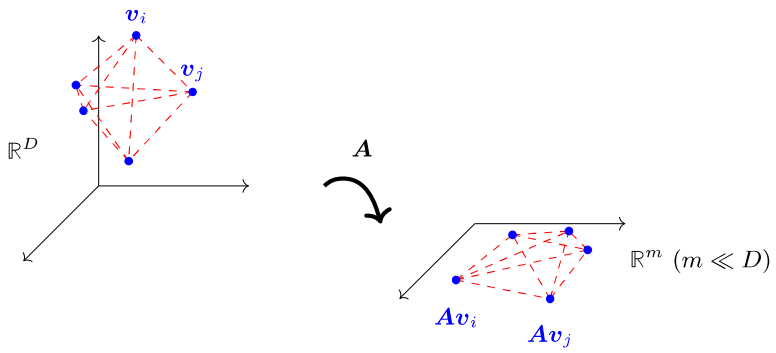


Figure 3.10 The Johnson-Lindenstrauss Lemma. Given a fixed collection of points $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ in a high-dimensional space $\mathbb{R}^D$, with high probability a random mapping into $m \sim \log n$ dimensions approximately preserves the distances between all pairs of points.

3.4.1 The Johnson-Lindenstrauss Lemma

Before proving Theorem 3.11, we will first state and prove a simpler result, as an illustration of the basic approach we will take to this result, and is very useful in its own right:

THEOREM 3.19 (Johnson-Lindenstrauss Lemma). *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^D$ for some D. Let $\boldsymbol{A} \in \mathbb{R}^{m \times D}$ be a random matrix whose entries are independent $\mathcal{N}(0, 1/m)$ random variables. Then for any $\varepsilon \in (0, 1)$, with probability at least $1 - 1/n^2$, the following holds:*

$$\forall\, i \neq j, \quad (1-\varepsilon)\,\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 \;\leq\; \|\boldsymbol{A}\boldsymbol{v}_i - \boldsymbol{A}\boldsymbol{v}_j\|_2^2 \;\leq\; (1+\varepsilon)\,\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2, \quad (3.4.2)$$

provided $m > 32\frac{\log n}{\varepsilon^2}$.

This result can be thought of as follows: we have a large database $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ of very high-dimensional vectors. We would like to embed them in a lower-dimensional space ($m \ll D$) such that the pairwise distances between the vectors are preserved. This is useful, for example, if we think of these as points in a database, and we imagine that we would like to be able to query the database to find points that are close to a given input $\boldsymbol{q}$ in norm – a good embedding will reduce both the storage and computation requirements for achieving this. If you think carefully, it should be clear that we can achieve a perfect (norm-preserving) embedding into $m = n$ dimensional space – simply project each point onto the span of the n points $\boldsymbol{v}_i$.

The surprise in the Johnson-Lindenstrauss lemma is that actually, if we allow some slack ε, the dimension can be much lower – only logarithmic in the number of points, and completely independent of the ambient data dimension D. It should not be too surprising that approaches (loosely) inspired by this

result have significant applications in search problems. Interestingly, with some additional clever ideas, it is possible to arrive at algorithms that can find approximate nearest neighbors in a database of points in a search time that depends sublinearly on the size of the dataset.

Proof Set $\boldsymbol{g}_{ij} = \boldsymbol{A}\frac{\boldsymbol{v}_i - \boldsymbol{v}_j}{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2}$. Notice that for any $\boldsymbol{v}_i \neq \boldsymbol{v}_j$, $\boldsymbol{g}_{ij}$ is distributed as an iid Gaussian vector, with entries $\mathcal{N}(0, 1/m)$. Applying Lemma 3.18, for each $i \neq j$, we have

$$\mathbb{P}\left[\left|\left\|\boldsymbol{g}_{ij}\right\|_2^2 - 1\right| > t\right] \;\leq\; 2\exp\left(-t^2 m/8\right). \tag{3.4.3}$$

Summing the probability of failure over all $i \neq j$, and then plugging in $t = \varepsilon$ and $m \geq 32 \log n/\varepsilon^2$, we get

$$\begin{aligned}\mathbb{P}\left[\exists\,(i,j)\;:\;\left|\left\|\boldsymbol{g}_{ij}\right\|_2^2 - 1\right| > t\right] &\leq \frac{n(n-1)}{2} \times 2\exp\left(-t^2 m/8\right) \\ &\leq n^{-2}.\end{aligned} \tag{3.4.4}$$

Whenever $\left|\left\|\boldsymbol{g}_{ij}\right\|_2^2 - 1\right| \leq \varepsilon$, we have

$$(1-\varepsilon)\left\|\boldsymbol{v}_i - \boldsymbol{v}_j\right\|_2^2 \;\leq\; \left\|\boldsymbol{A}\boldsymbol{v}_i - \boldsymbol{A}\boldsymbol{v}_j\right\|_2^2 \;\leq\; (1+\varepsilon)\left\|\boldsymbol{v}_i - \boldsymbol{v}_j\right\|_2^2, \tag{3.4.5}$$

as desired. □

Thus, the fairly powerful embedding result (Theorem 3.19) follows from a fairly straightforward pattern:

- **Discretization**: Argue that if $\boldsymbol{A}$ respects the norms of some finite set of vectors (here $\{\boldsymbol{v}_i - \boldsymbol{v}_j \mid i \neq j\}$), the desired property holds.
- **Tail bound**: Develop an upper bound on the probability that $\boldsymbol{A}$ fails to respect the norm of a single vector (here, this is Lemma 3.18).
- **Union bound**: Sum the failure probabilities over all of the finite set. Choose the embedding dimension m large enough that the total failure probability is small.

EXAMPLE 3.20 (p-Stable Distributions [DIIM04]). *From the above theorem, we see that a random Gaussian matrix has the property of preserving ℓ^2 distance between vectors. As it turns out that for $p \in (0, 2]$, there exist the so-called p-stable distributions such that a random matrix drawn from a p-stable distribution will preserve ℓ^p distance between vectors. For instance, the Cauchy distribution $p(x) = \frac{1}{\pi} \cdot \frac{1}{1+x^2}$ is 1-stable and a random Cauchy matrix preserves ℓ^1 distance. We leave this as an exercise.*

Fast Nearest Neighbor Methods.
The property of distance preserving (random) projections are the basis for developing most efficient codes and schemes for nearest neighbor search. The above JL Lemma works for a set of points of arbitrary configuration in $\mathbb{R}^D$. As it turns out, in many real applications, such as image search [MYW+10, LM16], the data

Algorithm 3.1 (Compact Code for Fast Nearest Neighbor)

Problem: Generate compact binary code for efficient nearest neighbor search of high-dimensional data points.
Input: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^D$ and $m = O(\log n)$.
Generate a random Gaussian matrix $\boldsymbol{R} \in \mathbb{R}^{m \times D}$ with entries i.i.d. $\mathcal{N}(0, 1)$.
for $i = 1, \dots, n$ **do**
    Compute $\boldsymbol{R}\boldsymbol{x}_i$,
    Set $\boldsymbol{y}_i = \sigma(\boldsymbol{R}\boldsymbol{x}_i)$ where $\sigma(\cdot)$ is the entry-wise binary thresholding.
end for
Output: $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \{0, 1\}^m$.

points are reasonably spread in space or have certain additional properties. Under such circumstances, approximate nearest neighbor search can be made even more memory and computation efficient – instead of $O(\log n)$ real numbers, one only needs $O(\log n)$ binary bits! We introduce one such property below as an example since it is related to the property of incoherence studied before.

Definition 3.21 (Weak Separability). *We say a set of points $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ in $\mathbb{R}^D$ is (Δ, l)-weakly separable if for any query point $\boldsymbol{q} \in \mathbb{R}^D$, we have*

$$|\{i \mid \angle(\boldsymbol{q}, \boldsymbol{x}_i) \leq \Delta\}| = O(n^l), \tag{3.4.6}$$

where typically $l \in [0, 1)$ is desired to be a small constant.

Although the above definition is defined in terms of arbitrary $\boldsymbol{q} \in \mathbb{R}^D$, the following lemma shows that it is sufficient to check this condition within the data set $\mathcal{X}$ itself.

Lemma 3.22. *If for every $\boldsymbol{x}_j \in \mathcal{X}$,*

$$|\{i \mid \angle(\boldsymbol{x}_j, \boldsymbol{x}_i) \leq 2\Delta\}| = O(n^l), \tag{3.4.7}$$

then $\mathcal{X}$ is (Δ, l)-weakly separable.

Proof We leave the proof as an exercise to the reader (see Exercise 3.14). □

Notice that weak separability of $\boldsymbol{x}_i$'s is similar to assuming that these data points (viewed as vectors) are weakly incoherent – majority of the angles between pairwise points are large.

Example 3.23 (Efficient c-Approximate Nearest Neighbor [MYW+10]). *Given a set of data points $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\}$ in $\mathbb{R}^D$ and a constant $c > 1$, the c-Approximate Nearest Neighbor (c-NN) problem is: for any query point $\boldsymbol{q} \in \mathbb{R}^D$, find $\boldsymbol{x}_\star$ such that*

$$\|\boldsymbol{q} - \boldsymbol{x}_\star\|_2 \leq c \cdot \min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{q} - \boldsymbol{x}\|_2.$$

As it turns out, for any (Δ, l)-weakly separable set $\mathcal{X}$, with the random binary

code generated by Algorithm 3.1, with probability $1-\delta$*, the c-NN problem can be solved with the number of binary bits* m *chosen in the order*

$$m = O(\log n) \quad \textit{(bits)}.$$

For any query point $\boldsymbol{q}$*, we may first compute its binary code with the same projection as in Algorithm 3.1:* $\boldsymbol{y_q} = \sigma(\boldsymbol{Rq})$ *where* $\sigma(\cdot)$ *is the binary thresholding function:* $\sigma(x) = 1$ *for* $x > 0$ *and* $\sigma(x) = 0$ *otherwise. Then we find a subset* $\tilde{\mathcal{X}}_{\boldsymbol{q}}$ *of points of size* $O(n^l)$ *which have the shortest Hamming distances to* $\boldsymbol{y_q}$ *in* $\mathcal{X}$*. One can show that:*

$$\boldsymbol{x}_\star = \arg\min_{\boldsymbol{x}\in\tilde{\mathcal{X}}_{\boldsymbol{q}}} \|\boldsymbol{q}-\boldsymbol{x}\|_2$$

gives the correct solution to the c-NN problem. We leave the proof for the correctness and efficiency of this simple scheme as an exercise to the reader, see Exercise 3.15.

3.4.2 RIP of Gaussian Random Matrices

To prove Theorem 3.11, we follow exactly the same pattern as we did for Johnson-Lindenstrauss. However, we will need to work a little bit harder in the discretization stage, since unlike the Johnson-Lindenstrauss Lemma, which was a statement about n (or $n(n-1)/2$) vectors, the RIP is a statement about an infinite family of vectors – all of the sparse vectors.

Discretization.
Let

$$\Sigma_k = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_0 \le k,\ \|\boldsymbol{x}\|_2 = 1\}. \tag{3.4.8}$$

Notice that $\delta_k(\boldsymbol{A}) \le \delta$ if and only if

$$\sup_{\boldsymbol{x}\in\Sigma_k} \left| \|\boldsymbol{Ax}\|_2^2 - 1 \right| \le \delta. \tag{3.4.9}$$

This is equivalent to

$$\sup_{\boldsymbol{x}\in\Sigma_k} |\langle \boldsymbol{A}^*\boldsymbol{Ax}, \boldsymbol{x}\rangle - 1| \le \delta. \tag{3.4.10}$$

Lemma 3.24 (Discretization). *Suppose we have a set* $\bar{\mathsf{N}} \subseteq \Sigma_k$ *with the following property: for all* $\boldsymbol{x} \in \Sigma_k$*, there exists* $\bar{\boldsymbol{x}} \in \bar{\mathsf{N}}$ *such that*

- $|\mathrm{supp}\,(\bar{\boldsymbol{x}}) \cup \mathrm{supp}\,(\boldsymbol{x})| \le k$
- $\|\boldsymbol{x}-\bar{\boldsymbol{x}}\|_2 \le \varepsilon$.

set

$$\delta_{\bar{\mathsf{N}}} = \max_{\bar{\boldsymbol{x}}\in\bar{\mathsf{N}}} \left| \|\boldsymbol{A}\bar{\boldsymbol{x}}\|_2^2 - 1 \right|. \tag{3.4.11}$$

Then

$$\delta_k(\boldsymbol{A}) \le \frac{\delta_{\bar{\mathsf{N}}} + 2\varepsilon}{1-2\varepsilon}. \tag{3.4.12}$$

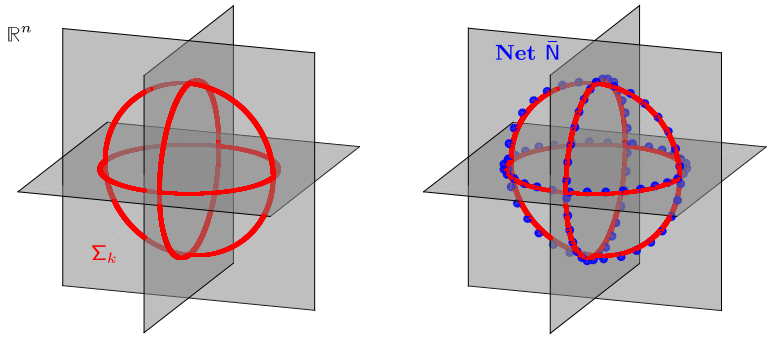


Figure 3.11 The Set Σ_k of Unit Norm Sparse Vectors. Left: visualization of the set $\Sigma_k = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_0 \leq k, \|\boldsymbol{x}\|_2 = 1\}$ of unit norm sparse vectors. Here, $k = 2$ and $n = 3$. **Right:** an ε-net $\bar{\mathsf{N}}$ for this set.

So, provided ε is small, not much changes if we restrict our calculation to the finite set $\bar{\mathsf{N}}$. The proof of this result uses the fact that if $\boldsymbol{x}$ and $\boldsymbol{z}$ are k-sparse vectors,

$$\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{z} \rangle \leq \sqrt{\|\boldsymbol{A}\boldsymbol{x}\|_2^2 \|\boldsymbol{A}\boldsymbol{z}\|_2^2} \leq (1 + \delta_k(\boldsymbol{A})) \|\boldsymbol{x}\|_2 \|\boldsymbol{z}\|_2 . \tag{3.4.13}$$

Proof Take any $\boldsymbol{x} \in \Sigma_k$ and choose $\bar{\boldsymbol{x}} \in \bar{\mathsf{N}}$ such that $\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_0 \leq k$ and $\|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2 \leq \varepsilon$. We have

$$\begin{aligned}
\left| \|\boldsymbol{A}\boldsymbol{x}\|_2^2 - 1 \right| &= |\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} \rangle - 1| && (3.4.14)\\
&= |\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} \rangle - \langle \boldsymbol{A}\bar{\boldsymbol{x}}, \boldsymbol{A}\bar{\boldsymbol{x}} \rangle + \langle \boldsymbol{A}\bar{\boldsymbol{x}}, \boldsymbol{A}\bar{\boldsymbol{x}} \rangle - 1| && (3.4.15)\\
&\leq |\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} \rangle - \langle \boldsymbol{A}\bar{\boldsymbol{x}}, \boldsymbol{A}\bar{\boldsymbol{x}} \rangle| + \delta_{\bar{\mathsf{N}}} && (3.4.16)\\
&= |\langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}(\boldsymbol{x} - \bar{\boldsymbol{x}}) \rangle - \langle \boldsymbol{A}\bar{\boldsymbol{x}}, \boldsymbol{A}(\bar{\boldsymbol{x}} - \boldsymbol{x}) \rangle| + \delta_{\bar{\mathsf{N}}} && (3.4.17)\\
&\leq 2 (1 + \delta_k(\boldsymbol{A})) \varepsilon + \delta_{\bar{\mathsf{N}}}. && (3.4.18)
\end{aligned}$$

Since this inequality holds for all $\boldsymbol{x} \in \Sigma_k$, we obtain that

$$\delta_k(\boldsymbol{A}) \leq 2 (1 + \delta_k(\boldsymbol{A})) \varepsilon + \delta_{\bar{\mathsf{N}}}, \tag{3.4.19}$$

from which the target inequality follows. □

The next task is to construct a set $\bar{\mathsf{N}}$ which has the desired good properties. We call a set N *an ε-net* for a given set S if

$$\forall\, \boldsymbol{x} \in \mathsf{S}, \quad \exists\, \bar{\boldsymbol{x}} \in \mathsf{N} \quad \text{such that} \quad \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2 \leq \varepsilon. \tag{3.4.20}$$

Let

$$\mathsf{B}(\boldsymbol{x}, r) = \{\boldsymbol{z} \in \mathbb{R}^d \mid \|\boldsymbol{z} - \boldsymbol{x}\|_2 \leq r\} \tag{3.4.21}$$

denote the ℓ^2 ball of center $\boldsymbol{x}$ and radius r, in $\mathbb{R}^d$. The following clever argument

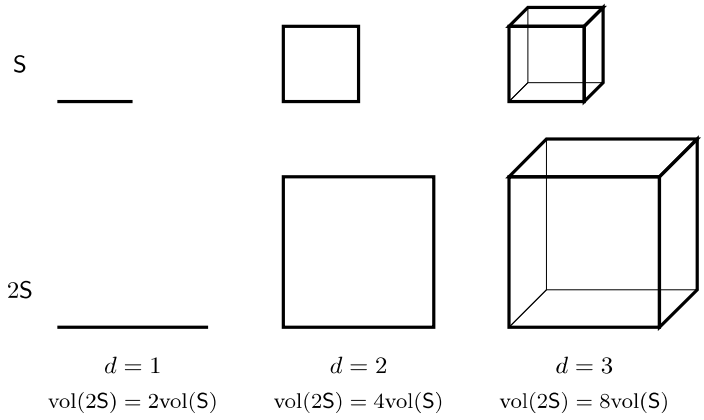


Figure 3.12 Volumes Scale as α^d.

shows that there exists an ε-net for the ℓ^2 ball $\mathsf{B}(\mathbf{0}, 1)$ of size at most $(3/\varepsilon)^d$. It uses the fact that if $\mathsf{S} \subset \mathbb{R}^d$ is a set, and

$$\alpha\mathsf{S} = \{\alpha \boldsymbol{s} \mid \boldsymbol{s} \in \mathsf{S}\} \tag{3.4.22}$$

denotes its α dilation, then

$$\mathrm{vol}(\alpha\mathsf{S}) \;\leq\; \alpha^d \mathrm{vol}(\mathsf{S}). \tag{3.4.23}$$

See Figure 3.12 for a visualization of this.

LEMMA 3.25 (ε-Nets for the Unit Ball). *There exists an ε-net for the unit ball $\mathsf{B}(0, 1) \subset \mathbb{R}^d$ of size at most $(3/\varepsilon)^d$.*

Proof Call a set ε-separated if every pair of distinct points in M has distance at least ε. Let $\mathsf{N} \subset \mathsf{B}(\mathbf{0}, 1)$ be a *maximal* ε-separated set. Here, maximal means that it is not contained in any larger ε-separated set.

We claim that N is an ε-net for $\mathsf{B}(\mathbf{0}, 1)$. Indeed, if it is not an ε-net, then there exists some point $\boldsymbol{x} \in \mathsf{B}(\mathbf{0}, 1)$ with distance greater than ε to each element of N. Adding $\boldsymbol{x}$ to N, we obtain a larger ε-separated set, contradicting maximality of N.

Since N is ε-separated, the balls $\mathsf{B}(\boldsymbol{x}, \varepsilon/2)$ and $\mathsf{B}(\boldsymbol{x}', \varepsilon/2)$ are disjoint, for any pair of distinct elements $\boldsymbol{x} \neq \boldsymbol{x}' \in \mathsf{N}$. Moreover, the union of these balls is contained in $\mathsf{B}(\mathbf{0}, 1 + \varepsilon/2)$. Thus,

$$|\mathsf{N}|\, \mathrm{vol}(\mathsf{B}(\mathbf{0}, \varepsilon/2)) \;\leq\; \mathrm{vol}(\mathsf{B}(\mathbf{0}, 1 + \varepsilon/2)). \tag{3.4.24}$$

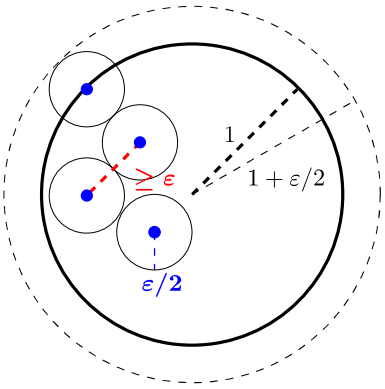


Figure 3.13 Volume Calculation for an ε-Net. An ε-separated set. The interiors of $\varepsilon/2$ balls around the points do not intersect. The union of the $\varepsilon/2$ balls is contained in an $(1+\varepsilon/2)$-ball.

Hence,

$$|\mathsf{N}| \leq \frac{\mathrm{vol}(\mathsf{B}(\mathbf{0}, 1+\varepsilon/2))}{\mathrm{vol}(\mathsf{B}(\mathbf{0}, \varepsilon/2))} \tag{3.4.25}$$

$$= \left(\frac{1+\varepsilon/2}{\varepsilon/2}\right)^d = (1+2/\varepsilon)^d \tag{3.4.26}$$

$$\leq (3/\varepsilon)^d \tag{3.4.27}$$

as desired. □

To construct our target set $\bar{\mathsf{N}}$, we simply consider each support pattern I of size $|\mathsf{I}| = k$ individually. There are $\binom{n}{k}$ such patterns. For each pattern, we use the previous lemma to build an ε-net N for the unit ball of vectors of ℓ^2 norm at most one, whose support is contained in I. Each of these nets has size at most $(3/\varepsilon)^k$. So, finally, we obtain

Lemma 3.26. *There exists an ε-net $\bar{\mathsf{N}}$ for Σ_k satisfying the two properties required in Lemma 3.24, with*

$$\left|\bar{\mathsf{N}}\right| \;\leq\; \exp\Big(k\log(3/\varepsilon) + k\log(n/k) + k\Big). \tag{3.4.28}$$

Proof The construction follows the above discussion. Using the Stirling's formula,[7] we can estimate

$$\left|\bar{\mathsf{N}}\right| \leq (3/\varepsilon)^k \binom{n}{k} \tag{3.4.29}$$

$$\leq (3/\varepsilon)^k \left(\frac{ne}{k}\right)^k \tag{3.4.30}$$

[7] Stirling's formula gives the bounds for factorials: $\sqrt{2\pi k}(\frac{k}{e})^k \leq k! \leq e\sqrt{k}(\frac{k}{e})^k$.

as desired. □

Union Bound.

Proof For each $\boldsymbol{x} \in \bar{\mathsf{N}}$, $\boldsymbol{A}\boldsymbol{x}$ is a random vector with entries independent $\mathcal{N}(0, 1/m)$. We have

$$\mathbb{P}\left[\left|\|\boldsymbol{A}\boldsymbol{x}\|_2^2 - 1\right| > t\right] \leq 2\exp(-mt^2/8). \tag{3.4.31}$$

Hence, summing over all elements of $\bar{\mathsf{N}}$, we have

$$\mathbb{P}\left[\delta_{\bar{\mathsf{N}}} > t\right] \leq 2\left|\bar{\mathsf{N}}\right| \exp\left(-mt^2/8\right) \tag{3.4.32}$$

$$\leq 2\exp\Big(-\frac{mt^2}{8} + k\log\big(\frac{n}{k}\big) + k(\log\big(\frac{3}{\varepsilon}\big) + 1)\Big). \tag{3.4.33}$$

On the complement of the event $\delta_{\bar{\mathsf{N}}} > t$, we have

$$\delta_k(\boldsymbol{A}) < \frac{2\varepsilon + t}{1 - 2\varepsilon}. \tag{3.4.34}$$

Setting $\varepsilon = \delta/8$, $t = \delta/4$, and ensuring that $m \geq Ck\log(n/k)/\delta^2$ for sufficiently large numerical constant C, we obtain the result. □

In the above derivation, especially from equation (3.4.33), we see that a slight more tight bound for m is of the form

$$m \geq 128k\log(n/k)/\delta^2 + (\log(24/\delta) + 1)k/\delta^2 \doteq C_1 k\log(n/k) + C_2 k.$$

However, for a small δ, the constants C_1 and C_2 can be rather large. Although qualitatively this bound is in the right form, it does not reflect exactly when ℓ^1 minimization works. In the work of [RV08], a much tighter bound for m is given as:

$$m \geq 8k\log(n/k) + 12k.$$

This is one of the best known bounds given through the RIP properties of Gaussian matrices. Nevertheless, as we will see later, using more advanced tools, ultimately we will be able to derive for Gaussian matrices a precise condition that characterizes the "phase-transition" behavior for the success of ℓ^1 minimization that we can observe through simulations.

3.4.3 RIP of Non-Gaussian Matrices

In many applications of interest, the matrix $\boldsymbol{A}$ cannot be assumed to be iid Gaussian. Perhaps surprisingly, often the theory developed for the Gaussian model is predictive of the behavior of ℓ^1 minimization in other models. However, it is still desirable to have a precise understanding (and corresponding mathematical guarantees) to describe what happens when the model is not so homogeneous.

Random Submatrices of a Unitary Matrix.
One model that occurs quite often posits that we generate $\boldsymbol{A}$ by randomly sampling some rows of an orthognonal matrix (in the real case) or a unitary matrix (in the complex case). Actually, we have already seen such a model in our brief discussion of MRI applications. There, we generated $\boldsymbol{A}$ as a row submatrix of $\boldsymbol{F}\boldsymbol{\Psi}$, where $\boldsymbol{F}$ was the DFT matrix, and $\boldsymbol{\Psi} \in \mathbb{C}^{n\times n}$ was a matrix whose columns formed an orthonormal wavelet basis for $\mathbb{C}^{n\times n}$. Since both $\boldsymbol{F}$ and $\boldsymbol{\Psi}$ were unitary, their product is unitary. In the work [CRT06a], it has been shown that for a given k-sparse vector $\boldsymbol{x} \in \mathbb{R}^n$, if $\boldsymbol{A}$ randomly takes $m = O\big(k\log(n)\big)$ rows of a unitary matrix, then with high probability the ℓ^1 minimization $\min \|\boldsymbol{x}\|_1$ s.t. $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ recovers the sparse vector. However, this result does not imply that with the same $\boldsymbol{A}$, the ℓ^1 minimization succeeds for all k-sparse vectors.[8]

The following theorem, according to [RV08], shows that if we sample a random row submatrix from a unitary matrix, it also has RIP with high probability, provided enough rows are chosen. We know that for a matrix satisfying the RIP condition, it is guaranteed that the associated ℓ^1 minimization succeeds for all k-sparse vectors.

THEOREM 3.27. *Let $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ be unitary ($\boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{I}$) and Ω is a random subset of m elements from $\{1,\dots,n\}$. Suppose that*

$$\|\boldsymbol{U}\|_\infty \;\le\; \zeta/\sqrt{n}. \tag{3.4.35}$$

If

$$m \;\ge\; \frac{C\zeta^2}{\delta^2} k\log^4(n), \tag{3.4.36}$$

then with high probability, $\boldsymbol{A} = \sqrt{\frac{n}{m}}\boldsymbol{U}_{\Omega,\bullet}$ satisfies the RIP of order k, with constant $\delta_k(\boldsymbol{A}) \le \delta$.

For simplicity, here we do not give a proof to this theorem and interested readers may refer to the work of [RV08].

In our context, there are two very salient points about this result. The first is the dependence on $\|\boldsymbol{U}\|_\infty$. It is worth noting that for any unitary matrix $\boldsymbol{U}$, $\|\boldsymbol{U}\|_\infty \;\ge\; 1/\sqrt{n}$. So, the parameter ζ measures how much we lose with respect to this optimal bound. The bound is clearly achievable in some cases – the DFT matrix $\boldsymbol{F}$ has $\|\boldsymbol{F}\|_\infty = 1/\sqrt{n}$, which follows directly from its definition (A.7.13) in Appendix A. If we are willing to interpret the result a bit, the idea that $\boldsymbol{U}$ should have uniformly bounded elements leads to a very nice intuition about sampling. Namely, if we wish to reconstruct an element that is sparse in some basis $\boldsymbol{\Psi}$, and we can take whatever linear samples $\langle \boldsymbol{f}_i, \boldsymbol{y}\rangle$ we want, we should

[8] To see the difference, one can recall in the Johnson-Lindenstrauss Lemma, the task is not just to show that given any pair of points, with high probability there exists a projection that approximately preserves the distance. We need to use the union bound to show that with high probability there exists a projection that approximately preserves the distance between all pairs simultaneously.

take samples that are as *incoherent* with the basis of sparsity as possible, in the sense that

$$\langle \boldsymbol{f}_i, \boldsymbol{\psi}_j \rangle \tag{3.4.37}$$

is uniformly small. This is in contrast to our usual intuition from signal processing, which might suggest that some sort of matched filter would be the best here. The challenge is that there are actually an exponentially large number of potential support patterns for $\boldsymbol{x}$, and hence an exponentially large number of signals to match. If, instead, we let each (incoherent) measurement collect information across all of the basis elements, we can then, using efficient computation, decide which elements of $\boldsymbol{\Psi}$ are active.

The second salient point is that the number of measurements, $k \log^4(n)$ is visually similar to the $k \log(n/k)$ that we saw for the Gaussian ensemble. It is currently conjectured that here $k \log n$ measurements suffice. It is currently an open problem to show this; it is considered hard, and known to connect to a number of interesting questions in probability and functional analysis. In fact, in [RV08] a more precise expression is given as: $m = O\big(k \log(n) \log^2(k) \log(k \log n)\big)$. This bound, against the conjectured optimal bound, is within a $\log \log(n)$ factor for n and within a $\log^3(k)$ factor for k.

Random Convolutions.
Another model that occurs quite frequently in engineering practice involves sampling the convolution of the input signal $\boldsymbol{x}$ with some filter $\boldsymbol{r}$. Formally, we can imagine that

$$\boldsymbol{y} = \mathcal{P}_\Omega[\boldsymbol{r} * \boldsymbol{x}] = \boldsymbol{A}\boldsymbol{x}, \tag{3.4.38}$$

where $\boldsymbol{x} \in \mathbb{C}^n$, $\boldsymbol{r} \in \mathbb{C}^n$, and $\Omega \subseteq [n]$ is our collection of sampling locations. Here, $*$ denotes circular convolution:

$$(\boldsymbol{r} * \boldsymbol{x})_i = \sum_{j=0}^{n-1} x_j r_{i+n-j \bmod n}. \tag{3.4.39}$$

This leads to a highly structured linear operator on $\boldsymbol{x}$ since we can represent the convolution in a circulant form as

$$\boldsymbol{r} * \boldsymbol{x} \quad = \quad \begin{bmatrix} r_0 & r_{n-1} & \dots & r_2 & r_1 \\ r_1 & r_0 & r_{n-1} & & r_2 \\ \vdots & r_1 & r_0 & \ddots & \vdots \\ r_{n-2} & & \ddots & \ddots & r_{n-1} \\ r_{n-1} & r_{n-2} & \dots & r_1 & r_0 \end{bmatrix} \boldsymbol{x} \quad \doteq \quad \boldsymbol{R}\boldsymbol{x}. \tag{3.4.40}$$

Such a matrix $\boldsymbol{R}$ is called a *circulant matrix*. One may see Appendix A for more nice properties of this type of matrices. In particular, any circulant matrix can be diagonalized by the discrete Fourier transform: $\boldsymbol{R} = \boldsymbol{F}\boldsymbol{D}\boldsymbol{F}^*$ for some diagonal matrix $\boldsymbol{D}$ (see Theorem A.32 of Appendix A). Here, we can view the sampling matrix $\boldsymbol{A}$ as taking a subset of rows of the circulant matrix $\boldsymbol{R}$, that is $\boldsymbol{A} = \boldsymbol{R}_{\Omega,\bullet}$.

The filter $\boldsymbol{r}$ can be rather general as well. For instance, it could as simple as a random Rademacher vector, i.e., a random vector with independent entries distributed according to $\mathbb{P}(r_i = \pm 1) = 0.5$, or it could be a random vector with independent zero-mean, subgaussian random variables of variance one. The exact randomness of $\boldsymbol{r}$ is not critical.

For this model, the work of [KMR14] has shown that essentially the following statement is true:

THEOREM 3.28. *Let $\Omega \subseteq \{1, \dots, n\}$ be any fixed subset of size $|\Omega| = m$. Then if*

$$m \geq \frac{Ck \log^2(k) \log^2(n)}{\delta^2}, \tag{3.4.41}$$

then with high probability, $\boldsymbol{A}$ has RIP of order k with $\delta_k(\boldsymbol{A}) \leq \delta$.

Notice that the above statement is rather strong in the following sense: Firstly, it states that even for a highly structured sampling matrix (a circulant matrix versus a random Gaussian matrix studied in previous section), we only lose a small factor of $\log^2(k)\log(n)$ in the required number of samples. Secondly, it claims that any subset of rows of $\boldsymbol{R}$ has the RIP property, not just a random subset with high probability. Thirdly, the RIP property ensures recoverability of any k-sparse vectors $\boldsymbol{x}$ uniformly not just for a fixed k-sparse vector. It has been shown in [Rau09] that, if one relaxes the uniform recoverability requirement, considering only a fixed k-sparse vector, it can be recovered via ℓ^1-minimization from a partial random circulant matrix with $m \geq Ck \log^2(n)$ measurements. This bound is slightly better than the one given in the theorem, but it is not uniform for all k-sparse vectors.

3.5 Noisy Observations or Approximate Sparsity

Thus far, we have been very idealistic in our model. We have assumed that the target $\boldsymbol{x}_o$ is perfectly sparse, and that there is no noise in the measurements, so $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ exactly. These assumptions are clearly violated in many practical applications. In practice, the observation $\boldsymbol{y}$ is usually perturbed by some amount of noise $\boldsymbol{z}$, which we assume to be small:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}, \quad \|\boldsymbol{z}\|_2 \leq \varepsilon. \tag{3.5.1}$$

In other practical scenarios, the ground truth signal $\boldsymbol{x}_o$ may not be perfectly sparse and may be only approximately so.

This motivates two natural questions. First, on the practical side, is it possible to modify our approaches to be stable under noise or for imperfect sparse signals? Second, what should we expect of their performance? Do the conditions and guarantees we introduced in previous sections remain meaningful?

To clearly state our assumptions and goals, we can consider the following three scenarios (or some combination of them):

- **Deterministic (worst case) noise**: $\boldsymbol{z}$ is bounded: $\|\boldsymbol{z}\|_2 \leq \varepsilon$, and ε is known.
- **Stochastic noise**: entries of $\boldsymbol{z} \sim_{iid} \mathcal{N}(0, \frac{\sigma^2}{m})$. Notice that under this random model, a typical noise vector $\boldsymbol{z}$ is of norm $\|\boldsymbol{z}\|_2 \approx \sigma$.[9] Gaussian noise is a very natural assumption; the results obtained here also extend to other noise models.
- **Inexact sparsity**: $\boldsymbol{x}_o$ is not perfectly sparse. Technically speaking, this is not noise, but rather a violation of our sparse modeling assumption. In this scenario, it may be meaningful to assume that $\boldsymbol{x}_o$ is *close* to a k-sparse vector. We can formalize this by letting $[\boldsymbol{x}_o]_k$ denote a best k-term approximation to $\boldsymbol{x}_o$:

$$[\boldsymbol{x}_o]_k \in \arg\min_{\|\boldsymbol{z}\|_0 \leq k} \|\boldsymbol{x}_o - \boldsymbol{z}\|_2^2 . \tag{3.5.2}$$

  This just keeps the k largest elements of $\boldsymbol{x}_o$. $\boldsymbol{x}_o$ is said to be "approximately sparse" if $\|\boldsymbol{x}_o - [\boldsymbol{x}_o]_k\|$ is small.

In all of these scenarios, we might hope to still "recover" a sparse estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}_o$ in some sense. There are (perhaps) three natural senses to consider:

- **Estimation**: Is $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2$ small?
- **Prediction**: Is $\boldsymbol{A}\hat{\boldsymbol{x}} \approx \boldsymbol{A}\boldsymbol{x}_o$?
- **Support recovery**: Is $\text{supp}(\hat{\boldsymbol{x}}) = \text{supp}(\boldsymbol{x}_o)$?

For engineering practice, we often care about either estimating the signal $\boldsymbol{x}_o$ (for sensing problems) or recovering its support $\text{supp}(\boldsymbol{x}_o)$ (for recognition problems). Nevertheless, statisticians sometimes also care about the prediction error $\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)$.

In the following subsections, we discuss results on stable estimation under (i) deterministic noise, (ii) stochastic noise and (iii) deterministic noise *and* inexact sparsity. Results on support recovery are discussed briefly in Section 3.6 and in the Notes section of this chapter.

3.5.1 Stable Recovery of Sparse Signals

In the ideal sensing model, the observation equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$ holds exactly for a sparse signal $\boldsymbol{x}_o$. In this subsection, we consider a more practical situation in which the observation $\boldsymbol{y}$ is perturbed by some amount of noise. For simplicity, we still assume the signal $\boldsymbol{x}_o$ is perfectly sparse. We can model the noise as an additive error $\boldsymbol{z}$, which we will assume to have a small magnitude:[10]

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}, \quad \|\boldsymbol{z}\|_2 \leq \varepsilon. \tag{3.5.3}$$

[9] We scale the variance of the normal distribution by $1/m$ on purpose, so that σ is directly comparable to ε in the deterministic noise case.

[10] This is similar to the setting in conventional signal processing problems where we typically assume the signal to noise ratio (SNR) is large.

To recover a sparse solution from the above observation, we may extend ℓ^1 minimization to this new setting and solve

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \varepsilon. \end{array} \tag{3.5.4}$$

In words, this program asks us to (try to) find the sparest $\boldsymbol{x}$ that agrees with the observation up to the noise level. Almost equally popular is the Lagrangian relaxation of this problem, which introduces a penalty parameter $\lambda \geq 0$, and solves the unconstrained optimization problem[11]

$$\min \quad \lambda \|\boldsymbol{x}\|_1 + \tfrac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 . \tag{3.5.5}$$

The optimization (3.5.4) is almost uniformly referred to as "Basis Pursuit Denoising" [CDS01], while the problem (3.5.5) is almost uniformly referred to as the "Lasso (Least absolute shrinkage and selection operator)" [Tib96]. These two problems are completely equivalent, in the sense that there is a calibration $\lambda \leftrightarrow \varepsilon$ such that if $\boldsymbol{x}$ is a solution to the Lasso problem for some choice of λ, then there exists an ε such that $\boldsymbol{x}$ is also a solution to the BPDN problem with parameter ε, and conversely, whenever $\boldsymbol{x}$ is a solution to BPDN with parameter ε, there exists a corresponding λ such that $\boldsymbol{x}$ also solves the Lasso problem with parameter λ. So, from a theoretical perspective, these two problems are completely equivalent.

On the other hand, from a practical perspective, they may be quite different, since the calibration $\lambda \leftrightarrow \varepsilon$ depends on the problem data $(\boldsymbol{y}, \boldsymbol{A})$, and no explicit form is known. In some situations, it may be easier to tune λ than ε, or vice versa. In particular, in situations in which the norm of the noise is known or can be estimated, the BPDN formulation may be more attractive, since its parameter can be set to be the noise level.[12] The optimal choice of the regularization parameter λ (or ε) is a surprisingly tricky issue in practice. In general, we have to either use generic statistical rules such as cross validation, or resort to theoretical analysis to get some insight into what scalings make sense.

Despite their conceptual equivalence, these problems may require rather different optimization techniques. In Chapter 8, we will discuss in more details about how to solve both (and many related problems!).

Deterministic Noise.

To account for measurement noise, we can simply solve one of (3.5.4) or (3.5.5). Both are convex problems. Any global minimizer gives an estimate $\hat{\boldsymbol{x}}$. Unlike the previous two sections, under noise we cannot expect $\hat{\boldsymbol{x}} = \boldsymbol{x}_o$ exactly. However, we *can* hope that if the noise level ε is small, the estimation error $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2$ will also be small.

[11] One may compare this with the ridge regression that regularizes the ℓ^2 norm of $\boldsymbol{x}$, which we have introduced in Exercise 1.8 of Chapter 1.

[12] Historically, the Lasso is preferred by statisticians, and BPDN by engineers, although confusingly, in the original papers the names Lasso and BPDN are not used to refer to these problems, but rather different equivalent problems!

How well do we expect to do? Imagine that we somehow knew the support I of $\boldsymbol{x}_o$. In this situation, we could form another estimate $\hat{\boldsymbol{x}}'$, by setting

$$\begin{cases} \hat{\boldsymbol{x}}'(\mathsf{I}) = (\boldsymbol{A}_\mathsf{I}^* \boldsymbol{A}_\mathsf{I})^{-1} \boldsymbol{A}_\mathsf{I}^* \boldsymbol{y}, \\ \hat{\boldsymbol{x}}'(\mathsf{I}^c) = \boldsymbol{0}. \end{cases} \tag{3.5.6}$$

This is just the least squares estimate, restricted to the set I. It is not difficult to argue that it is optimal, in the sense that it minimizes over all estimators, the worst error $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2$ over all $\boldsymbol{x}_o$ supported on I and $\boldsymbol{z}$ of norm at most ε. This "oracle" estimator produces an estimate $\hat{\boldsymbol{x}}'$ that satisfies

$$\left\|\hat{\boldsymbol{x}}' - \boldsymbol{x}_o\right\|_2 \;\leq\; \frac{\varepsilon}{\sigma_{\min}(\boldsymbol{A}_\mathsf{I})}, \tag{3.5.7}$$

and this bound can be tight.

So, the best we can possibly hope for in general is

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 \sim c\varepsilon,$$

with $c = \sigma_{\min}(\boldsymbol{A}_\mathsf{I})^{-1}$. As above, if we restrict ourselves to efficient algorithms, this is too much to hope for in general. However, can we still hope that under the same hypotheses as above,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 \leq C\varepsilon? \tag{3.5.8}$$

That is to say, the solution is at least *stable*: the error in estimating $\boldsymbol{x}$ is proportional to the size ε of the perturbation, even though the constant might not be as small as when we know the oracle of the correct support of $\boldsymbol{x}_o$.

The theorem below, which is similar to that in [CRT06b],[13] makes this precise:

THEOREM 3.29 (Stable Sparse Recovery via BPDN). *Suppose that* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, *with* $\|\boldsymbol{z}\|_2 \leq \varepsilon$, *and let* $k = \|\boldsymbol{x}_o\|_0$. *If* $\delta_{2k}(\boldsymbol{A}) < \sqrt{2} - 1$, *then any solution* $\hat{\boldsymbol{x}}$ *to the optimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \varepsilon \end{array} \tag{3.5.9}$$

satisfies

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 \leq C\varepsilon. \tag{3.5.10}$$

Here, C *is a constant which depends only on* $\delta_{2k}(\boldsymbol{A})$ *(and not on the noise level* ε*).*

Proof Because $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o\|_2 = \|\boldsymbol{z}\|_2 \leq \varepsilon$. Since $\hat{\boldsymbol{x}}$ is feasible, we have $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2 \leq$

[13] The condition on RIP constant in [CRT06b] was $\delta_{4k}(\boldsymbol{A}) < 1/4$, which is more restrictive than the one shown here.

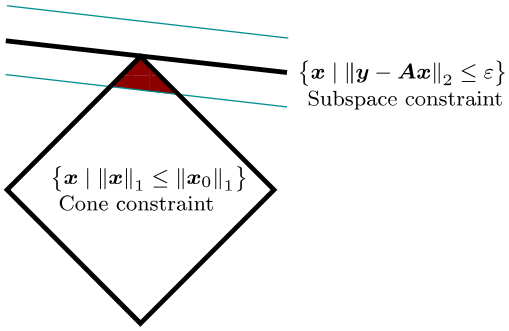


Figure 3.14 Geometry of the proof of Theorem 3.29.

ε as well. Using the triangle inequality,

$$\begin{aligned}\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2 &= \|(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}) - (\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o)\|_2 \\ &\leq \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2 + \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o\|_2 \\ &\leq 2\varepsilon.\end{aligned}$$

Let $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}_o$, we have $\|\boldsymbol{A}\boldsymbol{h}\|_2 \leq 2\varepsilon$. Geometrically, this means that the perturbation $\boldsymbol{h}$ must be close to the null space of $\boldsymbol{A}$.

Because $\boldsymbol{x}_o$ is feasible for the optimization problem, and $\hat{\boldsymbol{x}}$ is optimal, $\hat{\boldsymbol{x}}$ must have a lower objective function value than $\boldsymbol{x}_o$:

$$\|\hat{\boldsymbol{x}}\|_1 \;\leq\; \|\boldsymbol{x}_o\|_1 . \tag{3.5.11}$$

Let I denote the support of $\boldsymbol{x}_o$. We have

$$\begin{aligned}\|\boldsymbol{x}_o\|_1 &\geq \|\boldsymbol{x}_o + \boldsymbol{h}\|_1 \\ &\geq \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 ,\end{aligned}$$

and so

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \;\leq\; \|\boldsymbol{h}_{\mathsf{I}}\|_1 . \tag{3.5.12}$$

Geometrically, this means that $\hat{\boldsymbol{x}}$ lives in an ℓ^1 ball of radius $\|\boldsymbol{x}_o\|_1$, centered at the origin. Locally, this set looks like a convex cone (the "descent cone" of the ℓ^1 norm), hence the constraint $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \;\leq\; \|\boldsymbol{h}_{\mathsf{I}}\|_1$ is also known as a "cone constraint". It describes the set of all possible perturbations of $\hat{\boldsymbol{x}}$ from $\boldsymbol{x}_o$ that would decrease the value of the objective function. The geometric intuition behind the two constraints on the perturbation $\boldsymbol{h}$ is shown in Figure 3.14.

Note that the matrix $\boldsymbol{A}$ satisfies RIP. According to Theorem 3.17, we know that if $\delta_{2k} < \sqrt{2} - 1$, $\boldsymbol{A}$ satisfies the restricted strong convexity property with constant $\alpha = 1$ (which is the case for the restriction condition (3.5.12) on $\boldsymbol{h}$ above). Therefore, we have

$$\|\boldsymbol{A}\boldsymbol{h}\|_2^2 \geq \mu\|\boldsymbol{h}\|_2^2 \tag{3.5.13}$$

for some $\mu > 0$. Combining this with $\|\boldsymbol{A}\boldsymbol{h}\|_2 \leq 2\varepsilon$, we have

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 = \|\boldsymbol{h}\|_2 \leq \frac{2}{\sqrt{\mu}}\varepsilon. \tag{3.5.14}$$

Choosing $C = \frac{2}{\sqrt{\mu}}$ completes the proof. □

Notice that in the above proof, the constant C can be rather large if μ is very small. According to the proof of Theorem 3.17, we know

$$\sqrt{\mu} = \frac{1 - \delta_{2k}(1+\sqrt{2})}{\sqrt{2(1+\delta_{2k})}}.$$

The quantity μ becomes small if δ_{2k} is close to $\sqrt{2}-1$. Therefore, if we do not want the constant C in the above theorem to be too large, we need to ensure that δ_{2k} is significantly smaller than $\sqrt{2}-1$. However, no matter how small δ_{2k} is, we always have $\sqrt{\mu} < 1/\sqrt{2}$. Hence, based on this proof, the smallest that the constant C can be in the theorem is $2\sqrt{2}$.

Random Noise.
Above, we have shown that for any additive noise $\boldsymbol{z}$ in the observation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, we can estimate $\boldsymbol{x}_o$ with an error of size controlled by $C\|\boldsymbol{z}\|_2$ for some constant C. Based on our discussion before the theorem, this error bound is already close to the best possible.

For random noise, we might hope that if $m \gg k$, most of the energy of $\boldsymbol{z}$ would "miss" the k-dimensional subspace range($\boldsymbol{A}_\mathsf{I}$). If so, the accuracy in the estimated $\hat{\boldsymbol{x}}$ can improve as m grows. More precisely, the coefficient C in the error bound $C\|\boldsymbol{z}\|_2$ decreases as m increases. This turns out to be the case. For simplicity, we here state a theorem for random $\boldsymbol{A}$.[14] More precisely, we assume that the measurement model:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}. \tag{3.5.15}$$

where $\boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{x}_o$ k-sparse, and the matrix $\boldsymbol{A} \sim_{iid} \mathcal{N}(0, \frac{1}{m})$ and $\boldsymbol{z} \sim_{iid} \mathcal{N}(0, \frac{\sigma^2}{m})$. Notice that in the study of the deterministic case, we have assumed the measurement matrix $\boldsymbol{A}$ is a matrix that satisfies RIP conditions. Hence the norm of the columns of $\boldsymbol{A}$ there is typically normalized to one. Here the scaling factor $\frac{1}{m}$ in the variance is to ensure the columns of $\boldsymbol{A}$ is typically of length one and the noise vector of length σ so that the model and the results will be directly comparable to those for the deterministic case.[15]

As we have discussed earlier, with noisy measurements, we could find an estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}_o$ that strikes a balance between sparsity and minimizing the error. In particular, we would like to solve the following Lasso program for $\hat{\boldsymbol{x}}$:

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_m \|\boldsymbol{x}\|_1 . \tag{3.5.16}$$

[14] An analogous result holds for $\boldsymbol{A}$ satisfying the RIP.
[15] The variance σ replaces the role of ε in Theorem 3.29.

As usual, for convenience, we let $\mathsf{I} = \text{supp}(\boldsymbol{x}_o)$, let I^c denote its complement, and $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}_o \in \mathbb{R}^n$ the difference between the estimate and the ground truth. We also define $L(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$. Notice that $\nabla L(\boldsymbol{x}) = -\boldsymbol{A}^*(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})$ and in particular $\nabla L(\boldsymbol{x}_o) = -\boldsymbol{A}^*(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o) = -\boldsymbol{A}^*\boldsymbol{z}$ according to (3.5.15).

We want to know how small the difference $\|\boldsymbol{h}\| = \|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|$ is for a given λ_m. First we show that for a properly chosen λ_m, the difference vector $\boldsymbol{h}$ is highly *restricted* in the way that $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \alpha\|\boldsymbol{h}_{\mathsf{I}}\|_1$ for some constant α, i.e., the error off the support I of $\boldsymbol{x}_o$ is controlled by that on I.[16] More precisely, we have the following lemma.

LEMMA 3.30. *For the optimization problem (3.5.16), if we choose the regularization parameter* $\lambda_m \geq c \cdot 2\sigma\sqrt{\frac{\log n}{m}}$, *then with high probability,* $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}_o$ *satisfies the cone condition:*

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \frac{c+1}{c-1} \cdot \|\boldsymbol{h}_{\mathsf{I}}\|_1, \tag{3.5.17}$$

where I *is the support of the sparse* $\boldsymbol{x}_o$.

Proof Note that the difference between $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}_o$ is related to the difference between the values of the objective function in (3.5.16). Since $\hat{\boldsymbol{x}}$ minimizes the objective function, we have:

$$\begin{aligned} 0 &\geq L(\hat{\boldsymbol{x}}) + \lambda_m\|\hat{\boldsymbol{x}}\|_1 - L(\boldsymbol{x}_o) - \lambda_m\|\boldsymbol{x}_o\|_1 \\ &\geq \langle \nabla L(\boldsymbol{x}_o), \hat{\boldsymbol{x}} - \boldsymbol{x}_o\rangle + \lambda_m(\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1) \\ &\geq -\,|\,\langle \boldsymbol{A}^*\boldsymbol{z}, \boldsymbol{h}\rangle\,| + \lambda_m(\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1) \\ &\geq -\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty\|\boldsymbol{h}\|_1 + \lambda_m(\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1), \end{aligned} \tag{3.5.18}$$

where the second inequality we used the fact that $L(\boldsymbol{x})$ is a convex function. It remains to be seen how the two terms in the last inequality interact. Obviously we need to have a good idea about the value of $\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty$. This is where we need to resort to results about measure concentration of high-dimensional statistics.

Notice that the column $\boldsymbol{a}_i$ of $\boldsymbol{A}$ is typically of norm $\|\boldsymbol{a}_i\|_2 \approx 1$. Hence here we may assume the columns of $\boldsymbol{A}$ are all normalized to one. Therefore $\boldsymbol{a}_i^*\boldsymbol{z}$ is a Gaussian random variable of variance σ^2/m. We have

$$\mathbb{P}\left[|\boldsymbol{a}_i^*\boldsymbol{z}| \geq t\right] \leq 2\exp\left(-\frac{mt^2}{2\sigma^2}\right). \tag{3.5.19}$$

By union bound on the n columns, we have

$$\mathbb{P}\left[\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty \geq t\right] \leq 2\exp\left(-\frac{mt^2}{2\sigma^2} + \log n\right). \tag{3.5.20}$$

As we may see, as long as we choose t^2 to be in the order of $C\frac{\sigma^2\log n}{m}$ for a large enough constant C, the exponent will be negative and the event $\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty \geq t$

[16] Notice that a similar restriction on $\boldsymbol{h}$ was derived in (3.5.12). There the constant is $\alpha = 1$ and as we will soon see, here the constant needs to be 3.

will be of low probability. In particular we may choose $t^2 = 4\frac{\sigma^2 \log n}{m}$, and we know that with high probability at least $1 - cn^{-1}$, we have

$$\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty \leq 2\sigma\sqrt{\frac{\log n}{m}}.$$

So to make the two terms in (3.5.18) comparable in scale, it is natural to choose λ_m of the scale $\sigma\sqrt{\frac{\log n}{m}}$. In particular, we choose $\lambda_m \geq c \cdot 2\sigma\sqrt{\frac{\log n}{m}}$ for some $c > 0$. Then from the last inequality of (3.5.18), we have

$$\begin{aligned}
0 &\geq -\|\boldsymbol{A}^*\boldsymbol{z}\|_\infty\|\boldsymbol{h}\|_1 + \lambda_m(\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1) \\
&\geq -\frac{\lambda_m}{c}\|\boldsymbol{h}\|_1 + \lambda_m(\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1) \\
&\geq -\frac{\lambda_m}{c}\|\boldsymbol{h}_{\mathsf{I}}\|_1 - \frac{\lambda_m}{c}\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 + \lambda_m\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 - \lambda_m\|\boldsymbol{h}_{\mathsf{I}}\|_1 \\
&= \lambda_m\left(\left(1 - \frac{1}{c}\right)\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 - \left(1 + \frac{1}{c}\right)\|\boldsymbol{h}_{\mathsf{I}}\|_1\right),
\end{aligned} \tag{3.5.21}$$

where in the second to last inequality we used the fact that $\boldsymbol{x}_o$ is zero on I^c and $\|\hat{\boldsymbol{x}}_{\mathsf{I}}\|_1 - \|\boldsymbol{x}_{o\mathsf{I}}\|_1 \geq -\|\boldsymbol{h}_{\mathsf{I}}\|_1$. Therefore we have

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \frac{c+1}{c-1} \cdot \|\boldsymbol{h}_{\mathsf{I}}\|_1. \tag{3.5.22}$$

Notice that if we choose c to be large, $\frac{c+1}{c-1}$ can be arbitrarily close to 1. □

As we have discussed in the deterministic case, since $\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2 \leq \|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2 + \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o\|_2$, it suggests that $\|\boldsymbol{A}\boldsymbol{h}\|_2$ is typically very small and of the scale $C\sigma$. If the norm $\|\boldsymbol{A}\boldsymbol{h}\|_2$ upper bounds the norm $\|\boldsymbol{h}\|_2$, then the estimate is stable. Of course, this cannot be true for any $\boldsymbol{h} \in \mathbb{R}^n$ since the matrix $\boldsymbol{A}$ is typically severely under-determined and for any $\boldsymbol{h}$ in the null space of $\boldsymbol{A}$, the norm $\|\boldsymbol{A}\boldsymbol{h}\|$ is zero but the norm $\|\boldsymbol{h}\|$ can be arbitrarily large.

Nevertheless, due to the above lemma, we could hope that for $\boldsymbol{h}$ that satisfies the cone restriction $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \leq \alpha\|\boldsymbol{h}_{\mathsf{I}}\|_1$ for $\alpha = \frac{c+1}{c-1}$, $\|\boldsymbol{A}\boldsymbol{h}\|_2$ controls $\|\boldsymbol{h}\|_2$. Due to Theorem 3.11, we know that with high probability, $\boldsymbol{A}$ as a random Gaussian matrix satisfies RIP. Then Theorem 3.17 ensures that when $\boldsymbol{h}$ is restricted in such a cone, $\|\boldsymbol{A}\boldsymbol{h}\|_2$ controls the norm $\|\boldsymbol{h}\|_2$. This leads to the following theorem.[17]

THEOREM 3.31 (Stable Sparse Recovery via Lasso). *Suppose that* $\boldsymbol{A} \sim_{iid} \mathcal{N}(0, \frac{1}{m})$, *and* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, *with* $\boldsymbol{x}_o$ *k-sparse and* $\boldsymbol{z} \sim_{iid} \mathcal{N}(0, \frac{\sigma^2}{m})$. *Solve the Lasso*

$$\min \ \ \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_m\|\boldsymbol{x}\|_1, \tag{3.5.23}$$

with regularization parameter $\lambda_m = c \cdot 2\sigma\sqrt{\frac{\log n}{m}}$ *for a large enough* c. *Then with high probability,*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 \ \leq \ C'\sigma\sqrt{\frac{k\log n}{m}}. \tag{3.5.24}$$

[17] This result and its proof essentially follows that of [CT07] and [BRT09].

Generally, we are interested in the regime $m \geq k \log n$, because this is when the measurement matrix $\boldsymbol{A}$ satisfies RIP (due to Theorem 3.11). The above theorem indicates that in this case, we actually do much better under random noise than the deterministic noise: the estimation error in the random case can be the noise norm σ scaled by a diminishing factor[18] whereas in the deterministic case the error is the noise norm ε scaled by a constant factor (see Theorem 3.29 for comparison).

Proof With $L(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, we have

$$L(\hat{\boldsymbol{x}}) = L(\boldsymbol{x}_o) + \langle \nabla L(\boldsymbol{x}_o), \hat{\boldsymbol{x}} - \boldsymbol{x}_o \rangle + \frac{1}{2}\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2^2.$$

We now use this equality to better estimate the difference between the values of the objective function at $\hat{\boldsymbol{x}}$ and at $\boldsymbol{x}_o$ than that done in (3.5.18):

$$\begin{aligned} 0 &\geq L(\hat{\boldsymbol{x}}) + \lambda_m \|\hat{\boldsymbol{x}}\|_1 - L(\boldsymbol{x}_o) - \lambda_m \|\boldsymbol{x}_o\|_1 \\ &\geq \frac{1}{2}\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2^2 + \langle \nabla L(\boldsymbol{x}_o), \hat{\boldsymbol{x}} - \boldsymbol{x}_o \rangle + \lambda_m (\|\hat{\boldsymbol{x}}\|_1 - \|\boldsymbol{x}_o\|_1) \\ &\geq \frac{1}{2}\|\boldsymbol{A}\boldsymbol{h}\|_2^2 + \lambda_m \left(\left(1 - \frac{1}{c}\right)\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 - \left(1 + \frac{1}{c}\right)\|\boldsymbol{h}_{\mathsf{I}}\|_1 \right), \end{aligned} \tag{3.5.25}$$

where the last inequality follows exactly the same derivation that we have done in (3.5.18) and (3.5.21) for other terms without the term $\frac{1}{2}\|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2^2 = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{h}\|_2^2$.

From the last inequality we have

$$\frac{1}{2}\|\boldsymbol{A}\boldsymbol{h}\|_2^2 \leq \lambda_m \left(1 + \frac{1}{c}\right)\|\boldsymbol{h}_{\mathsf{I}}\|_1.$$

According to Theorem 3.11 and Theorem 3.17, with high probability, the random Gaussian matrix $\boldsymbol{A}$ satisfies the restricted strong convexity property, we have $\|\boldsymbol{A}\boldsymbol{h}\|_2^2 \geq \mu \|\boldsymbol{h}\|_2^2$ for some constant μ.[19] Also from the relationship between 1-norm and 2-norm, we have $\|\boldsymbol{h}_{\mathsf{I}}\|_1 \leq \sqrt{k}\|\boldsymbol{h}_{\mathsf{I}}\|_2 \leq \sqrt{k}\|\boldsymbol{h}\|_2$. Finally, with the choice $\lambda_m = c \cdot 2\sigma\sqrt{\frac{\log n}{m}}$, the above inequality leads to:

$$\frac{\mu}{2}\|\boldsymbol{h}\|_2^2 \leq 2(c+1)\sigma\sqrt{\frac{k \log n}{m}}\|\boldsymbol{h}\|_2 \quad \Rightarrow \quad \|\boldsymbol{h}\|_2 \leq C'\sigma\sqrt{\frac{k \log n}{m}}$$

for some constant $C' = \frac{4(c+1)}{\mu} \in \mathbb{R}_+$. □

The error bound given in the above theorem is actually nearly optimal as it is close to the best error that one can achieve by considering all possible estimators:

THEOREM 3.32 ([CD13]). *Suppose that we will observe $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{z}$. Set*

$$M^\star(\boldsymbol{A}) = \inf_{\hat{\boldsymbol{x}}} \sup_{\|\boldsymbol{x}\|_0 \leq k} \mathbb{E}\,\|\hat{\boldsymbol{x}}(\boldsymbol{y}) - \boldsymbol{x}\|_2^2 . \tag{3.5.26}$$

[18] As $\sqrt{\frac{k \log n}{m}}$ can be chosen to be arbitrarily small

[19] Notice that μ depends on the RIP constant $\delta_{2k}(\boldsymbol{A})$ and the constant $C = \frac{c+1}{c-1}$ of the cone restriction.

Then for any $\boldsymbol{A}$ *with* $\|\boldsymbol{e}_i^* \boldsymbol{A}\|_2 \le \sqrt{n}$ *for each* i*, we have*

$$M^\star(\boldsymbol{A}) \;\ge\; C\sigma^2 \frac{k\log(n/k)}{m}. \tag{3.5.27}$$

Proof of this theorem is beyond the scope of this book; we refer interested readers to the original paper for a proof. According to Theorem 3.31, the error bound $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2^2 \sim O(\sigma^2 \frac{k\log n}{m})$ achieved by Lasso is within a difference of $O(\sigma^2 \frac{k\log k}{m})$ from the best achievable bound above. When $m \gg k$, such a difference is negligible.

3.5.2 Recovery of Inexact Sparse Signals

In all the above analysis, we have assumed that in the observation model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, the signal $\boldsymbol{x}_o$ is perfectly k-sparse. In many cases, $\boldsymbol{x}_o$ might not be so sparse and even all entries could be nonzero. Then a question naturally arises: for $\boldsymbol{x}_o$ that is close to a k-sparse signal, can we still expect good recovery performance in some sense?

Let $[\boldsymbol{x}_o]_k$ be the best k-sparse signal that approximates $\boldsymbol{x}_o$. Then we can rewrite the observation model as:

$$\boldsymbol{y} = \boldsymbol{A}[\boldsymbol{x}_o]_k + \boldsymbol{A}(\boldsymbol{x}_o - [\boldsymbol{x}_o]_k) + \boldsymbol{z}.$$

Strictly speaking the term $\boldsymbol{w} = \boldsymbol{A}(\boldsymbol{x}_o - [\boldsymbol{x}_o]_k)$ is not noise. It is more of a deviation from our idealistic sparse signal assumption. But we may view it as introducing a deterministic error to the observation. Hence, if the norm of $\boldsymbol{w}$ is small, we should expect to obtain an estimate $\hat{\boldsymbol{x}}$ whose error from $\boldsymbol{x}_o$ is proportional to this norm.

The following is a typical result on estimation with inexact sparsity, which also allows deterministic noise.[20]

THEOREM 3.33 ([CRT06b]). *Let* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$*, with* $\|\boldsymbol{z}\|_2 \le \varepsilon$*. Let* $\hat{\boldsymbol{x}}$ *solve the basis pursuit denoising problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \le \varepsilon. \end{array} \tag{3.5.28}$$

Then for any k *such that* $\delta_{2k}(\boldsymbol{A}) < \sqrt{2} - 1$*,*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2 \;\le\; C\frac{\|\boldsymbol{x}_o - [\boldsymbol{x}_o]_k\|_1}{\sqrt{k}} + C'\varepsilon \tag{3.5.29}$$

for some constants C *and* C' *which only depend on* $\delta_{2k}(\boldsymbol{A})$*.*

How should we interpret this result? One way of reading it is to say that if we are working in a regime where noise-free sparse recovery would have succeeded ($\delta_{2k}(\boldsymbol{A}) < \sqrt{2} - 1$), then even if our modeling assumptions are violated (due to

[20] In fact, similar statements hold for random noise. The proof requires slight modification to that of Theorem 3.31. We leave the details to the reader as an exercise.

the introduction of noise and inexact sparsity), we can still *stably* estimate $\boldsymbol{x}_o$. Moreover, the error in our estimate is proportional to the degree to which our assumptions are violated and proportional to the noise level. When the original signal $\boldsymbol{x}_o$ is indeed k-sparse, we have $\boldsymbol{x}_o - [\boldsymbol{x}_o]_k = \boldsymbol{0}$ and the above result reduces to the deterministic noise case, i.e. Theorem 3.29.

Proof As usual, we denote $\boldsymbol{h} = \hat{\boldsymbol{x}} - \boldsymbol{x}_o$. We also denote the support of $[\boldsymbol{x}_o]_k$ as I so that we have $[\boldsymbol{x}_o]_k = \boldsymbol{x}_{o\mathsf{I}}$. Because $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_o\|_2 = \|\boldsymbol{z}\|_2 \le \varepsilon$. Since $\hat{\boldsymbol{x}}$ is feasible, we have $\|\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\|_2 \le \varepsilon$ as well. Using the triangle inequality,

$$\|\boldsymbol{A}\boldsymbol{h}\|_2 = \|\boldsymbol{A}(\hat{\boldsymbol{x}} - \boldsymbol{x}_o)\|_2 \le 2\varepsilon.$$

Therefore, in the inexact sparse case, the prediction error $\|\boldsymbol{A}\boldsymbol{h}\|_2$ is again bounded by the noise level.

Since $\hat{\boldsymbol{x}}$ minimizes the objective function, we have

$$\begin{aligned} 0 &\le \|\boldsymbol{x}_o\|_1 - \|\hat{\boldsymbol{x}}\|_1 \\ &= \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{x}_{o\mathsf{I}} + \boldsymbol{h}_{\mathsf{I}}\|_1 - \|\boldsymbol{x}_{o\mathsf{I}^c} + \boldsymbol{h}_{\mathsf{I}^c}\|_1 \\ &\le \|\boldsymbol{x}_o\|_1 - \|\boldsymbol{x}_{o\mathsf{I}}\|_1 + \|\boldsymbol{h}_{\mathsf{I}}\|_1 + \|\boldsymbol{x}_{o\mathsf{I}^c}\|_1 - \|\boldsymbol{h}_{\mathsf{I}^c}\|_1 . \end{aligned}$$

Thus we have

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \le \|\boldsymbol{h}_{\mathsf{I}}\|_1 + 2\,\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1 , \tag{3.5.30}$$

where $\boldsymbol{x}_{o\mathsf{I}^c} = \boldsymbol{x}_o - \boldsymbol{x}_{o\mathsf{I}}$. So in the inexact sparse case, the feasible perturbation $\boldsymbol{h}$ no longer satisfies the cone condition as that in the exact sparse case (see Theorem 3.29). Therefore, to establish the result of this theorem, we need to modify the proof of Theorem 3.17 to accommodate the extra term $2\,\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1$ in estimating the bounds for $\|\boldsymbol{A}\boldsymbol{h}\|_2$ and $\|\boldsymbol{h}\|_2$.

The proof essentially follows the same steps as in the proof for Theorem 3.17. The only difference is that in places where we used to apply the cone condition $\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \le \alpha\,\|\boldsymbol{h}_{\mathsf{I}}\|_1$, we now need to replace it with the new condition (3.5.30). Therefore, instead of (3.3.34), the new condition (3.5.30) implies

$$\|\boldsymbol{h}_{\mathsf{I}^c}\|_1 \le \sqrt{k}\,\|\boldsymbol{h}_{\mathsf{I}}\|_2 + 2\,\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1 \le \sqrt{k}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + 2\,\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1 . \tag{3.5.31}$$

Substituting this into (3.3.33) to establish a bound for $\|\boldsymbol{A}\boldsymbol{h}\|_2$, we obtain

$$(1-\delta_{2k})\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 \le \sqrt{2}\delta_{2k}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + 2\sqrt{2}\delta_{2k}\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}} + (1+\delta_{2k})^{1/2}\,\|\boldsymbol{A}\boldsymbol{h}\|_2 . \tag{3.5.32}$$

This gives

$$\|\boldsymbol{A}\boldsymbol{h}\|_2 \ge \frac{1-(1+\sqrt{2})\delta_{2k}}{(1+\delta_{2k})^{1/2}}\,\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 - \frac{2\sqrt{2}\delta_{2k}}{(1+\delta_{2k})^{1/2}}\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}}. \tag{3.5.33}$$

Now, to establish a bound for $\|\boldsymbol{h}\|_2$, in (3.3.40) where we have applied the cone condition in the second inequality, we also need to replace the cone condition

with the new condition (3.5.30) and that gives:

$$\|\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}\|_2 \le \frac{\|\boldsymbol{h}_{\mathsf{I}^c}\|_1}{\sqrt{k}} \le \frac{\|\boldsymbol{h}_{\mathsf{I}}\|_1 + 2\,\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}} \tag{3.5.34}$$

$$\le \|\boldsymbol{h}_{\mathsf{I}}\|_2 + 2\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}} \tag{3.5.35}$$

$$\le \|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + 2\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}}. \tag{3.5.36}$$

This gives

$$\|\boldsymbol{h}\|_2 \le \|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + \|\boldsymbol{h}_{(\mathsf{I}\cup\mathsf{J}_1)^c}\|_2 \le 2\|\boldsymbol{h}_{\mathsf{I}\cup\mathsf{J}_1}\|_2 + 2\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}}. \tag{3.5.37}$$

Combining this with (3.5.33) and the fact that $\|\boldsymbol{A}\boldsymbol{h}\|_2 \le 2\varepsilon$, we get

$$\|\boldsymbol{h}\|_2 \le \left(\frac{2+2(\sqrt{2}-1)\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}}\right)\frac{\|\boldsymbol{x}_{o\mathsf{I}^c}\|_1}{\sqrt{k}} + \left(\frac{4(1+\delta_{2k})^{1/2}}{1-(1+\sqrt{2})\delta_{2k}}\right)\varepsilon, \tag{3.5.38}$$

where we note $\boldsymbol{x}_{o\mathsf{I}^c} = \boldsymbol{x}_o - [\boldsymbol{x}_o]_k$. Therefore, as long as $1-(1+\sqrt{2})\delta_{2k} > 0$ or equivalently $\delta_{2k} < \sqrt{2}-1$, the conclusion of the theorem holds. □

Note that from the above proof, we know that the two constants in the Theorem can be chosen to be:

$$C = \frac{2-2(\sqrt{2}-1)\delta_{2k}}{1-(1+\sqrt{2})\delta_{2k}} \quad \text{and} \quad C' = \frac{4(1+\delta_{2k})}{1-(1+\sqrt{2})\delta_{2k}}. \tag{3.5.39}$$

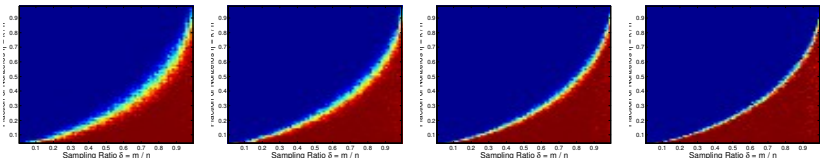
If δ_{2k} is very small, say approaching to zero, then C approaches to 2 and C' to 4. Those constants give the smallest possible bound for the error $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2$ based on this proof.

3.6 Phase Transitions in Sparse Recovery

Above, we showed that sparse vectors $\boldsymbol{x}_o$ can be accurately estimated from linear observations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$. One of the surprises was that in the noise-free case ($\boldsymbol{z} = \boldsymbol{0}$), k-sparse vectors could be exactly recovered from just slightly more than k measurements – to be precise, $m \ge Ck\log(n/k)$ measurements, where C is a constant. The key technical tool for doing this was the restricted isometry property (RIP). The RIP and related properties enable simple proofs, with correct orders of growth (i.e., $m \sim k\log(n/k)$), but are not intended to give precise estimates of the constant C.

For some applications, it can be important to know C. In sampling and reconstruction, this tells us precisely how many samples we need to acquire to accurately estimate a sparse signal; in error correction, this tells us precisely how many errors the system can tolerate.

Put another way, we would like to obtain precise relationships between the dimensionality n, the number of measurements m, and the number of nonzero



$n = 50$ $n = 100$ $n = 200$ $n = 400$

Figure 3.15 Phase Transition in Sparse Recovery with Gaussian Matrices. Each display plots the fraction of correct recoveries using ℓ^1 minimization, over a suite of randomly generated problems. The vertical axis represents the fraction of nonzero entries $\eta = k/n$ in the target vector $\boldsymbol{x}_o$ – the bottom corresponds to very sparse vectors, while the top corresponds to fully dense vectors. The horizontal axis represents the sampling ratio $\delta = m/n$ – the left corresponds to drastically under sampled problems ($m \ll n$), while the right corresponds to almost fully observed problems. For each (η, δ) pair, we generate 200 random problems, which we solve using CVX. We declare success if the recovered vector is accurate up to a relative error $\leq 10^{-6}$. Several salient features emerge: first, there is an easy regime (lower right corner) in which ℓ^1 minimization always succeeds. Second, there is a hard regime (upper left corner) in which ℓ^1 minimization always fails. Finally, as n increases, this transition between success and failure becomes increasingly sharp.

entries k that we can recover. We would like these relationships to be as sharp and explicit as possible. To get some intuition for what to expect, we again resort to numerical simulation. We fix n, and consider different levels of sparsity k, and numbers of measurements m. For each pair (k, m), we generate a number of random ℓ^1 minimization problems, with noiseless Gaussian measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$, and ask *"For what fraction of these problems does ℓ^1 minimization correctly recover $\boldsymbol{x}_o$?"*

Figure 3.15 displays the result as a two dimensional image. Here, the horizontal axis is the sampling ratio $\delta = m/n$. This ranges from zero on the left (a very short, wide $\boldsymbol{A}$) to one on the right (a nearly square $\boldsymbol{A}$). The vertical axis is the fraction of nonzeros $\eta = k/n$. Again, this ranges from zero at the bottom (very sparse problems) to one at the top (denser problems). For each pair (η, δ), we generate 200 random problems. The intensity is the fraction of problems for which ℓ^1 minimization succeeds. The four graphs, from left to right, show the result for $n = 50, 100, 200, 400$.

This figure conveys several important pieces of information. First, as expected, when m is large and k is small (the lower right corner of each graph), ℓ^1 minimization always succeeds. Conversely, when m is small and k is large (the upper left corner of each graph), ℓ^1 minimization always fails. Moreover, as n grows, the transition between success and failure becomes increasingly abrupt. Put another way, for high-dimensional problems, the behavior of ℓ^1 minimization is surprisingly predictable: it either almost always succeeds, or almost always fails. The line demarcating the sharp boundary between success and failure is known as a *phase transition*.

3.6.1 Phase Transitions: Main Conclusions

In this section, we state a result that precisely specifies the location of the phase transition. Namely, we will show that a sharp transition from failure to success occurs when the sampling ratio $\delta = m/n$ exceeds a certain function $\psi(\eta)$ of the sparsity ratio $\eta = k/n$. This result will be sharper than the ones we stated above using incoherence and RIP, in the sense that it identifies the precise number of measurements $m^\star = \psi(k/n)n$ required for success. To obtain such sharp results, we need to make two changes to our setting. First, we will make stronger assumptions on the matrix $\boldsymbol{A}$. Second, we will weaken the goal of our performance guarantee.

Random vs. Deterministic $\boldsymbol{A}$.
Thus far, we have focused on deterministic properties of the matrix $\boldsymbol{A}$, such as (in)-coherence and the RIP. These properties do not depend on any random model for the matrix $\boldsymbol{A}$, although they are easiest to verify for random $\boldsymbol{A}$. Obtaining sharp estimates on the location of the phase transition requires more sophisticated probabilistic tools, which intrinsically require $\boldsymbol{A}$ to be a random matrix. We will sketch this theory under the assumption that $A_{ij} \sim_{iid} \mathcal{N}(0, \frac{1}{m})$, i.e., $\boldsymbol{A}$ is a standard Gaussian random matrix. We will also briefly describe experiments and theoretical results which show that the results we will obtain for Gaussian $\boldsymbol{A}$ are "universal", in the sense that they precisely describe the behavior of ℓ^1 minimization for a fairly broad family of matrices $\boldsymbol{A}$. Nevertheless, all currently known theory which is sharp enough to precisely characterize the phase transition requires $\boldsymbol{A}$ to be a random matrix.

Recovering a Particular Sparse $\boldsymbol{x}_o$ vs. Recovering All Sparse $\boldsymbol{x}_o$.
Incoherence and RIP allow one to prove "for all" results, which say that for a given matrix $\boldsymbol{A}$, ℓ^1 minimization recovers *every* sparse $\boldsymbol{x}_o$ from $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$. The strongest and most general known results for phase transitions pertain to a slightly weaker statement: for a given, *fixed* $\boldsymbol{x}_o$, with high probability in the random matrix $\boldsymbol{A}$, ℓ^1 minimization recovers that particular $\boldsymbol{x}_o$ from the measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$.

A variety of mathematical tools have been brought to bear on the analysis of phase transitions in ℓ^1 minimization.[21] Historically, the phenomenon has been characterized using several different approaches, by different sets of authors. In the following two sections, we describe briefly two representative approaches, which correspond roughly to the two geometric pictures in Section 3.1, which describe the behavior of ℓ^1 minimization in terms of the space $\mathbb{R}^n$ of coefficient vectors $\boldsymbol{x}$ and the space $\mathbb{R}^m$ of observation vectors $\boldsymbol{y}$. We leave a more general and rigorous theory of the phase transition for a broad family of low-dimensional models to in Chapter 6.

[21] as well as phase transition phenomena for recovering broader family of low-dimensional structures, as we will see in Chapter 6.

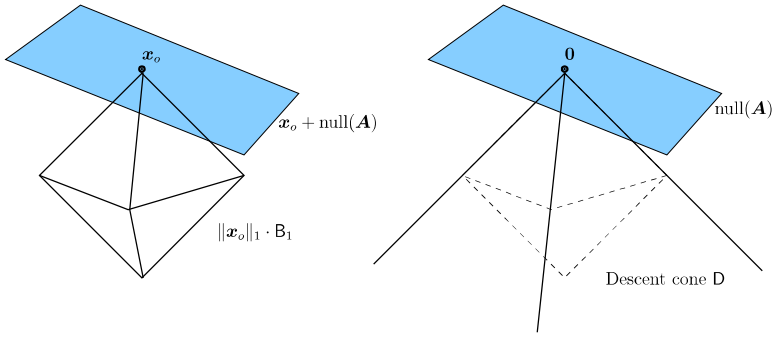


Figure 3.16 Cones and the Coefficient Space Geometry. ℓ^1 minimization uniquely recovers $\boldsymbol{x}_o$ if and only if the intersection of the descent cone D with null($\boldsymbol{A}$) is $\{\mathbf{0}\}$.

3.6.2 Phase Transitions via Coefficient-Space Geometry

Suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$. Recall the geometric picture in Figure 3.16 (left), which we introduced in Section 3.1. There, we argued that $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 minimization problem if and only if the affine subspace

$$\boldsymbol{x}_o + \text{null}(\boldsymbol{A}) \tag{3.6.1}$$

of feasible solutions $\boldsymbol{x}$ intersects the scaled ℓ^1 ball

$$\|\boldsymbol{x}_o\|_1 \cdot \mathsf{B}_1 = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_1 \leq \|\boldsymbol{x}_o\|_1\} \tag{3.6.2}$$

only at $\boldsymbol{x}_o$.

We can express the same geometry more cleanly in terms of the *descent cone*:

$$\mathsf{D} = \{\boldsymbol{v} \mid \|\boldsymbol{x}_o + t\boldsymbol{v}\|_1 \leq \|\boldsymbol{x}_o\|_1 \text{ for some } t > 0\}\,. \tag{3.6.3}$$

This is the set of directions $\boldsymbol{v}$ for which a small (but nonzero) perturbation of $\boldsymbol{x}_o$ in the $\boldsymbol{v}$ direction does not increase the objective function $\|\cdot\|_1$. The descent cone D is visualized in Figure 3.16 (right).

Notice that the perturbation $\boldsymbol{x}_o + t\boldsymbol{v}$ is feasible for $t \neq 0$ if and only if $\boldsymbol{v} \in \text{null}(\boldsymbol{A})$. The feasible perturbations which do not increase the objective function reside in the intersection $\mathsf{D} \cap \text{null}(\boldsymbol{A})$. Because D is a convex cone and null($\boldsymbol{A}$) is a subspace, D and null($\boldsymbol{A}$) always intersect at $\mathbf{0}$. It is not difficult to see that $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 problem if and only if $\mathbf{0}$ is the only point of intersection between null($\boldsymbol{A}$) and D:

LEMMA 3.34. *Suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o$. Then $\boldsymbol{x}_o$ is the unique optimal solution to the ℓ^1 minimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{x}\|_1 \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \end{array} \tag{3.6.4}$$

if and only if $\mathsf{D} \cap \mathrm{null}(\boldsymbol{A}) = \{\mathbf{0}\}$.

Proof First, suppose that $\mathsf{D} \cap \mathrm{null}(\boldsymbol{A}) = \{\mathbf{0}\}$. Consider any alternative solution $\boldsymbol{x}'$. Then $\boldsymbol{x}' - \boldsymbol{x}_o \in \mathrm{null}(\boldsymbol{A}) \setminus \{\mathbf{0}\}$. Since $\mathsf{D} \cap \mathrm{null}(\boldsymbol{A}) = \{\mathbf{0}\}$, $\boldsymbol{x}' - \boldsymbol{x}_o \notin \mathsf{D}$, and so $\|\boldsymbol{x}'\|_1 > \|\boldsymbol{x}_o\|_1$, and $\boldsymbol{x}'$ is not an optimal solution. Since this holds for any feasible $\boldsymbol{x}'$, $\boldsymbol{x}_o$ is the unique optimal solution.

Conversely, suppose $\boldsymbol{x}_o$ is not the unique optimal solution. Then there exists $\boldsymbol{x}' \neq \boldsymbol{x}_o$ with $\|\boldsymbol{x}'\|_1 \leq \|\boldsymbol{x}_o\|_1$. Thus $\boldsymbol{x}' - \boldsymbol{x}_o \in \mathsf{D}$. By feasibility, $\boldsymbol{x}' - \boldsymbol{x}_o \in \mathrm{null}(\boldsymbol{A})$, and so $\mathsf{D} \cap \mathrm{null}(\boldsymbol{A}) \neq \{\mathbf{0}\}$. □

Hence, to study whether ℓ^1 minimization succeeds, we may equivalently study whether the subspace $\mathrm{null}(\boldsymbol{A})$ has nontrivial intersection with the cone D. Because $\boldsymbol{A}$ is a random matrix, $\mathrm{null}(\boldsymbol{A})$ is a random subspace, of dimension $n - m$. If $\boldsymbol{A}$ is Gaussian, then $\mathrm{null}(\boldsymbol{A})$ follows the uniform distribution on the set of subspaces $\mathsf{S} \subset \mathbb{R}^n$ of dimension $n - m$.[22] Clearly, the probability that the random subspace $\mathrm{null}(\boldsymbol{A})$ intersects the descent cone D depends on properties of D. Intuitively, we would expect intersections to be more likely if D is "big" in some sense.

In Chapter 6, we will generalize the notion of "dimension" to all closed convex cones and show that this dimension precisely characterizes the probability of a convex cone intersecting with a subspace (or another convex cone). The same techniques actually apply to a broad family of norms that promote sparsity or low-dimensionality. In particular, we will show that the probability of correct recovery for ℓ^1 minimization undergoes a sharp transition at

$$m^\star = \psi\left(\frac{k}{n}\right) n. \tag{3.6.5}$$

Here, $\psi : [0,1] \to [0,1]$ a function which takes as input the fraction $\eta = k/n$ of nonzeros, and describes the ratio $m^\star/n$ of number of measurements to the ambient dimension. The precise location ψ of the transition is given by the expression:

$$\psi(\eta) = \min_{t \geq 0} \left\{ \eta(1+t^2) + (1-\eta)\sqrt{\frac{2}{\pi}} \int_t^\infty (s-t)^2 \exp\left(-\frac{s^2}{2}\right) ds \right\}. \tag{3.6.6}$$

The function ψ is somewhat complicated; in Chapter 6, we will demonstrate how it arises naturally from the geometry of ℓ^1 minimization. While there is no closed form solution for the minimization over t in this formula, it can be calculated numerically. Figure 3.17 displays this curve (red) superimposed over the empirical fraction of successes (grayscale) in our experiment. Clearly, there is a very good agreement between this theoretical prediction and our previous

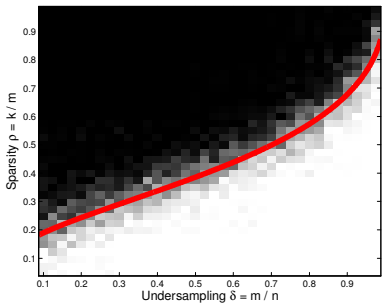


Figure 3.17 Phase Transitions: Agreement between Theory and Experiment. Theoretical phase transition predicted by (3.6.5) and (3.6.6), overlaid on fraction of successes in 200 experiments, for varying sparsities $\rho = k/m$ and aspect ratios $\delta = n/m$.

experiment: the empirical fraction of successes transitions rapidly from 0 to 1 as m/n exceeds $\psi(k/n)$.[23]

In fact, one can do slightly more: in addition to showing that $\psi(t)$ determines a point of transition between likely success and likely failure, we can give lower bounds on the probability of success (below the phase transition) and failure (above the phase transition) which quantify how sharp the transition is, for finite n. The following theorem makes all of this precise:

THEOREM 3.35. *Let $\boldsymbol{x}_o \in \mathbb{R}^n$ be k-sparse, and suppose that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o \in \mathbb{R}^{m \times n}$, with $\boldsymbol{A} \sim_{\text{iid}} \mathcal{N}(0, \frac{1}{m})$. Let $m^\star = \psi(k/n)n$, with ψ as in (3.6.6). Then*

$$\mathbb{P}\left[\ell^1 \text{ recovers } \boldsymbol{x}_o\right] \geq 1 - C\exp\left(-c\frac{(m-m^\star)^2}{n}\right), \qquad m > m^\star,$$
$$\mathbb{P}\left[\ell^1 \text{ does not recover } \boldsymbol{x}_o\right] \geq 1 - c'\exp\left(-C'\frac{(m^\star-m)^2}{n}\right), \qquad m < m^\star,$$

where C, c, c', C' are positive numerical constants.

Again, we leave the proof to Chapter 6 where we study phase transition in a more general setting. This result implies that a sharp transition indeed occurs at $m^\star$ measurements: when $m/n > m^\star/n + C''/\sqrt{n}$, the probability of failure is bounded by a small constant (which can be made arbitrarily small by choosing C'' large). Conversely, when $m/n < m^\star/n - C''/\sqrt{n}$, the probability of success is bounded by a small constant. Hence, the transition region observed in Figure 3.15 has width $O(1/\sqrt{n})$ – in particular, it vanishes as $n \to \infty$.

[22] To be more precise, $\text{null}(\boldsymbol{A})$ is distributed according to the Haar (uniform) measure on the Grassmannian manifold $\mathsf{G}_{n,n-m}$, the set of $(n-m)$-dimensional subspaces in $\mathbb{R}^n$.

[23] Figure 3.17 displays the same phase transition as in Figure 3.15 in a different parameterization, in which the vertical axis is $\rho = k/m$ and the horizontal axis is $\delta = m/n$.

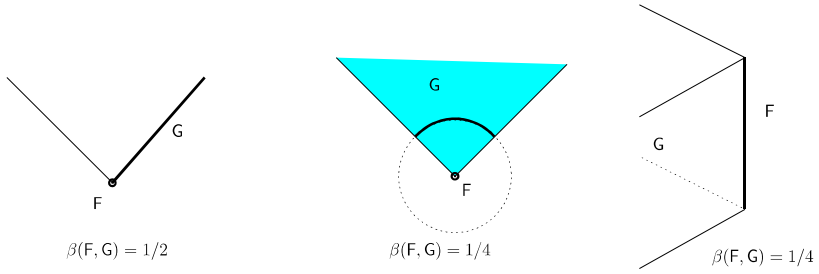


Figure 3.18 Internal Angles of Convex Polytopes. The internal angle $\beta(\mathsf{F},\mathsf{G})$ of a face $\mathsf{F} \subseteq \mathsf{G}$ with respect to another face G containing it is the fraction of the linear span of $\mathsf{G} - \boldsymbol{x}$ occupied by $\mathsf{G} - \boldsymbol{x}$, where $\boldsymbol{x}$ is any point in the relative interior of F.

3.6.3 Phase Transitions via Observation-Space Geometry

Historically, the first sharp estimates of the location of the phase transition were derived using the "observation space" geometric picture of ℓ^1 minimization, which we reproduce in Figure 3.4. In this picture, ℓ^1 minimization is visualized through the relationship between two convex polytopes, the unit ℓ^1 ball

$$\mathsf{B}_1 \doteq \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_1 \leq 1\} \tag{3.6.7}$$

and its projection into $\mathbb{R}^m$,

$$\mathsf{P} \doteq \boldsymbol{A}(\mathsf{B}_1) = \{\boldsymbol{A}\boldsymbol{x} \mid \|\boldsymbol{x}\|_1 \leq 1\}. \tag{3.6.8}$$

Namely, ℓ^1 minimization uniquely recovers any $\boldsymbol{x}$ with support I and signs $\boldsymbol{\sigma}$ if and only if

$$\mathsf{F} \doteq \mathrm{conv}(\{\sigma_i \boldsymbol{a}_i \mid i \in \mathsf{I}\}) \tag{3.6.9}$$

forms a face of the polytope P. Conversely, if F intersects the interior of P, then ℓ^1 minimization does not recover $\boldsymbol{x}_o$ with support I and signs $\boldsymbol{\sigma}$.

The first results bounding the phase transition derived from remarkable results in stochastic geometry, which give exact formulas for the expected number of k-dimensional faces of a randomly projected polytope $\mathsf{P} = \boldsymbol{A}(\mathsf{Q})$. This expectation depends two notions of the "size" of the polytope Q: the *internal angle* and *external angle*.

Definition 3.36 (Internal Angle). *The internal angle* $\beta(\mathsf{F},\mathsf{G})$ *of a face* F *of a polytope* G *is the fraction of* $\mathrm{span}(\mathsf{G}-\boldsymbol{x})$ *occupied by* $\mathsf{G}-\boldsymbol{x}$*, where* $\mathrm{span}(\cdot)$ *denotes the linear span, and* $\boldsymbol{x}$ *is any point in* $\mathrm{relint}(\mathsf{F})$.

The internal angle is visualized for several examples in Figure 3.18. Informally speaking, the internal angle measures the fraction of the space cut out by G, when viewed from F. There is a complementary notion of angle, called the external angle, which captures the fraction of the space cut out by the *normal cone* to G at a point in the relative interior of F:

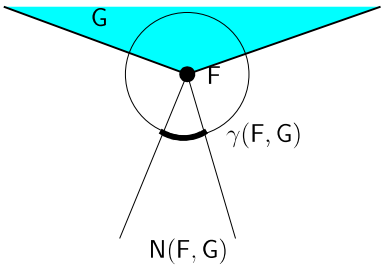


Figure 3.19 External Angles of Convex Polytopes. The external angle $\gamma(\mathsf{F},\mathsf{G})$ of a face $\mathsf{F} \subseteq \mathsf{G}$ with respect to another face G containing it is the fraction of the linear span of $\mathsf{G} - \boldsymbol{x}$ occupied by the normal cone $\mathsf{N}(\mathsf{F},\mathsf{G})$.

DEFINITION 3.37 (External Angle). *The external angle* $\gamma(\mathsf{F},\mathsf{G})$ *of a face* $\mathsf{F} \subseteq \mathsf{G}$ *is the fraction of* $\operatorname{span}(\mathsf{G} - \boldsymbol{x})$ *occupied by the normal cone*

$$\mathsf{N}(\mathsf{F},\mathsf{G}) = \{\boldsymbol{v} \in \operatorname{span}(\mathsf{G} - \boldsymbol{x}) \mid \langle v - \boldsymbol{x}, \boldsymbol{x}' - \boldsymbol{x}\rangle \leq 0 \; \forall \; \boldsymbol{x}' \in \mathsf{G}\}\,,$$

where $\boldsymbol{x}$ *is any point in* $\operatorname{relint}(\mathsf{F})$.

Figure 3.19 visualizes the external angle. There is an exquisite characterization of the expected number of k-dimensional faces of a random projection of a convex polytope P, in terms of its internal and external angles. Let $f_k(\mathsf{P})$ denote the number of k-dimensional faces of a polytope P, and let F_k denote the collection of such faces. Then for an $m \times n$ Gaussian matrix $\boldsymbol{A}$,

$$\mathbb{E}_{\boldsymbol{A}}[f_k(\boldsymbol{A}\mathsf{P})] = f_k(\mathsf{P}) - 2 \underbrace{\sum_{\ell=m+1,m+3,\ldots} \sum_{\mathsf{F}\in\mathsf{F}_k(\mathsf{P})} \sum_{\mathsf{G}\in\mathsf{F}_\ell(\mathsf{P})} \beta(\mathsf{F},\mathsf{G})\gamma(\mathsf{G},\mathsf{P})}_{\Delta=\text{Expected number of faces lost}}\,.$$

This formula arises out of a line of work in discrete geometry, which aims at understanding the behavior of "typical" point clouds, and studying the simplex method for linear programming for "typical" inputs. One remarkable aspect is that it gives the *exact* value of the expected face count. The connection to ℓ^1 minimization is that ℓ^1 successfully recovers every $k+1$-sparse vector $\boldsymbol{x}_o$ from measurements $\boldsymbol{A}\boldsymbol{x}_o$ if and only if $f_k(\boldsymbol{A}(\mathsf{P})) = f_k(\mathsf{P})$. This can be observed from the observation-space geometry described above. This event can be studied through the quantity Δ – the expected number of faces lost. Whenever $\Delta < 1$, there exists an $\boldsymbol{A}$ such that $f_k(\boldsymbol{A}(\mathsf{P})) = f_k(\mathsf{P})$; when Δ is substantially smaller than one, we can use the Markov inequality to argue that the probability that any face is lost in the projection is small.

3.6.4 Phase Transitions in Support Recovery

Thus far, we have focused on the problem of *estimating* a sparse vector $\boldsymbol{x}_o$. We showed that from noisy observations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, convex optimization produces

a vector $\hat{\boldsymbol{x}}$ such that $\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_2$ is small. For many engineering applications, where $\boldsymbol{x}_o$ is represents a signal to be sensed or an error to be corrected, this is exactly what we need. However, in some applications, the goal is not so much to estimate $\boldsymbol{x}_o$ as to determine which of the entries of $\boldsymbol{x}_o$ are nonzero. A good example, which we will revisit in later chapters, is in spectrum sensing for wireless communications. Here, the entries of $\boldsymbol{x}_o$ represent frequency bands which might be available for transmission, or which might be occupied. The goal is to know which frequency bands are available, so that we can avoid interfering with other users. In this setting, it is much more important to know which entries of $\boldsymbol{x}_o$ are nonzero than to estimate the particular values.

Support Recovery: Desiderata.
In this section, we consider the problem of estimating the signed support

$$\boldsymbol{\sigma}_o = \operatorname{sign}(\boldsymbol{x}_o), \tag{3.6.10}$$

from noisy observations

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}. \tag{3.6.11}$$

We will derive theory under the assumptions that the noise $\boldsymbol{z}$ is iid $\mathcal{N}(0, \frac{\sigma^2}{m})$. Let $\hat{\boldsymbol{x}}$ solve the Lasso problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \tfrac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 . \tag{3.6.12}$$

We can distinguish between two conclusions:

- **Partial support recovery**: $\operatorname{supp}(\hat{\boldsymbol{x}}) \subseteq \operatorname{supp}(\boldsymbol{x}_o)$. Our estimator exhibits no "false positives": every element of the estimated support is an element of the true support.
- **Signed support recovery**: $\operatorname{sign}(\hat{\boldsymbol{x}}) = \boldsymbol{\sigma}_o$. Our estimator correctly determines the nonzero entries of $\boldsymbol{x}_o$ and their signs.

Signed support recovery is clearly more desirable than partial support recovery. Signed support recovery requires stronger assumptions of the signal $\boldsymbol{x}_o$ than partial support recovery – if the nonzero entries of $\boldsymbol{x}_o$ are too small relative to the noise level σ, no method of any kind will be able to reliably determine the support.

In contrast, partial support recovery can be studied without additional assumptions on the signal $\boldsymbol{x}_o$. We will assume that $\boldsymbol{A} \sim_{\text{iid}} \mathcal{N}(0, \frac{1}{m})$. We will first derive a sharp phase transition for partial support recovery, at

$$m_\star = 2k \log(n - k) \tag{3.6.13}$$

measurements. The main result of this section will show that when m significantly exceeds this threshold, partial support recovery obtains with high probability. Moreover, through further analysis, we will show that when m significantly exceeds $m_\star$, *and* all of the nonzero entries of $\boldsymbol{x}_o$ are significantly larger than λ, *signed support recovery* also obtains with high probability. Conversely,

if m is significantly smaller than $m_\star$, the probability of signed support recovery is vanishingly small. Thus, $m_\star$ indeed gives a sharp threshold for support recovery. Notice that (3.6.13) grows roughly as $k \log n$, rather than $k \log(n/k)$. So, if m, n, k, grow in fixed ratios, support recovery is unlikely. In this sense, support recovery is a more challenging problem than estimation.

The following theorem makes the above discussion precise:

THEOREM 3.38 (Phase Transition in Partial Support Recovery). *Suppose that* $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ *with entries iid* $\mathcal{N}(0, \frac{1}{m})$ *random variables, and let* $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$, *with* $\boldsymbol{x}_o$ *a* k*-sparse vector and* $\boldsymbol{z} \sim_{\text{iid}} \mathcal{N}\left(0, \frac{\sigma^2}{m}\right)$. *If*

$$m \geq \left(1 + \frac{\sigma^2}{\lambda^2 k} + \varepsilon\right) 2k \log(n-k), \tag{3.6.14}$$

then with probability at least $1 - Cn^{-\varepsilon}$, *any solution* $\hat{\boldsymbol{x}}$ *to the Lasso problem*

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \tfrac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{3.6.15}$$

satisfies $\operatorname{supp}(\hat{\boldsymbol{x}}) \subseteq \operatorname{supp}(\boldsymbol{x}_o)$. *Conversely, if*

$$m < \left(1 + \frac{\sigma^2}{\lambda^2 k} - \varepsilon\right) 2k \log(n-k), \tag{3.6.16}$$

then the probability that there exists a solution $\hat{\boldsymbol{x}}$ *of the Lasso which satisfies* $\operatorname{sign}(\hat{\boldsymbol{x}}) = \operatorname{sign}(\boldsymbol{x}_o)$ *is at most* $Cn^{-\varepsilon}$. *Above,* $C > 0$ *is a positive numerical constant.*

Partial vs. (Exact) Signed Support Recovery.
The notion of support recovery in Theorem 3.38 is somewhat weak: it only demands that

$$\operatorname{supp}(\hat{\boldsymbol{x}}) \subseteq \operatorname{supp}(\boldsymbol{x}_o). \tag{3.6.17}$$

Put another way, *the support contains no false positives*. In many applications, we would like to *exactly* recover the support – i.e., we would like

$$\operatorname{supp}(\hat{\boldsymbol{x}}) = \operatorname{supp}(\boldsymbol{x}_o). \tag{3.6.18}$$

For this, we need that the nonzero entries of $\boldsymbol{x}_o$ are not too small, so that they do not become "lost" in the noise. Under (3.6.14), it is possible to show that exact support recovery occurs, as long as the smallest nonzero entry of $\boldsymbol{x}_o$ is larger than λ: if

$$\min_{i \in I} |\boldsymbol{x}_{oi}| > C\lambda, \tag{3.6.19}$$

then $\operatorname{sign}(\hat{\boldsymbol{x}}) = \boldsymbol{\sigma}_o$ with high probability. In the remainder of this section, we will prove Theorem 3.38. Exercise 3.18 guides the reader through an extension of this argument, which shows that under the same assumptions,

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_o\|_\infty < C\lambda. \tag{3.6.20}$$

When the nonzero entries of $\boldsymbol{x}_o$ have magnitude at least $C\lambda$, this implies that $\operatorname{sign}(\hat{\boldsymbol{x}}) = \boldsymbol{\sigma}_o$, as desired.

Main Ideas of the Proof of Theorem 3.38.
The phase transition in Theorem 3.38 has a strikingly simple formula: $m_\star = 2k\log(n-k)$. The proof of this result is similar in spirit to our first proof of the correctness of ℓ^1-minimization, Theorem 3.3, which directly manipulated the optimality conditions for the recovery program.

By differentiating the objective function (3.6.15), we can show that a given vector $\hat{\boldsymbol{x}}$ is optimal if and only if

$$\boldsymbol{A}^*\left(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\right) \in \lambda \partial \left\|\cdot\right\|_1 (\hat{\boldsymbol{x}}). \tag{3.6.21}$$

Let $\mathsf{J} = \operatorname{supp}(\hat{\boldsymbol{x}})$. Recall that the subdifferential $\partial \left\|\cdot\right\|_1 (\hat{\boldsymbol{x}})$ consists of those vectors $\boldsymbol{v} \in \mathbb{R}^n$ such that $\boldsymbol{v}_\mathsf{J} = \operatorname{sign}(\hat{\boldsymbol{x}}_\mathsf{J})$ and $\left\|\boldsymbol{v}_{\mathsf{J}^c}\right\|_\infty \leq 1$. Hence, the condition (3.6.21) decomposes into two conditions:

$$\boldsymbol{A}_\mathsf{J}^*\left(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\right) = \lambda \operatorname{sign}(\hat{\boldsymbol{x}}_\mathsf{J}), \tag{3.6.22}$$

$$\left\|\boldsymbol{A}_{\mathsf{J}^c}^*\left(\boldsymbol{y} - \boldsymbol{A}\hat{\boldsymbol{x}}\right)\right\|_\infty \leq \lambda. \tag{3.6.23}$$

Much like the proof of Theorem 3.3, we will proceed as follows: we will construct a guess at a solution vector $\boldsymbol{x}_\star$ such that the equality constraints in (3.6.22) are automatically satisfied. We will then be left to check the inequality constraints (3.6.23). In particular, we will construct our guess $\boldsymbol{x}_\star$ at the solution by solving a *restricted* Lasso problem

$$\boldsymbol{x}_\star \in \arg\min\nolimits_{\operatorname{supp}(\boldsymbol{x}) \subseteq \mathsf{I}} \left\{ \tfrac{1}{2}\left\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\right\|_2^2 + \lambda \left\|\boldsymbol{x}\right\|_1 \right\}, \tag{3.6.24}$$

where $\mathsf{I} = \operatorname{supp}(\boldsymbol{x}_o)$.

Recall that that $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$. We can write

$$\boldsymbol{r} \doteq \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star = \boldsymbol{A}_\mathsf{I}(\boldsymbol{x}_{o\mathsf{I}} - \boldsymbol{x}_{\star\mathsf{I}}) + \boldsymbol{z}. \tag{3.6.25}$$

Notice that $\boldsymbol{r}$ depends only on $\boldsymbol{A}_\mathsf{I}$ and $\boldsymbol{z}$; it is probabilistically independent of $\boldsymbol{A}_{\mathsf{I}^c}$. The key work that we will do in proving Theorem 3.38 is to determine whether the ℓ^∞ norm constraint is satisfied on I^c. That is to say, we need to study

$$\left\|\boldsymbol{A}_{\mathsf{I}^c}^*\left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right)\right\|_\infty = \left\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\right\|_\infty. \tag{3.6.26}$$

The matrix $\boldsymbol{A}_{\mathsf{I}^c}$ is a Gaussian matrix; moreover, it is probabilistically independent of $\boldsymbol{r}$. Conditioned on $\boldsymbol{r}$, $\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}$ is distributed as an $(n-k)$-dimensional iid $\mathcal{N}\left(0, \frac{\left\|\boldsymbol{r}\right\|_2^2}{m}\right)$ random vector. We will see that the ℓ^∞ norm of such a vector is sharply concentrated about $\left\|\boldsymbol{r}\right\|_2 \sqrt{\frac{2\log(n-k)}{m}}$. The following lemma provides the control that we need:

Lemma 3.39. *Suppose that $\boldsymbol{q} = [q_1, \ldots, q_d]^* \in \mathbb{R}^d$ is a $d \geq 2$-dimensional random*

vector, whose elements are independent $\mathcal{N}(0, \xi^2)$ *random variables. Then, for any* $\varepsilon \in [0, 1)$,

$$\mathbb{P}\left[\|\boldsymbol{q}\|_\infty < \xi\sqrt{(2-\varepsilon)\log d}\right] \le \exp\left(-\frac{d^{\varepsilon/2}}{4\sqrt{2\log d}}\right), \tag{3.6.27}$$

$$\mathbb{P}\left[\|\boldsymbol{q}\|_\infty > \xi\sqrt{(2+\varepsilon)\log d}\right] \le 2d^{-\varepsilon/2}. \tag{3.6.28}$$

This lemma can be proved using relatively elementary ideas (the union bound for the upper bound, a direct calculation for the lower bound). Using this lemma, we conclude that, conditioned on $\boldsymbol{r}$ (i.e., with high probability in $\boldsymbol{A}_{\mathsf{I}^c}$), $\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty$ is very close to $\|\boldsymbol{r}\|_2\sqrt{\frac{2\log(n-k)}{m}}$. To understand whether this quantity is smaller than λ (and hence recovery succeeds) or larger than λ (and hence recovery fails), we will need to study the norm of $\boldsymbol{r}$.

Notice that $\boldsymbol{r} = \boldsymbol{A}_{\mathsf{I}}(\boldsymbol{x}_{o\mathsf{I}} - \boldsymbol{x}_{\star\mathsf{I}}) + \boldsymbol{z}$. To study the size of $\boldsymbol{r}$ it will be important to understand the properties of the random matrix $\boldsymbol{A}_{\mathsf{I}}$ and the random vector $\boldsymbol{z}$. Because $\boldsymbol{A}_{\mathsf{I}} \in \mathbb{R}^{m\times k}$ is a "tall", random matrix, it is well-conditioned, in a sense that the following lemma makes precise:

LEMMA 3.40. *Let* $\boldsymbol{G} \in \mathbb{R}^{m\times k}$ *be a random matrix whose entries are iid* $\mathcal{N}(0, \frac{1}{m})$ *random variables. Then, with high probability*

$$\|\boldsymbol{G}^*\boldsymbol{G} - \boldsymbol{I}\|_{\ell^2\to\ell^2} \le C\sqrt{\tfrac{k}{m}}. \tag{3.6.29}$$

The proof of this lemma follows similar lines to our proof of the RIP property of Gaussian matrices (discretization, tail bound, union bound). Using this lemma, we can control $\|\boldsymbol{r}\|_2$; combining with the above calculations, we obtain control on $\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty$. The prescription for the required number of measurements m follows by demanding that this quantity be smaller than λ. To formally prove Theorem 3.38, we need to do a bit more. First, we need to formally control $\|\boldsymbol{r}\|_2$ and $\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty$. This is sufficient to show that our putative solution $\boldsymbol{x}_\star$ is indeed optimal. Second, we need to argue that under the same conditions, *every* solution $\hat{\boldsymbol{x}}$ indeed satisfies $\operatorname{supp}(\hat{\boldsymbol{x}}) \subseteq \operatorname{supp}(\boldsymbol{x}_o)$. This will follow from some auxiliary reasoning about the subdifferential of the ℓ^1 norm. Finally, we obtain the converse portion of Theorem 3.38 by showing that when the number of measurements $m \ll m_\star$, with high probability $\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty > \lambda$, and hence no putative solution $\boldsymbol{x}_\star$ with $\operatorname{sign}(\boldsymbol{x}_\star) = \boldsymbol{\sigma}_o$ can be optimal. We carry through all of this reasoning rigorously below.

Proof of Theorem 3.38: We proceed as follows.

i. Sufficient condition for partial support recovery.
Let $\mathsf{I} = \operatorname{supp}(\boldsymbol{x}_o)$. We wish to show that every solution $\hat{\boldsymbol{x}}$ to the Lasso problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \varphi(\boldsymbol{x}) \doteq \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{3.6.30}$$

satisfies $\mathrm{supp}(\boldsymbol{x}) \subseteq \mathsf{I}$. To do this, we will generate a vector $\boldsymbol{x}_\star$ with $\mathrm{supp}(\boldsymbol{x}_\star) \subseteq \mathsf{I}$, such that the residual

$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star \tag{3.6.31}$$

satisfies

$$\boldsymbol{A}^*\boldsymbol{r} \in \lambda \partial \left\|\cdot\right\|_1 (\boldsymbol{x}_\star), \tag{3.6.32}$$

$$\text{and} \qquad \left\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\right\|_\infty < \lambda. \tag{3.6.33}$$

The first property implies that $\boldsymbol{x}_\star$ is optimal for the Lasso problem, since it implies that

$$\begin{aligned} \boldsymbol{0} \in \partial\varphi(\boldsymbol{x}_\star) &= \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{x}_\star - \boldsymbol{y}) + \lambda\partial \left\|\cdot\right\|_1 (\boldsymbol{x}_\star) \\ &= -\boldsymbol{r} + \lambda\partial \left\|\cdot\right\|_1 (\boldsymbol{x}_\star). \end{aligned} \tag{3.6.34}$$

The property $\left\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\right\|_\infty < \lambda$ implies that any other optimal solution *also* has support contained in I. The reason is as follows: let $\lambda' = \lambda - \left\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\right\|_\infty > 0$. Then for any vector $\boldsymbol{v}$ supported on I^c, with $\left\|\boldsymbol{v}\right\|_\infty < \lambda'$, we have that

$$\boldsymbol{v} \in \partial\varphi_{\text{Lasso}}(\boldsymbol{x}_\star). \tag{3.6.35}$$

For any $\boldsymbol{x}'$ with $\boldsymbol{x}'_{\mathsf{I}^c} \neq \boldsymbol{0}$, set $\boldsymbol{v} = \lambda'\mathrm{sign}(\boldsymbol{x}'_{\mathsf{I}^c})/2$ and note that by the subgradient inequality,

$$\begin{aligned} \varphi(\boldsymbol{x}') &\geq \varphi(\boldsymbol{x}_\star) + \langle \boldsymbol{x}' - \boldsymbol{x}_\star, \boldsymbol{v}\rangle \\ &= \varphi(\boldsymbol{x}_\star) + \tfrac{\lambda'}{2}\left\|\boldsymbol{x}'_{\mathsf{I}^c}\right\|_1 \\ &> \varphi(\boldsymbol{x}_\star), \end{aligned} \tag{3.6.36}$$

and hence, $\boldsymbol{x}'$ is not optimal. Thus, if there exists an $\boldsymbol{x}_\star$ satisfying (3.6.32)–(3.6.33), then every solution $\hat{\boldsymbol{x}}$ to the Lasso problem satisfies $\mathrm{supp}(\hat{\boldsymbol{x}}) \subseteq \mathrm{supp}(\boldsymbol{x}_o)$.

ii. Constructing the putative solution $\boldsymbol{x}_\star$.
Let

$$\boldsymbol{x}_\star \in \mathrm{argmin}_{\mathrm{supp}(\boldsymbol{x})\subseteq\mathsf{I}} \tfrac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\right\|_2^2 + \lambda\left\|\boldsymbol{x}\right\|_1. \tag{3.6.37}$$

Let $\mathsf{J} = \mathrm{supp}(\boldsymbol{x}_\star) \subseteq \mathsf{I}$. The KKT optimality conditions for this problem give that

$$\boldsymbol{A}_{\mathsf{J}}^*(\boldsymbol{y} - \boldsymbol{A}_{\mathsf{I}}\boldsymbol{x}_{\star\mathsf{I}}) = \lambda\,\mathrm{sign}(\boldsymbol{x}_{\star\mathsf{J}}), \tag{3.6.38}$$

$$\left\|\boldsymbol{A}_{\mathsf{I}\setminus\mathsf{J}}^*(\boldsymbol{y} - \boldsymbol{A}_{\mathsf{I}}\boldsymbol{x}_{\star\mathsf{I}})\right\|_\infty \leq \lambda. \tag{3.6.39}$$

An equivalent way of expressing these conditions is to say that

$$\boldsymbol{A}_{\mathsf{I}}^*(\boldsymbol{y} - \boldsymbol{A}_{\mathsf{I}}\boldsymbol{x}_{\star\mathsf{I}}) = \lambda\boldsymbol{\nu}, \tag{3.6.40}$$

for some $\boldsymbol{\nu} \in \partial \left\|\cdot\right\|_1 (\boldsymbol{x}_{\star\mathsf{I}})$.

Because $\boldsymbol{y} = \boldsymbol{A}_{\mathsf{I}}\boldsymbol{x}_{o\mathsf{I}} + \boldsymbol{z}$, we can use (3.6.40) to express the difference $\boldsymbol{x}_{o\mathsf{I}} - \boldsymbol{x}_{\star\mathsf{I}}$ in terms of the subgradient $\boldsymbol{\nu}$ and the noise $\boldsymbol{z}$:

$$\boldsymbol{x}_{o\mathsf{I}} - \boldsymbol{x}_{\star\mathsf{I}} = (\boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{A}_{\mathsf{I}})^{-1}(\lambda\boldsymbol{\nu} - \boldsymbol{A}_{\mathsf{I}}^*\boldsymbol{z}). \tag{3.6.41}$$

Notice that since $m > k$, with probability one, $\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}}$ is invertible, and so this expression indeed makes sense.

iii. Verifying the KKT conditions.
We will prove that the restricted solution $\boldsymbol{x}_\star$ is indeed optimal for the full problem (3.6.30). The KKT conditions for *this problem* give that $\boldsymbol{x}_\star$ is optimal if and only if

$$\boldsymbol{A}^* \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right) \in \lambda \partial \left\|\cdot\right\|_1 (\boldsymbol{x}_\star). \tag{3.6.42}$$

Let $\mathsf{J} = \operatorname{supp}(\boldsymbol{x}_\star)$. The above expression can be broken into two parts as

$$\boldsymbol{A}_{\mathsf{J}}^* \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right) = \lambda \operatorname{sign}(\boldsymbol{x}_{\star \mathsf{J}}), \tag{3.6.43}$$

$$\left\|\boldsymbol{A}_{\mathsf{I} \cap \mathsf{J}^c}^* \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right)\right\|_\infty \leq \lambda, \tag{3.6.44}$$

$$\left\|\boldsymbol{A}_{\mathsf{I}^c}^* \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right)\right\|_\infty \leq \lambda. \tag{3.6.45}$$

Because $\boldsymbol{x}_{\star \mathsf{I}}$ satisfies the restricted KKT conditions, the first two conditions are automatically satisfied; to complete the proof, we establish the stronger version

$$\left\|\boldsymbol{A}_{\mathsf{I}^c}^* \left(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star\right)\right\|_\infty < \lambda \tag{3.6.46}$$

of the third – this is the condition (3.6.33) that $\left\|\boldsymbol{r}\right\|_\infty < \lambda$. Using (3.6.41), we can express the residual $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star$ as

$$\begin{aligned} \boldsymbol{r} &\doteq \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_\star \\ &= \left[\boldsymbol{I} - \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \boldsymbol{A}_{\mathsf{I}}^*\right] \boldsymbol{z} + \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \lambda \boldsymbol{\nu}. \end{aligned} \tag{3.6.47}$$

The two components of $\boldsymbol{r}$ are orthogonal, and so

$$\begin{aligned} \left\|\boldsymbol{r}\right\|_2 &= \sqrt{\left\|\left[\boldsymbol{I} - \boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \boldsymbol{A}_{\mathsf{I}}^*\right] \boldsymbol{z}\right\|_2^2 + \left\|\boldsymbol{A}_{\mathsf{I}} (\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})^{-1} \lambda \boldsymbol{\nu}\right\|_2^2} \\ &\leq \sqrt{\left\|\boldsymbol{z}\right\|_2^2 + \lambda^2 \frac{\left\|\boldsymbol{\nu}\right\|_2^2}{\sigma_{\min}(\boldsymbol{A}_{\mathsf{I}}^* \boldsymbol{A}_{\mathsf{I}})}} \\ &\leq \sqrt{\sigma^2 + \frac{\lambda^2 k}{1 - Ck/m}} \qquad \text{with high probability} \\ &\leq \sqrt{\sigma^2 + \lambda^2 k + C' \lambda^2 k^2 / m}. \end{aligned} \tag{3.6.48}$$

Applying the above lemma, with high probability in $\boldsymbol{A}_{\mathsf{I}^c}$,

$$\begin{aligned} \left\|\boldsymbol{A}_{\mathsf{I}^c}^* \boldsymbol{r}\right\|_\infty &< \sqrt{\frac{(2+\varepsilon)\log(n-k)}{m}} \left\|\boldsymbol{r}\right\|_2 \\ &\leq \lambda \left(\frac{\left(2k \log(n-k) \left(1 + \frac{\sigma^2}{\lambda^2 k} + \varepsilon\right)\right)}{m} \right)^{1/2}. \end{aligned} \tag{3.6.49}$$

Under our hypothesis on m, this is strictly smaller than λ, and so indeed (3.6.33) is verified.

iv. No signed support recovery when $m \ll m_\star$.
We next prove that when m is significantly smaller than $2k\log(n-k)$, no vector $\boldsymbol{x}$ satisfying

$$\operatorname{sign}(\boldsymbol{x}) = \operatorname{sign}(\boldsymbol{x}_o) \tag{3.6.50}$$

can be a solution to the Lasso problem. Without loss of generality, we can assume that $m \geq k$.[24] Suppose on the contrary that $\boldsymbol{x}$ was the solution to the Lasso problem. Then $\boldsymbol{x}$ is also the solution to the restricted Lasso problem. Moreover, since $\operatorname{sign}(\boldsymbol{x}_\mathsf{I}) = \boldsymbol{\sigma}_\mathsf{I}$ has no zero entries, we have

$$\boldsymbol{r} = \left[\boldsymbol{I} - \boldsymbol{A}_\mathsf{I}(\boldsymbol{A}_\mathsf{I}^*\boldsymbol{A}_\mathsf{I})^{-1}\boldsymbol{A}_\mathsf{I}^*\right]\boldsymbol{z} + \lambda\boldsymbol{A}_\mathsf{I}(\boldsymbol{A}_\mathsf{I}^*\boldsymbol{A}_\mathsf{I})^{-1}\boldsymbol{\sigma}_\mathsf{I}. \tag{3.6.52}$$

With high probability,

$$\left\|\left[\boldsymbol{I} - \boldsymbol{A}_\mathsf{I}(\boldsymbol{A}_\mathsf{I}^*\boldsymbol{A}_\mathsf{I})^{-1}\boldsymbol{A}_\mathsf{I}^*\right]\boldsymbol{z}\right\|_2^2 > (1-\varepsilon)(n-k)\sigma^2 \tag{3.6.53}$$

and

$$\left\|\boldsymbol{A}_\mathsf{I}(\boldsymbol{A}_\mathsf{I}^*\boldsymbol{A}_\mathsf{I})^{-1}\lambda\boldsymbol{\sigma}_\mathsf{I}\right\|_2^2 > \frac{\lambda^2 k}{1+Ck/m}, \tag{3.6.54}$$

whence, with high probability,

$$\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty > \sqrt{\frac{(2-\varepsilon)\log(n-k)}{m}}\,\|\boldsymbol{r}\|_2 \tag{3.6.55}$$

and

$$\|\boldsymbol{r}\|_2 \;\geq\; \sqrt{\sigma^2(1-ck/m) + \lambda^2 k(1-c'k/m)}. \tag{3.6.56}$$

Combining, we obtain

$$\begin{aligned}\|\boldsymbol{A}_{\mathsf{I}^c}^*\boldsymbol{r}\|_\infty &> \lambda\sqrt{\frac{(2-\varepsilon)k\log(n-k)\left(1+\frac{\sigma^2}{\lambda^2 k}+\varepsilon\right)}{m}} \\ &\geq \lambda.\end{aligned} \tag{3.6.57}$$

Hence, the putative solution $\boldsymbol{x}$ *is not* optimal for the full Lasso problem, with high probability in the matrix $\boldsymbol{A}$ and the noise $\boldsymbol{z}$. The above argument depends on $\boldsymbol{x}$ only through its sign and support pattern, and so on the same (large probability) bad event, *every* $\boldsymbol{x}$ having this sign and support pattern is suboptimal for the full Lasso problem. □

[24] If on the contrary, $m < k$, then the KKT conditions for the restricted problem become

$$\underbrace{\boldsymbol{A}_\mathsf{I}^*\boldsymbol{A}_\mathsf{I}}_{\text{Rank deficient}}\;\boldsymbol{x}_\mathsf{I} = \boldsymbol{A}_\mathsf{I}^*\boldsymbol{y} - \lambda\boldsymbol{\sigma}_\mathsf{I}. \tag{3.6.51}$$

This equation admits a solution if and only if $\boldsymbol{\sigma}_\mathsf{I} \in \operatorname{range}(\boldsymbol{A}_\mathsf{I}^*)$. Because $\boldsymbol{A}_\mathsf{I}^*$ is a tall Gaussian matrix, the probability that its range contains the fixed vector $\boldsymbol{\sigma}_\mathsf{I}$ is zero. So, when $m < k$, the probability that the Lasso problem admits a solution $\hat{\boldsymbol{x}}$ with $\operatorname{sign}(\hat{\boldsymbol{x}}) = \boldsymbol{\sigma}_o$ is zero.

3.7 Summary

In this chapter, we have provided a rather extensive and thorough study of conditions under which we can expect the ℓ^1 minimization:

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

to recover a k-sparse vector $\boldsymbol{x}_o \in \mathbb{R}^n$ from the observation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o \in \mathbb{R}^m$. Such conditions are developed through three different perspectives that give increasingly sharper characterization about the conditions.

Mutual Coherence.
The first approach is based on the notion of *mutual coherence* $\mu(\boldsymbol{A})$ of the measurement matrix $\boldsymbol{A}$, given in Definition 3.1. Theorem 3.3 shows that the ℓ^1 minimization finds the correct solution $\boldsymbol{x}_o$ if $k \leq \frac{1}{2\mu(\boldsymbol{A})}$. Based on an upper bound of $\mu(\boldsymbol{A})$ for a random matrix, Theorem 3.5, and a lower bound for an arbitrary matrix, Theorem 3.7, mutual coherence in general ensures that ℓ^1 minimization succeeds when

$$m = O(k^2).$$

Restricted Isometry Property.
The *restricted isometric* measure $\delta_k(\boldsymbol{A})$ of a matrix $\boldsymbol{A}$, given in Definition 3.8, provides a more refined characterization of the incoherence property of the measurement $\boldsymbol{A}$, by restricting the notion of isometry to the k-dimensional structures of interest. Theorem 3.10 and Theorem 3.11 show that with high-probability the ℓ^1 minimization can succeed in recovering a k-sparse vector from a generic $m \times n$ matrix $\boldsymbol{A}$ with

$$m = O\big(k \log(n/k)\big).$$

In the proportional growth model when $k \propto n$, this means the number of random measurements needed is $m = O(k)$.

Sharp Phase Transition.
While the above two approaches give qualitative bounds on the number of random measurements needed for ℓ^1 to succeed, Section 3.6 gives a precise characterization of the sharp *phase transition behavior* for success or failure of ℓ^1 minimization around a critical number of measures

$$m^\star = \psi\Big(\frac{k}{n}\Big)n.$$

An explicit expression (3.6.6) for the function ψ can be derived from the statistical relationships between high-dimensional convex cones and subspaces, as we will study systematically in Chapter 6.

Sensitivity Analysis.
Results given in Section 3.5 show that under similar conditions, ℓ^1 minimization, with slight modification, can recover sufficiently accurate estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}_o$ when there is noise in the measurement $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_o + \boldsymbol{z}$ or the signal $\boldsymbol{x}_o$ is only approximately sparse. These results ensure that ℓ^1 minimization is not sensitive to the modeling assumption that the ground truth vector $\boldsymbol{x}_o$ needs to be perfectly sparse. Theorem 3.38 shows that when the measurements are noisy, phase transition also occurs when we only care about recovering the correct sign and support of $\boldsymbol{x}_o$.

3.8 Notes

As we have mentioned before, historically ℓ^1 minimization was suggested to be beneficial as early as in the work of Boscovitch [Bos50] and later Laplace [Lap74]. To our knowledge, the first result that offers a guarantee for exact recovery of sparse signals via ℓ^1 minimization was obtained by B. Logan [Log65]. The advancement in computational power in recent years has made it possible to harness the tremendous benefits of ℓ^1 minimization in high-dimensional spaces, which has led to the revived interests in analyzing its sample and computational complexity more precisely.

Analyses of sparse recovery based on mutual coherence/incoherence are due to [GN03, DE03]. The proof approach described here is due to [Fuc04]. The stronger guarantee of ℓ^1 minimization via the notion of restricted isometry property (RIP) is due to the seminal work [CT05]. Our proof here follows closely to that of [CRT06b, Can08]. The analysis of phase transitions via observation space geometry was developed in a series of work [Don05, DT09, DT10]. The approach to phase transitions via coefficient space geometry follows mainly the work of [ALMT14]. We will give a more detailed account of this approach in Chapter 6 where we justify why phase transitions occur for the recovery of a broad family of low-dimensional models. The analysis of phase transitions in support recovery is due to [Wai09b].

3.9 Exercises

3.1 (Projection of Polytopes). *Notice that in $\mathbb{R}^3$, when we project an ℓ^1 ball B_1 to $\mathbb{R}^2$, in general all the vertices (1-faces) will be preserved. Does this generalize to higher-dimensional spaces? That is if we project an ℓ^1 ball in $\mathbb{R}^n$ to $\mathbb{R}^{n-1}$, can we expect all $(n-2)$-faces be preserved by a generic projection? You may run some simulations and argue if your hypothesis is true or false.*

3.2 (Mutual Coherence). *Compute by hand the mutual coherence of the matrix in Exercise 2.5. Then, program an algorithm that calculates the mutual coherence*

of a matrix. Generate an $n \times n$ Discrete Fourier Transform matrix $\boldsymbol{F}$ for a very large n. Randomly select $1/2$ of its rows and compute its mutual coherence.

3.3 (Comparisons between Norms). *Show that for all $\boldsymbol{x} \in \mathbb{R}^n$, we have the following relationships among the three norms $\|\cdot\|_1, \|\cdot\|_2$, and $\|\cdot\|_\infty$:*

1 $\|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1 \leq \sqrt{n}\,\|\boldsymbol{x}\|_2$.
2 $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_2 \leq \sqrt{n}\,\|\boldsymbol{x}\|_\infty$.
3 $\|\boldsymbol{x}\|_\infty \leq \|\boldsymbol{x}\|_1 \leq n\,\|\boldsymbol{x}\|_\infty$.

3.4 (Singular Values of Matrices). *Show that given a positive definite matrix $\boldsymbol{S} \in \mathbb{R}^{n\times n}$:*

1 $\sigma_{\max}(\boldsymbol{S}^{-1}) = \sigma_{\min}(\boldsymbol{S})^{-1}$.
2 $trace(\boldsymbol{S}) = \sum_{i=1}^n \sigma_i(\boldsymbol{S})$.
3 $\|\boldsymbol{S}\|_F = \sqrt{\sum_{i=1}^n \sigma_i^2(\boldsymbol{S})}$.

3.5. *Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$,*

1 What is the relationship between singular values of a matrix $\boldsymbol{A}$ and $\boldsymbol{A}^*\boldsymbol{A}$?
2 What is the comparison between the spectral norm $\|\boldsymbol{A}\|$ and the Frobenius norm of $\|\boldsymbol{A}\|_F$?

3.6. *Prove the inequalities in (3.2.3).*

3.7 (Constrained Optimization). *Consider the program:*

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \textit{subject to} \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0},$$

where $f(\cdot) \in \mathbb{R}$ and $\boldsymbol{h}(\cdot) \in \mathbb{R}^m$ are all C^1-differentiable. Show that if $\boldsymbol{x}_\star$ is an optimal solution, we must have

$$\nabla f(\boldsymbol{x}_\star) = \frac{\partial \boldsymbol{h}(\boldsymbol{x}_\star)}{\partial \boldsymbol{x}} \boldsymbol{\lambda}$$

for $\boldsymbol{\lambda} \in \mathbb{R}^m$, where $\frac{\partial \boldsymbol{h}(\boldsymbol{x}_\star)}{\partial \boldsymbol{x}}$ is the Jacobian of $\boldsymbol{h}(\cdot)$ at $\boldsymbol{x}_\star$. Notice that in our context:

1 The constraints $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}$. What is its Jacobian, and what the above conditions have become?
2 The function $f(\cdot)$ is not necessarily differentiable at $\boldsymbol{x}_\star$. Discuss how the above condition needs to be changed?

A less relevant but otherwise useful question for bonus points: What if the constraints are replaced with inequalities $\boldsymbol{h}(\boldsymbol{x}) \geq \boldsymbol{0}$?

3.8. *Prove equation (3.2.34).*

3.9. *In this exercise, use the sphere measure concentration Theorem 3.6 to prove a fact mentioned in the Introduction chapter, equation (1.3.5): in $\mathbb{R}^m$ when the*

dimension m is high, a randomly chosen unit vector $\boldsymbol{v} \in \mathbb{S}^{m-1}$ is with high probability highly incoherent (nearly orthogonal) to any of the standard base vectors $\boldsymbol{e}_i, i = 1, \ldots, m$. More precisely, given any small $\varepsilon > 0$, we have

$$|\langle \boldsymbol{e}_i, \boldsymbol{v} \rangle| \leq \varepsilon, \quad \forall i = 1, \ldots, m,$$

with high probability as m is large enough. (Hint: the proof should be very similar to, actually simpler than, the proof for Theorem 3.5. You only need to apply the measure concentration result to the functions $|\langle \boldsymbol{e}_i, \boldsymbol{v} \rangle|$ and characterize the union bound for the failure probability of all m functions.)

3.10 (ℓ^1 Minimization Experiments). *Program an algorithm to solve the ℓ^1 minimization problem.*

1. *Set $m = n/2$ and set $k = \|\boldsymbol{x}_o\|_0$ proportional to m – say, $k = m/4$. Then, try different aspect ratios $m = \alpha n$ and sparsity ratios $k = \beta m$.*
2. *Validate the Phase Transition in Figure 3.15.*

3.11. *Let $\boldsymbol{A}$ be a large $m \times n$ matrix with $m = n/4$. If you are told that any submatrix $\boldsymbol{A}_{\mathsf{I}}$ with $|\mathsf{I}| = k < m$ columns of $\boldsymbol{A}$ satisfies:*

$$\forall \boldsymbol{x} \in \mathbb{R}^k \quad (1-\delta)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}_{\mathsf{I}}\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2$$

with $\delta \leq 3\sqrt{k/m}$. Use this fact and Theorem 3.10 to give your best estimate of k as a fraction of n such that ℓ^1 minimization succeeds for all k-sparse vectors?

3.12. *Prove Lemma 3.15.*

3.13 (Johnson-Lindenstrauss). *Program an algorithm to validate the Johnson-Lindenstrauss Lemma.*

3.14. *Prove Lemma 3.22.*

3.15 (Compact Projection). *In this exercise, we use the properties of random projection to develop simple but efficient algorithm for computing approximate nearest neighbors for a high-dimensional dataset. In particular, prove that the scheme described in Example 3.23 is correct and most efficient. Show that:*

1. *With the random binary code generated by Algorithm 3.1, with probability $1-\delta$, the c-NN problem can be solved on any (Δ, l)-weakly separable set $\mathcal{X}$ with the number of binary bits m chosen to be in the order:*

$$m = O\left(\frac{\log(2/\delta) + \log n}{(1-1/c)^2\Delta}\right).$$

2. *The correct solution to the c-NN problem is given by*

$$\boldsymbol{x}_\star = \arg\min_{\boldsymbol{x} \in \tilde{\mathcal{X}}} \|\boldsymbol{x} - \boldsymbol{q}\|_2$$

where $\tilde{\mathcal{X}}$ is the subset of points of size $O(n^l)$ in $\mathcal{X}$ which have the shortest Hamming distances to $\boldsymbol{y}_q = \sigma(\boldsymbol{R}\boldsymbol{q})$.

3 With the above results, show that the c-NN problem can be solved with the following complexity[25]*:*

- *Code construction:* $O(Dn \log n)$*;*
- *Computation per query:* $O(n + Dn^l)$*;*
- *Index space:* $O(n)$*.*

3.16. *Given a matrix* $\boldsymbol{A}$ *of full column rank, show that*

$$\|(\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*\boldsymbol{z}\|_2 \leq \frac{1}{\sigma_{\min}(\boldsymbol{A})}\|\boldsymbol{z}\|_2.$$

3.17 (Restricted Isometry Property*). *Program an algorithm that calculates the order-k RIP constant of a matrix:*

```
delta = rip(A, k).
```

Generate an $n \times n$ *Discrete Fourier Transform matrix* $\boldsymbol{F}$*. Randomly select* $1/2$ *of its rows and compute its RIP constant. How large* n *can your algorithm go? Compare that with the case with mutual coherence.*

3.18. *Under the same assumption of Theorem 3.38, sketch a proof of signed support recovery in the sense of equation* (3.6.20).

[25] Note that, here, one can adopt the standard $(\log n)$-RAM computational model, in which arithmetic operations with $\log n$ bits can be performed in $O(1)$ time.

4 Convex Methods for Low-Rank Matrix Recovery

"*Mathematics is the art of giving the same name to different things.*"
– Henri Poincaré, *L'avenir des mathématiques, 1905*

In this chapter, we will branch out from sparse signals to a broader class of models: the low-rank matrices. Similar to the problem of recovering sparse signals, we consider how to recover a matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ from linear measurements $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}] \in \mathbb{R}^m$. This problem can be phrased as searching for a solution $\boldsymbol{X}$ to a linear system of equations

$$\mathcal{A}\Big[\underbrace{\boldsymbol{X}}_{\text{unknown}}\Big] = \underbrace{\boldsymbol{y}}_{\text{observation}}. \tag{4.0.1}$$

Here, $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is a linear map.

We will see that much of the mathematical structure in the sparse vector recovery problem carries over in a very natural way to this more general setting. In particular, in many interesting instances, we need to recover $\boldsymbol{X}$ from far fewer measurements than the number of entries in the matrix, i.e., $m \ll n_1 \times n_2$. Unless we can leverage some additional prior information about $\boldsymbol{X}$, the problem of recovering $\boldsymbol{X}$ from the linear measurements $\boldsymbol{y}$ is ill-posed.

We will consider applications in which we can leverage the following powerful piece of structural information: the target matrix $\boldsymbol{X}$ is *low-rank* or approximately so. Recall that the rank of a matrix $\boldsymbol{X}$ is the dimension of the linear subspace $\mathrm{col}(\boldsymbol{X})$ spanned by the columns of $\boldsymbol{X}$. If $\boldsymbol{X} = [\boldsymbol{x}_1 \mid \cdots \mid \boldsymbol{x}_{n_2}] \in \mathbb{R}^{n_1 \times n_2}$ is a data matrix whose columns are n_1-dimensional vectors, then $\mathrm{rank}(\boldsymbol{X}) = r \ll n_1$ if and only if the columns of $\boldsymbol{X}$ lie on an r-dimensional linear subspace of the data space $\mathbb{R}^{n_1}$ – see Figure 4.1 for an illustration. Low-rank matrix recovery problems arise in a broad range of application areas. We sketch a few of these below.

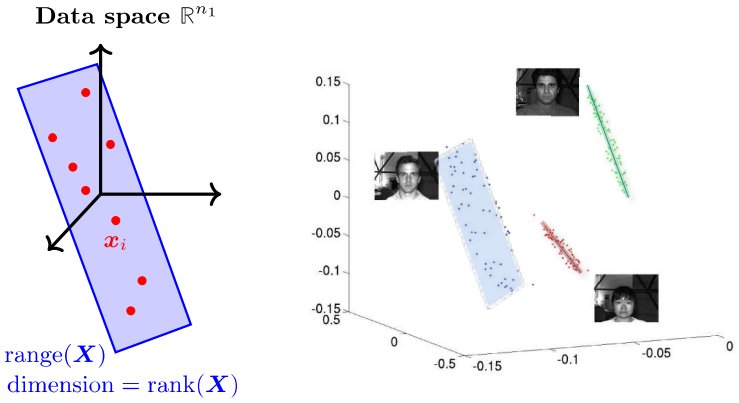


Figure 4.1 Low-rank Data Matrices. If matrix $\boldsymbol{X}$ with columns $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_2}$ has rank r, its columns lie on an r-dimensional subspace range($\boldsymbol{X}$). Many naturally occurring data matrices approximately satisfy this property. Right: low-dimensional approximations to images of faces under different lighting conditions.

4.1 Motivating Examples of Low-Rank Modeling

4.1.1 3D Shape from Photometric Measurements

As mentioned in the introduction, there are many situations in which low-rank data models arise due to the physical processes that generate the data. If the generative process has limited degrees of freedom, the data we observe would intrinsically be low dimensional, regardless of the dimension of the ambient space in which such data are observed or measured. For example, in computer vision, low rank models arise in a number of problems in reconstructing three-dimensional shape of a scene from two-dimensional images.[1] In *photometric stereo* [Woo80], we obtain images $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{n_2} \in \mathbb{R}^{n_1}$ of an object, say a face, illuminated by different distant point light sources. Write $\boldsymbol{Y} = [\boldsymbol{y}_1 \mid \cdots \mid \boldsymbol{y}_{n_2}] \in \mathbb{R}^{n_1 \times n_2}$. Let $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_{n_2} \in \mathbb{S}^2$ denote the directions of these light sources. The *Lambertian model* for reflectance models the reflected light intensity as

$$Y_{ij} = \alpha_i [\langle \boldsymbol{\nu}_i, \boldsymbol{l}_j \rangle]_+,$$

where $\boldsymbol{\nu}_i \in \mathbb{S}^2$ is the surface normal at the i-th pixel, α_i is a nonnegative scalar known as the *albedo*, and $[\cdot]_+$ takes the positive part of its argument. This model is appropriate for matte objects. See Figure 4.2 for a visualization of this model.

[1] Do not confuse the dimension of the measurements, in this case, the number of pixels with the physical dimension of the image array, which is two.

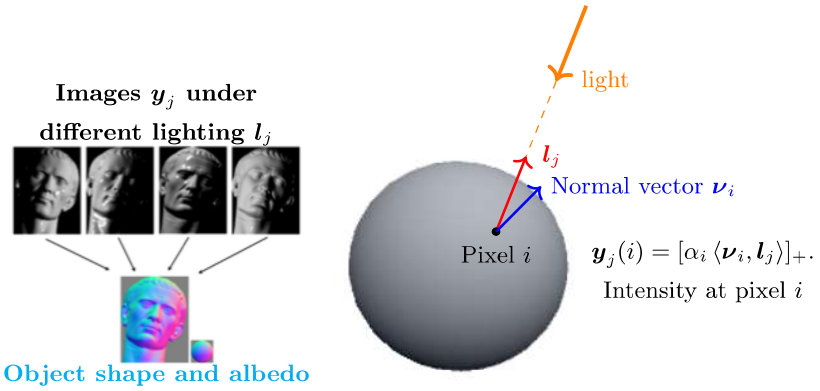


Figure 4.2 Photometric Stereo as Low-rank Matrix Recovery. Photometric stereo (left) seeks to recover object shape from images taken under different illuminations. Under a diffuse reflective (Lambertian) model (right), this leads directly to a low-rank recovery problem.

Under this model, if we let

$$\boldsymbol{N} = \begin{bmatrix} \alpha_1 \boldsymbol{\nu}_1^* \\ \vdots \\ \alpha_m \boldsymbol{\nu}_m^* \end{bmatrix} \in \mathbb{R}^{n_1 \times 3}, \quad \text{and} \quad \boldsymbol{L} = \begin{bmatrix} \boldsymbol{l}_1 \mid \cdots \mid \boldsymbol{l}_{n_2} \end{bmatrix} \in \mathbb{R}^{3 \times n_2},$$

then we have

$$\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{N}\boldsymbol{L}],$$

where

$$\Omega \doteq \{(i,j) \mid \langle \boldsymbol{\nu}_i, \boldsymbol{l}_j \rangle \geq 0\}.$$

If we can recover the low-rank matrix $\boldsymbol{X} = \boldsymbol{N}\boldsymbol{L}$ (of maximum rank 3), we can then recover information about the shape and reflectance of the object. Again, a useful heuristic is to look for a solution of minimum rank consistent with the observations [WGS+10]:

$$\begin{array}{ll} \min & \operatorname{rank}(\boldsymbol{X}), \\ \text{subject to} & \mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}. \end{array} \tag{4.1.1}$$

The reader can obtain an open-source implementation of this example from: https://github.com/yasumat/RobustPhotometricStereo. More detailed discussion will be covered in Chapter 14.

4.1.2 Recommendation Systems

In this example, imagine that we have n_2 products of interest, and n_1 users. Users consume products and rate them based on the quality of their experience. Our

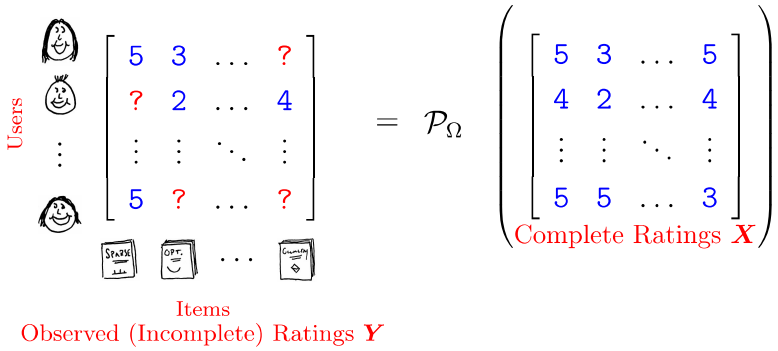


Figure 4.3 Collaborative Filtering as Low-rank Matrix Completion. Consider a universe of n_1 users and n_2 items. Users experience items, and then rate their experience. Our observation $\boldsymbol{Y}$ consists of those ratings that user have provided: Y_{ij} is user i's rating of item j. We wish to predict users ratings of items that they have not yet rated. This can be viewed as attempting to recover a large matrix $\boldsymbol{X}$ from a subset $\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{X}]$ of its entries.

goal is to use the information of all the users' ratings to predict which products will appeal to a given user. Formally, our object of interest is a large, unknown matrix

$$\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2},$$

whose (i, j) entry contains user i's degree of preference for item j. If we let

$$\Omega \doteq \Big\{(i, j) \mid \text{user } i \text{ has rated product } j\Big\},$$

then we observe

$$\underset{\text{Observed ratings}}{\boldsymbol{Y}} = \mathcal{P}_\Omega\left[\underset{\text{Complete ratings}}{\boldsymbol{X}}\right].$$

Here, $\mathcal{P}_\Omega$ is the projection operator onto the subset Ω:

$$\mathcal{P}_\Omega[\boldsymbol{X}](i, j) = \begin{cases} X_{ij} & (i, j) \in \Omega, \\ 0 & \text{else.} \end{cases}$$

See Figure 4.3 for a schematic representation of this scenario.

Our goal is to fill in the missing entries of $\boldsymbol{X}$. This problem is encountered in online recommendation systems – the most famous recent instance being the "Netflix Prize" competition conducted between 2006 and 2009. See the Wikipedia page: https://en.wikipedia.org/wiki/Netflix_Prize for details. Obviously, with no additional assumptions, the problem of filling in the missing entries of $\boldsymbol{X}$ is ill-posed. One popular assumption is that the ratings of distinct users (or distinct products) are correlated, and hence the target matrix $\boldsymbol{X}$ is low-rank, or approximately so. The relevant mathematical problem then becomes filling in

the missing entries of a low-rank matrix, or, somewhat equivalently, looking for the matrix $\boldsymbol{X}$ of minimum rank that is consistent with our given observations:

$$\begin{array}{ll} \text{min} & \text{rank}(\boldsymbol{X}), \\ \text{subject to} & \mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}. \end{array} \tag{4.1.2}$$

This problem is often referred to as *matrix completion* [CR09].

4.1.3 Euclidean Distance Matrix Embedding

This useful problem can be stated as follows: assume that we have n points $\boldsymbol{X} = [\boldsymbol{x}_1 \mid \cdots \mid \boldsymbol{x}_n]$ living in $\mathbb{R}^d$. We can define a matrix $\boldsymbol{D}$ via

$$D_{ij} = d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2.$$

Here $\boldsymbol{D}$ is known as a *Euclidean distance matrix*. Now imagine the following scenario: rather than observing the $\boldsymbol{x}_i$ themselves, we instead see their pairwise distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$. How can we tell if these distances were generated by some configuration of points living in $\mathbb{R}^d$? A necessary and sufficient condition is given by the following classical result:

THEOREM 4.1 (Schoenberg Theorem). *$\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is a Euclidean distance matrix for some set of n points in $\mathbb{R}^d$ if and only if the following conditions hold:*

- *$\boldsymbol{D}$ is symmetric.*
- *$D_{ii} = 0$ for all $i \in \{1, \ldots, n\}$.*
- *$\boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^* \preceq \boldsymbol{0}$, where $\boldsymbol{\Phi} = \boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^*$ is the* centering matrix *(here $\boldsymbol{1} \in \mathbb{R}^n$ is the vector whose entries are all ones).*
- *$\text{rank}\,(\boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^*) \leq d$.*

We leave the proof of this theorem as an exercise to the reader. See Exercise 4.1.

Now imagine we only know D_{ij} for some subset $\Omega \subset \{1, \ldots, n\} \times \{1, \ldots, n\}$, i.e., we observe $\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{D}]$. We can cast the problem of looking for a Euclidean distance matrix that agrees with our observations as a *rank minimization problem*:

$$\begin{array}{ll} \text{min} & \text{rank}\,(\boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^*)\,, \\ \text{subject to} & \boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^* \preceq \boldsymbol{0},\ \ \boldsymbol{D} = \boldsymbol{D}^*,\ \ \mathcal{P}_\Omega[\boldsymbol{D}] = \boldsymbol{Y},\ \ \forall i\, D_{ii} = 0. \end{array} \tag{4.1.3}$$

4.1.4 Latent Semantic Analysis

Low-dimensional models are very popular in document analysis. Consider an idealized problem in search or document retrieval. The system has access to n_2 documents (say, news articles), each of which is viewed as a collection of words in a dictionary of size n_1. For the j-th document, we compute a histogram of word

occurrences, giving an n_1-dimensional vector $\boldsymbol{y}_j$ whose i-th entry is the fraction of occurrences of word i in document j. Set

$$\underset{\text{Word occurrences}}{\boldsymbol{Y}} = \overset{\text{Words}}{} \Big[\boldsymbol{y}_1 \mid \cdots \mid \boldsymbol{y}_{n_2} \Big]. \quad \text{Documents}$$

We model these observations as follows. We imagine that there exists a set of "topics" $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r$. Each topic is a probability distribution on $\{1, 2, \ldots, n_1\}$. We may imagine that the $\boldsymbol{t}_l$ corresponds loosely to our informal notion of what a topic is – say, architecture or New York city. An article on architecture in New York would involve multiple topics. We model this as a mixture distribution, writing

$$\underset{\text{Word distribution for document } j}{\boldsymbol{p}_j} = \sum_{l=1}^{r} \underset{\text{topic}}{\boldsymbol{t}_l} \underset{\text{abundance}}{\alpha_{l,j}},$$

where $\alpha_{1,j} + \alpha_{2,j} + \cdots + \alpha_{r,j} = 1$. We imagine that $\boldsymbol{y}_j$ is generated by sampling words independently at random from the mixture distribution $\boldsymbol{p}_j$ and computing a histogram.[2] If the number of words sampled is large, we can imagine $\boldsymbol{y}_j \approx \boldsymbol{p}_j$. So, if we write $\boldsymbol{T} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_r]$ and $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$, then we have

$$\underset{\text{Word occurrences}}{\boldsymbol{Y}} \approx \underset{\text{Topics}}{\boldsymbol{T}} \underset{\text{Abundances}}{\boldsymbol{A}}. \tag{4.1.4}$$

Notice that $\operatorname{rank}(\boldsymbol{TA}) \leq r$: *the rank is bounded by the number of topics. Latent semantic analysis* computes a best low-rank approximation to $\boldsymbol{Y}$ and then uses it for search and indexing [DFL+88, DDF+90]. There are several advanced extensions to the basic latent semantic indexing (LSI) model, such as probabilistic LSI (pLSI) [Hof99, Hof04], Latent Dirichlet Allocation (LDA) [BNJ03], and a joint topic-document model (via low-rank and sparse matrix) [MZWM10].

Many additional examples arise, for example in solving positioning problems, problems in system identification, quantum state tomography, image and video alignment, etc. We will survey more of these in the coming application chapters.

4.2 Representing Low-Rank Matrix via SVD

In all of the applications described above, our goal is to recover an unknown $\boldsymbol{X}$ whose columns live on an r-dimensional linear subspace of the data space $\mathbb{R}^{n_1}$. This subspace can be characterized via the *singular value decomposition* (SVD) of $\boldsymbol{X}$ (see Appendix A.8 for a more detailed review):

THEOREM 4.2 (Compact SVD). *Let $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ be a matrix, and $r = \operatorname{rank}(\boldsymbol{X})$. Then there exist $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_r)$ with numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and*

[2] In practice, researchers have observed that more complicated methods of constructing $\boldsymbol{Y}$ (say, using the *TF-IDF* weighting) improves performance compared to just using the histogram.

matrices $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}$, $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r}$, *such that* $\boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{V}^*\boldsymbol{V} = \boldsymbol{I}$ *and*

$$\boldsymbol{X} \;=\; \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \;=\; \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*. \tag{4.2.1}$$

Exercise 4.2 gives a guided proof of this result. This construction turns out to be a very versatile tool both for theory and for numerical computation. The *full* singular value decomposition extends the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ to complete orthonormal bases for $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively, by adding bases for the left and right null spaces of $\boldsymbol{X}$:

THEOREM 4.3 (SVD). *Let* $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ *be a matrix. Then there exist orthogonal matrices* $\boldsymbol{U} \in \mathsf{O}(n_1)$ *and* $\boldsymbol{V} \in \mathsf{O}(n_2)$*, and numbers*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{n_1,n_2\}}$$

such that if we let $\boldsymbol{\Sigma} \in \mathbb{R}^{n_1 \times n_2}$ *with* $\Sigma_{ii} = \sigma_i$ *and* $\Sigma_{ij} = 0$ *for* $i \neq j$*,*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*. \tag{4.2.2}$$

FACT 4.4 (Properties of the SVD). *We note the following properties of the construction in Theorem 4.2:*

- *The left singular vectors* $\boldsymbol{u}_i$ *are the eigenvectors of* $\boldsymbol{X}\boldsymbol{X}^*$ *(check this!).*
- *The right singular vectors* $\boldsymbol{v}_i$ *are the eigenvectors of* $\boldsymbol{X}^*\boldsymbol{X}$*.*
- *The nonzero singular values* σ_i *are the positive square roots of the positive eigenvalues* λ_i *of* $\boldsymbol{X}^*\boldsymbol{X}$*.*
- *The nonzero singular values* σ_i *are also the positive square roots of the positive eigenvalues* λ_i *of* $\boldsymbol{X}\boldsymbol{X}^*$*.*

Notice that since $\boldsymbol{U}$ and $\boldsymbol{V}$ are nonsingular, the $\operatorname{rank}(\boldsymbol{X}) = \operatorname{rank}(\boldsymbol{\Sigma})$. Since $\boldsymbol{\Sigma}$ is diagonal, this quantity is especially simple – it is simply the number of nonzero entries σ_i! Here, and below, we will let $\boldsymbol{\sigma}(\boldsymbol{X}) = (\sigma_1, \ldots, \sigma_{\min\{n_1,n_2\}}) \in \mathbb{R}^{\min\{n_1,n_2\}}$ denote the vector of singular values of $\boldsymbol{X}$. Then, in the language that we've been developing thus far,

$$\operatorname{rank}(\boldsymbol{X}) = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_0 . \tag{4.2.3}$$

Hence any problem that minimizes the rank of an unknown matrix $\boldsymbol{X}$ is essentially minimizing the number of nonzero singular values of $\boldsymbol{X}$ – the "sparsity" of singular values, subject to data constraints.

4.2.1 Singular Vectors via Nonconvex Optimization

The SVD can be computed in time $O(\max\{n_1, n_2\} \min\{n_1, n_2\}^2)$. The first r singular value/vector triples can be computed in time $O(n_1 n_2 r)$. Hence, the problem of finding a linear subspace that best fits a given set of data can be solved in polynomial time. On the surface this is quite remarkable – the problem of

computing singular vectors is nonconvex. We briefly describe why this nonconvex problem can be solved globally in an efficient manner.

We give a brief indication of *why* it is possible to efficiently compute singular vectors of a matrix $\boldsymbol{X}$. Consider the matrix $\boldsymbol{\Gamma} \doteq \boldsymbol{X}\boldsymbol{X}^*$. Let $\boldsymbol{\Gamma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^*$ be the eigenvalue decomposition of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_1})$ be the eigenvalues. It is obvious that the left singular vectors $\boldsymbol{u}_i$ of $\boldsymbol{X}$ are the eigenvectors of $\boldsymbol{\Gamma}$. Because our goal in this paragraph is merely to convey intuition, we make the simplifying assumption that $\boldsymbol{\Gamma}$ has no repeated eigenvalues and λ_1 is the largest. We show how to use nonconvex optimization to compute the leading eigenvector $\boldsymbol{u}_1$ – see Exercise 4.5 for extensions to repeated leading eigenvectors.

Consider the optimization problem

$$\begin{array}{ll} \min & \varphi(\boldsymbol{q}) \equiv -\frac{1}{2}\boldsymbol{q}^*\boldsymbol{\Gamma}\boldsymbol{q}, \\ \text{subject to} & \|\boldsymbol{q}\|_2^2 = 1. \end{array} \tag{4.2.4}$$

The gradient and Hessian of the function $\varphi(\boldsymbol{q})$ are

$$\nabla\varphi(\boldsymbol{q}) = -\boldsymbol{\Gamma}\boldsymbol{q} \quad \text{and} \quad \nabla^2\varphi(\boldsymbol{q}) = -\boldsymbol{\Gamma}, \tag{4.2.5}$$

respectively. A point $\boldsymbol{q}$ is a *critical point* of the function φ over the sphere:

$$\mathbb{S}^{n-1} = \left\{ \boldsymbol{q} \mid \|\boldsymbol{q}\|_2^2 = 1 \right\}$$

if there is no direction $\boldsymbol{v} \perp \boldsymbol{q}$ (i.e., no direction that is tangent to the sphere at $\boldsymbol{q}$) along which the function decreases. Equivalently, $\boldsymbol{q}$ is a critical point of φ over the sphere if and only if the gradient is proportional to $\boldsymbol{q}$:

$$\nabla\varphi(\boldsymbol{q}) \propto \boldsymbol{q}. \tag{4.2.6}$$

Using our expression for $\nabla\varphi$, this is true if and only if $\boldsymbol{\Gamma}\boldsymbol{q} = \lambda\boldsymbol{q}$ for some λ: *The critical points of φ over $\mathbb{S}^{n-1}$ are precisely the eigenvectors $\pm\boldsymbol{u}_i$ of $\boldsymbol{\Gamma}$.*

Which critical points $\pm\boldsymbol{u}_i$ are actual local or global minimizers (instead of saddle points)? To answer this question, we need to study the curvature of the function $\varphi(\boldsymbol{q})$ around a critical point $\bar{\boldsymbol{q}}$. In Euclidean space, the correct tool for studying curvature is the Hessian, as is justified by the second order Taylor expansion of the function along the curve $\boldsymbol{x}(t) = \boldsymbol{x} + t\boldsymbol{v}$:

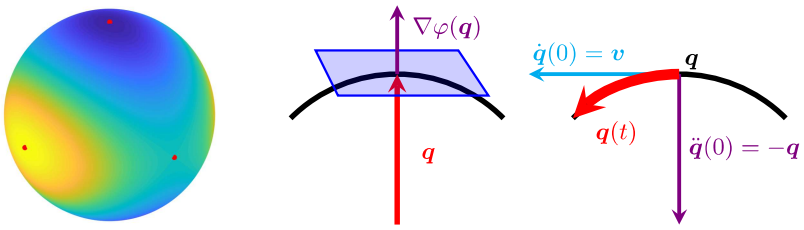
$$f(\boldsymbol{x} + t\boldsymbol{v}) = f(\boldsymbol{x}) + \underbrace{t\,\langle \nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle}_{=0 \text{ at any critical point}} + \frac{1}{2}t^2\boldsymbol{v}^*\nabla^2 f(\boldsymbol{x})\boldsymbol{v} + o(t^2).$$

In Euclidean space, a critical point $\bar{\boldsymbol{x}}$ is a local minimizer if $\nabla^2 f(\bar{\boldsymbol{x}}) \succ \boldsymbol{0}$. Conversely, if $\nabla^2 f(\bar{\boldsymbol{x}})$ has a negative eigenvalue, the point is not a local minimizer.

Over the sphere, we can perform a similar Taylor expansion, but we need to replace the straight line $\boldsymbol{x}(t) = \boldsymbol{x} + t\boldsymbol{v}$ with a great circle[3]

$$\boldsymbol{q}(t) = \boldsymbol{q}\cos(t) + \boldsymbol{v}\sin(t), \tag{4.2.7}$$

[3] Curves of this form are *geodesics* on $\mathbb{S}^{n-1}$.



Objective: $\varphi(\boldsymbol{q}) = -\frac{1}{2}\boldsymbol{q}^*\boldsymbol{\Gamma}\boldsymbol{q}$

Critical point: $\nabla\varphi(\boldsymbol{q}) = \lambda\boldsymbol{q}$

Curvature:

$$\begin{aligned}&\tfrac{d^2}{dt^2}\varphi(\boldsymbol{q}(t))\big|_{t=0}\\ =\ &\boldsymbol{v}^*\nabla^2\varphi(\boldsymbol{q})\boldsymbol{v}\\ +\ &\langle\nabla\varphi(\boldsymbol{q}), -\boldsymbol{q}\rangle\,\|\boldsymbol{v}\|_2^2.\end{aligned}$$

Figure 4.4 Eigenvector Computation as Nonconvex Optimization over the Sphere. We plot $\varphi(\boldsymbol{q}) = -\frac{1}{2}\boldsymbol{q}^*\boldsymbol{\Gamma}\boldsymbol{q}$ over the sphere, for one particular $\boldsymbol{\Gamma}$. Red dots represent the eigenvectors of $\boldsymbol{\Gamma}$. Critical points (middle) are points $\boldsymbol{q}$ for which $\nabla\varphi(\boldsymbol{q})$ is proportional to $\boldsymbol{q}$. Every critical point is an eigenvector of $\boldsymbol{\Gamma}$; the only local minimizers are eigenvectors that correspond to the largest eigenvalue $\lambda_1(\boldsymbol{\Gamma})$. Right: curvature of the φ over the sphere comes from both curvature $\nabla^2\varphi$ of φ and the curvature of the sphere.

where $\boldsymbol{v} \perp \boldsymbol{q}$ and $\|\boldsymbol{v}\|_2 = 1$. Calculus shows that the second directional derivative of $\varphi(\boldsymbol{q}(t))$ is given by

$$\frac{d^2}{dt^2}\varphi(\boldsymbol{q}(t))\Big|_{t=0} = \underbrace{\boldsymbol{v}^*\nabla^2\varphi(\boldsymbol{q})\boldsymbol{v}}_{\text{Curvature of } \varphi} - \underbrace{\langle\nabla\varphi(\boldsymbol{q}), \boldsymbol{q}\rangle\,\boldsymbol{v}^*\boldsymbol{v}}_{\text{Curvature of the sphere}}. \tag{4.2.8}$$

This formula contains two terms, which combine the usual Hessian of φ (accounting for the curvature of φ) and a second correction term involving $-\langle\nabla\varphi(\boldsymbol{q}), \boldsymbol{q}\rangle$ which accounts for the fact that the curve $\boldsymbol{q}(t)$ curves in the $-\boldsymbol{q}$ direction in order to stay on the sphere.

Noting that $\nabla^2\varphi(\boldsymbol{q}) = -\boldsymbol{\Gamma}$. So we have $\langle\nabla\varphi(\boldsymbol{u}_i), \boldsymbol{u}_i\rangle = -\boldsymbol{u}_i^*\boldsymbol{\Gamma}\boldsymbol{u}_i = -\lambda_i$, we observe that at a critical point $\bar{\boldsymbol{q}} = \pm\boldsymbol{u}_i$, the second derivative in the $\boldsymbol{v}$ direction is

$$\frac{d^2}{dt^2}\varphi(\boldsymbol{q}(t))\Big|_{t=0} = \boldsymbol{v}^*\Big(-\boldsymbol{\Gamma} + \lambda_i\boldsymbol{I}\Big)\boldsymbol{v}. \tag{4.2.9}$$

The eigenvalues of the operator $-\boldsymbol{\Gamma} + \lambda_i$ take the form $-\lambda_j + \lambda_i$; there is a strictly negative eigenvalue if $\boldsymbol{u}_i$ is an eigenvector that does not correspond to the largest eigenvalue λ_1. So $\pm\boldsymbol{u}_1$ are the only local minimizers of φ. All other critical points have a direction of strict negative curvature. This benign geometry implies that a simple projected gradient method converges to a global optimizer from almost any initialization. This phenomenon turns out to be rather representative of optimization problems associated with learning low-dimensional models for high-dimensional data, as we will return more formally to study them in Chapter 7.

For computing leading eigenvalue and eigenvector, we can do more than employing the generic gradient descent. Exercise 4.6 gives a more specific algorithm,

the *power iteration* method, which is much faster and more commonly used. In Section 9.3.2 of Chapter 9, we will give a precise characterization of the computational complexity of this method as well as its more efficient variant.[4] For now, we take these observations as an intuitive indication of why the SVD is amenable to efficient computation.

Implications and History.
Whichever rationale we adopt, the fact that the SVD can be both optimal (in a precisely defined and often quite relevant sense) and efficient (at least for moderate problems) makes it a very useful element in the numerical computing toolbox. The canonical example application of the SVD is *Principal Component Analysis* (PCA). Outlined in 1901 and 1933 papers by Pearson and Hotelling [Pea01, Hot33], respectively, PCA finds a best-fitting low-dimensional subspace, which can be computed via the SVD, as suggested by Theorem 4.5 below. Remarkably, Pearson's 1901 paper asserts that PCA is "well-suited to numerical computation" – meaning hand calculations!

4.2.2 Best Low-Rank Matrix Approximation

We are interested in recovering a low-rank matrix that is consistent with certain linear observations. Because the rank has a similar characteristics to the ℓ^0 norm, one should expect that these problems would be computationally intractable in general, as in the case with recovering a sparse solution (see Theorem 2.8).

Remarkably, there *are* however a few special instances of rank minimization that we *can* solve efficiently, with virtually no assumptions on the input. The most important is the *best rank-r approximation* problem, in which we try to approximate an arbitrary input matrix $\boldsymbol{Y}$ with a matrix $\boldsymbol{X}$ of rank at most r such that the approximation error $\|\boldsymbol{X} - \boldsymbol{Y}\|_F$ is as small as possible. The optimal solution to this problem can be obtained by simply retaining the first r leading singular values/vectors of $\boldsymbol{Y}$:

THEOREM 4.5 (Best Low-rank Approximation). *Let $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$, and consider the following optimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{X} - \boldsymbol{Y}\|_F, \\ \text{subject to} & \operatorname{rank}(\boldsymbol{X}) \leq r. \end{array} \tag{4.2.10}$$

Every optimal solution $\hat{\boldsymbol{X}}$ to the above problem has the form $\hat{\boldsymbol{X}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$, where $\boldsymbol{Y} = \sum_{i=1}^{\min(n_1,n_2)} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$ is a (full) singular value decomposition of $\boldsymbol{Y}$.

In fact, the same solution (truncating the SVD) also solves the low-rank approximation problem when the error is measured in the operator norm, or any other orthogonal-invariant matrix norm (see Appendix A). Please see Exercise 4.3 for guidance on how to prove Theorem 4.5.

[4] The Lanczos method for computing the leading eigenvalue and eigenvector.

The problem (4.2.10) can be turned around and cast as one of minimizing the rank of the unknown matrix, subject to a data fidelity constraint:

$$\begin{array}{ll} \text{min} & \text{rank}(\boldsymbol{X}), \\ \text{subject to} & \|\boldsymbol{X} - \boldsymbol{Y}\|_F \leq \varepsilon. \end{array} \tag{4.2.11}$$

This is an example of a *matrix rank minimization* problem – we seek a matrix of minimum rank that is consistent with some given observations. Because of its very special nature, this particular rank minimization can be solved optimally via the SVD. We leave the solution to this problem as an exercise to the reader (see Exercise 4.4).[5]

4.3 Recovering a Low-Rank Matrix

4.3.1 General Rank Minimization Problems

In the previous section, we saw that for certain very specific rank minimization problems, globally optimal solutions could be obtained using efficient algorithms based on the singular value decomposition. However, all of the applications discussed above (and many others!) force us to attempt to minimize the rank of $\boldsymbol{X}$ over much more complicated sets. One model example problem is the *affine rank minimization* problem [FHB04]:

$$\begin{array}{ll} \text{min} & \text{rank}(\boldsymbol{X}), \\ \text{subject to} & \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \end{array} \tag{4.3.1}$$

Here $\boldsymbol{y} \in \mathbb{R}^m$ is an observation, and $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ is a linear map. When $m \ll n_1 n_2$, the linear system of equations $\mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}$ is underdetermined. The notion of a linear map $\mathcal{A}$ from $n_1 \times n_2$ matrices to m-dimensional vectors may seem somewhat abstract. Any linear map of this form can be represented using the matrix inner product[6]:

$$\mathcal{A}[\boldsymbol{X}] = (\langle \boldsymbol{A}_1, \boldsymbol{X} \rangle, \ldots, \langle \boldsymbol{A}_m, \boldsymbol{X} \rangle). \tag{4.3.2}$$

Here, the set of matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m \in \mathbb{R}^{n_1 \times n_2}$ define our "measurements" $\boldsymbol{y}$, through their inner products with the unknown matrix $\boldsymbol{X}$.[7]

A mathematically simple and natural assumption on these "measurement" matrices is that they are i.i.d. Gaussian matrices. Such an assumption will allow us to understand the conditions under which one could expect to recover a low-rank matrix with generic measurements. Hence, our first attempt to understand the low-rank recovery problem will rely on such a simplifying assumption. However, in many practical problems of interest, the operator $\mathcal{A}$ has particular

[5] Hint: you may first try to guess what the optimal solution is and then show its optimality.

[6] Recall that the standard inner product between matrices $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{R}^{n_1 \times n_2}$ is defined by $\langle \boldsymbol{P}, \boldsymbol{Q} \rangle = \sum_{ij} P_{ij} Q_{ij} = \text{trace}[\boldsymbol{Q}^* \boldsymbol{P}]$.

[7] You can think of the measurements $\boldsymbol{A}_i$ as analogous to the *rows* $\boldsymbol{a}_i^*$ of the matrix $\boldsymbol{A}$ in the equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ studied in Chapters 2-3.

structures that make it behave differently. As a concrete example, in the matrix completion problems discussed above, we would have $m = |\Omega|$, and $\boldsymbol{A}_l = \boldsymbol{e}_{i_l}\boldsymbol{e}_{j_l}^*$, with $\Omega = \{(i_1, j_1), \ldots, (i_m, j_m)\}$. We will also thoroughly analyze this important special case and provide conditions under which the recovery can be successful.

Connection to ℓ^0, NP-hardness.
To make the connection to sparse recovery explicit, using the observation that $\operatorname{rank}(\boldsymbol{X}) = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_0$, we can rewrite the affine rank minimization problem as

$$\begin{array}{ll} \min & \|\boldsymbol{\sigma}(\boldsymbol{X})\|_0 \\ \text{subject to} & \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \end{array} \tag{4.3.3}$$

Moreover, *if* $\boldsymbol{X}$ is a diagonal matrix, then $\operatorname{rank}(\boldsymbol{X}) = \|\boldsymbol{X}\|_0$. So, every ℓ^0 minimization problem can be converted into a rank minimization problem with a diagonal constraint. This means that in the worst case, the rank minimization problem is at least as hard as the ℓ^0 minimization problem: it is NP-hard (as shown in Theorem 2.8).

As was the case for ℓ^0 minimization, we could simply give up here in searching for tractable algorithms. However, given the close analogy between rank minimization and ℓ^0 minimization, we might hope that there could be some fairly broad subclass of "nice enough" instances that we *can* solve efficiently.

4.3.2 Convex Relaxation of Rank Minimization

The close analogy to ℓ^0 minimization suggests a natural strategy: replace the rank, which is the ℓ^0 norm $\boldsymbol{\sigma}(\boldsymbol{X})$ with the ℓ^1 norm of $\boldsymbol{\sigma}(\boldsymbol{X})$:

$$\|\boldsymbol{\sigma}(\boldsymbol{X})\|_1 = \sum_i \sigma_i(\boldsymbol{X}). \tag{4.3.4}$$

We call this function the *nuclear norm* of $\boldsymbol{X}$, and reserve the special notation

$$\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X}). \tag{4.3.5}$$

When $\boldsymbol{X}$ is a symmetric positive semidefinite matrix, $\boldsymbol{X}$ has real nonnegative eigenvalues, and $\sigma_i(\boldsymbol{X}) = \lambda_i(\boldsymbol{X})$. Since $\sum_i \lambda_i(\boldsymbol{X}) = \operatorname{trace}(\boldsymbol{X})$, in the special case when $\boldsymbol{X}$ is semidefinite, $\|\boldsymbol{X}\|_* = \operatorname{trace}[\boldsymbol{X}]$. For this reason, the nuclear norm is sometimes also referred to as the *trace norm*. Other names in various literatures include the *Schatten 1-norm* and *Ky-Fan k-norm*.[8]

When $\boldsymbol{X}$ is not a semidefinite matrix, the function $\|\boldsymbol{X}\|_*$ depends on the entries in a very complicated way. The results below give a couple of equivalent characterizations of the nuclear norm, which will be useful later in this book when we deal with certain nonconvex formulation of rank minimization (in Chapter 7).

[8] For $p \in [1, \infty]$, the Schatten p-norm of a matrix is $\|\boldsymbol{X}\|_{S_p} = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_p$. The Ky-Fan k-norm is $\|\boldsymbol{X}\|_{KF_k} = \sum_{i=1}^k \sigma_i(\boldsymbol{X})$. Both of these functions are examples of *orthogonal invariant matrix norms*, see Appendix A.9 for more details.

PROPOSITION 4.6 (Variational Forms of Nuclear Norm). *The nuclear norm of a matrix* $\|\boldsymbol{X}\|_*$ *is equivalent to the following variational forms:*

1 $\|\boldsymbol{X}\|_* = \min_{\boldsymbol{U},\boldsymbol{V}} \frac{1}{2}(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2),\ s.t.\ \boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^*.$
2 $\|\boldsymbol{X}\|_* = \min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{U}\|_F\|\boldsymbol{V}\|_F,\ s.t.\ \boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^*.$
3 $\|\boldsymbol{X}\|_* = \min_{\boldsymbol{U},\boldsymbol{V}} \sum_k \|\boldsymbol{u}_k\|_2\|\boldsymbol{v}_k\|_2,\ s.t.\ \boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^* \doteq \sum_k \boldsymbol{u}_k\boldsymbol{v}_k^*.$

This proposition can be proved by showing that the global minimum of each of these problems is reached when $\boldsymbol{U}_\star = \boldsymbol{U}_o\sqrt{\boldsymbol{\Sigma}_o}$ and $\boldsymbol{V}_\star = \boldsymbol{V}_o\sqrt{\boldsymbol{\Sigma}_o}$, where $\boldsymbol{X} = \boldsymbol{U}_o\boldsymbol{\Sigma}_o\boldsymbol{V}_o^*$ is any singular value decomposition of $\boldsymbol{X}$. This can be readily shown, by noting that each of the objective functions is invariant to orthogonal transformations, and reducing to the case when $\boldsymbol{X}$ is a diagonal matrix, and carefully examining this special case. We leave the details as an exercise for the reader.

Notice that in the above variational forms, there is *no* restriction on the dimensions of the two factors $\boldsymbol{U}, \boldsymbol{V}$ as long as the equality $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^*$ holds. Hence choosing $\boldsymbol{U}, \boldsymbol{V}$ to be matrices of larger sizes does not affect the minimization. These forms will become very useful when we consider alternative ways to minimize the nuclear norm for promoting low-rank property, as we will examine later in Chapter 7.

Despite the above characterization, it remains not obvious at all that the sum of singular values is a norm, or even is indeed a convex function of the matrix. To allay any suspicion, we give a quick proof that $\|\cdot\|_*$ is indeed a norm:

THEOREM 4.7. *For* $\boldsymbol{M} \in \mathbb{R}^{n_1\times n_2}$, *let* $\|\boldsymbol{M}\|_* = \sum_{i=1}^{\min\{n_1,n_2\}} \sigma_i(\boldsymbol{M})$. *Then* $\|\cdot\|_*$ *is a norm. Moreover, the nuclear norm and the* ℓ^2 *operator norm (or the spectral norm) are* dual norms*:*

$$\|\boldsymbol{M}\|_* = \sup_{\|\boldsymbol{N}\|\le 1} \langle \boldsymbol{M}, \boldsymbol{N}\rangle, \quad \textit{and} \quad \|\boldsymbol{M}\| = \sup_{\|\boldsymbol{N}\|_*\le 1} \langle \boldsymbol{M}, \boldsymbol{N}\rangle. \tag{4.3.6}$$

Proof We begin by proving the first equality in (4.3.6). Let

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \tag{4.3.7}$$

be a full singular value decomposition of $\boldsymbol{M}$, with $\boldsymbol{U} \in \mathsf{O}(n_1)$, $\boldsymbol{V} \in \mathsf{O}(n_2)$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{n_1\times n_2}$, and note that

$$\begin{aligned}
\sup_{\|\boldsymbol{N}\|\le 1} \langle \boldsymbol{N}, \boldsymbol{M}\rangle &= \sup_{\|\boldsymbol{N}\|\le 1} \langle \boldsymbol{N}, \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*\rangle \\
&= \sup_{\|\boldsymbol{N}\|\le 1} \left\langle \boldsymbol{U}^*\boldsymbol{N}\boldsymbol{V}, \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_{n_2} \\ 0 & 0 & 0 \\ & \vdots & \end{bmatrix} \right\rangle \\
&\ge \sum_{i=1}^{n_2} \sigma_i,
\end{aligned} \tag{4.3.8}$$

where the last line follows by making a particular choice

$$\boldsymbol{N} = \boldsymbol{U} \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ 0 & 0 & 0 & \\ & \vdots & & \end{bmatrix} \boldsymbol{V}^*. \tag{4.3.9}$$

So, $\sup_{\boldsymbol{N}} \langle \boldsymbol{N}, \boldsymbol{M} \rangle \geq \|\boldsymbol{M}\|_*$.

For the opposite direction, notice if matrix $\boldsymbol{N} \in \mathbb{R}^{n_1 \times n_2}$ satisfies $\|\boldsymbol{N}\| \leq 1$, then $\bar{\boldsymbol{N}} \doteq \boldsymbol{U}^* \boldsymbol{N} \boldsymbol{V}$ has columns of ℓ^2 norm at most one. Thus, for each i, $\bar{N}_{ii} \leq 1$, and

$$\langle \boldsymbol{N}, \boldsymbol{M} \rangle = \langle \bar{\boldsymbol{N}}, \boldsymbol{\Sigma} \rangle = \sum_{i=1}^{n_2} \bar{N}_{ii} \sigma_i \;\leq\; \sum_i \sigma_i = \|\boldsymbol{M}\|_* . \tag{4.3.10}$$

This establishes the result.

For the second equality in (4.3.6), notice that for any nonzero $\boldsymbol{M}$,

$$\langle \boldsymbol{M}, \boldsymbol{N} \rangle = \|\boldsymbol{M}\| \left\langle \frac{\boldsymbol{M}}{\|\boldsymbol{M}\|}, \boldsymbol{N} \right\rangle \quad \leq \quad \|\boldsymbol{M}\| \, \|\boldsymbol{N}\|_* . \tag{4.3.11}$$

Hence, $\sup_{\|\boldsymbol{N}\|_* \leq 1} \langle \boldsymbol{M}, \boldsymbol{N} \rangle \leq \|\boldsymbol{M}\|$. To show that this inequality is actually an equality, let's take $\boldsymbol{N} = \boldsymbol{u}_1 \boldsymbol{v}_1^*$, and notice that $\|\boldsymbol{N}\|_* = 1$ and $\langle \boldsymbol{M}, \boldsymbol{N} \rangle = \boldsymbol{u}_1^* \boldsymbol{M} \boldsymbol{v}_1 = \sigma_1(\boldsymbol{M}) = \|\boldsymbol{M}\|$. This completes the proof of (4.3.6).

To see that $\|\cdot\|_*$ is indeed a norm, we just use (4.3.6) to verify that the three axioms of a norm are satisfied. Since the singular values are nonnegative, and $\sigma_1(\boldsymbol{M}) = 0$ if and only if $\boldsymbol{M} = \boldsymbol{0}$, it is immediate that $\|\boldsymbol{M}\|_* \geq 0$ with equality iff $\boldsymbol{M} = \boldsymbol{0}$. For nonnegative homogeneity, notice that for $t \in \mathbb{R}_+$,

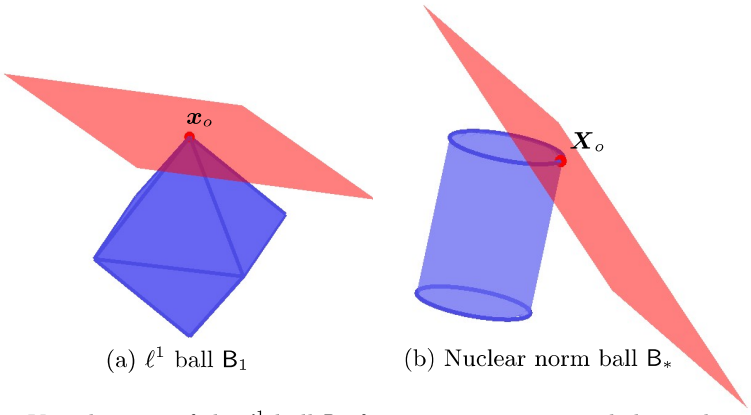
$$\|t\boldsymbol{M}\|_* = \sup_{\|\boldsymbol{N}\| \leq 1} \langle t\boldsymbol{M}, \boldsymbol{N} \rangle = t \sup_{\|\boldsymbol{N}\| \leq 1} \langle \boldsymbol{M}, \boldsymbol{N} \rangle = t \|\boldsymbol{M}\|_* . \tag{4.3.12}$$

Finally, for the triangle inequality, consider two matrices $\boldsymbol{M}$ and $\boldsymbol{M}'$, and notice that

$$\begin{aligned} \left\|\boldsymbol{M} + \boldsymbol{M}'\right\|_* &= \sup_{\|\tilde{\boldsymbol{N}}\| \leq 1} \left\langle \boldsymbol{M} + \boldsymbol{M}', \tilde{\boldsymbol{N}} \right\rangle \\ &\leq \sup_{\|\boldsymbol{N}\| \leq 1} \langle \boldsymbol{M}, \boldsymbol{N} \rangle \quad + \quad \sup_{\|\boldsymbol{N}'\| \leq 1} \left\langle \boldsymbol{M}', \boldsymbol{N}' \right\rangle \\ &= \|\boldsymbol{M}\|_* + \left\|\boldsymbol{M}'\right\|_* , \end{aligned} \tag{4.3.13}$$

verifying the triangle inequality. This shows that $\|\cdot\|_*$ is indeed a norm. □

The above proof highlights a useful fact about $\|\cdot\|_*$: it is the dual norm of the operator norm $\|\boldsymbol{X}\| = \sigma_1(\boldsymbol{X})$. The fact that $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$ explains the $*$ notation – this symbol is often used for duality.



(a) ℓ^1 ball B_1 (b) Nuclear norm ball B_*

Figure 4.5 Visualization of the ℓ^1 ball B_1 for sparse vectors $\boldsymbol{x}$ and the nuclear norm ball B_* for symmetry 2×2 matrices. The red affine subspace represents the solution space to the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x}_o$ for vectors (left) and the equation $\mathcal{A}[\boldsymbol{X}] = \mathcal{A}[\boldsymbol{X}_o]$ for matrices (right). The target low-rank matrix $\boldsymbol{X}_o$ is the unique minimum nuclear norm solution to this equation if and only the only intersects B_* at $\boldsymbol{X}_o$.

Because $\|\cdot\|_*$ is a norm, it is convex. Hence, a natural convex replacement for the rank minimization problem is the *nuclear norm minimization* problem

$$\begin{array}{ll} \min & \|\boldsymbol{X}\|_*, \\ \text{subject to} & \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \end{array} \tag{4.3.14}$$

This problem is convex, and moreover is efficiently solvable. In Chapter 8, we will see how to use the special structure of this problem to give practical, efficient algorithms which work well at moderate scales.

Example 4.8 (Nuclear Norm Ball). *To visualize the nuclear norm, let us consider the set of* 2×2 *symmetric matrices, parameterized as*

$$\boldsymbol{M} = \begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{4.3.15}$$

We leave as an exercise for the reader to find out the conditions on the three coordinates $(x, y, z) \in \mathbb{R}^3$ *such that* $\|\boldsymbol{M}\|_* = 1$. *Let* $\mathsf{B}_* = \{\boldsymbol{M} \mid \|\boldsymbol{M}\|_* \leq 1\}$ *be the unit ball defined by the nuclear norm. If we visualize such points in* $\mathbb{R}^3$, *the nuclear norm ball looks like a cylinder shown in Figure 4.5. The two circles at both ends of the cylinder correspond to matrices of rank 1, which has a high chance to meet the affine subspace containing all solutions satisfying* $\mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}$.

4.3.3 Nuclear Norm as a Convex Envelope of Rank

From the analogy to ℓ^0/ℓ^1 minimization, we might guess that the nuclear norm is a good convex surrogate for the rank, over some appropriate set. Recall that

we have proved in Theorem 2.11 that the ℓ^1 norm was the convex envelope of the ℓ^0 norm over the ℓ^∞ ball. Since for a matrix $\boldsymbol{X}$, $\|\boldsymbol{\sigma}(\boldsymbol{X})\|_\infty = \sigma_1(\boldsymbol{X}) = \|\boldsymbol{X}\|$, you might guess the following relationship:

THEOREM 4.9. *$\|\boldsymbol{M}\|_*$ is the convex envelope of* $\operatorname{rank}(\boldsymbol{M})$ *over*

$$\mathsf{B}_{op} \doteq \{\boldsymbol{M} \mid \|\boldsymbol{M}\| \leq 1\}. \tag{4.3.16}$$

Proof We prove that any convex function $f(\cdot)$ which satisfies

$$f(\boldsymbol{M}) \leq \operatorname{rank}(\boldsymbol{M}) \tag{4.3.17}$$

for all $\boldsymbol{M} \in \mathsf{B}_{op}$, is dominated by the nuclear norm: $f(\boldsymbol{M}) \leq \|\boldsymbol{M}\|_*$.

Write the SVD $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. Notice that

$$\boldsymbol{\Sigma} \in \operatorname{conv}\left\{\operatorname{diag}(\boldsymbol{w}) \mid \boldsymbol{w} \in \{0,1\}^{\min\{n_1,n_2\}}\right\}, \tag{4.3.18}$$

and for any $\boldsymbol{w} \in \{0,1\}^{\min\{n_1,n_2\}}$,

$$\|\boldsymbol{U}\operatorname{diag}(\boldsymbol{w})\boldsymbol{V}^*\|_* = \sum_i w_i = \operatorname{rank}(\boldsymbol{U}\operatorname{diag}(\boldsymbol{w})\boldsymbol{V}^*). \tag{4.3.19}$$

Writing

$$\boldsymbol{\Sigma} = \sum_i \lambda_i \operatorname{diag}(\boldsymbol{w}_i) \tag{4.3.20}$$

with $\boldsymbol{w}_i \in \{0,1\}^{\min\{n_1,n_2\}}$ with $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$, and applying Jensen's inequality, we obtain

$$\begin{aligned}
f(\boldsymbol{M}) &= f\left(\boldsymbol{U}\sum_i \lambda_i \operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{V}^*\right) && (4.3.21)\\
&\leq \sum_i \lambda_i f\left(\boldsymbol{U}\operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{V}^*\right) && (4.3.22)\\
&\leq \sum_i \lambda_i \operatorname{rank}\left(\boldsymbol{U}\operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{V}^*\right) && (4.3.23)\\
&= \sum_i \lambda_i \|\boldsymbol{w}_i\|_1 && (4.3.24)\\
&= \left\|\boldsymbol{U}\sum_i \lambda_i \operatorname{diag}(\boldsymbol{w}_i)\boldsymbol{V}^*\right\|_* && (4.3.25)\\
&= \|\boldsymbol{M}\|_* && (4.3.26)
\end{aligned}$$

as desired. □

Note that this proof essentially mirrored our argument for ℓ^1 and ℓ^∞. This is not a coincidence!

4.3.4 Success of Nuclear Norm under Rank-RIP

For now, assuming that we can solve nuclear norm minimization problems efficiently (say with algorithms given in Chapter 8), we turn our attention to whether nuclear norm minimization actually gives the correct answers. Namely, if we know that $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o]$, with $r = \operatorname{rank}(\boldsymbol{X}_o) \ll n$, is it true that $\boldsymbol{X}_o$ is the unique optimal solution to the nuclear norm minimization problem (4.3.14)? What we can say depends strongly on what we know about the operator $\mathcal{A}$.

By analogy to the *sparse* recovery problem, we can ask if it is enough for $\mathcal{A}$ to preserve the geometry of a small set of structured objects – here, the low-rank matrices. Formally, we can define a *rank-restricted isometry property*, under which for every rank-r $\boldsymbol{X}$, $\|\mathcal{A}[\boldsymbol{X}]\|_2 \approx \|\boldsymbol{X}\|_F$.

Definition 4.10 (Rank-Restricted Isometry Property [RFP10]). *The operator $\mathcal{A}$ has the rank-restricted isometry property of rank r with constant δ, if $\forall\, \boldsymbol{X}$'s that satisfy $\operatorname{rank}(\boldsymbol{X}) \le r$, we have*

$$(1-\delta)\|\boldsymbol{X}\|_F^2 \;\le\; \|\mathcal{A}[\boldsymbol{X}]\|_2^2 \;\le\; (1+\delta)\|\boldsymbol{X}\|_F^2. \tag{4.3.27}$$

The rank-r restricted isometry constant $\delta_r(\mathcal{A})$ is the smallest δ such that the above property holds.

As with the RIP for sparse vectors, the rank-RIP implies uniqueness of structured (low-rank) solutions:

Theorem 4.11. *If $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o]$, with $r = \operatorname{rank}(\boldsymbol{X}_o)$ and $\delta_{2r}(\mathcal{A}) < 1$, then $\boldsymbol{X}_o$ is the unique optimal solution to the rank minimization problem*

$$\begin{array}{ll} \min & \operatorname{rank}(\boldsymbol{X}) \\ \text{subject to} & \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \end{array} \tag{4.3.28}$$

We leave the proof of this claim as an exercise to the reader (see Exercise 4.14). The key property is the subadditivity of the matrix rank, namely,

$$\operatorname{rank}(\boldsymbol{X} + \boldsymbol{X}') \;\le\; \operatorname{rank}(\boldsymbol{X}) + \operatorname{rank}(\boldsymbol{X}'). \tag{4.3.29}$$

Moreover, like the RIP for sparse vectors, when the rank-RIP holds with sufficiently small constant δ, we can conclude that nuclear norm minimization will recover the desired low-rank solution:

Theorem 4.12 (Nuclear Norm Minimization [RFP10]). *Suppose that $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o]$ with $\operatorname{rank}(\boldsymbol{X}_o) \le r$, and that $\delta_{4r}(\mathcal{A}) \le \sqrt{2}-1$. Then $\boldsymbol{X}_o$ is the unique optimal solution to the nuclear norm minimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{X}\|_* \\ \text{subject to} & \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \end{array} \tag{4.3.30}$$

There is nothing special here about the numbers $4r$ and $\sqrt{2}-1$. The interesting part is the qualitative statement: if $\mathcal{A}$ respects the geometry of low-rank matrices in a sufficiently strong sense, then nuclear norm minimization succeeds. The

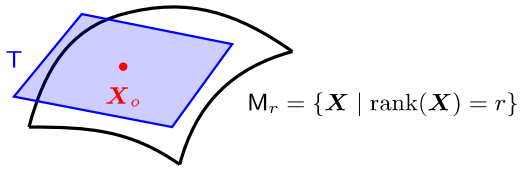


Figure 4.6 "Support" of a Low-rank Matrix X_o. Consider a rank-r matrix X_o with compact singular value decomposition $X_o = U\Sigma V^*$. The subspace $\mathsf{T} = \{UR^* + QV^*\}$ can be interpreted as the *tangent space* to the collection M_r of rank-r matrices at X_o.

proof is analogous to the proof we gave in the previous chapter for the success of ℓ^1 minimization for recovering sparse signals. However, to extend the proof techniques from ℓ^1 to nuclear norm, we need to generalize a few concepts from vectors to matrices.

"Support" and "Signs" of a Low-rank Matrix.
Let $X_o = U\Sigma V^*$ denote the compact SVD of the true solution X_o. Let

$$\mathsf{T} \doteq \left\{UR^* + QV^* \mid R \in \mathbb{R}^{n_2 \times r},\ Q \in \mathbb{R}^{n_1 \times r}\right\} \subseteq \mathbb{R}^{n_1 \times n_2}. \tag{4.3.31}$$

Notice that T is a linear subspace. In the analogy between ℓ^1 minimization and nuclear norm minimization, the subspace T plays the role of the "support" of X_o. Geometrically, T represents the *tangent space* to the set of rank-r matrices at X_o – see Figure 4.6 and Exercise 4.11. The subspace T is generated by matrices UR^* whose column space is contained in $\mathrm{col}(X_o)$ and matrices QV^* whose row space is contained in $\mathrm{row}(X_o)$. Notice that elements in T have rank no more than $2r$. Meanwhile the matrix UV^* plays the role of the "signs" of X_o since $UV^* \in \mathsf{T}$ and

$$\langle X_o, UV^* \rangle = \|X_o\|_* . \tag{4.3.32}$$

The orthogonal complement of T is

$$\mathsf{T}^\perp \doteq \{M \mid \mathrm{col}(M) \perp \mathrm{col}(X), \mathrm{row}(M) \perp \mathrm{row}(X)\} . \tag{4.3.33}$$

Let $P_U = UU^*$ and $P_V = VV^*$ be the orthogonal projections onto the column space and row space of X_o, respectively. Then the orthogonal projections onto these subspaces are given by[9]

$$\mathcal{P}_\mathsf{T}[M] = P_U M + M P_V - P_U M P_V, \tag{4.3.34}$$

and

$$\mathcal{P}_{\mathsf{T}^\perp}[M] = (I - P_U) M (I - P_V). \tag{4.3.35}$$

[9] Equations (4.3.34) and (4.3.35) can be derived from the condition that at $\mathcal{P}_\mathsf{T}[M]$, the error $M - \mathcal{P}_\mathsf{T}[M]$ is orthogonal to T.

Notice that because the orthogonal projections $\boldsymbol{P}_{U^\perp} = \boldsymbol{I} - \boldsymbol{P}_U$ and $\boldsymbol{P}_{V^\perp} = \boldsymbol{I} - \boldsymbol{P}_V$ have norm at most one, $\mathcal{P}_{\mathsf{T}^\perp}$ does not increase the operator norm:

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{M}]\| \leq \|\boldsymbol{M}\| . \tag{4.3.36}$$

Feasible Cone Restriction.
Note that any matrix $\boldsymbol{M} \in \mathsf{T}^\perp$ has columns that are orthogonal to the columns of $\boldsymbol{U}$ and rows that are orthogonal to the rows of $\boldsymbol{V}^*$. This implies that

$$\|\boldsymbol{M} + \boldsymbol{U}\boldsymbol{V}^*\| = \max\{\|\boldsymbol{M}\|, \|\boldsymbol{U}\boldsymbol{V}^*\|\} = \max\{\|\boldsymbol{M}\|, 1\} . \tag{4.3.37}$$

So, for any matrix $\boldsymbol{X}$,

$$\begin{aligned}
\|\boldsymbol{X}\|_* &= \sup_{\|\boldsymbol{Q}\| \leq 1} \langle \boldsymbol{X}, \boldsymbol{Q} \rangle && (4.3.38)\\
&\geq \sup_{\|\boldsymbol{M}\| \leq 1} \langle \boldsymbol{X}, \boldsymbol{U}\boldsymbol{V}^* + \mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{M}] \rangle && (4.3.39)\\
&= \langle \boldsymbol{X}, \boldsymbol{U}\boldsymbol{V}^* \rangle + \sup_{\|\boldsymbol{M}\| \leq 1} \langle \mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{X}], \boldsymbol{M} \rangle && (4.3.40)\\
&= \langle \boldsymbol{X}, \boldsymbol{U}\boldsymbol{V}^* \rangle + \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{X}]\|_* . && (4.3.41)
\end{aligned}$$

Let $\hat{\boldsymbol{X}}$ be any optimal solution to our problem (4.3.30). It can be written as $\hat{\boldsymbol{X}} = \boldsymbol{X}_o + \boldsymbol{H}$, with $\boldsymbol{H} = \hat{\boldsymbol{X}} - \boldsymbol{X}_o \in \mathrm{null}(\mathcal{A})$. From the above calculation, we have

$$\begin{aligned}
\|\boldsymbol{X}_o + \boldsymbol{H}\|_* &\geq \langle \boldsymbol{X}_o + \boldsymbol{H}, \boldsymbol{U}\boldsymbol{V}^* \rangle + \left\| \mathcal{P}_{\mathsf{T}^\perp}[\hat{\boldsymbol{X}}] \right\|_* && (4.3.42)\\
&= \|\boldsymbol{X}_o\|_* + \langle \boldsymbol{H}, \boldsymbol{U}\boldsymbol{V}^* \rangle + \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* && (4.3.43)\\
&\geq \|\boldsymbol{X}_o\|_* - \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_* + \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* . && (4.3.44)
\end{aligned}$$

So, if a better solution than $\boldsymbol{X}_o$ exists, the feasible perturbation $\boldsymbol{H}$ must satisfy the following cone restriction:

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* \leq \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_* . \tag{4.3.45}$$

Matrix Restricted Strong Convexity Property.
As in the proof of ℓ^1 success, we want to show that feasible perturbations $\boldsymbol{H} \in \mathrm{null}(\mathcal{A})$ must have $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* > \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_*$. This is true if the operator $\mathcal{A}$ satisfies the following (uniform) *matrix restricted strong convexity* property (RSC) property:

Definition 4.13 (Matrix Restricted Strong Convexity). *The linear operator $\mathcal{A}$ satisfies the matrix* restricted strong convexity *(RSC) condition of rank r with constant α if for the support T of every matrix of rank r and for all nonzero $\boldsymbol{H}$ satisfying*

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* \leq \alpha \cdot \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_* . \tag{4.3.46}$$

with some constant $\alpha \geq 1$, we have

$$\|\mathcal{A}[\boldsymbol{H}]\|_2^2 > \mu \cdot \|\boldsymbol{H}\|_F^2 \tag{4.3.47}$$

for some constant $\mu > 0$.

The following theorem says that if $\mathcal{A}$ satisfies the rank-RIP, then it satisfies the matrix RSC:

THEOREM 4.14 (Rank-RIP Implies Matrix RSC). *If a linear operator* $\mathcal{A}$ *satisfies rank-RIP with* $\delta_{4r} < \frac{1}{1+\alpha\sqrt{2}}$, *then* $\mathcal{A}$ *satisfies the matrix-RSC condition of rank* r *with constant* α.

Both the statement and proof of Theorem 4.14 parallel Theorem 3.17 for the ℓ^1 norm. Theorem 4.14 involves δ_{4r}, as opposed to δ_{2k} for k-sparse vectors. The bigger constant in $4r = r+3r$ reflects the need to account for all three components of the singular value decomposition in the proof:

Proof Using the parallelogram identity, similar to Lemma 3.16, it is not difficult to show that for any $\boldsymbol{Z}$, $\boldsymbol{Z}'$ such that $\boldsymbol{Z} \perp \boldsymbol{Z}'$, and $\operatorname{rank}(\boldsymbol{Z}) + \operatorname{rank}(\boldsymbol{Z}') \le 4r$,

$$\left|\left\langle \mathcal{A}[\boldsymbol{Z}], \mathcal{A}[\boldsymbol{Z}']\right\rangle\right| \le \delta_{4r}(\mathcal{A}) \left\|\boldsymbol{Z}\right\|_F \left\|\boldsymbol{Z}'\right\|_F . \tag{4.3.48}$$

Let T denote the support subspace for some matrix of rank r. Take any $\boldsymbol{H}$ that satisfies the cone restriction $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{Z}]\|_* \le \alpha \cdot \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{Z}]\|_*$, and write

$$\boldsymbol{H} = \mathcal{P}_{\mathsf{T}}[\boldsymbol{H}] + \mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]. \tag{4.3.49}$$

Let $\boldsymbol{H}_{\mathsf{T}}$ denote $\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]$. For the second term, $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]$, write its compact singular value decomposition

$$\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] = \sum_i \eta_i \boldsymbol{\phi}_i \boldsymbol{\zeta}_i^*, \tag{4.3.50}$$

where $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots$ are the left singular vectors, $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \dots$ the right singular vectors, and $\eta_1 \ge \eta_2 \ge \cdots > 0$ the singular values. From the variational characterization of the singular vectors, each $\boldsymbol{\phi}_i$ is orthogonal to the columns of $\boldsymbol{U}$, and each $\boldsymbol{\zeta}_i$ is orthogonal to the columns of $\boldsymbol{V}$. So, if we partition $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]$ as

$$\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] \;=\; \underbrace{\sum_{i=1}^{r} \eta_i \boldsymbol{\phi}_i \boldsymbol{\zeta}_i^*}_{\doteq \boldsymbol{\Phi}_1} + \underbrace{\sum_{i=r+1}^{2r} \eta_i \boldsymbol{\phi}_i \boldsymbol{\zeta}_i^*}_{\doteq \boldsymbol{\Phi}_2} + \cdots , \tag{4.3.51}$$

we have $\boldsymbol{\Phi}_i \perp \boldsymbol{\Phi}_j$ for $i \ne j$, $\boldsymbol{\Phi}_i \perp \boldsymbol{H}_{\mathsf{T}}$ for every T.

Since the singular values η_i are non-increasing, the largest singular value of the $(i+1)$-th block is bounded by the average of the singular values in the i-th block.

$$\forall i \ge 1, \quad \|\boldsymbol{\Phi}_{i+1}\| \le \frac{\|\boldsymbol{\Phi}_i\|_*}{r}. \tag{4.3.52}$$

So, noting that as an element in T, we have $\operatorname{rank}(\boldsymbol{H}_{\mathsf{T}}) \le 2r$ and so $\operatorname{rank}(\boldsymbol{H}_{\mathsf{T}} + \boldsymbol{\Phi}_1) \le 3r$. Notice that

$$\mathcal{A}[\boldsymbol{H}_{\mathsf{T}}] + \mathcal{A}[\boldsymbol{\Phi}_1] = \mathcal{A}[\boldsymbol{H}] - \mathcal{A}[\boldsymbol{\Phi}_2] - \mathcal{A}[\boldsymbol{\Phi}_3] - \cdots . \tag{4.3.53}$$

Then, very similar to the derivation of inequalities (3.3.32) in Theorem 3.17, and by applying the rank-RIP to matrices of rank bounded by at most $4r$, we have

$$\begin{aligned}
&(1-\delta_{4r})\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F^2 \\
&\le \langle \mathcal{A}[\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1], \mathcal{A}[\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1]\rangle \\
&= \langle \mathcal{A}[\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1], \mathcal{A}[\boldsymbol{H}]-\mathcal{A}[\boldsymbol{\Phi}_2]-\mathcal{A}[\boldsymbol{\Phi}_3]-\cdots\rangle \\
&\le \sum_{j\ge 2} |\langle \mathcal{A}[\boldsymbol{H}_{\mathsf{T}}], \mathcal{A}[\boldsymbol{\Phi}_j]\rangle| + |\langle \mathcal{A}[\boldsymbol{\Phi}_1], \mathcal{A}[\boldsymbol{\Phi}_j]\rangle| + \langle \mathcal{A}[\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1], \mathcal{A}[\boldsymbol{H}]\rangle \\
&\le \delta_{4r}\,(\|\boldsymbol{H}_{\mathsf{T}}\|_F + \|\boldsymbol{\Phi}_1\|_F)\sum_{j\ge 2}\|\boldsymbol{\Phi}_j\|_F + \|\mathcal{A}[\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1]\|_2\,\|\mathcal{A}[\boldsymbol{H}]\|_2 \\
&\le \delta_{4r}\sqrt{2}\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\sum_{j\ge 2}\|\boldsymbol{\Phi}_j\|_F + (1+\delta_{4r})\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\,\|\mathcal{A}[\boldsymbol{H}]\|_2 \\
&\le \delta_{4r}\sqrt{2}\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\,\frac{\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*}{\sqrt{r}} + (1+\delta_{4r})\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\,\|\mathcal{A}[\boldsymbol{H}]\|_2\,.
\end{aligned}$$

Note that $\boldsymbol{H}$ is restricted by the cone condition (4.3.46), which leads to:

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* \le \alpha\,\|\boldsymbol{H}_{\mathsf{T}}\|_* \le \alpha\sqrt{r}\,\|\boldsymbol{H}_{\mathsf{T}}\|_F \le \alpha\sqrt{r}\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\,. \tag{4.3.54}$$

Combining this with the previous inequality, we obtain:

$$\|\mathcal{A}[\boldsymbol{H}]\|_2 \ge \frac{1-\delta_{4r}(1+\alpha\sqrt{2})}{1+\delta_{4r}}\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F\,. \tag{4.3.55}$$

Since the singular values η_i are non-increasing, the ith singular value in $\boldsymbol{\Phi}_2+\boldsymbol{\Phi}_3+\cdots$ is no larger than the mean of the first i singular values in $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]$. So we have

$$\forall i \ge r+1,\ \eta_i \le \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*/i. \tag{4.3.56}$$

This leads to

$$\begin{aligned}
\|\boldsymbol{\Phi}_2+\boldsymbol{\Phi}_3+\cdots\|_F^2 &= \sum_{i=r+1}^{\infty}\eta_i^2 && (4.3.57)\\
&\le \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*^2\sum_{i=r+1}^{\infty}\frac{1}{i^2} && (4.3.58)\\
&\le \frac{\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*^2}{r} \le \frac{\alpha^2\,\|\boldsymbol{H}_{\mathsf{T}}\|_*^2}{r} && (4.3.59)\\
&\le \alpha^2\,\|\boldsymbol{H}_{\mathsf{T}}\|_F^2 \le \alpha^2\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F^2\,. && (4.3.60)
\end{aligned}$$

Since $\boldsymbol{\Phi}_i$ with $i\ge 2$ are orthogonal to $\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1$, this gives us

$$\|\boldsymbol{H}\|_F^2 \le (1+\alpha^2)\,\|\boldsymbol{H}_{\mathsf{T}}+\boldsymbol{\Phi}_1\|_F^2\,. \tag{4.3.61}$$

Combining this with the previous bound (4.3.55) on $\|\mathcal{A}[\boldsymbol{H}]\|_2$, we obtain

$$\|\mathcal{A}[\boldsymbol{H}]\|_2 \ge \frac{1-\delta_{4r}(1+\alpha\sqrt{2})}{(1+\delta_{4r})\sqrt{1+\alpha^2}}\,\|\boldsymbol{H}\|_F\,. \tag{4.3.62}$$

This concludes the proof. □

Note that for the nuclear norm minimization problem, the feasible perturbation $\boldsymbol{H}$ satisfies the cone restriction (4.3.45). Thus Theorem 4.12 is essentially a corollary to Theorem 4.14 with constant $\alpha = 1$ for the cone restriction.

4.3.5 Rank-RIP of Random Measurements

Theorem 4.12 indicates that the rank-RIP implies a very strong conclusion: nuclear norm minimization exactly recovers low-rank matrices. Moreover the recovery is *uniform* in the sense that a single set of measurements $\mathcal{A}$ suffices to recover any sufficiently low-rank matrix $\boldsymbol{X}_o$. The remaining question is what measurement operators satisfy the rank-RIP?

Random Gaussian Measurements.
A simple and natural choice is to consider the random Gaussian measurements:

$$\mathcal{A}[\boldsymbol{X}] = (\langle \boldsymbol{A}_1, \boldsymbol{X}\rangle, \dots, \langle \boldsymbol{A}_m, \boldsymbol{X}\rangle), \tag{4.3.63}$$

where the entries of the matrices $\boldsymbol{A}_1, \dots, \boldsymbol{A}_m \in \mathbb{R}^{n_1 \times n_2}$ are all i.i.d. Gaussian $\mathcal{N}(0, \frac{1}{m})$. This is equivalent to viewing $\mathcal{A}$ as an $m \times n_1 n_2$ matrix with entries $\mathcal{A}_{ij}$ sampled i.i.d. $\mathcal{N}(0, \frac{1}{m})$. We demonstrate that such random maps satisfy the rank-RIP with high probability, using ideas and techniques similar to the proof of the (regular) RIP of random Gaussian matrices in Section 3.4.2:

THEOREM 4.15 (Rank-RIP of Gaussian Measurements). *If the measurement operator $\mathcal{A}$ is a random Gaussian map with entries i.i.d. $\mathcal{N}(0, \frac{1}{m})$, then $\mathcal{A}$ satisfies the rank-RIP with constant $\delta_r(\mathcal{A}) \le \delta$ with high probability, provided $m \ge Cr(n_1 + n_2) \times \delta^{-2} \log \delta^{-1}$, where $C > 0$ is a numerical constant.*

Proof Let

$$\mathsf{S}_r \doteq \{\boldsymbol{X} \mid \operatorname{rank}(\boldsymbol{X}) \le r, \|\boldsymbol{X}\|_F = 1\}.$$

Notice that $\delta_r(\mathcal{A}) \le \delta$ if and only if

$$\sup_{\boldsymbol{X} \in \mathsf{S}_r} \mid \langle \mathcal{A}[\boldsymbol{X}], \mathcal{A}[\boldsymbol{X}]\rangle - 1 \mid \le \delta. \tag{4.3.64}$$

We complete the rest of the proof in three steps.

1. Constructing a covering ε-net for S_r.
Notice that for any rank-r matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$, it can be represented by its SVD; $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. So to construct a covering of all rank-r matrices, we can try to construct a covering for each of the terms $\boldsymbol{U}, \boldsymbol{V}$ and $\boldsymbol{\Sigma}$, respectively.

LEMMA 4.16. *There is a covering ε-net N_U for the $\mathsf{H} = \{\boldsymbol{U} \in \mathbb{R}^{n_1 \times r} \mid \boldsymbol{U}^*\boldsymbol{U} = \boldsymbol{I}\}$ in operator norm, i.e.,*

$$\forall \boldsymbol{U} \in \mathsf{H},\ \exists \boldsymbol{U}' \in \mathsf{N}_U \quad \textit{satisfying} \quad \|\boldsymbol{U} - \boldsymbol{U}'\| \le \varepsilon, \tag{4.3.65}$$

of size $|\mathsf{N}_U| \le (6/\varepsilon)^{n_1 r}$.

Proof Let N' be an $\varepsilon/2$-net for $\{\boldsymbol{U} \in \mathbb{R}^{n_1 \times r} \mid \|\boldsymbol{U}\| \leq 1\}$ of size $|\mathsf{N}'| \leq (6/\varepsilon)^{n_1 r}$. The existence of such a net follows immediately from the volumetric argument used in the proof of Lemma 3.25. Let

$$\mathsf{Q} \doteq \{\boldsymbol{U}' \in \mathsf{N}' \mid \exists\, \boldsymbol{U} \in \mathsf{H} \text{ with } \|\boldsymbol{U} - \boldsymbol{U}'\| \leq \varepsilon/2\}.$$

For each $\boldsymbol{U}' \in \mathsf{Q}$, let $\hat{\boldsymbol{U}}(\boldsymbol{U}')$ be the nearest element of H. Set $\mathsf{N}_U = \{\hat{\boldsymbol{U}}(\boldsymbol{U}') \mid \boldsymbol{U}' \in \mathsf{Q}\} \subseteq \mathsf{H}$. By the triangle inequality, N_U is an ε-net for H. □

Similarly, one can construct an ε-net N_V for $\mathsf{H}' = \{\boldsymbol{V} \in \mathbb{R}^{n_2 \times r} \mid \boldsymbol{V}^*\boldsymbol{V} = \boldsymbol{I}\}$ of size $|\mathsf{N}_V| \leq (6/\varepsilon)^{n_2 r}$. With this lemma, we have the following result.

LEMMA 4.17. *There is a covering ε-net N_r for the set S_r, of size* $|\mathsf{N}_r| \leq \exp\big((n_1 + n_2)r\log(18/\varepsilon) + r\log(9/\varepsilon)\big)$.

Proof Choose $\varepsilon/3$-nets N_U and N_V that cover H and H', respectively, in operator norm. According to the above lemma, the sizes of the nets can be less than $(18/\varepsilon)^{n_1 r}$ and $(18/\varepsilon)^{n_2 r}$, respectively. Form a covering $\varepsilon/3$-net N_Σ for

$$\mathsf{D} \doteq \{\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r} \mid \boldsymbol{\Sigma} \text{ diagonal}, \|\boldsymbol{\Sigma}\|_F = 1\},$$

in Frobenius norm. According to Lemma 3.25, the size of the net can be less than $|\mathsf{N}_\Sigma| \leq (9/\varepsilon)^r$.

Now consider the following net for the whole set S_r:

$$\mathsf{N}_r \doteq \{\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^* \mid \boldsymbol{U} \in \mathsf{N}_U, \boldsymbol{\Sigma} \in \mathsf{N}_\Sigma, \boldsymbol{V} \in \mathsf{N}_V\}.$$

Its size is bounded by the product of all three nets, hence the expression in the Lemma. Now we only have to show that this is indeed a covering ε-net for S_r. For any given $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$, we can find $\hat{\boldsymbol{X}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^* \in \mathsf{N}_r$ with $\|\boldsymbol{U} - \hat{\boldsymbol{U}}\| \leq \varepsilon/3$, $\|\boldsymbol{V} - \hat{\boldsymbol{V}}\| \leq \varepsilon/3$, and $\|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|_F \leq \varepsilon/3$.

The triangle inequality gives

$$\begin{aligned}
&\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F \\
&\leq \|\boldsymbol{U} - \hat{\boldsymbol{U}}\|\|\boldsymbol{\Sigma}\boldsymbol{V}^*\|_F + \|\hat{\boldsymbol{U}}\|\|\boldsymbol{\Sigma} - \hat{\boldsymbol{\Sigma}}\|_F\|\boldsymbol{V}^*\| + \|\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\|_F\|\boldsymbol{V}^* - \hat{\boldsymbol{V}}^*\| \\
&\leq \varepsilon,
\end{aligned}$$

where we have used that each of the approximation errors is bounded by $\varepsilon/3$, $\|\hat{\boldsymbol{U}}\| = \|\boldsymbol{V}\| = 1$, and $\|\boldsymbol{\Sigma}\boldsymbol{V}^*\|_F = \|\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\|_F = 1$. □

2. Discretization.

As in the ℓ^1 case for sparse signals in Section 3.4.2, the goal of discretization is trying to show that if $\mathcal{A}$ is restricted isometric on the finite set of (discrete) points in the covering net N_r with a constant δ_{N_r}, so is $\mathcal{A}$ on the whole set S_r, with a constant δ_r possibly slightly larger than δ_{N_r}.

Now consider a point $\boldsymbol{X}$ in S_r and its closest point $\hat{\boldsymbol{X}}$ in N_r. Thus, we have

$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F \le \varepsilon$. Also we have[10]

$$\begin{aligned}
&|\langle \mathcal{A}[\boldsymbol{X}], \mathcal{A}[\boldsymbol{X}]\rangle - \langle \mathcal{A}[\hat{\boldsymbol{X}}], \mathcal{A}[\hat{\boldsymbol{X}}]\rangle| \\
=\ &|\langle \mathcal{A}[\boldsymbol{X}], \mathcal{A}[\boldsymbol{X} - \hat{\boldsymbol{X}}\boldsymbol{P}_V]\rangle + \langle \mathcal{A}[\boldsymbol{X} - \boldsymbol{P}_{\hat{U}}\boldsymbol{X}], \mathcal{A}[\hat{\boldsymbol{X}}\boldsymbol{P}_V]\rangle \\
&+\langle \mathcal{A}[\boldsymbol{P}_{\hat{U}}\boldsymbol{X} - \hat{\boldsymbol{X}}], \mathcal{A}[\hat{\boldsymbol{X}}\boldsymbol{P}_V]\rangle + \langle \mathcal{A}[\hat{\boldsymbol{X}}], \mathcal{A}[\hat{\boldsymbol{X}}\boldsymbol{P}_V - \hat{\boldsymbol{X}}]\rangle|.
\end{aligned}$$

To bound the first term in the above expression, notice that

$$\|\boldsymbol{X} - \hat{\boldsymbol{X}}\boldsymbol{P}_V\|_F = \|(\boldsymbol{X} - \hat{\boldsymbol{X}})\boldsymbol{P}_V\|_F \le \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F \le \varepsilon.$$

Also, $\boldsymbol{X} - \hat{\boldsymbol{X}}\boldsymbol{P}_V$ is of rank r. So we have

$$|\langle \mathcal{A}[\boldsymbol{X}], \mathcal{A}[\boldsymbol{X} - \hat{\boldsymbol{X}}\boldsymbol{P}_V]\rangle| \le (1 + \delta_r(\mathcal{A}))\varepsilon.$$

For the second term, since $\boldsymbol{P}_{\hat{U}}$ is an orthogonal projection onto the space of matrices whose columns are the same as $\hat{\boldsymbol{X}}$, we have

$$\|\boldsymbol{X} - \boldsymbol{P}_{\hat{U}}\boldsymbol{X}\|_F \le \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F \le \varepsilon.$$

Also, since $\boldsymbol{X}$ and $\boldsymbol{P}_{\hat{U}}\boldsymbol{X}$ have the same row space, so $\boldsymbol{X} - \boldsymbol{P}_{\hat{U}}\boldsymbol{X}$ is of rank r or less. Therefore, we also have

$$|\langle \mathcal{A}[\boldsymbol{X} - \boldsymbol{P}_{\hat{U}}\boldsymbol{X}], \mathcal{A}[\hat{\boldsymbol{X}}\boldsymbol{P}_V]\rangle| \le (1 + \delta_r(\mathcal{A}))\varepsilon.$$

Similarly for the third and fourth terms, each is bounded by the same bound. Therefore, we get

$$|\langle \mathcal{A}[\boldsymbol{X}], \mathcal{A}[\boldsymbol{X}]\rangle - \langle \mathcal{A}[\hat{\boldsymbol{X}}], \mathcal{A}[\hat{\boldsymbol{X}}]\rangle| \le 4(1 + \delta_r(\mathcal{A}))\varepsilon.$$

From this we have

$$\delta_r(\mathcal{A}) - \delta_{\mathsf{N}_r} \le 4(1 + \delta_r(\mathcal{A}))\varepsilon. \tag{4.3.66}$$

This gives

$$\delta_r(\mathcal{A}) \le \frac{4\varepsilon + \delta_{\mathsf{N}_r}}{1 - 4\varepsilon}. \tag{4.3.67}$$

3. Union bound.
For each $\boldsymbol{X} \in \mathsf{N}_r$, $\mathcal{A}[\boldsymbol{X}] \in \mathbb{R}^m$ is a random vector with entries independent $\mathcal{N}(0, 1/m)$. We have

$$\mathbb{P}\left[\left|\|\mathcal{A}[\boldsymbol{X}]\|_2^2 - 1\right| > t\right] \ \le\ 2\exp(-mt^2/8). \tag{4.3.68}$$

Hence, summing the probabilities over all elements of N_r, we have

$$\begin{aligned}
\mathbb{P}\left[\delta_{\mathsf{N}_r} > t\right] &\le 2\,|\mathsf{N}_r| \exp\left(-mt^2/8\right) \\
&= 2\exp\left(-\frac{mt^2}{8} + (n_1 + n_2)r\log(18/\varepsilon) + r\log(9/\varepsilon)\right).
\end{aligned}$$

If we choose $\varepsilon = c \cdot \delta$ and $t = c \cdot \delta$ for some small constant c and ensure $m \ge Cr(n_1 + n_2)\delta^{-2}\log\delta^{-1}$ for some large enough C, the above failure probability is

[10] Notice that here the derivation is more subtle than the ℓ^1 case because $\boldsymbol{X} - \hat{\boldsymbol{X}}$ is not necessarily of rank r.

bounded by $2\exp(-c'm\delta^2)$. On the complement of this "failure" event, $\delta_{\mathsf{N}_r} \leq c\cdot\delta$, and due to (4.3.67) we have $\delta_r(\mathcal{A}) \leq \delta$. This concludes the proof of the Theorem 4.15. □

The number of measurements $m = O(r(n_1 + n_2))$ required is nearly optimal, since an $n_1 \times n_2$ rank-r matrix has $r(n_1 + n_2 - r)$ degrees of freedom. Of course, the big O notation hides a numerical constant. Like the ℓ^1 minimization for sparse recovery, when the dimension is high, nuclear norm minimization exhibits a phase transition between success and failure. Identifying this transition yields more precise estimates of the number m of measurements required to reconstruct a low rank matrix. We discuss this issue in more detail below.

Random Submatrix of a Unitary Basis.
Although random Gaussian measurements have very nice properties such as (rank) RIP, the lack of structure in such measurements makes it rather expensive to generate, store and apply such operators in practice. Hence it is natural to ask if there exist other more structured measurements that have similarly good RIP properties. In Section 3.4.3, we saw that given any unitary matrix that is incoherent from sparse signals, then a randomly selected subset of its rows will satisfy the RIP with high probability. An important special case that has been widely used in practice for compressive sensing is a randomly chosen submatrix of the discrete Fourier transform basis. It is then natural to ask what are the Fourier-type bases for matrices.

In the case of sparse recovery, we start with a unitary basis $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ and show that if the rows $\{\boldsymbol{u}_i\}_{i=1}^n$ of the basis is incoherent with sparse signals:

$$\forall i \quad \|\boldsymbol{u}_i\|_\infty = \sup_{\boldsymbol{x}:\|\boldsymbol{x}\|_2=1,\|\boldsymbol{x}\|_0=1} \langle \boldsymbol{u}_i, \boldsymbol{x}\rangle \leq \zeta/\sqrt{n}$$

for some constant ζ, then a randomly selected (sufficient) number of rows of $\boldsymbol{U}$ will satisfy RIP.

To simplify the discussion of matrices, we will assume $n_1 = n_2 = n$ for the rest of this subsection; a similar approach applies when $n_1 \neq n_2$. Let us assume $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_{n^2}\} \subset \mathbb{C}^{n\times n}$ form a unitary basis for the matrix space $\mathbb{C}^{n\times n}$. Similarly we want each of the matrix $\boldsymbol{U}_i$ to be incoherent with low-rank matrices. Note that for any $\boldsymbol{X} \in \mathbb{C}^{n\times n}$,

$$\|\boldsymbol{U}_i\| = \sup_{\boldsymbol{X}:\|\boldsymbol{X}\|_2=1,\mathrm{rank}(\boldsymbol{X})=1} \langle \boldsymbol{U}_i, \boldsymbol{X}\rangle. \tag{4.3.69}$$

Hence in order for each $\boldsymbol{U}_i$ to be incoherent with low-rank matrices, we could require:

$$\forall i \quad \|\boldsymbol{U}_i\| \leq \zeta/\sqrt{n}. \tag{4.3.70}$$

Then to construct the measurement operator $\mathcal{A}$, we randomly select a subset of

m bases from $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_{n^2}\}$ and properly scale them as:[11]

$$\mathcal{A}: \quad \boldsymbol{A}_i = \frac{n}{\sqrt{m}} \boldsymbol{U}_i, \quad i = 1, \ldots, m. \tag{4.3.71}$$

Then one should expect that when m is large enough, with high probability, the so-defined $\mathcal{A}$ satisfies the rank-RIP. The following theorem makes this precise:

THEOREM 4.18. *Let us assume* $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_{n^2}\} \subset \mathbb{C}^{n\times n}$ *be a unitary basis for the matrix space* $\mathbb{C}^{n\times n}$ *and with* $\|\boldsymbol{U}_i\| \le \zeta/\sqrt{n}$ *for some constant* ζ. *Let* $\mathcal{A}$ *to be defined as per* (4.3.71). *Then if*

$$m \ge C\zeta^2 \cdot rn \log^6 n, \tag{4.3.72}$$

then with high probability, $\mathcal{A}$ *satisfies the rank-RIP over the set of all rank-*r *matrices.*

The proof of this theorem is out of the scope of this book and interested readers may refer to the work of [Liu11].

According to this statement, from an incoherent unitary basis, with high probability we could find a (compressive) sensing operator $\mathcal{A}$ such that it is rank-RIP. Hence with this operator, one can recover all rank-r matrices via the nuclear norm minimization. The remaining question is what type of structured bases (of the matrix space) are rank-incoherent as per (4.3.70)? To this end, one should seek a matrix analogue to the Fourier basis.

In the case of MRI imaging, we have seen that measurements that one can physically take are essentially the Fourier coefficients of the brain image. As it turns out, the matrix analogue to Fourier basis also has a natural origin from physics. In quantum-state tomography, a system of k qubits is of dimension $n = 2^k$. The quantum state of such a system is described by a density matrix $\boldsymbol{X}_o \in \mathbb{C}^{n\times n}$ which is positive semidefinite with trace 1. When the state is early pure, $\boldsymbol{X}_o$ is a very low-rank matrix with $\operatorname{rank}(\boldsymbol{X}_o) = r \ll n$.

One problem in quantum physics is how to recover the quantum state $\boldsymbol{X}_o$ of a system from linear measurements. As it turns out, a set of experimentally feasible measurements are given by the so-called *Pauli observables*. Each Pauli measurement is given by the inner product of $\boldsymbol{X}_o$ with matrices of the form $\boldsymbol{P}_1 \otimes \cdots \otimes \boldsymbol{P}_k$ where $\otimes$ is the tensor (Kronecker) product and each $\boldsymbol{P}_i = \frac{1}{\sqrt{2}}\boldsymbol{\sigma}$ where $\boldsymbol{\sigma}$ is a 2×2 matrix chosen from the following four possibilities:

$$\boldsymbol{\sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{\sigma}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \boldsymbol{\sigma}_3 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \boldsymbol{\sigma}_4 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It is easy to see that there are a total of 4^k possible choices for the tensor product, denoted as $\{\boldsymbol{U}_i\}_{i=1}^{4^k}$ and they together form an orthonormal basis for the matrix space $\mathbb{C}^{n\times n}$ where $n = 2^k$.

One can show that for each basis $\boldsymbol{U}_i = \boldsymbol{P}_1 \otimes \cdots \otimes \boldsymbol{P}_k$, its operator norm is bounded as $\|\boldsymbol{U}_i\| \le 1/\sqrt{n}$ hence incoherent with low-rank matrices. Then

[11] The scaling is to ensure that the "column" of $\mathcal{A}$ to be of unit norm.

according to Theorem 4.18, a randomly selected $m \geq Crn \log^6 n$ rows of the Pauli bases will satisfy the rank-RIP property with high probability. Hence, such a sensing operator will be able to uniformly recover all pure quantum states less than rank r.

4.3.6 Noise, Inexact Low Rank, and Phase Transition

Above, we established that, under fairly broad conditions, nuclear norm minimization correctly recovers a low-rank matrix $\boldsymbol{X}_o$ from ideal measurements $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o]$. In practice, the measurements can be corrupted by noise or measurement errors. In some cases, $\boldsymbol{X}_o$ may not be exactly low rank. It is desirable to understand whether nuclear norm minimization still gives reasonably good estimates of $\boldsymbol{X}_o$ in these situations.

In Section 3.5, we established that ℓ^1 minimization accurately estimates sparse signals under deterministic noise, random noise, and even inexact sparsity. As we will see in this section, essentially the same analysis and results generalize to the case of nuclear norm minimization for recovering low-rank matrices.

Deterministic Noise.
Here we still assume the matrix $\boldsymbol{X}_o$ is perfectly low-rank, but the measurements $\boldsymbol{y}$ is corrupted by small additive noise:

$$\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}, \quad \|\boldsymbol{z}\|_2 \leq \varepsilon. \tag{4.3.73}$$

Similar to Theorem 3.29, for recovering low-rank matrices with (deterministic) noise, we have the following result.

THEOREM 4.19 (Stable Low-rank Recovery via BPDN). *Suppose that* $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}$, *with* $\|\boldsymbol{z}\|_2 \leq \varepsilon$, *and let* $\operatorname{rank}(\boldsymbol{X}_o) = r$. *If* $\delta_{4r}(\mathcal{A}) < \sqrt{2} - 1$, *then any solution* $\hat{\boldsymbol{X}}$ *to the optimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{X}\|_* \\ \text{subject to} & \|\mathcal{A}[\boldsymbol{X}] - \boldsymbol{y}\|_2 \leq \varepsilon. \end{array} \tag{4.3.74}$$

satisfies

$$\left\|\hat{\boldsymbol{X}} - \boldsymbol{X}_o\right\|_F \leq C\varepsilon. \tag{4.3.75}$$

Here, C *is a numerical constant.*

Proof The proof of this theorem parallels that for Theorem 3.29 and we leave the details for the reader as an exercise (see Exercise 4.17). □

Random Noise.
Now let us consider the case when the noise in the above measurement model (4.3.73) is random (Gaussian):

$$\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}, \tag{4.3.76}$$

where entries of $\boldsymbol{z}$ are random i.i.d. Gaussian $\mathcal{N}(0, \frac{\sigma^2}{m})$. Then we have the following theorem that parallels Theorem 3.31 for the ℓ^1 case.

THEOREM 4.20 (Stable Low-rank Recovery via Lasso). *Suppose that* $\mathcal{A} \sim_{\text{iid}} \mathcal{N}(0, \frac{1}{m})$, *and* $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}$, *with* $\boldsymbol{X}_o$ *of rank* r *and* $\boldsymbol{z} \sim_{\text{iid}} \mathcal{N}(0, \frac{\sigma^2}{m})$. *Solve the matrix Lasso*

$$\min \ \ \tfrac{1}{2} \left\| \boldsymbol{y} - \mathcal{A}[\boldsymbol{X}] \right\|_2^2 + \lambda_m \left\| \boldsymbol{X} \right\|_*, \tag{4.3.77}$$

with regularization parameter $\lambda_m = c \cdot 2\sigma\sqrt{\frac{(n_1+n_2)}{m}}$ *for a large enough* c. *Then with high probability,*

$$\left\| \hat{\boldsymbol{X}} - \boldsymbol{X}_o \right\|_F \ \le \ C'\sigma\sqrt{\frac{r(n_1+n_2)}{m}}. \tag{4.3.78}$$

Notice in contrast to deterministic noise, random noise leads to a much more favorable scaling $\sqrt{\frac{r(n_1+n_2)}{m}}$ in the estimation error: To see this, notice that in a typical compressive sensing setting (as suggested by Theorem 4.15), the sampling dimension m needs to be at least $C \cdot r(n_1+n_2)$ for some large constant C. Hence the scaling factor is proportional to $1/\sqrt{C}$ and it becomes small when C is large.

Proof The overall proof strategy is quite similar to that of Theorem 3.31 for the stability of Lasso estimate. We will lay out the key places that are different from the ℓ^1 case and leave the details to the reader as an exercise.

In the proof of Lemma 3.30, we see that in order to establish the cone condition for the Lasso type minimization, one of the key steps is to bound $|\langle \boldsymbol{A}^*\boldsymbol{z}, \boldsymbol{h}\rangle|$ via

$$|\langle \boldsymbol{A}^*\boldsymbol{z}, \boldsymbol{h}\rangle| \le \left\| \boldsymbol{A}^*\boldsymbol{z} \right\|_\infty \left\| \boldsymbol{h} \right\|_1.$$

Following similar arguments, in the matrix Lasso case here, we need to bound $|\langle \mathcal{A}^*\boldsymbol{z}, \boldsymbol{H}\rangle|$ instead as

$$|\langle \mathcal{A}^*\boldsymbol{z}, \boldsymbol{H}\rangle| \le \left\| \mathcal{A}^*\boldsymbol{z} \right\| \left\| \boldsymbol{H} \right\|_*,$$

where $\|\mathcal{A}^*\boldsymbol{z}\|$ is the operator norm (largest singular value) of the matrix $\mathcal{A}^*\boldsymbol{z} = \sum_{i=1}^m z_i \boldsymbol{A}_i$. To this end, we need to provide a tight bound for the operator norm of $\mathcal{A}^*\boldsymbol{z}$.

Notice that

$$M \doteq \left\| \sum_{i=1}^m z_i \boldsymbol{A}_i \right\| = \sup_{\boldsymbol{u}\in\mathbb{S}^{n_1-1}, \boldsymbol{v}\in\mathbb{S}^{n_2-1}} \boldsymbol{u}^* \sum_{i=1}^m z_i \boldsymbol{A}_i \boldsymbol{v} \tag{4.3.79}$$

$$= \sup_{\boldsymbol{u}\in\mathbb{S}^{n_1-1}, \boldsymbol{v}\in\mathbb{S}^{n_2-1}} \langle \boldsymbol{z}, \mathcal{A}[\boldsymbol{u}\boldsymbol{v}^*]\rangle. \tag{4.3.80}$$

The $\boldsymbol{u}_\star$ and $\boldsymbol{v}_\star$ that achieve the maximum value in (4.3.80) depend on $\boldsymbol{z}$ and $\mathcal{A}$. So in order to eliminate this dependency and provide a bound for $\|\sum_{i=1}^m z_i \boldsymbol{A}_i\|$, we cover the two spheres $\mathbb{S}^{n_1-1}$ and $\mathbb{S}^{n_2-1}$ with two ε-nets N_1 and N_2 respectively. According to Lemma 3.25, the sizes of the nets can be less than $(3/\varepsilon)^{n_1}$ and $(3/\varepsilon)^{n_2}$ respectively.

Let us denote

$$M_{\mathsf{N}} \doteq \sup_{\boldsymbol{u}\in\mathsf{N}_1,\boldsymbol{v}\in\mathsf{N}_2} \boldsymbol{u}^* \sum_{i=1}^m z_i \boldsymbol{A}_i \boldsymbol{v},$$

and then it is easy to show that[12]

$$M \le \frac{M_{\mathsf{N}}}{1-2\varepsilon}. \tag{4.3.81}$$

Notice that given any $\boldsymbol{u} \in \mathsf{N}_1, \boldsymbol{v} \in \mathsf{N}_2$, $\langle \boldsymbol{z}, \mathcal{A}[\boldsymbol{u}\boldsymbol{v}^*]\rangle$ is a Gaussian variable of distribution $\mathcal{N}(0, \|\mathcal{A}[\boldsymbol{u}\boldsymbol{v}^*]\|_2^2(\sigma^2/m))$. Since $\mathcal{A}$ is rank-RIP and $\boldsymbol{u}\boldsymbol{v}^*$ is a rank-1 matrix of unit Frobenius norm, we have

$$\|\mathcal{A}[\boldsymbol{u}\boldsymbol{v}^*]\|_2^2 \le (1+\delta) \le 2. \tag{4.3.82}$$

Thus, we have

$$\mathbb{P}\left[\left|\boldsymbol{u}^* \sum_{i=1}^m z_i \boldsymbol{A}_i \boldsymbol{v}\right| > t\right] \le 2\exp\left(-\frac{mt^2}{4\sigma^2}\right). \tag{4.3.83}$$

Apply the union bound on all possible pairs of $(\boldsymbol{u}, \boldsymbol{v})$ from the two nets and choose $t = \alpha\sigma\sqrt{\frac{n_1+n_2}{m}}$ for some large enough α, then we have $M_{\mathsf{N}} > t$ with diminishing probability as n_1 or n_2 becomes large. Therefore, we have

$$M = \left\|\sum_{i=1}^m z_i \boldsymbol{A}_i\right\| \le \beta\sigma\sqrt{\frac{n_1+n_2}{m}} \tag{4.3.84}$$

for some constant β with high probability.

Now, similar to the proof of Lemma 3.30, if we choose λ_m to be in the order of $O\big(\sigma\sqrt{\frac{n_1+n_2}{m}}\big)$, then the feasible perturbation $\boldsymbol{H}$ satisfies the cone restriction. Since $\mathcal{A}$ is rank-RIP, it implies that $\mathcal{A}$ satisfies the matrix restricted strong convexity (RSC) property. That leads to the bound on the estimation error:

$$\|\boldsymbol{H}\|_F = \left\|\hat{\boldsymbol{X}} - \boldsymbol{X}_o\right\|_F \le C'\sigma\sqrt{\frac{r(n_1+n_2)}{m}}. \tag{4.3.85}$$

The details of the proof for this follow essentially the same steps as those in the proof of Theorem 3.31 for the ℓ^1 case. We leave those to the reader as an exercise (see Exercise 4.19.) □

The error bound given in the above theorem can actually be shown to be nearly optimal as it is close to the best error that can be achieved by any estimator over all rank-r matrices. The following theorem, due to [CP11] makes this precise:

THEOREM 4.21. *Suppose that* $\mathcal{A} \sim_{\text{iid}} \mathcal{N}(0, \frac{1}{m})$ *and we observe* $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}$ *where entries of* $\boldsymbol{z}$ *are i.i.d.* $\mathcal{N}(0, \frac{\sigma^2}{m})$ *random variables. Set*

$$M^\star(\mathcal{A}) = \inf_{\hat{\boldsymbol{X}}(\boldsymbol{y})} \sup_{\operatorname{rank}(\boldsymbol{X})\le r} \mathbb{E}\left\|\hat{\boldsymbol{X}}(\boldsymbol{y}) - \boldsymbol{X}\right\|_F^2. \tag{4.3.86}$$

[12] We leave the details of proving this inequality to the reader as an exercise.

Then we have

$$M^\star(\mathcal{A}) \;\geq\; c\sigma^2 \frac{rn}{m}, \tag{4.3.87}$$

for $n = \max\{n_1, n_2\}$*, where* $c > 0$ *is a numerical constant.*

The proof of this theorem is beyond the scope of this book; we refer interested readers to [CP11] for a proof. According to Theorem 4.20, the worst error of the matrix Lasso matches the best achievable by any estimator, up to constants.

Inexact Low-rank Matrices.
In the case when $\boldsymbol{X}_o$ is not exactly low-rank, let $[\boldsymbol{X}_o]_r$ be the best rank-r approximation of $\boldsymbol{X}_o$. We can rewrite the observation model

$$\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}, \quad \|\boldsymbol{z}\|_2 \leq \varepsilon. \tag{4.3.88}$$

as:

$$\boldsymbol{y} = \mathcal{A}\left[[\boldsymbol{X}_o]_r\right] + \mathcal{A}\left[\boldsymbol{X}_o - [\boldsymbol{X}_o]_r\right] + \boldsymbol{z}, \quad \|\boldsymbol{z}\|_2 \leq \varepsilon.$$

THEOREM 4.22 (Inexact Low-rank Recovery). *Let* $\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}_o] + \boldsymbol{z}$*, with* $\|\boldsymbol{z}\|_2 \leq \varepsilon$*. Let* $\hat{\boldsymbol{X}}$ *solve the denoising problem*

$$\begin{array}{ll} \min & \|\boldsymbol{X}\|_* \\ \text{subject to} & \|\boldsymbol{y} - \mathcal{A}[\boldsymbol{X}]\|_2 \leq \varepsilon. \end{array} \tag{4.3.89}$$

Then for any r *such that* $\delta_{4r}(\boldsymbol{A}) \;<\; \sqrt{2} - 1$*,*

$$\left\|\hat{\boldsymbol{X}} - \boldsymbol{X}_o\right\|_2 \;\leq\; C\frac{\|\boldsymbol{X}_o - [\boldsymbol{X}_o]_r\|_*}{\sqrt{r}} + C'\varepsilon \tag{4.3.90}$$

for some constants C *and* C'*.*

Proof The proof of this theorem parallels that for Theorem 3.33 for the inexact sparse recovery problem. We here only setup some analogous concepts and key ideas that allow us to extend that proof to the matrix case here. But we leave details of the proof as an exercise to the reader.

Let $\boldsymbol{X}_o = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ denote the compact SVD of the true solution $\boldsymbol{X}_o$. Then its best rank-r approximation is $[\boldsymbol{X}_o]_r = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^*$. Now let

$$\mathsf{T} \doteq \left\{\boldsymbol{U}_r\boldsymbol{R}^* + \boldsymbol{Q}\boldsymbol{V}_r^* \mid \boldsymbol{R} \in \mathbb{R}^{n_2\times r},\, \boldsymbol{Q} \in \mathbb{R}^{n_1\times r}\right\} \subseteq \mathbb{R}^{n_1\times n_2}. \tag{4.3.91}$$

Show that in the inexact low-rank case, instead of the cone restriction (4.3.45), we have the following restriction for the feasible perturbation $\boldsymbol{H} = \hat{\boldsymbol{X}} - \boldsymbol{X}_o$:

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_* \leq \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_* + 2\left\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{X}_o]\right\|_*. \tag{4.3.92}$$

Notice that $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{X}_o] = \boldsymbol{X}_o - [\boldsymbol{X}_o]_r$. Then, similar to the proof of Theorem 3.33, simply carry the extra term $2\left\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{X}_o]\right\|_*$ at the places in the proof of Theorem 4.14 where the cone restriction is applied. One can reach the conclusion of the theorem. We leave details of the proof as an exercise to the reader (see Exercise 4.18). □

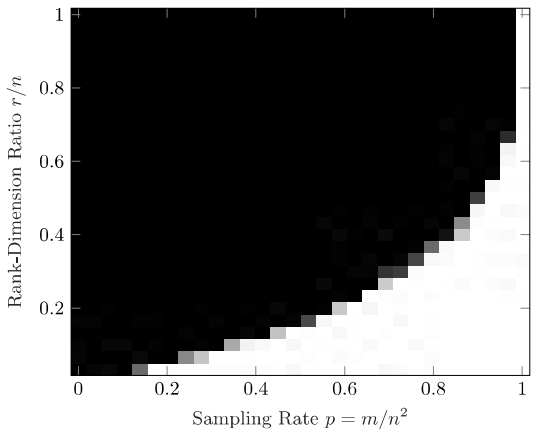


Figure 4.7 Phase Transitions in Low-rank Matrix Recovery. We plot the probability of successfully recovering an $n \times n$ low-rank matrix $\boldsymbol{X}_o$ from Gaussian measurements. Horizontal axis: sampling rate $p = m/n^2$. Vertical axis rank-dimension ratio r/n. The success of nuclear norm minimization exhibits a very sharp transition from success to failure.

Phase Transition in Low-rank Matrix Recovery.
Thus far, we have seen strong parallels between sparse vector recovery using ℓ^1 norm minimization and low-rank matrix recovery using nuclear norm minimization. In both cases, we saw how an appropriate notion of restricted isometry property could be used to guarantee exact recovery from a near-minimal number of random measurements – about $k \log(n/k)$ for k-sparse vectors, and about nr for rank-r matrices. However, just like in the sparse vector case, this tool does not yield sharp constants.

In fact, there is a phase transition phenomenon for low-rank recovery, which mirrors that for sparse recovery: as the dimension grows, the transition between success and failure in low-rank recovery becomes increasingly sharp. Figure 4.7 illustrates this.

Just as we did for sparse recovery, we can use the "coefficient space" geometry of the low-rank recovery problem to derive very sharp estimates of this transition. This geometry is phrased in terms of the descent cone D of the nuclear norm at the target solution $\boldsymbol{X}_o$:

$$\mathsf{D} \doteq \{\boldsymbol{H} \mid \|\boldsymbol{X}_o + \boldsymbol{H}\|_* \leq \|\boldsymbol{X}_o\|_*\}\,. \tag{4.3.93}$$

As for sparse recovery, $\boldsymbol{X}_o$ is the unique optimal solution to the nuclear norm minimization problem if and only if $\mathsf{D} \cap \mathsf{null}(\mathcal{A}) = \{\boldsymbol{0}\}$. Hence, quantifying the probability of success under a random linear projection becomes equivalent to quantifying the probability that the two convex cones D and $\mathsf{null}(\mathcal{A})$ have only trivial intersection. Deploying Theorem 6.14, we find that there is a sharp tran-

sition between success and failure around

$$m^\star \sim \delta(\mathsf{D}), \tag{4.3.94}$$

the statistical dimension of the descent cone. Moreover, the theorem tells us that the width of the transition region is roughly $O(\sqrt{n_1 n_2})$. The *location* of the transition region can be characterized using the same machinery that we deployed in Section 3.6 to estimate the statistical dimension of the descent cone of the ℓ^1 norm. This machinery involves estimating the expected squared distance of a random vector (here, random matrix) to the polar cone, which is spanned by the subdifferential of the nuclear norm. For convenience, for a matrix $\boldsymbol{M}$ with singular value decomposition $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$, let us define the *singular value thresholding operator* as

$$\mathcal{D}_\tau[\boldsymbol{M}] \doteq \boldsymbol{U}\mathcal{S}_\tau[\boldsymbol{\Sigma}]\boldsymbol{V}^*, \tag{4.3.95}$$

where $\mathcal{S}_\tau[\cdot]$ is the entry-wise *soft thresholding* operator:

$$\forall \boldsymbol{X}, \quad \mathcal{S}_\tau[\boldsymbol{X}] = \operatorname{sign}(\boldsymbol{X}) \circ (|\boldsymbol{X}| - \tau)_+,$$

where $\circ$ is the entry-wise (Hadamard) product of two matrices. An intermediate result produced by these calculations is as follows:

THEOREM 4.23 (Phase Transition in Low-rank Recovery). *Let* D *denote the descent cone of the nuclear norm at any matrix* $\boldsymbol{X}_o \in \mathbb{R}^{n_1 \times n_2}$ *of rank* r*. Let* $\boldsymbol{G}$ *be an* $(n_1 - r) \times (n_2 - r)$ *matrix with entries i.i.d.* $\mathcal{N}(0,1)$*. Set*

$$\psi(n_1, n_2, r) = \inf_{\tau \geq 0} \left\{ r(n_1 + n_2 - r + \tau^2) + \mathbb{E}_{\boldsymbol{G}} \left[\|\mathcal{D}_\tau[\boldsymbol{G}]\|_F^2 \right] \right\}. \tag{4.3.96}$$

Then

$$\psi(n_1, n_2, r) - 2\sqrt{n_2/r} \;\leq\; \delta(\mathsf{D}) \;\leq\; \psi(n_1, n_2, r). \tag{4.3.97}$$

This theorem identifies a sharp transition in low-rank recovery. It is possible to use asymptotic results on the limiting distribution of the singular values of a random matrix to give a formula for $\psi(n_1, n_2, r)/(n_1 n_2)$, which is valid when $n_1 \to \infty$, $n_1/n_2 \to \alpha \in (0, \infty)$ and $r/n_1 \to \rho \in (0, 1)$. In the exercises, we guide the interested reader through this derivation. Here, we merely display the result of this calculation in Figure 4.7, and note the excellent agreement between this theoretical prediction and numerical experiment: *for the idealized setting of "generic" measurements, we have a very precise prediction of the phase transition!*

4.4 Low-Rank Matrix Completion

We have seen how concepts from sparse recovery transpose directly to the low-rank recovery problem. The concept of sparsity had a natural analogue in the

concept of rank-deficiency. The ℓ^1 minimization problem for sparse recovery had a natural analogue in the nuclear norm minimization problem for low-rank recovery. Moreover, these convex relaxations succeed under analogous conditions involving restricted isometry properties of the observation operator.

However, in many of the most interesting applications of nuclear norm minimization, the RIP does not hold! In the introduction to this chapter, we sketched applications to recommendation systems, in which we had access to *a subset* of the entries of a low-rank user-item matrix. We also sketched problems in reconstructing 3D shape, in which we observed *a subset* of the pixels of the rank-3 matrix $\boldsymbol{NL}$. Finally, we sketched a problem in Euclidean embedding, in which we observe *a subset* of the distances between some objects of interest. In all of these problems, the object of interest is a low-rank matrix $\boldsymbol{X}_o \in \mathbb{R}^{n \times n}$; the observation selects a subset $\Omega \subset [n] \times [n]$ of the entries of $\boldsymbol{X}_o$. The *matrix completion* problem asks us to fill in the missing entries:

PROBLEM 4.24 (Matrix Completion). *Let $\boldsymbol{X}_o \in \mathbb{R}^{n \times n}$ be a low-rank matrix. Suppose we are given $\boldsymbol{y} = \mathcal{P}_\Omega[\boldsymbol{X}_o]$, where $\Omega \subseteq [n] \times [n]$. Fill in the missing entries of $\boldsymbol{X}_o$.*

In matrix completion, the observation operator $\mathcal{A} = \mathcal{P}_\Omega$ is the restriction onto some small subset $\Omega \subseteq [n] \times [n]$ of the entries. In this situation, if $(i, j) \notin \Omega$, $\mathcal{P}_\Omega[\boldsymbol{E}_{ij}] = \boldsymbol{0}$ where $\boldsymbol{E}_{ij}$ denotes the matrix with all zeros except for the (i, j)th entry being 1. That is to say, if Ω is a strict subset of $[n] \times [n]$, then $\mathcal{P}_\Omega$ has matrices of rank one in its null space! So, the rank-RIP cannot hold for any positive rank r with any nontrivial $\delta < 1$.

At a more basic level, the example of $\boldsymbol{X}_o = \boldsymbol{E}_{ij}$ suggests that there are some (very sparse) matrices that are impossible to complete from only a few entries. This is in contrast to our discussion of low-rank matrix recovery thus far, in which the only factor that dictates the ease or difficulty of recovering a target $\boldsymbol{X}_o$ is the complexity $\operatorname{rank}(\boldsymbol{X}_o)$. Nevertheless, our development thus far suggests that even for the more challenging problem of matrix completion, there may be some class of *well-structured* matrices $\boldsymbol{X}_o$ of interest for applications, which *can* be efficiently completed from just a few entries. In this section, we will see that this is indeed the case.

4.4.1 Nuclear Norm Minimization for Matrix Completion

In light of our previous study of matrix recovery, a natural approach to completing a low-rank matrix from a small subset $\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{X}_o]$ of its entries is to look for the matrix $\boldsymbol{X}$ of minimum nuclear norm that agrees with the observation:

$$\begin{array}{ll} \text{min} & \|\boldsymbol{X}\|_* \\ \text{subject to} & \mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}. \end{array} \tag{4.4.1}$$

This is a special instance of the general nuclear norm minimization problem (4.3.14), with observation operator $\mathcal{A} = \mathcal{P}_\Omega$. As such, it is a semidefinite program,

and can be solved with high accuracy in polynomial time. In practice, though, it is more important to have methods that scale to large problem instances. In the next section, we sketch one approach to achieving this, using Lagrange multiplier techniques. This approach has pedagogical value: it introduces several objects that will be used for analyzing when we can solve matrix completion problems efficiently. It also yields reasonably scalable algorithms. For practical matrix completion at the scale of $n \sim 10^6$ and beyond, even more scalable methods are needed; we discuss these issues in Chapters 8-9.

4.4.2 Algorithm via Augmented Lagrange Multiplier

There are two basic challenges in solving problem (4.4.1) at large scale. The first arises from the nonsmoothness of the nuclear norm $\|\cdot\|_*$; the second is due to the need to satisfy the constraint $\mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}$ exactly.[13] The fundamental technology for handling constraints in optimization is Lagrange duality.

The basic object is the *Lagrangian*, which introduces a matrix $\boldsymbol{\Lambda}$ of Lagrange multipliers for the constraint $\mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}$. The Lagrangian for (4.4.1) is

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\Lambda}) = \|\boldsymbol{X}\|_* + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}] \rangle \,. \tag{4.4.2}$$

As introduced in the Appendix C, the optimal $\boldsymbol{X}_\star$ solution is characterized as a *saddle point* of the Lagrangian which is *minimized* with respect to $\boldsymbol{X}$, and maximized with respect to $\boldsymbol{\Lambda}$. A basic approach to solving a constrained problem such as (4.4.1) is to seek such a saddle point. In practice, more robustly convergent algorithms can be derived by instead working with the *augmented* Lagrangian

$$\mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{\Lambda}) = \|\boldsymbol{X}\|_* + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}] \rangle + \tfrac{\mu}{2} \|\boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}]\|_F^2 \,, \tag{4.4.3}$$

which encourages satisfaction of the constraint by adding an additional quadratic penalty term $\frac{\mu}{2} \|\boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}]\|_F^2$. A more general introduction to augmented Lagrangian method (ALM) is given in section 8.4 of Chapter 8.

The augmented Lagrangian method seeks a saddle point of $\mathcal{L}_\mu$ by alternating between minimizing with respect to the "primal variables" $\boldsymbol{X}$ and taking one step of gradient ascent to increase $\mathcal{L}_\mu$ using the "dual variables" $\boldsymbol{\Lambda}$:

$$\boldsymbol{X}_{k+1} \in \arg\min_{\boldsymbol{X}} \; \mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{\Lambda}_k), \tag{4.4.4}$$

$$\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu \mathcal{P}_\Omega\big[\boldsymbol{Y} - \boldsymbol{X}_{k+1}\big]. \tag{4.4.5}$$

Here, $\mathcal{P}_\Omega\big[\boldsymbol{Y} - \boldsymbol{X}_{k+1}\big] = \nabla_{\boldsymbol{\Lambda}} \mathcal{L}_\mu(\boldsymbol{X}_{k+1}, \boldsymbol{\Lambda})$. The ALM algorithm makes a very special choice of the step size (μ) for updating $\boldsymbol{\Lambda}$. This choice is important in general: it ensures that $\boldsymbol{\Lambda}$ stays dual feasible, an issue that we will explain in more depth in section 8.4 of Chapter 8.

Under very general conditions, the iteration (4.4.4)-(4.4.5) converges to a primal dual optimal pair $(\boldsymbol{X}_\star, \boldsymbol{\Lambda}_\star)$, and hence yields a solution to (4.4.1). While

[13] In practice, when observations are noisy, exactly satisfying $\mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}$ is neither necessary nor desirable. We study the noisy matrix completion in Section 4.4.5, and develop dedicated algorithms for it in Chapter 8.

this algorithm appears simple, some caution is necessary: the first step is itself a nontrivial optimization problem! This subproblem has a characteristic form, which we encountered in our study of sparse recovery in noise: the objective function is a sum of a smooth convex term $f(\boldsymbol{X})$, and a nonsmooth convex function $g(\boldsymbol{X}) = \|\boldsymbol{X}\|_*$:

$$\min_{\boldsymbol{X}} \underbrace{\|\boldsymbol{X}\|_*}_{g(\boldsymbol{X}) \text{ convex}} + \underbrace{\langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}] \rangle + \tfrac{\mu}{2} \|\boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}]\|_F^2}_{f(\boldsymbol{X}) \text{ smooth, convex}}. \tag{4.4.6}$$

Here,

$$\nabla f(\boldsymbol{X}) = -\mathcal{P}_\Omega[\boldsymbol{\Lambda}] + \mu \mathcal{P}_\Omega[\boldsymbol{X} - \boldsymbol{Y}]. \tag{4.4.7}$$

This is μ-Lipschitz, in the sense that for any pair of matrices $\boldsymbol{X}$ and $\boldsymbol{X}'$,

$$\left\|\nabla f(\boldsymbol{X}) - \nabla f(\boldsymbol{X}')\right\|_F \leq \mu \left\|\boldsymbol{X} - \boldsymbol{X}'\right\|_F. \tag{4.4.8}$$

This class of problem is amenable to the *proximal gradient method*.

The general proximal gradient iteration applies to objectives of the form $F(\boldsymbol{X}) = g(\boldsymbol{X}) + f(\boldsymbol{X})$, where g is convex, and f is convex, smooth, and has L-Lipschitz gradient. See section 8.2 of Chapter 8. Here we have the Lipschitz constant $L = \mu$. So the iteration takes the form

$$\boldsymbol{X}_{k+1} = \arg\min_{\boldsymbol{X}} \left\{ g(\boldsymbol{X}) + \tfrac{\mu}{2} \left\| \boldsymbol{X} - \left(\boldsymbol{X}_k - \tfrac{1}{\mu} \nabla f(\boldsymbol{X}_k)\right) \right\|_F^2 \right\}. \tag{4.4.9}$$

In particular, it requires us to solve a sequence of "proximal problems"

$$\min_{\boldsymbol{X}} \left\{ g(\boldsymbol{X}) + \tfrac{\mu}{2} \|\boldsymbol{X} - \boldsymbol{M}\|_F^2 \right\}, \tag{4.4.10}$$

for particular choices of the matrix $\boldsymbol{M}$. When g is the nuclear norm, this problem can be solved in closed form from the SVD of $\boldsymbol{M}$. Recall from (4.3.95), for a matrix $\boldsymbol{M}$ with the singular value decomposition $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$, its singular value thresholding operator is defined to be

$$\mathcal{D}_\tau[\boldsymbol{M}] = \boldsymbol{U} \mathcal{S}_\tau[\boldsymbol{\Sigma}] \boldsymbol{V}^*,$$

where $\mathcal{S}_\tau[\boldsymbol{X}] = \operatorname{sign}(\boldsymbol{X}) \circ (|\boldsymbol{X}| - \tau)_+$ is the soft thresholding operator.

THEOREM 4.25. *The unique solution $\boldsymbol{X}_\star$ to the program:*

$$\min_{\boldsymbol{X}} \left\{ \|\boldsymbol{X}\|_* + \tfrac{\mu}{2} \|\boldsymbol{X} - \boldsymbol{M}\|_F^2 \right\}, \tag{4.4.11}$$

is given by

$$\boldsymbol{X}_\star = \mathcal{D}_{\mu^{-1}}[\boldsymbol{M}]. \tag{4.4.12}$$

The proof of this result follows from Exercise 4.13. The resulting procedures are stated as Algorithms 4.1-4.2. Here, for simplicity, we have neglected important issues such as the choice of stopping conditions, and the effect of inexact solution

Algorithm 4.1 (Matrix Completion by ALM)

1: **initialize:** $\boldsymbol{X}_0 = \boldsymbol{\Lambda}_0 = 0, \mu > 0$.
2: **while** not converged **do**
3: compute $\boldsymbol{X}_{k+1} \in \arg\min_{\boldsymbol{X}} \mathcal{L}_\mu\big(\boldsymbol{X}, \boldsymbol{\Lambda}_k\big)$ (say by Algorithm 4.2);
4: compute $\boldsymbol{\Lambda}_{k+1} = \boldsymbol{\Lambda}_k + \mu\big(\boldsymbol{Y} - \mathcal{P}_\Omega\big[\boldsymbol{X}_{k+1}\big]\big)$.
5: **end while**

Algorithm 4.2 (Proximal Gradient for Augmented Lagrangian)

1: **initialize:** $\boldsymbol{X}_0$ starts with the $\boldsymbol{X}_k$ from the outer loop of Algorithm 4.1.
2: **while** not converged **do**
3: compute

$$\begin{aligned}\boldsymbol{X}_{\ell+1} &= \operatorname{prox}_{g/\mu}\big(\boldsymbol{X}_\ell - \mu^{-1}\nabla f(\boldsymbol{X}_\ell)\big)\\ &= \mathcal{D}_{\mu^{-1}}\left[\mathcal{P}_{\Omega^c}[\boldsymbol{X}_\ell] + \boldsymbol{Y} + \mu^{-1}\mathcal{P}_\Omega\big[\boldsymbol{\Lambda}_k\big]\right].\end{aligned}$$

4: **end while**

to subproblem (4.4.4) on the convergence of the basic ALM iteration in Algorithm 4.1.

To understand when the convex program (4.4.1) and the above algorithm correctly recover a matrix $\boldsymbol{X} = \boldsymbol{X}_o$ from a part of its entries, we vary the rank r of the matrix $\boldsymbol{X}_o$ as a fraction of the dimension n and a fraction $p \in (0, 1)$ of (randomly chosen) observed entries. In other words, p is the probability that an entry is given. Figure 4.8 shows the simulation results of using the above algorithm to recover a random low-rank matrix $\boldsymbol{X}_o$ under different settings.

We may draw a few observations from the above simulations: 1. the convex program (4.4.1) and the above algorithm indeed succeed under a surprisingly wide range of conditions, as long as the rank of the matrix is relatively low and a fraction of the entries are observed. 2. the success and failure of the convex program (4.4.1) exhibit a sharp phase-transition phenomenon.

4.4.3 When Nuclear Norm Minimization Succeeds?

The above simulations encourage us to understand the conditions under which the nuclear norm minimization program (4.4.1) is guaranteed to succeed for matrix completion?[14] It may be easier to first think about when it fails. It may fail if (i) $\boldsymbol{X}_o$ is *sparse* (as in the example of $\boldsymbol{E}_{ij}$), or (ii) if the sampling pattern Ω is chosen adversarially (e.g., if we miss an entire row or column of $\boldsymbol{X}_o$). Below, we will state a theorem that makes this intuition precise – namely, if $\boldsymbol{X}_o$ is low-rank, and not too "spiky", and Ω is chosen at random, then nuclear norm

[14] Or ultimately, if possible, to precisely characterize the phase transition behavior we have observed through experiments.

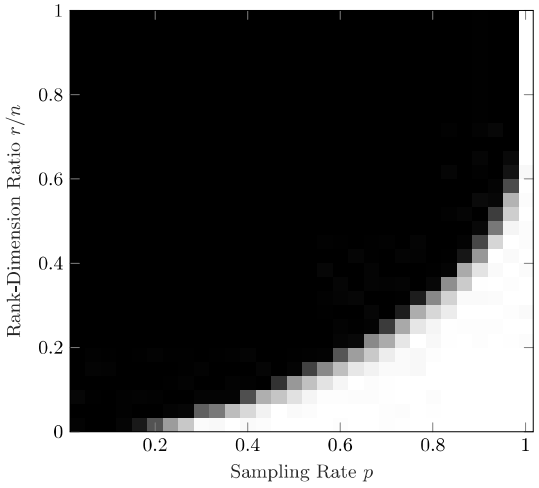


Figure 4.8 Matrix Completion for Varying Rank and Sampling Rate. Fraction of correct recoveries across 50 trials, as a function of the rank-dimension ratio r/n (vertical-axis) and fraction p of observed entries (horizontal-axis). Here, $n = 60$. In all cases, $\boldsymbol{X}_o = \boldsymbol{A}\boldsymbol{B}^*$ is a product of two independent $n \times r$ i.i.d. $\mathcal{N}(0, 1/n)$ matrices. Trials are considered successful if $\|\hat{\boldsymbol{X}} - \boldsymbol{X}_o\|_F / \|\boldsymbol{X}_o\|_F < 10^{-3}$.

minimization succeeds with high probability. Below we make these assumptions precise.

Incoherent Low-rank Matrices.
Although our intuition is that $\boldsymbol{X}_o$ itself should not be too "sparse", for technical reasons it will be necessary to enforce this condition on the singular vectors of $\boldsymbol{X}_o$, rather than on $\boldsymbol{X}_o$ itself. Let $\boldsymbol{X}_o = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ be the (reduced) singular value decomposition of $\boldsymbol{X}_o$. We say that $\boldsymbol{X}_o$ is ν-*incoherent* if the following hold:

$$\forall\, i \in [n], \quad \|\boldsymbol{e}_i^* \boldsymbol{U}\|_2^2 \leq \nu r/n, \tag{4.4.13}$$

$$\forall\, j \in [n], \quad \|\boldsymbol{e}_j^* \boldsymbol{V}\|_2^2 \leq \nu r/n. \tag{4.4.14}$$

These two conditions control the "spikiness" of the singular vectors of $\boldsymbol{X}_o$. To understand them better, note that $\boldsymbol{U}$ is an $n \times r$ matrix whose columns have unit ℓ^2 norm. Hence, $\sum_i \|\boldsymbol{e}_i^* \boldsymbol{U}\|_2^2 = \|\boldsymbol{U}\|_F^2 = r$. There are n rows, and so at least one of them must have ℓ^2 norm at least as large as the average, r/n. Hence, for any matrix $\boldsymbol{U}$ with unit norm columns, $\max_i \|\boldsymbol{e}_i^* \boldsymbol{U}\|_2^2 \geq r/n$. The *incoherence parameter* ν quantifies how much we lose with respect to this optimal bound. So, if ν is small, the singular vectors are, in a sense, spread around. To give a sense of scale, notice that it is always true that

$$1 \leq \nu \leq n/r. \tag{4.4.15}$$

If $\boldsymbol{U}$ and $\boldsymbol{V}$ are chosen uniformly at random (say by orthogonalizing the columns of a Gaussian matrix), then with high probability ν is bounded by $C \log(n)$. However, the definition does not require $\boldsymbol{U}$ and $\boldsymbol{V}$ to be random.

One important implication of this definition for matrix completion is that *when ν is small, there are no sparse matrices close to the tangent space* T. Indeed, let $\boldsymbol{E}_{ij} = \boldsymbol{e}_i \boldsymbol{e}_j^*$ denote the one-sparse matrix whose nonzero element occurs in entry (i, j). Then, using the expression (4.3.34) for the projection operator $\mathcal{P}_{\mathsf{T}}$ onto the tangent space T, we have that

$$\begin{aligned} \|\mathcal{P}_{\mathsf{T}}[\boldsymbol{E}_{ij}]\|_F^2 &= \|\boldsymbol{U}\boldsymbol{U}^* \boldsymbol{E}_{ij}\|_F^2 + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^*)\boldsymbol{E}_{ij}\boldsymbol{V}\boldsymbol{V}^*\|_F^2 \\ &\leq \|\boldsymbol{U}^* \boldsymbol{e}_i\|_2^2 + \left\|\boldsymbol{e}_j^* \boldsymbol{V}\right\|_2^2 \\ &\leq \frac{2\nu r}{n}. \end{aligned} \tag{4.4.16}$$

This indicates that no standard basis matrix $\boldsymbol{E}_{ij}$ is too close to the subspace T. Strangely enough, this implies the standard basis $\{\boldsymbol{E}_{ij}\}$ is a good choice for reconstructing elements from T. This is similar in spirit to our observations on incoherent operator bases: if no $\boldsymbol{E}_{ij}$ is too close to T, information about any particular element $\boldsymbol{X}_o \in \mathsf{T}$ must be spread across many different $\boldsymbol{E}_{ij}$. It will only take a few of these projections to be able to reconstruct $\boldsymbol{X}_o$. Note, however, a crucial difference between this notion of incoherence and our previous notions for matrix and vector recovery: here, the subspace T depends on $\boldsymbol{X}_o$ itself. The discussion in this section suggests that random sampling will be effective for reconstructing the particular matrix $\boldsymbol{X}_o$. We make this intuition formal below.

Exact Matrix Completion from Random Samples.
We assume that each entry (i, j) belongs to the set Ω independently with probability p. We call this a *Bernoulli* sampling model, since the indicators $\mathbb{1}_{(i,j)\in\Omega}$ are independent $\mathrm{Ber}(p)$ random variables. Under this model, the expected number of observed entries is

$$m = \mathbb{E}\big[\,|\Omega|\,\big] = pn^2. \tag{4.4.17}$$

Under this model, nuclear norm minimization succeeds even when the number m of observations is close to the number of intrinsic degrees of freedom in the rank-r matrix $\boldsymbol{X}_o$. The following theorem makes this precise:

THEOREM 4.26 (Matrix Completion via Nuclear Norm Minimization). *Let $\boldsymbol{X}_o \in \mathbb{R}^{n\times n}$ be a rank-r matrix with incoherence parameter ν. Suppose that we observe $\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{X}_o]$, with Ω sampled according to the Bernoulli model with probability*

$$p \geq C_1 \frac{\nu r \log^2(n)}{n}. \tag{4.4.18}$$

Then with probability at least $1 - C_2 n^{-c_3}$, $\boldsymbol{X}_o$ is the unique optimal solution to

$$\text{minimize} \quad \|\boldsymbol{X}\|_* \quad \text{subject to} \quad \mathcal{P}_\Omega[\boldsymbol{X}] = \boldsymbol{Y}. \tag{4.4.19}$$

There are several things to notice about the above theorem. First, the expected number of measurements is

$$m = pn^2 = C_1 \nu n r \log^2(n). \tag{4.4.20}$$

Since a rank-r matrix has $O(nr)$ degrees of freedom, the oversampling factor is only about $C\nu \log^2(n)$ – the number of samples we must see is nearly minimal.[15] Second, the number of samples required scales with the coherence of the matrix $\boldsymbol{X}_o$. So, if we want to recover a very coherent (think, "nearly sparse") $\boldsymbol{X}_o$, we will simply need more observations. Finally, the probability of success is in all the possible choices of the observed subset but is only for a given low-rank matrix $\boldsymbol{X}_o$. This is in contrast with the probability of success in the generic case studied in the previous sections, where an incoherent sampling operator is good for recovering the set of all matrices of rank less than r.

Of course, the precise conditions of the above theorem can only be interpreted as an idealized mathematical abstraction of real matrix completion or collaborative filtering problems. In particular, in real problems there may be noise in the observation, and, more importantly the observations may not be uniformly distributed.

4.4.4 Proving Correctness of Nuclear Norm Minimization

In this section, we prove Theorem 4.26. This section can be skipped for first time readers who are not theory oriented or are not strongly interested in the techniques needed for a rigorous proof of the theorem.

Our approach is analogous to our proof that ℓ^1 recovers sparse vectors under incoherence (in Section 3.2.2) – we simply write down the optimality conditions and try to show that they are satisfied! Carrying this program through will be trickier, though.

To get started, we need an optimality condition for the nuclear norm minimization problem (4.4.19). As mentioned in the previous section, the Lagrangian associated with the matrix completion problem (4.4.19) is

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\Lambda}) = \|\boldsymbol{X}\|_* + \langle \boldsymbol{\Lambda}, \boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{X}] \rangle \,, \tag{4.4.21}$$

[15] According to Theorem 1.7 of [CT09], if the sampling probability $p < \frac{\nu r \log(2n)}{2n}$, there will be infinitely many matrices of rank at most r that satisfy the incoherence condition and all have the same entries on Ω.

and the KKT conditions for the desired optimal $\boldsymbol{X}_o$ are such that there exist Lagrangian multipliers $\boldsymbol{\Lambda}$ that satisfy:

$$\mathcal{P}_\Omega[\boldsymbol{\Lambda}] = 0, \quad \boldsymbol{\Lambda} \in \partial \left\|\cdot\right\|_* (\boldsymbol{X}_o). \tag{4.4.22}$$

Similar to the ℓ^1 case in Section 3.2.2, such $\boldsymbol{\Lambda}$, if can be found, are called a *dual certificate* that certifies the optimality of the ground truth $\boldsymbol{X}_o$.

Subdifferential of the Nuclear Norm.
Similar to the case with ℓ^1 norm minimization, the above conditions suggest we need an expression for the subdifferential of the nuclear norm. The following lemma provides one:

LEMMA 4.27. *Let $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ have compact singular value decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. The subdifferential of the nuclear norm at $\boldsymbol{X}$ is given by*

$$\partial \left\|\cdot\right\|_* (\boldsymbol{X}) = \left\{\boldsymbol{Z} \mid \mathcal{P}_\mathsf{T}[\boldsymbol{Z}] = \boldsymbol{U}\boldsymbol{V}^*,\ \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{Z}]\| \le 1\right\}. \tag{4.4.23}$$

Proof Consider any $\boldsymbol{Z}$ satisfying $\mathcal{P}_\mathsf{T}[\boldsymbol{Z}] = \boldsymbol{U}\boldsymbol{V}^*$, and $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{Z}]\| \le 1$. Notice that $\|\boldsymbol{Z}\| = 1$. Since $\boldsymbol{X} \in \mathsf{T}$,

$$\langle \boldsymbol{X}, \boldsymbol{Z}\rangle = \langle \boldsymbol{X}, \boldsymbol{U}\boldsymbol{V}^*\rangle = \langle \boldsymbol{U}^*\boldsymbol{X}\boldsymbol{V}, \boldsymbol{I}\rangle = \langle \boldsymbol{\Sigma}, \boldsymbol{I}\rangle = \left\|\boldsymbol{X}\right\|_*. \tag{4.4.24}$$

For every $\boldsymbol{X}'$,

$$\|\boldsymbol{X}\|_* + \left\langle \boldsymbol{Z}, \boldsymbol{X}' - \boldsymbol{X}\right\rangle = \left\langle \boldsymbol{Z}, \boldsymbol{X}'\right\rangle \le \|\boldsymbol{Z}\| \left\|\boldsymbol{X}'\right\|_* = \|\boldsymbol{X}'\|_*. \tag{4.4.25}$$

Thus $\boldsymbol{Z}$ is a subgradient of the nuclear norm at $\boldsymbol{X}$: $\boldsymbol{Z} \in \partial\|\cdot\|_*(\boldsymbol{X})$. To complete the proof, we need to show that every element $\boldsymbol{Z} \in \partial\|\cdot\|_*(\boldsymbol{X})$ satisfies $\mathcal{P}_\mathsf{T}[\boldsymbol{Z}] = \boldsymbol{U}\boldsymbol{V}^*$ and $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{Z}]\| \le 1$. We leave the converse as an exercise (see Exercise 4.20). □

If we compare to the expression for the subdifferential of the ℓ^1 norm, here, the subspace T plays the role of the *support* of the matrix, while the matrix $\boldsymbol{U}\boldsymbol{V}^*$ is playing the role of the *signs*. Indeed, in this language, $\partial\left\|\cdot\right\|_*$ consists of those $\boldsymbol{Z}$ that are equal to the "sign" $\boldsymbol{U}\boldsymbol{V}^*$ on the support T, and whose dual norm $\|\cdot\|$ is bounded by one on the orthogonal complement $\mathsf{T}^\perp$ of the support.

Optimality Conditions.
Once we have the subdifferential in hand, we can fairly immediately write down an optimality condition for the convex program of interest. Indeed, consider the optimization problem

$$\begin{array}{ll} \min & \left\|\boldsymbol{X}\right\|_* \\ \text{subject to} & \mathcal{P}_\Omega[\boldsymbol{X}] = \mathcal{P}_\Omega[\boldsymbol{X}_o]. \end{array} \tag{4.4.26}$$

Any feasible $\boldsymbol{X}$ can be written as $\boldsymbol{X}_o + \boldsymbol{H}$, where $\boldsymbol{H} \in \operatorname{null}(\mathcal{P}_\Omega)$, i.e., $\boldsymbol{H}$ is supported on the set Ω^c of entries that we do not observe. Similar to the ℓ^1 case in Section 3.2.2, if we can find a dual certificate $\boldsymbol{\Lambda}$ such that it satisfies (the KKT condition):

- (i) $\boldsymbol{\Lambda}$ is supported on Ω and
- (ii) $\boldsymbol{\Lambda} \in \partial \|\cdot\|_* (\boldsymbol{X}_o)$ – i.e., $\mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] = \boldsymbol{U}\boldsymbol{V}^*$ and $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| \leq 1$,

then we have

$$\|\boldsymbol{X}_o + \boldsymbol{H}\|_* \geq \|\boldsymbol{X}_o\|_* + \langle \boldsymbol{\Lambda}, \boldsymbol{H} \rangle = \|\boldsymbol{X}_o\|_* , \tag{4.4.27}$$

where the final equality holds because $\boldsymbol{\Lambda}$ is supported on Ω and $\boldsymbol{H}$ is supported on Ω^c. In addition, if we further have $\|\mathcal{P}_{\Omega^c}\mathcal{P}_{\mathsf{T}}\| < 1$ and $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| < 1$, then one can show that $\boldsymbol{X}_o$ is the *unique* optimal solution. The proof is similar to that in the ℓ^1 case (see the proof of Theorem 3.3) and we leave to the reader as an exercise (see Exercise 4.16).

A natural idea for constructing $\boldsymbol{\Lambda}$ might be to simply follow the program that has worked before (in the ℓ^1 minimization case) and look for a matrix $\boldsymbol{\Lambda}$ of smallest 2-norm that satisfies the equality constraints

$$\mathcal{P}_{\Omega^c}[\boldsymbol{\Lambda}] = \boldsymbol{0}, \quad \mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] = \boldsymbol{U}\boldsymbol{V}^*, \tag{4.4.28}$$

and then hope to check that it satisfies the inequality constraints

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| \leq 1.$$

For example, we could take $\boldsymbol{\Lambda} = \mathcal{P}_{\Omega}[\boldsymbol{G}]$, with $\boldsymbol{G} = (\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega})^{\dagger}[\boldsymbol{U}\boldsymbol{V}^*]$, where $(\cdot)^{\dagger}$ denotes the pseudo inverse. We are then left to check that

$$\left\|\mathcal{P}_{\mathsf{T}^\perp}\mathcal{P}_{\Omega}(\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega})^{\dagger}[\boldsymbol{U}\boldsymbol{V}^*]\right\| \tag{4.4.29}$$

is small. This is a random matrix, but it is an exceedingly complicated one. It actually *is* possible to analyze its norm, but the analysis is quite intricate. The challenge arises because the thing that is random here is the support Ω. It is repeated in several places, creating probabilistic dependencies, which complicates the analysis.

Relaxed Optimality Conditions.
As it is difficult to directly find a dual certificate satisfying the KKT conditions exactly, we might want to relax these conditions and see if we could still find another certificate for the optimality. The following proposition suggests that we can ensure the optimality of $\boldsymbol{X}_o$ with an alternative set of (relaxed) conditions:

PROPOSITION 4.28 (KKT Conditions – Approximate Version). *The matrix $\boldsymbol{X}_o$ is the unique optimal solution to the nuclear minimization problem (4.4.19) if the following set of conditions hold*

1 The operator norm of the operator $p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}$ is small:

$$\left\|p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}\right\| \leq \tfrac{1}{2}.$$

2 There exists a dual certificate $\boldsymbol{\Lambda} \in \mathbb{R}^{n\times n}$ that satisfies $\mathcal{P}_{\Omega}[\boldsymbol{\Lambda}] = \boldsymbol{\Lambda}$ and
   1 $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| \leq \frac{1}{2}$;
   2 $\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \frac{1}{4n}$.

Conditions 2(a) and 2(b) above trade off between the degree of satisfaction of the equality constraint $\mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] = \boldsymbol{U}\boldsymbol{V}^*$ and the inequality constraint for the dual norm $\|\mathcal{P}_{\mathsf{T}^\perp}\boldsymbol{\Lambda}\| \leq 1$ in the original KKT conditions. This is possible under the additional assumption that $\left\|p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}\right\|$ is not too large. This assumption is satisfied whenever the sampling map $p^{-1}\mathcal{P}_\Omega$ nearly preserves the length of all elements $\boldsymbol{X} \in \mathsf{T}$ – in other words, restricted on T the operator $p^{-1}\mathcal{P}_\Omega$ is nearly *isometric*. It can be considered a strengthening of the condition that $\mathsf{T} \cap \Omega^\perp = \{\mathbf{0}\}$, which was needed for unique optimality.

To prove Proposition 4.28, we will need another lemma. This says that provided $\mathcal{P}_\Omega$ acts nicely on matrices from T, every feasible perturbation $\boldsymbol{H}$ (i.e., $\boldsymbol{H}$ such that $\mathcal{P}_\Omega[\boldsymbol{H}] = \mathbf{0}$) must have a non-negligible component along $\mathsf{T}^\perp$:

Lemma 4.29. *Suppose that the operator $\mathcal{P}_\Omega$ satisfies*

$$\left\|\mathcal{P}_{\mathsf{T}} - p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}\right\| \leq \frac{1}{2}. \tag{4.4.30}$$

Then for any $\boldsymbol{H}$ satisfying $\mathcal{P}_\Omega[\boldsymbol{H}] = \mathbf{0}$, we have

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_F \geq \sqrt{\frac{p}{2}}\,\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F\,. \tag{4.4.31}$$

Proof We have

$$\begin{aligned}
\langle \mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}], \mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\rangle &= \langle \mathcal{P}_{\mathsf{T}}[\boldsymbol{H}], \mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\rangle \\
&= p\langle \mathcal{P}_{\mathsf{T}}[\boldsymbol{H}], p^{-1}\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\rangle \\
&= p\langle \mathcal{P}_{\mathsf{T}}[\boldsymbol{H}], \mathcal{P}_{\mathsf{T}}p^{-1}\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\rangle \\
&\geq p\Big(1 - \left\|\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}p^{-1}\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}\right\|\Big)\,\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F^2 \\
&\geq \frac{p}{2}\,\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F^2\,,
\end{aligned} \tag{4.4.32}$$

Then from $\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}] + \mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] = \mathcal{P}_\Omega[\boldsymbol{H}] = \mathbf{0}$, we have

$$\begin{aligned}
0 &= \|\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}] + \mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_F \\
&\geq \|\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F - \|\mathcal{P}_\Omega\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_F \\
&\geq \sqrt{\frac{p}{2}}\,\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F - \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_F\,,
\end{aligned} \tag{4.4.33}$$

giving the conclusion. □

We are now ready to prove the optimality of $\boldsymbol{X}_o$ under the conditions given by Proposition 4.28.

Proof We want to show that under the above conditions, for any feasible perturbation $\boldsymbol{H} \neq \mathbf{0}$ and $\boldsymbol{X} = \boldsymbol{X}_o + \boldsymbol{H}$, we have $\|\boldsymbol{X}\|_* > \|\boldsymbol{X}_o\|_*$. Let $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] = \bar{\boldsymbol{U}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^*$. Then we have $\bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^* \in \mathsf{T}^\perp$ and $\left\|\bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^*\right\| \leq 1$. Therefore, we have $\boldsymbol{U}\boldsymbol{V}^* + \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^* \in \partial\,\|\cdot\|_*(\boldsymbol{X}_o)$ is a subgradient of the nuclear norm at $\boldsymbol{X}_o$.

Also, we have $\langle \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^*, \mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\rangle = \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*$ and $\langle \boldsymbol{\Lambda}, \boldsymbol{H}\rangle = 0$ and apply them to the following inequalities:

$$\begin{aligned}
\|\boldsymbol{X}_o + \boldsymbol{H}\|_* &\geq \|\boldsymbol{X}_o\|_* + \langle \boldsymbol{U}\boldsymbol{V}^* + \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^*, \boldsymbol{H}\rangle, \\
&= \|\boldsymbol{X}_o\|_* + \langle \boldsymbol{U}\boldsymbol{V}^* + \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^* - \boldsymbol{\Lambda}, \boldsymbol{H}\rangle, \\
&= \|\boldsymbol{X}_o\|_* + \langle \boldsymbol{U}\boldsymbol{V}^* - \mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}], \boldsymbol{H}\rangle + \langle \bar{\boldsymbol{U}}\bar{\boldsymbol{V}}^* - \mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}], \boldsymbol{H}\rangle, \\
&\geq \|\boldsymbol{X}_o\|_* - \frac{1}{4n}\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{H}]\|_F + \frac{1}{2}\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_*, \\
&\geq \|\boldsymbol{X}_o\|_* + \underbrace{\left(\tfrac{1}{2} - \tfrac{1}{4n}\sqrt{\tfrac{2}{p}}\right)}_{>0, \text{ since } p > n^{-2}.} \|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}]\|_F. \qquad (4.4.34)
\end{aligned}$$

In the final inequality, we have invoked Lemma 4.29.

Hence, for feasible perturbations $\boldsymbol{H}$, $\|\boldsymbol{X}_o + \boldsymbol{H}\|_* \geq \|\boldsymbol{X}_o\|_*$, with equality if and only if $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] = \boldsymbol{0}$. But via Lemma 4.29, $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{H}] = \boldsymbol{0} \implies \boldsymbol{H} = \boldsymbol{0}$. Thus, for any nonzero feasible perturbation $\boldsymbol{H}$, $\|\boldsymbol{X}_o + \boldsymbol{H}\|_* > \|\boldsymbol{X}_o\|_*$, establishing the desired condition. □

The Optimality Condition is Satisfied with High Probability.
To complete the proof, we simply need to show that the optimality condition can be satisfied with high probability. To do this, we need to verify two claims: first, that with high probability the sampling operator Ω acts nicely on T, in the sense that $\left\|p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}\right\|$ is small. We then need to show that with high probability we can construct the desired dual certificate $\boldsymbol{\Lambda}$.

1. The sampling operator acts nicely on T:
We next prove that the sampling operator $\mathcal{P}_{\Omega}$ preserves some part of every element of T, in the sense that $\left\|p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathsf{T}} - \mathcal{P}_{\mathsf{T}}\right\|$ is small. This phenomenon is a consequence of the incoherence of the matrix $\boldsymbol{X}_o$ and the uniform random model on Ω. The proof of the following lemma uses the matrix (operator) Bernstein inequality to show this rigorously.

LEMMA 4.30. *Let $\mathcal{P}_{\Omega} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ denote the operator*

$$\mathcal{P}_{\Omega}[\boldsymbol{X}] = \sum_{ij} X_{ij}\mathbb{1}_{(i,j)\in\Omega}\boldsymbol{E}_{ij} \qquad (4.4.35)$$

with $\mathbb{1}_{(i,j)\in\Omega}$ independent Bernoulli random variables with probability p. Fix any ε with $c\frac{\sqrt{\log n}}{n} \leq \varepsilon \leq 1$. There is a numerical constant C such that if $p > C\frac{\nu r \log n}{\varepsilon^2 n}$, then with high probability,

$$\left\|\mathcal{P}_{\mathsf{T}} - p^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega}\mathcal{P}_{\mathsf{T}}\right\| \leq \varepsilon. \qquad (4.4.36)$$

Proof We apply the matrix Bernstein inequality in Theorem E.8 to bound the

norm of

$$\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_\Omega\mathcal{P}_T = \sum_{ij} \underbrace{\mathcal{P}_T\left(\frac{\mathcal{I}}{n^2} - p^{-1}\mathbb{1}_{(i,j)\in\Omega}\boldsymbol{E}_{ij}\langle \boldsymbol{E}_{ij},\cdot\rangle\right)\mathcal{P}_T}_{\doteq \mathcal{W}_{ij}}.$$

Here, $\mathcal{W}_{ij} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ are independent random linear maps, and $\mathbb{E}\left[\sum_{ij}\mathcal{W}_{ij}\right] = 0$. The matrix Bernstein inequality requires (i) an almost sure bound R on $\max_{ij}\|\mathcal{W}_{ij}\|$, and (ii) control of the "variance"

$$\sum_{ij}\mathbb{E}\left[\mathcal{W}_{ij}^*\mathcal{W}_{ij}\right]. \tag{4.4.37}$$

We provide these as follows.
(i) Almost sure control of the summands:

$$\begin{aligned}
\|\mathcal{W}_{ij}\| &\le \max\left\{\left\|n^{-2}\mathcal{P}_T\right\|, \left\|p^{-1}\mathcal{P}_T[\boldsymbol{E}_{ij}]\langle\mathcal{P}_T[\boldsymbol{E}_{ij}],\cdot\rangle\right\|\right\}, \quad \text{almost surely}\\
&= \max\left\{n^{-2}, p^{-1}\|\mathcal{P}_T[\boldsymbol{E}_{ij}]\|_F^2\right\},\\
&\le \max\left\{n^{-2}, \frac{2\nu r}{np}\right\},\\
&\le \max\left\{\frac{1}{n^2}, \frac{2\varepsilon^2}{C\log n}\right\},\\
&= \frac{2\varepsilon^2}{C\log n}.
\end{aligned} \tag{4.4.38}$$

We may take $R = \frac{2\varepsilon^2}{C\log n}$.
(ii) Control of the "operator variance". Note that

$$\begin{aligned}
&\sum_{ij}\mathbb{E}\left[\mathcal{W}_{ij}^*\mathcal{W}_{ij}\right]\\
&= \sum_{ij}\mathbb{E}\Big[\frac{1}{n^4}\mathcal{P}_T - \frac{2p^{-1}}{n^2}\mathbb{1}_{(i,j)\in\Omega}\mathcal{P}_T\boldsymbol{E}_{ij}\langle\boldsymbol{E}_{ij},\cdot\rangle\mathcal{P}_T\\
&\qquad\qquad + \mathbb{1}_{(i,j)\in\Omega}p^{-2}\mathcal{P}_T\boldsymbol{E}_{ij}\|\mathcal{P}_T\boldsymbol{E}_{ij}\|_F^2\langle\boldsymbol{E}_{ij},\cdot\rangle\mathcal{P}_T\Big]\\
&\preceq p^{-1}\sum_{ij}\mathcal{P}_T\boldsymbol{E}_{ij}\|\mathcal{P}_T\boldsymbol{E}_{ij}\|_F^2\langle\boldsymbol{E}_{ij},\cdot\rangle\mathcal{P}_T\\
&\preceq p^{-1}\frac{2\nu r}{n}\sum_{ij}\mathcal{P}_T\boldsymbol{E}_{ij}\langle\boldsymbol{E}_{ij},\cdot\rangle\mathcal{P}_T\\
&\preceq \frac{2\varepsilon^2}{C\log n}\mathcal{P}_T.
\end{aligned} \tag{4.4.39}$$

The operator $\sum_{ij}\mathbb{E}\left[\mathcal{W}_{ij}^*\mathcal{W}_{ij}\right]$ is self-adjoint and positive semidefinite. The above

calculation therefore implies that

$$\sigma^2 = \max\left\{\left\|\sum_{ij}\mathbb{E}\left[\mathcal{W}_{ij}^*\mathcal{W}_{ij}\right]\right\|, \left\|\sum_{ij}\mathbb{E}\left[\mathcal{W}_{ij}\mathcal{W}_{ij}^*\right]\right\|\right\} \leq \frac{2\varepsilon^2}{C\log n}. \tag{4.4.40}$$

Using these calculations, we obtain a bound

$$\mathbb{P}\left[\left\|\sum_{ij}\mathcal{W}_{ij}\right\| > t\right] \leq 2n\exp\left(\frac{-t^2/2}{\frac{2\varepsilon^2}{C\log n} + t\frac{2\varepsilon^2}{3C\log n}}\right). \tag{4.4.41}$$

The probability of failure for $t = \varepsilon$ is bounded by $n^{-\rho}$; the exponent ρ can be made as large as desired by choosing C appropriately. □

Choosing $\varepsilon = 1/2$ in the statement of the above lemma, we obtain the desired condition needed for Lemma 4.29.

2. Construction of a dual certificate by the golfing scheme.
From the above discussion, in order to prove Theorem 4.26, we only have to show that under the conditions of the theorem, we can find a dual certificate that satisfies two conditions $2(a)$ and $2(b)$ of Proposition 4.28. In this section, we show how to construct such a dual certificate, $\boldsymbol{\Lambda}$. In the next chapter, we will reuse this construction to analyze the related problem of *robust matrix recovery*, in which a fraction of the entries of a low-rank matrix have been corrupted. For this purpose, we give a complete summary of the properties of our construction in the following proposition. Here, properties (i) and (ii) are essential for matrix completion; property (iii) will be used in the following chapters for robust matrix recovery.

PROPOSITION 4.31 (Dual Certificate for Low-rank Recovery). *Let $\boldsymbol{X}_o \in \mathbb{R}^{n\times n}$ be a rank-r matrix, with coherence ν. Let $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$ be matrices whose columns are leading left- and right singular vectors of $\boldsymbol{X}_o$. Let*

$$\mathsf{T} = \left\{\boldsymbol{U}\boldsymbol{X}^* + \boldsymbol{Y}\boldsymbol{V}^* \mid \boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{n\times r}\right\}. \tag{4.4.42}$$

Then if $\Omega \sim \mathrm{Ber}(p)$, with

$$p > C_0\frac{\nu r\log^2 n}{n}, \tag{4.4.43}$$

there exists a matrix $\boldsymbol{\Lambda}$ supported on Ω, satisfying

1 $\|\mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \frac{1}{4n}$,
2 $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| \leq \frac{1}{4}$,
3 $\|\boldsymbol{\Lambda}\|_\infty < \frac{C_1\log n}{p} \times \|\boldsymbol{U}\boldsymbol{V}^*\|_\infty$,

with high probability. Here, C_1 is a positive numerical constant.

We prove this proposition using an iterative construction. Let

$$\Omega_1, \ldots, \Omega_k \tag{4.4.44}$$

be *independent* random subsets, chosen according to the Bernoulli model with parameter q. Set

$$\Omega = \bigcup_{i=1}^{k} \Omega_i. \tag{4.4.45}$$

Then Ω is *also* a Bernoulli subset, with parameter

$$p = 1 - (1-q)^k. \tag{4.4.46}$$

The parameter p is the probability that a given entry is in *at least one* of the subsets Ω_i. Hence, $p \leq kq$. The argument that we develop below will lead us to choose $k = C_g \log(n)$, with C_g a constant. Because k is not too large, this implies that the parameter q is also not too small:

$$q \geq \frac{p}{k} = \frac{C_0}{C_g} \frac{\nu r \log n}{n}. \tag{4.4.47}$$

Provided C_0 is large enough compared to C_g, the subsets Ω_i *all* satisfy the conditions of Lemma 4.30, and so with high probability

$$\left\| \mathcal{P}_{\mathsf{T}} - q^{-1} \mathcal{P}_{\mathsf{T}} \mathcal{P}_{\Omega_j} \mathcal{P}_{\mathsf{T}} \right\| \leq \tfrac{1}{2}, \quad j = 1, \ldots, k. \tag{4.4.48}$$

We will construct a sequence of matrices $\boldsymbol{\Lambda}_0, \boldsymbol{\Lambda}_1, \ldots, \boldsymbol{\Lambda}_k$, in which each $\boldsymbol{\Lambda}_j$ depends only on $\Omega_1, \ldots, \Omega_j$. We let $\boldsymbol{\Lambda}_0 = \mathbf{0}$. And let

$$\boldsymbol{E}_j = \mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}_j] - \boldsymbol{U}\boldsymbol{V}^*. \tag{4.4.49}$$

Since our goal is to obtain $\boldsymbol{\Lambda}$ such that $\mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}] \approx \boldsymbol{U}\boldsymbol{V}^*$, $\boldsymbol{E}_j$ should be considered the *error* at iteration j. To get our next $\boldsymbol{\Lambda}$, we simply try to correct the error:

$$\boldsymbol{\Lambda}_j = \boldsymbol{\Lambda}_{j-1} - \left(q^{-1}\mathcal{P}_{\Omega_j}\right)[\boldsymbol{E}_{j-1}]. \tag{4.4.50}$$

This construction is known as the *golfing scheme*, as it tries to reach the goal by reducing error step by step.

There are several things worth noting about this construction. First, it produces $\boldsymbol{\Lambda}_j$ supported only on $\Omega_1 \cup \cdots \cup \Omega_j$. Thus, as desired, $\boldsymbol{\Lambda}_k$ is supported on Ω. Second, because $\boldsymbol{U}\boldsymbol{V}^* \in \mathsf{T}$, $\boldsymbol{E}_j \in \mathsf{T}$ for each j. This means that

$$\begin{aligned}
\boldsymbol{E}_j &= \mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}_j] - \boldsymbol{U}\boldsymbol{V}^* \\
&= \mathcal{P}_{\mathsf{T}}[\boldsymbol{\Lambda}_{j-1}] - \boldsymbol{U}\boldsymbol{V}^* - q^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_j}[\boldsymbol{E}_{j-1}] \\
&= \boldsymbol{E}_j - q^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_j}[\boldsymbol{E}_{j-1}] \\
&= (\mathcal{P}_{\mathsf{T}} - q^{-1}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_j}\mathcal{P}_{\mathsf{T}})[\boldsymbol{E}_{j-1}].
\end{aligned}$$

Since $\mathbb{E}\left[q^{-1}\mathcal{P}_{\Omega_j}\right] = \mathcal{I}$, in expectation, this iterative process drives the error to zero: $\mathbb{E}[\boldsymbol{E}_j] = \mathbf{0}$.

As it turns out, due to the fact that $\left\|\mathcal{P}_T - q^{-1}\mathcal{P}_T\mathcal{P}_{\Omega_j}\mathcal{P}_T\right\| \leq \frac{1}{2}$, after k steps, the error reduces to

$$\|\mathcal{P}_T[\boldsymbol{\Lambda}_k] - \boldsymbol{U}\boldsymbol{V}^*\|_F = \|\boldsymbol{E}_k\|_F \leq 2^{-k}\|\boldsymbol{E}_0\|_F \tag{4.4.51}$$

with high probability.

So, based on the golfing scheme, to achieve the desired accuracy as suggested by the above lemma, we want $2^{-k}\|\boldsymbol{E}_0\|_F = 2^{-k}\sqrt{r} \leq \frac{1}{4n}$. Since $r < n$, we only need to have $2^{-k} \sim O(1/n^2)$, that is to choose $k = C_g \log(n)$ for some large enough constant C_g, say $C_g = 20$. Therefore, under these conditions, the dual certificate constructed after k iterations $\boldsymbol{\Lambda}_k$ satisfies condition 2 (b) of Proposition 4.28:

$$\|\mathcal{P}_T[\boldsymbol{\Lambda}_k] - \boldsymbol{U}\boldsymbol{V}^*\|_F \leq \frac{1}{4n}. \tag{4.4.52}$$

Finally, to satisfy Condition 2(a) of Proposition 4.28, we need to show that the operator norm of the random matrix $\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}_k]$ is bounded as

$$\|\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}_k]\| \leq 1/4.$$

Notice that from the construction of $\boldsymbol{\Lambda}_k$, we have

$$\begin{aligned}
\boldsymbol{\Lambda}_k &= \sum_{j=1}^{k} -q^{-1}\mathcal{P}_{\Omega_j}[\boldsymbol{E}_{j-1}], \\
\boldsymbol{E}_j &= (\mathcal{P}_T - \mathcal{P}_T q^{-1}\mathcal{P}_{\Omega_j}\mathcal{P}_T)[\boldsymbol{E}_{j-1}], \quad \text{with } \boldsymbol{E}_0 = -\boldsymbol{U}\boldsymbol{V}^*.
\end{aligned}$$

The matrix of interest can be expressed as

$$\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}_k] = \sum_{j=1}^{k} -q^{-1}\mathcal{P}_{T^\perp}\mathcal{P}_{\Omega_j}[\boldsymbol{E}_{j-1}] = \sum_{j=1}^{k} \mathcal{P}_{T^\perp}(\mathcal{P}_T - q^{-1}\mathcal{P}_{\Omega_j}\mathcal{P}_T)[\boldsymbol{E}_{j-1}], \tag{4.4.53}$$

where the second identity is due to $\mathcal{P}_{T^\perp}\mathcal{P}_T = 0$ and $\mathcal{P}_T[\boldsymbol{E}_j] = \boldsymbol{E}_j$.

Since we are interested in bounding the norm of $\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}_k]$, it would help if we know good bounds on various norms of $\mathcal{P}_{\Omega_j}$ and its interaction with the operator $\mathcal{P}_T$ or $\mathcal{P}_{T^\perp}$. Notice each $\mathcal{P}_{\Omega_j}$ is a summation of independent random operators. A very powerful tool we can use to bound the norm of summation of random matrices (or operators) is the so-called matrix Bernstein inequality introduced in the Appendix E, which we have used once before in Lemma 4.30.

To bound the norm of $\mathcal{P}_{T^\perp}[\boldsymbol{\Lambda}_k]$, we need good bounds on three additional operators similar to that in Lemma 4.30. The proofs of these bounds[16] are all similar to that of Lemma 4.30 by utilizing the matrix Bernstein inequality. We hence leave their derivations as exercises to the reader to get familiar with the matrix Bernstein inequality.

We phrase these bounds in terms of

$$\|\boldsymbol{Z}\|_\infty = \max_{ij}|Z_{ij}|, \tag{4.4.54}$$

[16] following the work of [CJSC13].

and the maximum of the largest ℓ^2 norm of a row and the largest ℓ^2 norm of a column, which we denote by $\|\cdot\|_{rc}$:

$$\|\boldsymbol{Z}\|_{rc} = \max\left\{\max_i \|\boldsymbol{e}_i^*\boldsymbol{Z}\|_2\,,\ \max_j \|\boldsymbol{Z}\boldsymbol{e}_j\|_2\right\}. \tag{4.4.55}$$

LEMMA 4.32. *Let $\boldsymbol{Z}$ be any fixed $n \times n$ matrix, and Ω a $\mathrm{Ber}(q)$ subset, with*

$$q > C_0 \frac{\nu r \log n}{n}. \tag{4.4.56}$$

Then with high probability

$$\left\|\left(q^{-1}\mathcal{P}_\Omega - \mathcal{I}\right)[\boldsymbol{Z}]\right\| \le C\Big(\frac{n}{C_0\nu r}\|\boldsymbol{Z}\|_\infty + \sqrt{\frac{n}{C_0\nu r}}\|\boldsymbol{Z}\|_{rc}\Big), \tag{4.4.57}$$

where C is a numerical constant.

Proof Exercise 4.23. □

LEMMA 4.33. *Let $\boldsymbol{Z}$ be any fixed $n \times n$ matrix. There exists a numerical constant C_0 such that if Ω is a $\mathrm{Ber}(q)$ subset with*

$$q > C_0 \frac{\nu r \log n}{n}, \tag{4.4.58}$$

then with high probability

$$\left\|(q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega - \mathcal{P}_\mathsf{T})[\boldsymbol{Z}]\right\|_{rc} \le \frac{1}{2}\Big(\sqrt{\frac{n}{\nu r}}\|\boldsymbol{Z}\|_\infty + \|\boldsymbol{Z}\|_{rc}\Big). \tag{4.4.59}$$

Proof Exercise 4.24. □

LEMMA 4.34. *Suppose $\boldsymbol{Z}$ is a fixed $n \times n$ matrix in T. There exists a constant C_0 such that if Ω is a Bernoulli(q) subset with*

$$q > C_0 \frac{\nu r \log n}{n}. \tag{4.4.60}$$

Then with high probability we have

$$\left\|(\mathcal{P}_\mathsf{T} - q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega\mathcal{P}_\mathsf{T})[\boldsymbol{Z}]\right\|_\infty \le \tfrac{1}{2}\|\boldsymbol{Z}\|_\infty. \tag{4.4.61}$$

Proof Exercise 4.25. □

With these three lemmas in hand, we are now ready to show that the spectral norm of $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]$ is very small, in particular can be bounded as $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]\| \le 1/4$:

Proof From the golfing construction, $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]$ can be expressed as the series given in (4.4.53). Hence we have

$$\begin{aligned}\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]\| &\le \sum_{j=1}^{k} \left\|\mathcal{P}_{\mathsf{T}^\perp}(\mathcal{P}_{\mathsf{T}} - q^{-1}\mathcal{P}_{\Omega_j}\mathcal{P}_{\mathsf{T}})[\boldsymbol{E}_{j-1}]\right\| \\ &\le \sum_{j=1}^{k} \left\|(\mathcal{P}_{\mathsf{T}} - q^{-1}\mathcal{P}_{\Omega_j}\mathcal{P}_{\mathsf{T}})[\boldsymbol{E}_{j-1}]\right\| \\ &= \sum_{j=1}^{k} \left\|(\mathcal{I} - q^{-1}\mathcal{P}_{\Omega_j})[\boldsymbol{E}_{j-1}]\right\|. \end{aligned} \tag{4.4.62}$$

Notice that in the construction of the golfing scheme, we have ensured that each subset Ω_j is sampled according to the Bernoulli model, with parameter $q > C_0\frac{\nu r \log n}{n}$ for some large enough C_0. This means each of the k subsets Ω_j satisfies the conditions of the above lemmas. We first apply Lemma 4.32 to the right hand side of the last inequality and obtain (assuming $C_0 > 1$):

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]\| \le \frac{C}{\sqrt{C_0}} \sum_{j=1}^{k} \Big(\frac{n}{\nu r} \|\boldsymbol{E}_{j-1}\|_\infty + \sqrt{\frac{n}{\nu r}}\, \|\boldsymbol{E}_{j-1}\|_{rc} \Big). \tag{4.4.63}$$

To bound $\|\boldsymbol{E}_{j-1}\|_\infty$ we apply Lemma 4.34 and obtain

$$\begin{aligned}\|\boldsymbol{E}_{j-1}\|_\infty &= \left\|(\mathcal{P}_{\mathsf{T}} - \tfrac{1}{q}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_{j-1}}\mathcal{P}_{\mathsf{T}}) \cdots (\mathcal{P}_{\mathsf{T}} - \tfrac{1}{q}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_1}\mathcal{P}_{\mathsf{T}})[\boldsymbol{E}_0]\right\|_\infty \\ &\le \Big(\frac{1}{2}\Big)^{j-1} \|\boldsymbol{U}\boldsymbol{V}^*\|_\infty . \end{aligned} \tag{4.4.64}$$

Using this together with the fact that $\boldsymbol{\Lambda}_k = -\sum_j q^{-1}\mathcal{P}_{\Omega_j}[\boldsymbol{E}_{j-1}]$, we obtain

$$\|\boldsymbol{\Lambda}_k\|_\infty \le q^{-1} \sum_j \|\boldsymbol{E}_{j-1}\|_\infty \tag{4.4.65}$$

$$\le 2q^{-1} \|\boldsymbol{U}\boldsymbol{V}^*\|_\infty . \tag{4.4.66}$$

Since $q > p/C_q \log n$, this establishes property (iii) of Proposition 4.31 for $\boldsymbol{\Lambda}_k$.

To bound $\|\boldsymbol{E}_{j-1}\|_{rc}$ we apply Lemma 4.33 and obtain

$$\begin{aligned}\|\boldsymbol{E}_{j-1}\|_{rc} &= \left\|(\mathcal{P}_{\mathsf{T}} - \tfrac{1}{q}\mathcal{P}_{\mathsf{T}}\mathcal{P}_{\Omega_{j-1}}\mathcal{P}_{\mathsf{T}})[\boldsymbol{E}_{j-2}]\right\|_{rc} \\ &\le \frac{1}{2}\sqrt{\frac{n}{\nu r}}\, \|\boldsymbol{E}_{j-2}\|_\infty + \frac{1}{2}\|\boldsymbol{E}_{j-1}\|_{rc} . \end{aligned} \tag{4.4.67}$$

Combine the above two inequalities and apply them recursively to $j-1, j-2, \ldots, 0$ and we obtain

$$\|\boldsymbol{E}_{j-1}\|_{rc} \le j\Big(\frac{1}{2}\Big)^{j-1} \sqrt{\frac{n}{\nu r}}\, \|\boldsymbol{U}\boldsymbol{V}^*\|_\infty + \Big(\frac{1}{2}\Big)^{j-1} \|\boldsymbol{U}V^*\|_{rc} . \tag{4.4.68}$$

Substitute the bounds (4.4.64) and (4.4.68) to the right and side of (4.4.63)

and we obtain

$$\begin{aligned}\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]\| &\le \frac{C}{\sqrt{C_0}}\frac{n}{\nu r}\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty \sum_{j=1}^{k}(j+1)\Big(\frac{1}{2}\Big)^{j-1} \\ &\quad + \frac{C}{\sqrt{C_0}}\sqrt{\frac{n}{\nu r}}\|\boldsymbol{U}\boldsymbol{V}^*\|_{rc}\sum_{j=1}^{k}\Big(\frac{1}{2}\Big)^{j-1} \\ &\le \frac{6C}{\sqrt{C_0}}\frac{n}{\nu r}\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty + \frac{2C}{\sqrt{C_0}}\sqrt{\frac{n}{\nu r}}\|\boldsymbol{U}\boldsymbol{V}^*\|_{rc}. \end{aligned} \tag{4.4.69}$$

As the matrix $\boldsymbol{X}_o$ satisfies the incoherence conditions (4.4.13) and (4.4.14), we have

$$\begin{aligned}\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty &\le \max_{i,j}\Big\{\|\boldsymbol{U}^*\boldsymbol{e}_i\|_2 \times \|\boldsymbol{V}^*\boldsymbol{e}_j\|_2\Big\} \le \frac{\nu r}{n}, \\ \|\boldsymbol{U}\boldsymbol{V}^*\|_{rc} &\le \max\left\{\max_i \|\boldsymbol{e}_i^*\boldsymbol{U}\boldsymbol{V}^*\|_2, \max_j \|\boldsymbol{U}\boldsymbol{V}^*\boldsymbol{e}_j\|_2\right\} \le \sqrt{\frac{\nu r}{n}}.\end{aligned}$$

Therefore,

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]\| \le \frac{6C}{\sqrt{C_0}} + \frac{2C}{\sqrt{C_0}} \le \frac{1}{4} \tag{4.4.70}$$

for large enough C_0. This establishes property (ii) of Proposition 4.31 for $\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}_k]$. □

The above derivations and results show that the relaxed KKT conditions in Proposition 4.28 can be satisfied with high probability, proving Theorem 4.26.

4.4.5 Stable Matrix Completion with Noise

So far in the matrix completion problem, we have assumed that the observed entries are precise. In real world matrix completion problems, the observed entries are often corrupted with some noise:

$$Y_{ij} = [\boldsymbol{X}_o]_{ij} + Z_{ij}, \quad (i,j) \in \Omega, \tag{4.4.71}$$

where Z_{ij} can be some small noise. Or equivalently, we can write

$$\mathcal{P}_\Omega[\boldsymbol{Y}] = \mathcal{P}_\Omega[\boldsymbol{X}_o] + \mathcal{P}_\Omega[\boldsymbol{Z}], \tag{4.4.72}$$

where $\boldsymbol{Z}$ is an $n \times n$ matrix of noises. We may assume that the overall noise level is small $\|\mathcal{P}_\Omega[\boldsymbol{Z}]\|_F < \varepsilon$. As in the stable matrix recovery case, we could expect to recover a low rank matrix matrix close to $\boldsymbol{X}_o$ via solving the following convex program:

$$\begin{aligned} &\min && \|\boldsymbol{X}\|_* \\ &\text{subject to} && \|\mathcal{P}_\Omega[\boldsymbol{X}] - \mathcal{P}_\Omega[\boldsymbol{Y}]\|_F < \varepsilon. \end{aligned} \tag{4.4.73}$$

The following theorem states that under the same conditions of Theorem 4.26 when the nuclear norm minimization recovers the correct low rank matrix from

noiseless measurements, the above program gives a stable estimate $\hat{\boldsymbol{X}}$ of the true low-rank matrix $\boldsymbol{X}_o$:

THEOREM 4.35 (Stable Matrix Completion). *Let $\boldsymbol{X}_o \in \mathbb{R}^{n\times n}$ be a rank-r, ν-incoherent matrix. Suppose that we observe $\mathcal{P}_\Omega[\boldsymbol{Y}] = \mathcal{P}_\Omega[\boldsymbol{X}_o] + \mathcal{P}_\Omega[\boldsymbol{Z}]$, where Ω is a subset of $[n]\times[n]$. If Ω is uniformly sampled from subsets of size*

$$m \geq C_1 \nu n r \log^2(n), \tag{4.4.74}$$

then with high probability, the optimal solution $\hat{\boldsymbol{X}}$ to the convex program (4.4.73) satisfies

$$\|\hat{\boldsymbol{X}} - \boldsymbol{X}_o\|_F \leq c\frac{n\sqrt{n}\log(n)}{\sqrt{m}}\varepsilon \leq c'\frac{n}{\sqrt{r}}\varepsilon \tag{4.4.75}$$

for some constant $c > 0$.

Proof Similar to the proof of Theorem 4.26 in the noiseless case which has the same incoherence condition on $\boldsymbol{X}_o$ and the sampling condition, we know the sampling operator $\mathcal{P}_\Omega$ and the dual certificate $\boldsymbol{\Lambda}_k$ constructed via the golfing scheme satisfies the properties in Proposition 4.28. All we need to show here is that these properties also imply the conclusion of this theorem for the case with noisy measurements.

Let $\boldsymbol{H} = \hat{\boldsymbol{X}} - \boldsymbol{X}_o$. Notice that we can split $\boldsymbol{H}$ into two parts $\boldsymbol{H} = \mathcal{P}_\Omega[\boldsymbol{H}] + \mathcal{P}_{\Omega^c}[\boldsymbol{H}]$. For the first part, we have

$$\begin{aligned}\|\mathcal{P}_\Omega[\boldsymbol{H}]\|_F &= \|\mathcal{P}_\Omega[\hat{\boldsymbol{X}} - \boldsymbol{X}_o]\|_F \\ &\leq \|\mathcal{P}_\Omega[\hat{\boldsymbol{X}} - \boldsymbol{Y}]\|_F + \|\mathcal{P}_\Omega[\boldsymbol{Y} - \boldsymbol{X}_o]\|_F \\ &\leq 2\varepsilon.\end{aligned} \tag{4.4.76}$$

Notice that the second part $\mathcal{P}_{\Omega^c}[\boldsymbol{H}]$ is a feasible perturbation to the noiseless matrix completion problem. From the proof of Proposition 4.28 and in particular (4.4.34), we have

$$\|\boldsymbol{X}_o + \mathcal{P}_{\Omega^c}[\boldsymbol{H}]\|_* \geq \|\boldsymbol{X}_o\|_* + \left(\frac{1}{2} - \frac{1}{4C_2\sqrt{nr}}\right)\|\mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F, \tag{4.4.77}$$

and based on triangle inequality, we also have

$$\|\hat{\boldsymbol{X}}\|_* = \|\boldsymbol{X}_o + \boldsymbol{H}\|_* \geq \|\boldsymbol{X}_o + \mathcal{P}_{\Omega^c}[\boldsymbol{H}]\|_* - \|\mathcal{P}_\Omega[\boldsymbol{H}]\|_*. \tag{4.4.78}$$

Since $\|\hat{\boldsymbol{X}}\|_* \leq \|\boldsymbol{X}_o\|_*$, we have

$$\|\mathcal{P}_\Omega[\boldsymbol{H}]\|_* \geq \left(\frac{1}{2} - \frac{1}{4C_2\sqrt{nr}}\right)\|\mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F. \tag{4.4.79}$$

This leads to

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F \leq 4\|\mathcal{P}_\Omega[\boldsymbol{H}]\|_* \leq 4\sqrt{n}\|\mathcal{P}_\Omega[\boldsymbol{H}]\|_F \leq 4\sqrt{n}\varepsilon. \tag{4.4.80}$$

Since $\mathcal{P}_{\Omega^c}[\boldsymbol{H}] = \mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]] + \mathcal{P}_{\mathsf{T}}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]$, we remain to bound the term

$\mathcal{P}_{\mathsf{T}}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]$. Applying the proof of Lemma 4.29 to $\mathcal{P}_{\Omega^c}[\boldsymbol{H}]$, we have

$$\|\mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F \geq C_1 \frac{\sqrt{m}}{n\log(n)} \|\mathcal{P}_{\mathsf{T}}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F$$

for some large enough C_1. Therefore, we have

$$\|\mathcal{P}_{\mathsf{T}}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F \leq \frac{n\log(n)}{C_1\sqrt{m}} \|\mathcal{P}_{\mathsf{T}^\perp}[\mathcal{P}_{\Omega^c}[\boldsymbol{H}]]\|_F \leq c\frac{n\sqrt{n}\log(n)}{\sqrt{m}}\varepsilon. \tag{4.4.81}$$

This bound dominates the bounds of all the other terms, leading to the conclusion of the theorem. □

4.5 Summary

In this chapter, we have studied the problem of recovering a low-rank matrix from a number of m linear observations much fewer than its number of entries:

$$\boldsymbol{y} = \mathcal{A}[\boldsymbol{X}] \quad \in \mathbb{R}^m,$$

where $\mathcal{A}$ is a linear operator typically *incoherent* to the low-rank structure in $\boldsymbol{X} \in \mathbb{R}^{n\times n}$. This problem arises in a range of applications. It generalizes the problem of recovering a sparse vector. We described a convex relaxation of the low rank recovery problem, in which we minimize the nuclear norm, which is the sum (ℓ^1 norm) of the singular values of a matrix. We proved that, similar to the ℓ^1 minimization for recovering sparse vectors, if the measurements satisfy the *restricted isometry property* for low-rank matrices, then with a nearly minimum number of linear measurements in the order of

$$m = O(nr),$$

the convex program associated with nuclear norm minimization recovers all rank-r matrices correctly with high probability.

We have also studied a specific matrix completion problem with a more structured measurement model, in which we observe only a small subset of the entries of a low-rank matrix:

$$\boldsymbol{Y} = \mathcal{P}_\Omega[\boldsymbol{X}],$$

where $\mathcal{P}_\Omega$ samples a subset of entries of $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ in the support set Ω, with $|\Omega| = m < n^2$. This matrix completion problem captures the special structure of some of the most important practical low rank recovery applications, such as in the recommendation problem. It is mathematically more challenging, because certain sparse low-rank matrices cannot be completed without seeing almost all of their entries. Nevertheless, we observe that for low rank matrices *incoherent* to this measurement model, i.e.matrices whose singular vectors are not so concentrated on any coordinates, nuclear norm minimization succeeds with high probability with nearly minimum number of measurements in the order of

$$m = O(nr\log^2 n).$$

Almost parallel to the development for the recovery of sparse vectors, we have shown that these theoretical results and algorithms can be extended to cope with nuisance factors, such as measurement noise. The resulting algorithms are stable to small noise in the measurements. Moreover, in the next chapter we will see how to combine these ideas with those from sparse recovery to generate even richer classes of models and more robust algorithms.

4.6 Notes

As we have discussed in the beginning of this Chapter, rank minimization problems arise in a very broader range of engineering fields and applications. Arguably optimization issues associated with rank minimization have been studied most extensively and systematically in control [MP97] and identification [FHB01, FHB04] of dynamical systems. The fact that the nuclear norm is the convex envelope of the rank over the operator norm ball is due to [FHB01], leading to a convex formulation of the rank minimization problem. The extension of the restricted isometric property (RIP) to the matrix case is due to [RFP10] and it has helped characterized conditions under which the convex formulation succeeds, similar to the theory for sparse vectors studied the previous chapter.

For the matrix completion problem, the golfing scheme is due to Gross [Gro10]. Variants of Theorem 4.26 have been established by Gross [Gro10] and Recht [Rec10]; both include the extra assumption that $\|\boldsymbol{U}\boldsymbol{V}^*\|_\infty$ is small. The form stated here (without this assumption) is due to Chen [Che13]. It is easy to see that with little modification, the proofs and results established for matrix completion with respect to the standard basis can be generalized to any orthonormal (matrix) basis $\{\boldsymbol{B}_i\}_{i=1}^{n^2}$ as long as it is incoherent (inner product being small) with low-rank matrices. Since we have $|\langle \boldsymbol{B}_i, \boldsymbol{X}\rangle| \le \|\boldsymbol{B}_i\|\,\|\boldsymbol{X}\|_*$, for the basis to be incoherent with the low-rank matrix $\boldsymbol{X}$, we usually desire the base matrix $\boldsymbol{B}_i$ to have small operator norm. Fourier or Pauli bases are both such bases.

For the noisy matrix completion problem, the result in Theorem 4.35 is essentially attributed to the work of [CP10] but here the statement and proof are adapted to the weaker notion of incoherence required in the previous section. As result, we need an extra term of $\log(n)$ for the error bound, compared to that of [CP10].

Many methods have been developed in the literature that may sacrifice recoverability for computational efficiency or for measurement efficiency. To push for extreme scalability, the convex formulation that computes with the full $n \times n$ matrix might becomes unaffordable. In such cases, people start to investigate direct nonconvex formulations such as

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{Y} - \mathcal{P}_\Omega[\boldsymbol{U}\boldsymbol{V}^*]\|_2^2,$$

where $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times r}$ are rank-r matrices. Somewhat surprisingly, despite its

nonconvex nature, we will see in Chapter 7 that under fairly broad conditions, one can still find its optimal (and correct) low-rank solution using simple algorithms such as gradient descent.

4.7 Exercises

4.1 (Proof of Schoenberg's Theorem). *In this exercise, we invite the interested reader to prove Schoenberg's Euclidean embedding theorem (Theorem 4.1). Let* $\boldsymbol{D}$ *be a Euclidean distance matrix for some point set* $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, *i.e.,* $D_{ij} = \|\boldsymbol{x}_i\|_2^2 + \|\boldsymbol{x}_j\|_2^2 - 2\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. *Let* $\mathbf{1} \in \mathbb{R}^n$ *denote the vector of all ones, and* $\boldsymbol{\Phi} = \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^*$. *Using that* $\boldsymbol{\Phi}\mathbf{1} = \mathbf{0}$, *argue that* $\boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^*$ *satisfies the conditions of Schoenberg's theorem, i.e., it is negative semidefinite and has rank at most* d.

For the converse, let $\boldsymbol{D}$ *be a symmetric matrix with zero diagonal, and suppose that* $\boldsymbol{\Phi}\boldsymbol{D}\boldsymbol{\Phi}^*$ *is negative semidefinite and has rank at most* d. *Argue that there exists some matrix* $\boldsymbol{X} \in \mathbb{R}^{d \times n}$ *for which* $D_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$.

4.2 (Derivation of the SVD). *Let* $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ *be a matrix of rank* r. *Argue that there exists matrices* $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}$, $\boldsymbol{V} \in \mathbb{R}^{n_2 \times r}$, *with orthonormal columns and a diagonal matrix* $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$, *with* $\sigma_1 \geq \cdots \geq \sigma_r > 0$, *such that*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*. \tag{4.7.1}$$

Hint: what is the relationship between the singular values σ_i *and singular vectors* $\boldsymbol{v}_i$ *and the eigenvalues / eigenvectors of the matrix* $\boldsymbol{X}^*\boldsymbol{X}$?

4.3 (Best Rank-r Approximation). *We prove Theorem 4.5. First, consider the special case in which* $\boldsymbol{Y} = \boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_n)$ *with* $\sigma_1 > \sigma_2 > \cdots > \sigma_n$. *An arbitrary rank-*$r$ *matrix* $\boldsymbol{X}$ *can be expressed as* $\boldsymbol{X} = \boldsymbol{F}\boldsymbol{G}^*$ *with* $\boldsymbol{F} \in \mathbb{R}^{n_1 \times r}$, $\boldsymbol{F}^*\boldsymbol{F} = \boldsymbol{I}$ *and* $\boldsymbol{G} \in \mathbb{R}^{n_2 \times r}$.

1 *Argue that for any fixed* $\boldsymbol{F}$, *the solution to the optimization problem*

$$\min_{\boldsymbol{G} \in \mathbb{R}^{n_2 \times r}} \|\boldsymbol{F}\boldsymbol{G}^* - \boldsymbol{\Sigma}\|_F^2 \tag{4.7.2}$$

is given by $\hat{\boldsymbol{G}} = \boldsymbol{\Sigma}^*\boldsymbol{F}$, *and the optimal cost is*

$$\|(\boldsymbol{I} - \boldsymbol{F}\boldsymbol{F}^*)\boldsymbol{\Sigma}\|_F^2 . \tag{4.7.3}$$

2 *Let* $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{F}\boldsymbol{F}^*$, *and write* $\nu_i = \|\boldsymbol{P}\boldsymbol{e}_i\|_2^2$. *Argue that* $\sum_{i=1}^n \nu_i = n_1 - r$ *and* $\nu_i \in [0, 1]$. *Conclude that*

$$\|\boldsymbol{P}\boldsymbol{\Sigma}\|_F^2 = \sum_{i=1}^{n_1} \sigma^2 \nu_i \geq \sum_{i=r+1}^{n_1} \sigma_i^2, \tag{4.7.4}$$

with equality if and only if $\nu_1 = \nu_2 = \cdots = \nu_r = 0$ *and* $\nu_{r+1} = \cdots = \nu_n$. *Conclude that Theorem 4.5 holds in the special case* $\boldsymbol{Y} = \boldsymbol{\Sigma}$.

3 *Extend your argument to the situation in which the* σ_i *are not distinct (i.e.,* $\sigma_i = \sigma_{i+1}$ *for some* i).

4 Extend your argument to any $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$. Hint: use the fact that the Frobenius norm $\|\boldsymbol{M}\|_F$ is unchanged by orthogonal transformations of the rows and columns: $\|\boldsymbol{M}\|_F = \|\boldsymbol{RMS}\|_F$ for any orthogonal matrices $\boldsymbol{R}, \boldsymbol{S}$.

4.4 (Minimal Rank Approximation). *We consider a variant of Theorem 4.5 in which we are given a data matrix $\boldsymbol{Y}$ and we want to find a matrix $\boldsymbol{X}$ of minimum rank that approximates $\boldsymbol{Y}$ up to some given fidelity:*

$$\begin{array}{ll} \min & \operatorname{rank}(\boldsymbol{X}), \\ \text{subject to} & \|\boldsymbol{X}-\boldsymbol{Y}\|_F \le \varepsilon. \end{array} \tag{4.7.5}$$

Give an expression for the optimal solution(s) to this problem, in terms of the SVD of $\boldsymbol{Y}$. Prove that your expression is correct.

4.5 (Multiple and Repeated Eigenvalues). *Consider the eigenvector problem*

$$\min \quad -\tfrac{1}{2}\boldsymbol{q}^*\boldsymbol{\Gamma}\boldsymbol{q} \quad \textit{subject to} \quad \|\boldsymbol{q}\|_2^2 = 1, \tag{4.7.6}$$

where $\boldsymbol{\Gamma}$ is a symmetric matrix. In the text, we argued that when the eigenvalues of $\boldsymbol{\Gamma}$ are distinct, every local minimizer of this problem is global. (i) Argue that even when $\boldsymbol{\Gamma}$ has repeated eigenvalues, every local minimum of this problem is global. (ii) Now suppose we wish to find multiple eigenvector/eigenvalue pairs. Consider the optimization problem over the Stiefel manifold:

$$\begin{array}{ll} \min & -\tfrac{1}{2}\boldsymbol{Q}^*\boldsymbol{\Gamma}\boldsymbol{Q} \\ \text{subject to} & \boldsymbol{Q} \in \mathsf{St}(n,p) \doteq \{\boldsymbol{Q} \in \mathbb{R}^{n\times p} \mid \boldsymbol{Q}^*\boldsymbol{Q} = \boldsymbol{I}\}. \end{array} \tag{4.7.7}$$

Argue that every local minimizer of this problem has the form

$$\boldsymbol{Q} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p]\,\boldsymbol{\Pi}, \tag{4.7.8}$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$ are eigenvectors of $\boldsymbol{\Gamma}$ associated with the p largest eigenvalues, and $\boldsymbol{\Pi}$ is a permutation matrix.

4.6 (The Power Method). *In this exercise, we derive how to compute eigenvectors (and hence singular vectors) using the power method. Let $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite matrix. Let $\boldsymbol{q}_0$ be a random vector that is uniformly distributed on the sphere $\mathbb{S}^{n-1}$ (we can generate such a random vector by taking an n-dimensional iid $\mathcal{N}(0,1)$ vector and then normalizing it to have unit ℓ^2 norm). Generate a sequence of vectors $\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots$ via the iteration*

$$\boldsymbol{q}_{k+1} = \frac{\boldsymbol{\Gamma}\boldsymbol{q}_k}{\|\boldsymbol{\Gamma}\boldsymbol{q}_k\|_2}. \tag{4.7.9}$$

This iteration is called the power method.

Suppose that there is a gap between the first and second eigenvalues of $\boldsymbol{\Gamma}$: $\lambda_1(\boldsymbol{\Gamma}) > \lambda_2(\boldsymbol{\Gamma})$.

1 What does $\boldsymbol{q}_k$ converge to? Hint: write $\boldsymbol{\Gamma} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^*$ in terms of its eigenvectors/values. How does $\boldsymbol{V}^*\boldsymbol{q}_k$ evolve?

2 Obtain a bound on the error $\|\boldsymbol{q}_k - \boldsymbol{q}_\infty\|_2$ in terms of the spectral gap $\frac{\lambda_1-\lambda_2}{\lambda_1}$.

3 Your bound in 2 should suggest that as long as there is a gap between λ_1 and λ_2, the power method converges rapidly. How does the method behave if $\lambda_1 = \lambda_2$?
4 How can we use the power method to compute the singular values of a matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$?

4.7 (Variational Forms of Nuclear Norm). *Prove the statements of Proposition 4.6.*

4.8 (Convex Envelope Property via the Bidual). *In Theorem 4.9, we proved that the nuclear norm $\|\boldsymbol{X}\|_*$ is the convex envelope of* $\operatorname{rank}(\boldsymbol{X})$ *over the operator norm ball $\mathsf{B}_{\text{op}} = \{\boldsymbol{X} \mid \|\boldsymbol{X}\| \leq 1\}$. Here, we give an alternative derivation of this result, using the fact that the* biconjugate *of a function over a set B is the convex envelope. Let $f(\boldsymbol{X}) = \operatorname{rank}(\boldsymbol{X})$ denote the rank function.*

1 Prove that the Fenchel dual

$$f^*(\boldsymbol{Y}) = \sup_{\boldsymbol{X} \in \mathsf{B}} \{\langle \boldsymbol{X}, \boldsymbol{Y} \rangle - f(\boldsymbol{X})\}$$

can be expressed as

$$f^*(\boldsymbol{Y}) = \|\mathcal{D}_1[\boldsymbol{Y}]\|_*,$$

where $\mathcal{D}_\tau[\boldsymbol{M}]$ is the singular value thresholding operator, given by $\mathcal{D}_\tau[\boldsymbol{M}] = \boldsymbol{U}\mathcal{S}_\tau[\boldsymbol{S}]\boldsymbol{V}^*$ for any singular value decomposition $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$ of $\boldsymbol{M}$.
2 Prove that the dual of f^*,

$$f^{**}(\boldsymbol{X}) = \sup_{\boldsymbol{Y}} \langle \boldsymbol{X}, \boldsymbol{Y} \rangle - f^*(\boldsymbol{Y})$$

can satisfies

$$f^{**}(\boldsymbol{X}) = \|\boldsymbol{X}\|_*.$$

3 Use Proposition B.14 of Appendix B to conclude that $\|\cdot\|_*$ is the convex envelope of $\operatorname{rank}(\cdot)$ *over* B.

4.9 (Nuclear Norm of Submatrices). *Let $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{n \times m}$ be two matrices, and $\boldsymbol{M} = [\boldsymbol{M}_1, \boldsymbol{M}_2]$ be their concatenation. Show that:*

1 $\|\boldsymbol{M}\|_* \leq \|\boldsymbol{M}_1\|_* + \|\boldsymbol{M}_2\|_*$.
2 $\|\boldsymbol{M}\|_* = \|\boldsymbol{M}_1\|_* + \|\boldsymbol{M}_2\|_*$ if $\boldsymbol{M}_1^* \boldsymbol{M}_2 = \boldsymbol{0}$ (that is, the spans of $\boldsymbol{M}_1, \boldsymbol{M}_2$ are orthogonal).

4.10 (Convexifying Low-rank Approximation). *Consider the following optimization problem:*

$$\begin{array}{ll} \min & \|\boldsymbol{\Pi}\boldsymbol{Y}\|_F^2 \\ \text{subject to} & \boldsymbol{0} \preceq \boldsymbol{\Pi} \preceq \boldsymbol{I},\ \operatorname{trace}[\boldsymbol{\Pi}] = m - r. \end{array} \tag{4.7.10}$$

Prove that if $\sigma_r(\boldsymbol{Y}) > \sigma_{r+1}(\boldsymbol{Y})$, this problem has a unique optimal solution $\boldsymbol{\Pi}_\star$, which is the orthoprojector onto the linear span of the $n_1 - r$ trailing singular vectors $\boldsymbol{u}_{r+1}, \boldsymbol{u}_{r+2}, \ldots, \boldsymbol{u}_{n_1}$. The matrix $(\boldsymbol{I} - \boldsymbol{\Pi}_\star)\boldsymbol{Y}$ is the best rank-r approximation to $\boldsymbol{Y}$.

4.11 (Tangent Space to the Rank-r Matrices). *Consider a matrix $\boldsymbol{X}_o$ of rank r with compact singular value decomposition $\boldsymbol{X}_o = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$. Argue that the* tangent space *to the collection $\mathsf{M}_r = \{\boldsymbol{X} \mid \operatorname{rank}(\boldsymbol{X}) = r\}$ at $\boldsymbol{X}_o$ is given by $\mathsf{T} = \{\boldsymbol{U}\boldsymbol{R}^* + \boldsymbol{Q}\boldsymbol{V}^*\}$. Hint: consider generating a nearby low-rank matrix by writing $\boldsymbol{X}' = (\boldsymbol{U} + \boldsymbol{\Delta}_{\boldsymbol{U}})(\boldsymbol{\Sigma} + \boldsymbol{\Delta}_{\boldsymbol{\Sigma}})(\boldsymbol{V} + \boldsymbol{\Delta}_{\boldsymbol{V}})^*$.*

4.12 (Quadratic Measurements). *Consider a target vector $\boldsymbol{x}_o \in \mathbb{R}^{n\times n}$. In many applications, the observation can be modeled as a* quadratic *function of the vector $\boldsymbol{x}_o$. In notation, we see the squares*

$$y_1 = \langle \boldsymbol{a}_1, \boldsymbol{x}_o \rangle^2, \quad y_2 = \langle \boldsymbol{a}_2, \boldsymbol{x}_o \rangle^2, \quad \ldots, \quad y_m = \langle \boldsymbol{a}_m, \boldsymbol{x}_o \rangle^2$$

of the projections of $\boldsymbol{x}_o$ onto vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$. Notice that from this observation, it is only possible to reconstruct $\boldsymbol{x}_o$ up to a sign ambiguity: $-\boldsymbol{x}_o$ produces exactly the same observation.

1 *Consider the quadratic problem*

$$\min_{\boldsymbol{x}} \sum_{i=1}^{n} \left(y_i - \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2 \right)^2. \tag{4.7.11}$$

Is this problem convex in $\boldsymbol{x}$?

2 *Convert this to a convex problem, by replacing the vector valued variable $\boldsymbol{x}$ with a matrix valued variable $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^*$: convert the problem to*

$$\min_{\boldsymbol{X}} \sum_{i=1}^{n} \left(y_i - \langle \boldsymbol{A}_i, \boldsymbol{X} \rangle \right)^2. \tag{4.7.12}$$

How should we choose the matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_m$? Show that if $m < n^2$, $\boldsymbol{X}_o = \boldsymbol{x}_o\boldsymbol{x}_o^*$ is not *the unique optimal solution to this problem. How can we use the fact that $\operatorname{rank}(\boldsymbol{X}_o) = 1$ to improve this?*

3 *In the absence of noise, we can attempt to solve for $\boldsymbol{X}_o$ by solving the convex program*

$$\min \ \|\boldsymbol{X}\|_* \quad \text{such that} \quad \mathcal{A}[\boldsymbol{X}] = \boldsymbol{y}. \tag{4.7.13}$$

Implement this optimization using a custom algorithm or CVX. Does it typically recover $\boldsymbol{X}_o$?

4 *Does the operator $\mathcal{A}$ satisfy the rank RIP?*

4.13 (Proof of Theorem 4.25). *We prove Theorem 4.25. The goal here is to show that the solution to*

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* + \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{M}\|_F^2 \tag{4.7.14}$$

is given by $\mathcal{D}_1[\boldsymbol{M}]$.

1 *Argue that Problem (4.7.14) is strongly convex, and hence has a unique optimal solution.*

2 *Show that a solution $\boldsymbol{X}_\star$ is optimal if and only if $\boldsymbol{X}_\star \in \boldsymbol{M} - \partial \|\cdot\|_*(\boldsymbol{X}_\star)$.*

3 Using the condition from part 2, show that if $\boldsymbol{M}$ is diagonal, i.e., $M_{ij} = 0$ for $i \neq j$, then $\mathcal{S}_1[\boldsymbol{M}]$ is the unique optimal solution to (4.7.14).
4 Use the SVD to argue that in general, $\mathcal{D}_1[\boldsymbol{M}]$ is the unique optimal solution to (4.7.14).

4.14. *Prove Theorem 4.11.*

4.15 (Uniform Matrix Completion?). *Let Ω be a strict subset of $[n] \times [n]$. Show that there exist two matrix $\boldsymbol{X}_o$ and $\boldsymbol{X}'_o$ of rank one such that $\mathcal{P}_\Omega[\boldsymbol{X}_o] = \mathcal{P}_\Omega[\boldsymbol{X}'_o]$.* The implication of this is that it is not possible to reconstruct *all* low-rank matrices from the same observation Ω.

4.16 (Unique Optimality for Matrix Completion). *Consider the optimization problem*

$$\begin{array}{ll} \min & \|\boldsymbol{X}\|_* \\ \text{subject to} & \mathcal{P}_\Omega[\boldsymbol{X}] = \mathcal{P}_\Omega[\boldsymbol{X}_o]. \end{array} \tag{4.7.15}$$

Suppose that $\|\mathcal{P}_{\Omega^c}\mathcal{P}_\mathsf{T}\| < 1$. Assume that we can find some $\boldsymbol{\Lambda}$ such that

1 $\boldsymbol{\Lambda}$ is supported on Ω and
2 $\boldsymbol{\Lambda} \in \partial \|\cdot\|_* (\boldsymbol{X}_o)$ – i.e., $\mathcal{P}_\mathsf{T}[\boldsymbol{\Lambda}] = \boldsymbol{U}\boldsymbol{V}^*$ and $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| < 1$.

Show that $\boldsymbol{X}_o$ is the unique optimal solution to the optimization problem.

4.17. *Prove Theorem 4.19*

4.18. *Fill in the detailed steps of proof for Theorem 4.22.*

4.19. *Derive detailed steps that prove the error bound (4.3.85) in the proof of Theorem 4.20.*

4.20. *Show that in Lemma 4.27, any subdifferential of nuclear norm must be of the form given in (4.4.23).*

4.21. *Let $\mathcal{R}_\Omega[\boldsymbol{X}_o] = \sum_{\ell=1}^q [\boldsymbol{X}_o]_{i_\ell, j_\ell} \boldsymbol{e}_{i_\ell} \boldsymbol{e}_{j_\ell}^*$ with each (i_ℓ, j_ℓ) chosen iid at random from the uniform distribution on $[n] \times [n]$. Use the matrix Bernstein inequality to show that if $q > C\nu nr \log n$ for sufficiently large C, we have*

$$\left\| \mathcal{P}_{\mathsf{T}^\perp} \tfrac{n^2}{q} \mathcal{R}_\Omega \mathcal{P}_\mathsf{T} \right\| \leq t. \tag{4.7.16}$$

for any arbitrarily small constant t with high probability. [Hint: similar to the proof of Lemma 4.30.]

4.22. *For the dual certificate $\boldsymbol{\Lambda}$ constructed from the golfing scheme, use the fact in Exercise 4.21 and the fact that $\left\| \frac{n^2}{q} \mathcal{P}_{\mathsf{T}^\perp} \mathcal{R}_{\Omega_j}[\boldsymbol{E}_j] \right\|_F \leq \left\| \frac{n^2}{q} \mathcal{P}_{\mathsf{T}^\perp} \mathcal{R}_{\Omega_j} \mathcal{P}_\mathsf{T} \right\| \|\boldsymbol{E}_j\|_F$, show that if*

$$m \geq C\nu nr^2 \log^2 n$$

for a large enough constant C, we have $\|\mathcal{P}_{\mathsf{T}^\perp}[\boldsymbol{\Lambda}]\| \leq 1/2$ with high probability.

4.23. *Prove Lemma 4.32. Hint: write:*

$$(q^{-1}\mathcal{P}_\Omega - \mathcal{I})[\boldsymbol{Z}] = \sum_{ij} \underbrace{Z_{ij}\left(q^{-1}\mathbb{1}_{ij\in\Omega} - 1\right)\, \boldsymbol{E}_{ij}}_{\doteq \boldsymbol{W}_{ij}},$$

and apply the operator Bernstein inequality, controlling the operator norm of $\boldsymbol{W}_{ij}$ *in terms of* $\|\boldsymbol{Z}\|_\infty$ *and controlling the matrix variance in terms of* $\|\boldsymbol{Z}\|_{rc}$.

4.24. *Prove Lemma 4.33. Use the matrix Bernstein inequality to obtain a bound on the probability that the* ℓ*-th row* $\|\boldsymbol{e}_\ell^*\left(q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega - \mathcal{P}_\mathsf{T}\right)[\boldsymbol{Z}]\|$ *is large, repeat for each column, and then sum the failure probabilities over all rows and columns to obtain a bound on the probability that the* $\|\cdot\|_{rc}$ *is large. Hint: apply the matrix Bernstein inequality to the random vector:*

$$\boldsymbol{e}_\ell^*\left(q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega - \mathcal{P}_\mathsf{T}\right)[\boldsymbol{Z}] = \sum_{ij} \underbrace{Z_{ij}(q^{-1}\mathbb{1}_{ij\in\Omega} - 1)\boldsymbol{e}_\ell^*\mathcal{P}_\mathsf{T}[\boldsymbol{E}_{ij}]}_{\doteq \boldsymbol{w}_{ij}}.$$

4.25. *Prove Lemma 4.34. Apply the standard Bernstein inequality to bound the probability that the* k,l *entry of* $(\mathcal{P}_\mathsf{T} - q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega\mathcal{P}_\mathsf{T})[\boldsymbol{Z}]$ *is large, and then sum this probability over all entries* k,l *to bound the probability that the* ℓ^∞ *norm is large. For the* k,l *entry work with the sum of independent random variables*

$$\begin{aligned}\left[\left(\mathcal{P}_\mathsf{T} - q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega\mathcal{P}_\mathsf{T}\right)[\boldsymbol{Z}]\right]_{kl} &= Z_{kl} - [q^{-1}\mathcal{P}_\mathsf{T}\mathcal{P}_\Omega[\boldsymbol{Z}]_{kl}] \\ &= \sum_{ij} \underbrace{n^{-2}Z_{kl} - q^{-1}\mathbb{1}_{ij\in\Omega}\,\langle\mathcal{P}_\mathsf{T}[\boldsymbol{E}_{kl}], \mathcal{P}_\mathsf{T}[\boldsymbol{E}_{ij}]\rangle\, Z_{ij}}_{\doteq w_{ij}}.\end{aligned}$$

5 Decomposing Low-Rank and Sparse Matrices

"*The whole is greater than the sum of the parts.*"
– Aristotle, *Metaphysics*

In the previous chapters, we have studied how either a sparse vector or a low-rank matrix can be recovered from compressive or incomplete measurements. In this chapter, we will show that it is also possible to simultaneously recover a sparse signal and a low-rank signal from their superposition (mixture) or from highly compressive measurements of their superposition (mixture). This combination of rank and sparsity gives rise to a broader class of models that can be used to model richer structures underlying high-dimensional data, as we will see in examples in this chapter and later application chapters. Nevertheless, we are also faced with new technical challenges about whether and how such structures can be recovered correctly and effectively, from few observations.

5.1 Robust PCA and Motivating Examples

5.1.1 Problem Formulation

In this chapter, we study variants of the following problem. We are given a large data matrix $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ which is a superposition of two matrices:

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o, \tag{5.1.1}$$

where $\boldsymbol{L}_o \in \mathbb{R}^{n_1 \times n_2}$ is a low-rank matrix and $\boldsymbol{S}_o \in \mathbb{R}^{n_1 \times n_2}$ is a sparse matrix. Neither $\boldsymbol{L}_o$, nor $\boldsymbol{S}_o$ is known ahead of time. Can we hope to efficiently recover both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$?

This problem resembles another classical low-rank matrix recovery problem in which the observed data matrix $\boldsymbol{Y} \in \mathbb{R}^{n_1 \times n_2}$ is a superposition of two matrices:

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{Z}_o, \tag{5.1.2}$$

where as before $\boldsymbol{L}_o \in \mathbb{R}^{n_1 \times n_2}$ is a low-rank matrix but here $\boldsymbol{Z}_o \in \mathbb{R}^{n_1 \times n_2}$ is assumed to be a small, but dense perturbation matrix. For example, $\boldsymbol{Z}_o$ could be a Gaussian random matrix with small standard deviation. In other words, one wants to recover a low-rank matrix $\boldsymbol{L}_o$ (or the low-dimensional subspace spanned by the columns of $\boldsymbol{L}_o$) from noisy measurements. The classical *Principal*

Component Analysis (PCA) [Jol86] seeks the best rank-r estimate of $\boldsymbol{L}_o$ by solving

$$\min_{\boldsymbol{L}} \|\boldsymbol{Y} - \boldsymbol{L}\|_F \quad \text{subject to} \quad \operatorname{rank}(\boldsymbol{L}) \leq r. \tag{5.1.3}$$

This problem is also known as the best rank-r approximation problem. As we have seen in Section 4.2.2, it can be solved very efficiently via the *Singular Value Decomposition* (SVD): If $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ is the SVD of the matrix $\boldsymbol{Y}$, the optimal rank-r approximation to $\boldsymbol{Y}$ is

$$\hat{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Sigma}_r\boldsymbol{V}^*,$$

where $\boldsymbol{\Sigma}_r$ keeps only the first r leading singular values of the diagonal matrix $\boldsymbol{\Sigma}$. This solution enjoys a number of optimality properties when the perturbation in matrix $\boldsymbol{Z}_o$ is small or i.i.d. Gaussian [Jol02].

However, in the new measurement model (5.1.1), the perturbation term $\boldsymbol{S}_o$ can have elements with arbitrary magnitude and hence its ℓ^2 norm can be unbounded. In a sense, the measurement we observe

$$\boldsymbol{Y} = \boldsymbol{L}_o + \boldsymbol{S}_o$$

is a corrupted version of the low-rank matrix $\boldsymbol{L}_o$ – entries of $\boldsymbol{Y}$ where $\boldsymbol{S}_o$ is nonzero carry no information about $\boldsymbol{L}_o$. The problem of recovering the matrix $\boldsymbol{L}_o$ (and the associated low-dimensional subspace) from such highly corrupted measurements can be considered a form of *Robust Principal Component Analysis* (RPCA), as opposed to the classical PCA which is only stable to small noise or perturbation.

In this chapter, we use $\mathfrak{S}$ and $\boldsymbol{\Sigma}_o$ to denote the support and signs of the sparse matrix $\boldsymbol{S}_o$, respectively:

$$\mathfrak{S} \doteq \operatorname{supp}(\boldsymbol{S}_o) \quad \subseteq [n_1] \times [n_2], \tag{5.1.4}$$

$$\boldsymbol{\Sigma}_o \doteq \operatorname{sign}(\boldsymbol{S}_o) \quad \in \{-1, 0, 1\}^{n_1 \times n_2}. \tag{5.1.5}$$

We note that if we somehow knew the support $\mathfrak{S}$ of $\boldsymbol{S}_o$, we could potentially recover $\boldsymbol{L}_o$ by solving a matrix completion problem (as in the previous chapter) using $\mathcal{P}_\Omega[\boldsymbol{L}_o]$ with $\Omega = \mathfrak{S}^c$. But in the problems described above, both $\boldsymbol{L}_o$ and $\boldsymbol{S}_o$ are unknown.

5.1.2 Matrix Rigidity and Planted Clique

Using this connection to matrix completion, one can show that the Robust PCA problem is NP-Hard in general. The hardness can also be shown directly via a connection to the concept of *matrix rigidity*. We say a matrix $\boldsymbol{M}$ is *rigid* if it is far from a low-rank matrix in Hamming distance. Or more formally,

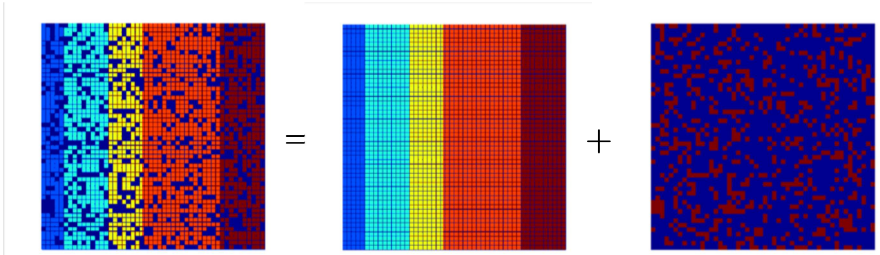


Figure 5.1 Superposition of a low-rank matrix $\boldsymbol{L}_o$ and a sparse matrix $\boldsymbol{S}_o$.

Definition 5.1 (Matrix Rigidity). *The* rigidity *of a matrix* $\boldsymbol{M}$ *(relative to rank* r *matrices) is defined to be:*

$$R_{\boldsymbol{M}}(r) \doteq \min\{\|\boldsymbol{S}\|_0 : \operatorname{rank}(\boldsymbol{M}+\boldsymbol{S}) \leq r\}, \tag{5.1.6}$$

the smallest number of entries that need to be modified in order to change $\boldsymbol{M}$ *to a rank* r *matrix.*

Matrix rigidity is an important concept in computational complexity theory: It has been shown by [Val77] that matrix rigidity gives a lower bound on the circuit complexity for computing the linear transform $\boldsymbol{M}\boldsymbol{x}$. Matrix rigidity is also related to the notion of communication complexity [Wun12]. Nevertheless, computing matrix rigidity is in general NP-hard [MSM07], and so it is hard to decompose a general matrix:

$$\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}$$

into a low-rank and sparse one. Exercise 5.2 studies the hardness of matrix rigidity and guides the interested reader through this connection.

The hardness of the Robust PCA problem can also be established through its connection to the *Planted Clique* problem [AB09].

Definition 5.2 (Planted Clique Problem). *Given a graph* $\mathcal{G}$ *with* n *nodes, randomly connect each pair of nodes with probability 1/2. Then select any* n_o *nodes and make them a clique – a fully connected subgraph. The goal is to find this hidden clique from the graph* $\mathcal{G}$*.*

It is known that with high probability the largest clique of the randomly generated graph (with $1/2$ connectivity) is $2\log_2 n$. Hence, theoretically, if

$$n_o > 2\log_2 n,$$

we should be able to identify such a planted clique and distinguish the graph from the randomly generated one. It is also known that if

$$n_o = \Omega(\sqrt{n}),$$

it is possible to efficiently identify the planted clique [Kuč95, AKS98] using spectral methods. The interesting and difficult part of this problem is for

$$2\log_2 n < n_o < \sqrt{n}.$$

There is a working conjecture about the complexity of this problem[1]:

Conjecture: $\forall \varepsilon > 0$, if $n_o < n^{0.5-\varepsilon}$, then there is *no* tractable algorithm that can find the hidden clique from $\mathcal{G}$ with high probability.

In our context, if we consider the adjacency matrix $\boldsymbol{A}$ of the graph $\mathcal{G}$, then we have

$$\boldsymbol{A} = \boldsymbol{L}_o + \boldsymbol{S}_o,$$

where $\boldsymbol{L}_o$ is a rank-1 matrix with an $n_o \times n_o$ block of all ones, and $\boldsymbol{S}_o$ is a relatively sparse matrix with around $(n - n_o)/2$ nonzero entries. Hence, given the difficulty of the planted clique problem, we should not expect that there exists an efficient algorithm to decompose the matrix $\boldsymbol{A}$ correctly to a rank-1 matrix $\boldsymbol{L}_o$ and a sparse $\boldsymbol{S}_o$ when $n_o < n^{0.5-\varepsilon}$. We leave more detailed study of the planted clique problem as exercises, which will help the reader understand better the working conditions of the method proposed in this chapter.

For our purposes here, however, we simply note that the situation for Robust PCA is analogous to that for low-rank recovery and for sparse recovery: *we should not expect to find an efficient algorithm which works for every problem instance.* Instead, the instances $\boldsymbol{Y}$ that are of practical interest are relatively "soft": they can be made significantly low-rank by correcting a small number of entries, as we will see in a number of important applications discussed below.

5.1.3 Applications of Robust PCA

Many important practical applications confront us with instances of the problem (5.1.1). We here give a few representative examples inspired by some contemporary challenges in data science. Notice that depending on the applications, either the low-rank component or the sparse component could be the object of interest.

Video Surveillance.
Given a sequence of surveillance video frames, we often need to identify activities that stand out from the background. If we stack the video frames as columns of a matrix $\boldsymbol{Y}$, then the low-rank component $\boldsymbol{L}_o$ represents the stationary background and the sparse component $\boldsymbol{S}_o$ captures the moving objects in the foreground. However, since each image frame may have thousands or millions of pixels and each video fragment may contain hundreds or thousands of frames, it would be

[1] For more evidence on the complexity of the planted clique problem around $n_o = \Theta(\sqrt{n})$, one may refer to the work of [GZ19]. The more recent work of [BB20] has further revealed the important role of the planted clique problem in characterizing computational hardness taxonomy among various statistical inference problems regarding low-dimensional models in high-dimensional spaces.

only possible to decompose $\boldsymbol{Y}$ this way if we have a truly scalable solution to this problem. The method developed in this chapter will enable us to achieve this goal, as we will later see in an example shown in Figure 5.3.

Face Recognition.
As we learned in Section 4.1.1, images of a convex, Lambertian surface under varying illuminations span a low-dimensional subspace [BJ03]. That is, if we stack face images of a person as column vectors of a matrix, then this matrix is (approximately) a low-rank matrix $\boldsymbol{L}_o$. This fact has been a major reason why low-dimensional models are effective for imagery data. In particular, images of a human's face can be well-approximated by a low-dimensional subspace. Being able to correctly retrieve this subspace is crucial in many applications such as face recognition and alignment. However, realistic face images often suffer from self-shadowing, specularities, or saturation in brightness (as we have seen in images on the left of Figure 4.2), which make this a difficult task and subsequently compromise the recognition performance. A more careful study shows that the face images are better modeled by a low-rank matrix $\boldsymbol{L}_o$ superposed with a sparse matrix $\boldsymbol{S}_o$ which models such imperfection [ZMKW13]. To be able to recover both components from occluded images will allow us to repair such images for better recognition, as we will soon see an example in Figure 5.4.

Latent Semantic Indexing.
Web search engines often need to analyze and index the content of an enormous corpus of documents. A popular scheme is the *Latent Semantic Indexing* (LSI), [DFL$^+$88, PTRV98] which we have discussed in the preceding chapter, Section 4.1.4. Recall that the basic idea is to gather a document-versus-term matrix $\boldsymbol{Y}$ whose entries typically encode the relevance of a term (or a word) to a document such as the frequency it appears in the document (e.g., term frequency-inverse document frequency, also known as TF-IDF). PCA (or SVD) has traditionally been used to decompose the matrix as a low-rank part plus a residual, which is not necessarily sparse (as we would like). If we were able to decompose $\boldsymbol{Y}$ as a sum of a low-rank component $\boldsymbol{L}_o$ and a sparse component $\boldsymbol{S}_o$, then $\boldsymbol{L}_o$ could capture a few topic models of all the documents while $\boldsymbol{S}_o$ captures the few keywords that best distinguish each document from others. See [MZWM10] for more details about such a *joint topic-document model* (via a superposition of a low-rank and sparse matrix).

Ranking and Collaborative Filtering.
As we have seen in Section 4.1.2, anticipating user preferences has been an important problem in online commerce and advertisement. Companies now routinely collect user rankings for various products, e.g., movies, books, games, or web tools, among which the Netflix Prize for movie ranking is the best known example. The problem posed in the Netflix Prize is to use very sparse and incomplete rankings provided by the users on some of the products to predict the preference

of any given user on any products, also known as collaborative filtering [Hof04]. In the previous chapter, this problem has been cast as a problem completing a low-rank matrix, say $\boldsymbol{L}_o$. However, in reality, as the data collection process often lacks control or is sometimes even *ad hoc*, a small portion of the available rankings could be rather random and even tampered with by malicious users or competitors. We may model those entries as a sparse matrix $\boldsymbol{S}_o$. The recommendation problem now becomes more challenging since we need to simultaneously complete a low-rank matrix $\boldsymbol{L}_o$ and correct these (sparse) errors $\boldsymbol{S}_o$. That is, we need to infer the low-rank matrix $\boldsymbol{L}_o$ from a set of incomplete and corrupted entries, a problem that methods introduced in the previous chapter are inadequate to solve.

Community Discovery and Data Clustering.
With the increasing popularity of social networks, one important task is to discover hidden patterns and structures in such networks. We model a social network as a graph $\mathcal{G}$, with a node representing a person and an edge representing friendship. Then the adjacency matrix of the graph is a symmetric matrix $\boldsymbol{A}$ with $a_{ij} = a_{ji} = 1$ if and only if i and j are friends, and 0 otherwise. A "community" in the network is a subgroup of nodes that have much higher density of connectivity among themselves than with others. Such a group of nodes is also known as a "cluster," as shown in Figure 5.2. Note that each cluster can be approximately modeled as a fully connected subgraph, also known as a "clique." Each clique corresponds to a rank-1 submatrix with all ones. Hence, for a graph that consists of multiple communities, the adjacent matrix $\boldsymbol{A}$ will be of the form:

$$\boldsymbol{A} = \boldsymbol{L}_o + \boldsymbol{S}_o,$$

where $\boldsymbol{L}_o$ is a low-rank matrix consisting of several blocks of rank-1 submatrices with all ones, and $\boldsymbol{S}_o$ is a sparse matrix that corresponds to the remaining few spurious or missing connections. This can be viewed as an extended (more challenging) version of the "planted clique" problem discussed earlier as here we allow multiple cliques in the graph. In data science and engineering, many tasks that try to cluster data into multiple subgroups, segments, subsystems, or subspaces, can be reduced to a problem of this nature [VMS16].

All the applications that we have listed above require solving the problem of decomposing a low-rank and sparse matrix possibly of very high dimension, under various conditions. As it turns out, mathematically, this class of problems is rather fundamental to machine learning and system theory. They are actually the underlying problem for correctly and robustly learning graphical models and identifying dynamical systems, as discussed in Chapter 1, the Introduction of this book.