Rigorous optimization recipes for sparse and low rank inverse problems with applications in data sciences

Thèse n. 6350 (2014)
présenté le 8 Septembre 2014

Laboratory for Information and Inference Systems
Doctoral program at School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences par

Anastasios Kyrillidis

acceptée sur proposition du jury:

Prof Emre Telatar, président du jury
Prof Volkan Cevher, directeur de thèse
Prof Mario Figueiredo, rapporteur
Prof Nikos Sidiropoulos, rapporteur
Prof Pierre Vandergheynst, rapporteur

Lausanne, EPFL, 2014

To my family

Acknowledgements

First, I would like to express my gratitude to my supervisor Volkan Cevher. Volkan, you both provided me with a deep perspective of signal processing, machine learning and optimization and guidance on many technical aspects of this thesis. Your inspiration and constant excitement for research have been truly invaluable to me. Moreover, I would like to thank Emre Telatar, Pierre Vandergheynst, Mario Figueiredo and Nicholas Sidiropoulos for agreeing to serve as members of my final examination committee. Moreover, I would like to sincerely thank Mario Figueiredo for his careful assessment of my work and for providing useful suggestions for improvements.

During my time in LIONS lab, I had the great pleasure to collaborate with great researchers. I would like to thank all my lab members: Cosimo Aprile, Bubacarr Bah, Luca Baldassarre, Gosia Baltaian, Ilija Bogunovic, Baran Gozcu, Marwa El-Halabi, Radu Christian Ionescu, Yen-Huan Li, Quoc Tran-Dihn, Alp Yurtsever. I truly enjoyed spending my time with them and I have learned a lot from each one of them. I'm also grateful to be lab-mate with Mitra Fatemi, Sylvia Sandoz and, Hemant Tyagi — thank you very much for the moments we spent together.

Many thanks go to IBM Research Zurich for hosting me as a research intern during the winter of 2013. In particular, I would like to thank Michail Vlachos, Anastasios Zouzias (always winning me at billiards games) and Vassilis Vasileidis (always winning him at FIFA games).

Of course, many thanks to all my friends in Lausanne for the great moments I had these 4 years: Alhussein, Anna, Alessandra, Andrea[2], Antonis, Christina, Dorina, Efi, Elena, Emre, Giannis, Giorgos, Iris, Ivan, Lorenzo, Marianna, Michalis, Nikos[2], Sofia, Vassilis, Vicky, Xiaowen.

Finally, my heartfelt thanks go out to my family and friends in Greece for their constant support and encouragement.

Lausanne, 11 September 2014 A. K.

Abstract

Many natural and man-made signals can be described as having a few degrees of freedom relative to their size due to natural parameterizations or constraints; examples include bandlimited signals, collections of signals observed from multiple viewpoints in a network-of-sensors, and per-flow traffic measurements of the Internet. Low-dimensional models (LDMs) mathematically capture the inherent structure of such signals via combinatorial and geometric data models, such as sparsity, unions-of-subspaces, low-rankness, manifolds, and mixtures of factor analyzers, and are emerging to revolutionize the way we treat inverse problems (e.g., signal recovery, parameter estimation, or structure learning) from dimensionality-reduced or incomplete data.

Assuming our problem resides in a LDM space, in this thesis we investigate how to integrate such models in convex and non-convex optimization algorithms for significant gains in computational complexity. We mostly focus on two LDMs: (i) sparsity and (ii) low-rankness. We study trade-offs and their implications to develop efficient and provable optimization algorithms, and–more importantly–to exploit convex and combinatorial optimization that can enable cross-pollination of decades of research in both.

Beaucoup de signaux naturels et artificiels peuvent être décrits comme ayant peu de degrés de liberté par rapport à leur taille en raison de paramétrages naturels ou des contraintes; des exemples comprennent les signaux à bande limitée, les collections de signaux observés à partir de plusieurs points de vue dans un réseau de capteurs, et les mesures de trafic d'Internet par flux. Les modèles de basse dimensionnalité (MBD) capturent mathématiquement la structure inhérente de ces signaux via des modèles de données combinatoires et géométriques, comme la parcimonie, les unions de sous-espaces, la faiblesse de rang, la variété, et les mélanges d'analyseurs factorielles, et ils émergent pour révolutionner la façon dont nous traitons les problèmes inverses (par exemple, la récupération de signal, l'estimation de paramètres, ou l'apprentissage de structure) à partir de données de dimensionnalité réduite ou incomplètes.

En supposant que notre problème réside dans un espace de MBD, dans cette thèse, nous étudions comment intégrer ces modèles dans les algorithmes d'optimisation convexes et non convexes pour des gains importants dans la complexité de calcul. Nous nous concentrons principalement sur deux MBDs: (i) la parcimonie et (ii) la faiblesse de rang. Nous étudions les compromis et leurs implications pour développer des algorithmes d'optimisation efficaces et prouvables, et – plus important encore – pour exploiter l'optimisation convexe et combinatoire qui peut permettre la pollinisation croisée de décennies de recherche à la fois.

Key words: Sparse Euclidean projections, sparse linear regression, compressed sensing, affine rank minimization, matrix completion, structured sparsity, convex composite minimization, self-concordance.

Contents

Introduction

Living in the "information age", we have witnessed an ever increasing interest in designing computing systems that can analyze large amount of information in reasonable time: organizations ranging from all-around data analytics companies to banks and from petroleum enterprises to bio-informatics research labs move towards this direction. In order to accomplish the desiderata, one has to take into consideration every aspect of data to conceive a viable and efficient system design: its *volume*, its *variety* and the need for quick analysis tools (*velocity*); see Figure 1.

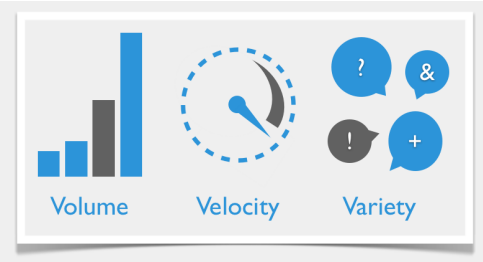


Figure 1: The "3 V.'s" description of data.

To quantify the importance of each attribute in such description, we highlight next some interesting statistics on the *volume* and the *variety* of data nowadays. Google processes more than 24 Petabytes of data per day and collects data originating from both social networks and multimedia portals (images, video, social network data, etc.). The main reason behind this huge data creation is the Web and its volunteered users: the crowd has become an important data provider. Subsumed under the term Volunteered Information (VI), non-expert users have been providing a wealth of data online. However, in order to exploit this data in its entirety, efficient *compression* algorithms are needed so that, e.g., search queries can be efficiently completed, even in the compressed domain [VFK]. Moreover, data mining methods should be able to operate with high accuracy in such compressed spaces via efficient feature selection or dimensionality reduction schemes.

Within the same context, companies such as Microsoft, Facebook and Twitter further collect data that can lead to social network inference (e.g., graph inference). As a representative example, Twitter collects more than 50 million "tweets" per day, a huge amount of information that can be exploited to infer inter-user dependencies and correlations.

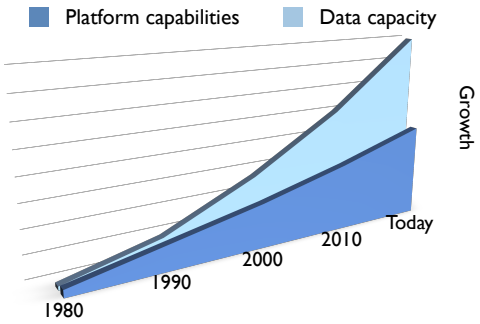
From a different perspective, there are many problems where data is represented in the most usual form (tables/matrices in databases); e.g., bank companies collect transaction data in database tables where querying, monitoring or even prediction tasks are performed (e.g., portfolio suggestion in finance optimization).

In addition to the above, recent studies have shown that more than 1.3 Exabytes of data are sampled, stored and transfered over the (wireless) communication network, due to the bloom of smart-phones in the phone industry. This underlines the need for more efficient sampling techniques—consider for example the *compressive sensing* paradigm [CDS98, CT06, CRT06], described in the next chapters—in

order to reduce the amount of data transmitted over the network, while pertaining the overall quality of service.

To this end, there is an ever increasing need for computer hardware designs that can accommodate such data "deluge". However, the rate of data growth is far higher than the dictated growth in platform computational capabilities—see Figure 2 for a relative comparison. Thus, by just using traditional statistical tools to compress, analyze and process data, one might not fully exploit the available data in its entirety, no matter how she/he designs and optimizes the computer hardware. Therefore, we need accurate, robust and scalable algorithms that can handle larger amount of data simultaneously.

Figure 2: Growth rate comparison.



Within this context, this thesis focuses on and evolves around *novel, fast and provable algorithms* for data analysis in large-scale problems. From our point of view, a brief description of data analysis in layers is given in Figure 3. While data querying and monitoring are two of the most used tasks in data analysis (e.g., most database systems rely heavily on such tasks), here we study the two out-most layers: *data preprocessing* and *prediction*. In particular, we concentrate on understanding and tackling challenges in large scale inference problems and we do so in view of diverse machine learning and signal processing applications.

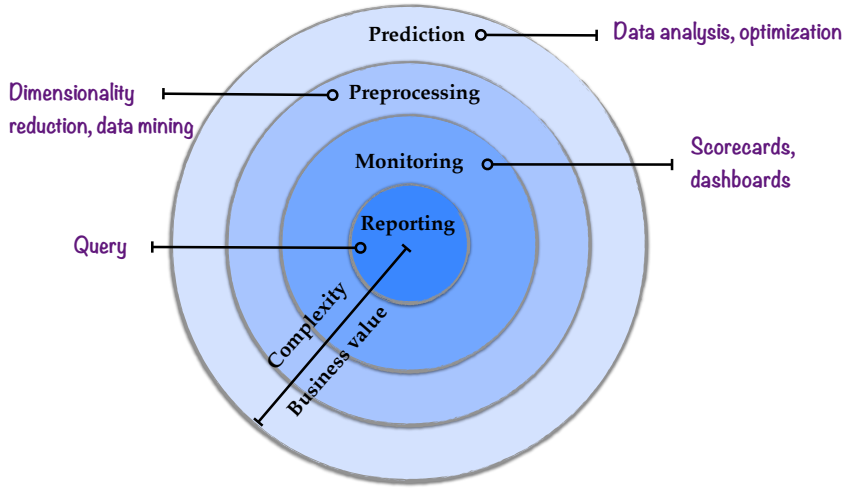


Figure 3: Layers of data analysis tasks. The further we move away from the center of the circles, the higher the complexity/business value of the task.

Our perspective can be summarized as follows: While the ambient dimension is vast in modern data analysis problems, the relevant information therein typically resides in a much lower dimensional space. To this end, given a data set to analyze, one is usually interested in the *simplest* model that well-characterizes the observations. This conclusion has lead to several new theoretical and algorithmic developments in different communities, including theoretical computer science [BGI+08], applied mathematics [RZMC11], and digital signal processing [Don06, CW08]. In practice, it turns out that it is sufficient to identify a *low*

dimensional model (LDM) that is inherent in the acquired data, formulate proper optimization criteria that disclose such LDMs from the observations and, develop fast and accurate algorithms to accomplish this task.

In the first two parts of this thesis (Chapters 1-4), we mainly focus on two mathematical problems from a *non-convex* perspective, with a wide range of real-world applications: (i) sparse linear regression and (ii) low rank matrix approximation from incomplete data. To motivate our discussion, we first present some sparsity primitives that evolve around special Euclidean projection operations and provide intuition for our developments later in the text. Besides the theory developed for this task, we describe a class of fast and accurate algorithms with low computational and space complexity (as compared to other convex and non-convex state-of-the-art approaches), aiming for their direct application in real-world engineering problems.

In the third part of this thesis (Chapter 5), we focus on convex optimization and propose an novel algorithmic framework that solves a wide range of problems with provable guarantees. The highlight of this attempt is the use of unconventional theoretical convex tools that lead to provably better and robust algorithms with attractive convergence guarantees. As we show, this scheme finds application in a wide range of problems, ranging from graphical modeling to low-light neuron image processing under Poisson noise and from sparse signal reconstruction in MRI images to sparse covariance estimation for portfolio optimization.

Next, we only "scratch the surface" of the topics covered in this thesis to stimulate the reader's interest in the chapters that follow.

Making inferences with low dimensional models

Greedy approaches in sparse signal approximation (Chapters 1-2): One outstanding application of LDMs is found in compressive sensing (CS), which exploits *sparse* representations in one-way signal arrays (i.e., vectors). It is well-known that signals such as images can be well-approximated and compressed as the sparse superposition of atoms/functions from an appropriate basis. Using this prior information, CS showcases that such signals can be reliably reconstructed from only a limited set of measurements, far fewer than what conventional wisdom dictates. However, while most CS recovery algorithms are based on convex optimization to seek sparse solutions, they-staggeringly-never take advantage of the crucial non-convex low-dimensional scaffold, upon which the CS problem resides. In [KPC12, KC12a, KC11], we investigate how to integrate combinatorial, sparse projections in convex optimization algorithms for better signal reconstruction and lower computational complexity. As a result, we introduce the ALPS, CLASH and NORM-PURSUITS classes of algorithms that enhance the performance of state-of-the-art algorithms by carefully selecting parameters and incorporating convex and non-convex constraints on the regression vector.

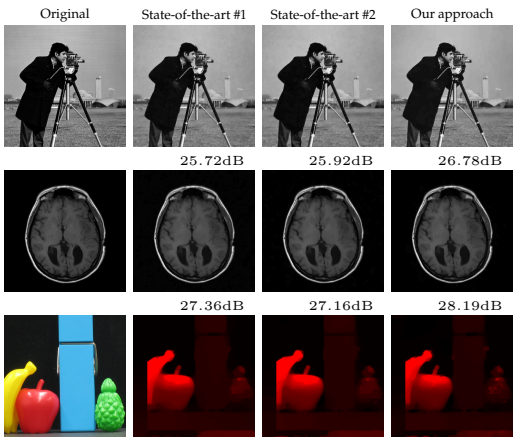


Figure 4: Image reconstruction results for three problem cases: real image data (Top row), MRI brain image data (Middle row) and spectral imaging real data (Bottom row).

In the context of image processing, Figure 4 shows some preliminary results of this attempt, highlighting the merits of our approach. For this case, we use the NORMED PURSUIT approach, where both sparsity and convex Total variation-norm (TV-norm) constraints are present in the optimization criterion. To study the performance of NORMED PURSUITS in the compressed domain, we conduct experiments on natural images, brain[1] images and Coded Aperture Snapshot Spectral Imager (CASSI) data[2]. Using measurements that correspond to the 25% of the full data, we obtain superior image reconstruction, as compared to state-of-the-art schemes —see Chapter 2 for more information.

Key ingredients for this type of optimization are the sparse Euclidean projections, probably accompanied with additional constraints. This observation has led us to study the behavior of such operations in Chapter 1. As an extension to this line of research, in [KBCK13], we consider the problem of sparse projections onto simplex-type of constraints and propose efficient sparse projections in solving high-dimensional learning problems such as sparse density estimation and portfolio selection.

Greedy approaches in affine rank minimization (Chapter 4): Within the context of affine rank minimization problems, where low-rankness constitutes a LDM for this case, we present and analyze a new set of low-rank *non-convex* recovery algorithms for linear inverse problems with applications in image denoising, background image subtraction (see Figure 5), and quantum state tomography [KC14, KC12b]. In Chapter 4, we provide strategies in order to achieve complexity vs. accuracy trade-offs in practice and propose acceleration schemes (via memory-based techniques and randomized, ϵ-approximate projections) to decrease the computational costs in the recovery process.

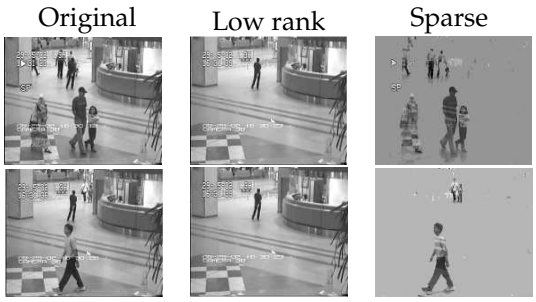


Figure 5: Background subtraction in video sequence.

In the context of quantum state tomography, in [BCK13] we further improve the low-rank recovery scheme to operate on space proportional to the degrees of freedom in the problem. This added twist decreases the per iteration requirements in terms of storage and computational complexity leading to an efficient solver working in extreme large scale problems. To test scaling to very large data, we exploit parallel computing capabilities of workstations provided in EPFL and demonstrate the performance of our algorithm under realistic scenarios of a 16 q-bit state quantum system (i.e., a 65536×65536 matrix), using a known quantum state as input with realistic quantum mechanical perturbations.

A glimpse in sparse LDMs: beyond simple sparsity

Compressive sensing (CS) exploits sparsity to recover sparse or compressible signals from dimensionality reducing, non-adaptive sensing mechanisms. Sparsity is also used to enhance interpretability in machine learning and statistics applications. However, many solutions proposed nowadays do not leverage the true underlying structure. Recent results in CS extend the simple sparsity idea to more sophisticated *structured* sparsity models, which describe the interdependency between the nonzero components of a signal, increasing the interpretability of the results and leading to better recovery performance. In order to better understand the impact of structured sparsity, in Chapter 3 we analyze the connections

[1] BRAINIX database: http://pubimage.hcuge.ch:8080/.
[2] http://www.disp.duke.edu/projects/CASSI

between the discrete models and their convex relaxations, highlighting their relative advantages. We start with the general group sparse model and then elaborate on two important special cases: the dispersive and the hierarchical models. For each, we present the models in their discrete nature, discuss how to solve the ensuing discrete problems and then describe convex relaxations. Further, we discuss efficient optimization solutions for structured sparsity problems and illustrate structured sparsity in action via two applications.

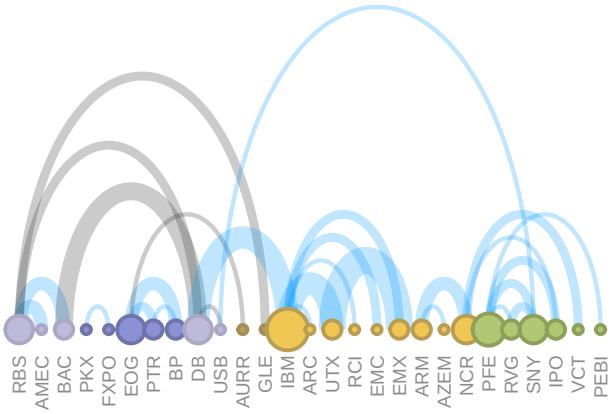
Convex optimization thrust

While *non-convex* approaches (such as the ones aforementioned) perform quite impressively in practice, they seldom come with rigorous global guarantees (or, in the best case, they rely on strong global assumptions), while they are more susceptible to model errors due to their "rigid" definition. Moreover, their applicability is usually restricted to the specific problem at hand.

In contrast, the literature on the formulation, analysis, and applications of *composite convex minimization* is ever expanding due to its broad applications in machine learning, signal processing, and statistics. By composite minimization, we refer to the following optimization problem:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^n} \{F(\mathbf{x}) \mid F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}, \tag{1}$$

where f and g are convex functions, characterizing our problem at hand. Here, f function represents the data fidelity term (e.g., least-squares objective function, logistic loss, etc.) and g "forces" the solutions in (1) to favor a low-dimensional model, depending on the nature of g.



Stock Abbr.	Company name	Stock Abbr.	Company name
RBS	Scotland Bank	ARC	Arc Document
AMEC	AMEC Group	UTX	United Tech.
BAC	Bank of America	RCI	Rogers Comm.
PKX	Posco	EMX	EMX Industries
FXPO	Ferrexpo	ARM	ARM Holdings
EOG	EOG Resources	AZEM	Azem Chemicals
PTR	PetroChina	NCR	NCR Electronics
BP	BP	PFE	Pfizer Inc.
DB	Deutsche Bank	RVG	Retro Virology
USB	U.S. Bank Corp.	SNY	Sanofi health
AURR	Aurora Russia	IPO	Intellectual Property
GLE	Glencore	VCT	Victrex Chemicals
IBM	IBM	PEBI	Port Erin BioFarma

Figure 6: We focus on four sectors: (i) bank industry (light purple), (ii) petroleum industry (dark purple), (iii) Computer science/microelectronics industry (light yellow), (iv) Pharmaceuticals/Chemistry industry (green). Using the proposed scheme, one is able to identify accurately strong correlations among stock assets from incomplete data: Positive and negative correlations are denoted with blue and black arcs, respectively. The width of the arcs denotes the strength of the correlation.

Within this context, in Chapter 5, we work in the convex domain to solve problems that are more-involved than the classical linear model, like the ones presented in Chapters 2 and 4. We propose a variable metric framework that solves instances of (1) and theoretically establish the convergence of our framework without relying on the usual Lipschitz gradient assumption on the objective. To support these theoretical developments, we describe concrete algorithmic instances of our framework for several interesting

large-scale applications and apply them on real data. As an example, in the portfolio optimization context, Figure 6 depicts some representative correlation estimates among stock assets by solving the *sparse covariance estimation problem* with the proposed algorithms. A non-exhaustive list of applications includes graphical modeling, low-light neuron image processing under Poisson noise and, sparse signal reconstruction in MRI images.

Notation and prerequisites

Throughout the thesis, plain and boldface lowercase letters represent scalars and vectors, respectively. Matrices are denoted with boldface uppercase letters. Since there are overlaps in the way we define notions for vector and matrix cases, we split the discussion in parts. Moreover, any nomenclature specific for a distinct topic is provided in the introduction of the corresponding chapter.

Vector notation: The i-th entry of a vector $\mathbf{w}$ is denoted as w_i, and $[w_i]_+ = \max(w_i, 0)$. We use superscripts or subscripts such as $\mathbf{w}^i$ or $\mathbf{w}_i$ to denote the estimate at the i-th iteration of an algorithm; the distinction is apparent from the context. Given a set $\mathcal{S} \subseteq \mathcal{N} = \{1, \ldots, n\}$, the complement $\mathcal{S}^c$ is defined with respect to $\mathcal{N}$, and the cardinality is $|\mathcal{S}|$. The support set of $\mathbf{w}$ is $\text{supp}(\mathbf{w}) = \{i : w_i \neq 0\}$. Given a vector $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{w}_\mathcal{S}$ is the projection (in $\mathbb{R}^n$) of $\mathbf{w}$ onto $\mathcal{S}$, i.e. $(\mathbf{w}_\mathcal{S})_{\mathcal{S}^c} = \mathbf{0}$, whereas $\mathbf{w}_{|\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ is $\mathbf{w}$ limited to $\mathcal{S}$ entries. The all-ones column vector is $\mathbb{1}$, with dimensions apparent from the context. We define Σ_k as the set of all k-sparse subsets of $\mathcal{N}$, and we sometimes write $\mathbf{x} \in \Sigma_k$ to mean $\text{supp}(\mathbf{x}) \in \Sigma_k$. With a slight abuse of notation, we also use the notation $\Sigma_k := \{\mathbf{x} : \|\mathbf{x}\|_0 \leq k,\ \mathbf{x} \in \mathbb{R}^n\}$; the distinction is apparent from the context.

Matrix notation: The rank of $\mathbf{X} \in \mathbb{R}^{p\times n}$ is denoted as $\text{rank}(\mathbf{X}) \leq \min\{p, n\}$. The inner product between matrices $\boldsymbol{A},\ \boldsymbol{B} \in \mathbb{R}^{p\times n}$ is denoted as $\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \text{trace}(\boldsymbol{B}^T\boldsymbol{A})$, where ^{T} represents the transpose operation. $\mathbf{I}$ represents an identity matrix with dimensions apparent from the context.

Prerequisites: To maintain a good "reading flow" in this thesis, we moved to Appendix A at the end of the thesis the necessary background to support the developments presented in the rest of the thesis. This includes norm definitions, convexity definitions and tools, an abstract definition of LDMs, projection and proximity operations and some optimization basics used in the main text. Our intension is to provide a complete set of preliminary tools that makes reading this dissertation easy. In case we believe that the reader should consult the Appendix A, we mention it explicitly in the main text.

1 Sparse Euclidean projections onto sets

Introduction

Sparse signal approximation lies at the heart of exciting developments in the areas of signal processing, machine learning, theoretical computer science and optimization. It roughly states that a sparse signal, i.e., the number of its nonzero entries is small as compared to its ambient dimension, can be perfectly reconstructed from far fewer samples than dictated by the well-known Nyquist-Shannon theorem, i.e., uniformly taking samples with frequency at least twice its highest frequency in the Fourier domain.

However, since Nyquist-Shannon theorem guarantees perfect reconstruction, *why has sparse signal approximation gained such attention?* It is well-known that sampling at Nyquist rate might be prohibitive for computationally demanding applications and creates vast amount of data which must be stored or transmitted [TLD+10]. Sparse approximation theory proposes an alternative sampling scheme (when the signal of interest can be sparsely represented using an appropriate basis [Don06, CW08]) where (i) sampling might not be periodic, (ii) samples are usually acquired through linear "sketches" of the signal of interest with an appropriate measurement matrix and, (iii) the number of measurements needed for accurate signal recovery is much less compared to traditional sampling techniques. To accomplish the above, sparsity-based optimization algorithms are required, accompanied with strong theoretical convergence and approximation guarantees.

From a different perspective, sparsity is also used as a means of solution parsimony in machine learning / applied mathematics applications [Ng04, Tib96, Nat95]. While classical criteria result in good model-fitting solutions, i.e., solutions that minimize a selected data fidelity term (e.g, the square loss, the logistic loss, etc.), in many cases they hardly provide any interpretability of the data-generating model. In fact, this inefficiency has been the center of attention over the past decade: a broad range of applications, from medical imaging [LDP07, LSDP06] and communications [CR02, HS09] to graph learning [DVR08, BEGd08, TDKC13c] and portfolio optimization [BDDM+09, KBCK13], attempt to exploit sparsity in order to reduce the solution's degrees of freedom and improve its utility and interpretability, as compared to classical approaches.

However, sparsity is generally difficult to handle in its pure form: it inherently introduces non-convexity into learning problems, which is undesirable according to the conventional wisdom. To formulate our discussion, a n-dimensional vector $\mathbf{x} \in \mathbb{R}^n$ is k-sparse with $k \leq n$ if it contains at most k non-zero entries,

i.e.,

$$\|\mathbf{x}\|_0 := |\{i \ : \ x_i \neq 0\}| = |\text{supp}(\mathbf{x})| \leq k.$$

Here, $\|\mathbf{x}\|_0$ is known as the ℓ_0 "norm"; as its name indicates, it is not a proper norm and has combinatorial nature. Furthermore, many optimization instances that include the ℓ_0 "norm" are known to be in general *combinatorially* hard to solve [Nat95, GJY11, BAd10].

While this observation jeopardizes the use of sparsity in optimization procedures, fortunately the success of sparse approximation theory lies also in the computational tractability of such task through *relaxations*. Tibshirani [Tib96], Donoho [Don06] and Candes et al. [CRT06] utilize convex relaxations of the ℓ_0 "norm" to compute a sparse solution in the underdetermined linear regression setting, with attractive approximation guarantees in polynomial time. Specifically, the ℓ_0 "norm" is replaced by its convex surrogate[1] ℓ_1-norm, since:

$$\|\mathbf{x}\|_1 \leq \max_i |x_i| \cdot \|\mathbf{x}\|_0, \quad \text{for} \quad \|\mathbf{x}\|_1 := \sum_{i=1}^{n} |x_i|, \tag{1.1}$$

where efficient convex solvers can be utilized.

Using sparsity in applications

Based on the above notions, a prevalent approach for prediction / decision-making problems with sparse solutions is to model such tasks with a convex ℓ_1*-norm constrained* optimization criterion; see [Tib96]. In particular, we are provided with a (usually) convex data fidelity / risk function $f : \mathbb{R}^n \to \mathbb{R}$ and we are interested in finding the minimizer $\mathbf{x}^\star$ such that:

$$\mathbf{x}^\star \in \underset{\mathbf{x}: \ \|\mathbf{x}\|_1 \leq \lambda}{\arg\min} \ f(\mathbf{x}), \tag{1.2}$$

where $\lambda > 0$ is a user-defined parameter that controls the sparsity of the solution. According to (1.1), smaller values of λ lead to sparser solutions.[2]

A popular scheme for solving (1.2), known for its simplicity and ease of implementation, is the *projected (sub)gradient descent* algorithm [Gol64, LP66, Ber82]:[3] Per iteration, the main computational complexity is due to the calculation of the (sub)gradient of f and the projection operation onto the set $\mathcal{IC}_\lambda^1$, according to the next definition.

Definition 1 (Inequality-norm sets)**.** *We use $\mathcal{IC}_\lambda^\alpha$ to denote an* inequality-constrained ℓ_α*-norm set with parameter λ such that $\mathcal{IC}_\lambda^\alpha := \{\mathbf{w} \ : \ \|\mathbf{w}\|_\alpha \leq \lambda\}$.*

The above lead to the following projected (sub)gradient descent recursion for solving (1.2):

$$\mathbf{x}_{i+1} := \mathcal{P}_{\mathcal{IC}_\lambda^1}\left(\mathbf{x}_i - \frac{\mu}{2}\mathbf{v}_i\right), \quad \mathbf{v}_i \in \partial f(\mathbf{x}_i), \tag{1.3}$$

[1] For signals with *bounded* energy, ℓ_1-norm is the closest convex surrogate to ℓ_0 "norm". We highlight that, while ℓ_0 "norm" does not depend on the scaling of the individual entries of $\mathbf{x}$, ℓ_1-norm inserts the notion of scaling in the optimization procedure.

[2] However, one can easily observe that such reasoning is misleading (it is not a *necessary* condition). E.g., a solution with many small entries such that $\|\mathbf{x}\|_1$ is small does not imply sparsity in $\mathbf{x}$ for arbitrary n.

[3] Please, also refer to the Appendix A for further information.

where $\mathbf{x}_i$ denotes the current estimate, $\mu > 0$ is a properly selected step size, $\mathbf{v}_i$ represents a (sub)gradient of f around $\mathbf{x}_i$ and, $\mathcal{P}_{\mathcal{IC}^1_\lambda}(\cdot)$ is defined as:

$$\mathcal{P}_{\mathcal{IC}^1_\lambda}(\mathbf{y}) = \underset{\mathbf{x}:\ \mathbf{x}\in\mathcal{IC}^1_\lambda}{\arg\min} \|\mathbf{x}-\mathbf{y}\|_2^2; \tag{1.4}$$

in (1.3), the input signal $\mathbf{y}$ is $\mathbf{y} := \mathbf{x}_i - \frac{\mu}{2}\mathbf{v}_i$ at the i-th iteration. In the $\mathcal{IC}^1_\lambda$ setting, [DSSSC08] shows that the projection (1.4) can be efficiently computed in at most $\mathcal{O}(n \log n)$ time-complexity. [LY09] casts (1.4) as a root finding bisection method that achieves an ε-close solution vector in linear time complexity, where $\varepsilon > 0$ is a user-defined parameter.

While such sparsity-inducing norm approaches are impressive in practice, the solution returned by (1.2) might not have the desired sparsity for data interpretation. Fine-tuning of the parameter λ usually leads to solutions with sparsity level close to the desired one, but successive application of (1.2) is required to achieve the desiderata. Moreover, there are problem cases where any additional constraints might conflict with $\mathcal{IC}^1_\lambda$, thus leading to invalid or non-sparse solutions, as we show next.

Chapter roadmap

Our intention in this chapter is to leverage both *combinatorial* (such as the ℓ_0 "norm") and *norm constraints* to guide such variate selection processes under different settings. A key actor for this task is an efficient *sparse projection operation over norm and linear constraints* that goes beyond simple selection heuristics, with provable solution quality as well as attractive runtime/memory performance. In the next sections, we first provide some non-trivial and useful key lemmas, regarding sparse Euclidean projections onto ℓ_2- and ℓ_∞-norm constraints (Section 1.3). We continue our discussions with sparse Euclidean projections onto simplex-type of constraints in Section 1.4. Finally, this chapter concludes with real application examples where some of the aforementioned projections are used and compared with state of the art algorithms; see Section 1.5.

This chapter is based on the joint work with Volkan Cevher, Stephen Becker and Christoph Koch [KBCK13].

1.1 Preliminaries

We first define the pure sparse projection operator, $\mathcal{P}_{\Sigma_k}(\cdot)$, where k denotes the desired sparsity level:

$$\mathcal{P}_{\Sigma_k}(\mathbf{y}) = \underset{\mathbf{x}:\ \mathbf{x}\in\Sigma_k}{\arg\min} \|\mathbf{x}-\mathbf{y}\|_2^2. \tag{1.5}$$

This projection is generally known as the *hard-thresholding operation*. Here, $\mathcal{P}_{\Sigma_k}(\mathbf{y})$ represents the best k-sparse approximation of $\mathbf{y} \in \mathbb{R}^n$ over all vectors in Σ_k, the non-convex set of k-sparse vectors with appropriate dimensions. While $\mathcal{P}_{\Sigma_k}(\cdot)$ is a *combinatorial* operation, there is an obvious and intuitive solution to it: one only requires to determine the k largest in magnitude elements of $\mathbf{y}$ in $\mathcal{O}(n \log n)$ time-cost.

In stark contrast to (1.2), in the discussions below we deal with the more demanding and non-convex case:

$$\mathbf{x}^\star \in \underset{\mathbf{x}:\ \Sigma_k \cap \#}{\arg\min} f(\mathbf{x}), \tag{1.6}$$

where # represents additional structure constraints on the variate solution vector; here, Σ_k forces sparsity in the solution such that $\|\mathbf{x}^\star\|_0 \leq k$. The key ingredient in solving (1.6) is the following non-convex projection operation:

$$\mathcal{P}_{\Sigma_k \cap \#}(\mathbf{y}) = \underset{\mathbf{x}:\ \mathbf{x} \in \Sigma_k \cap \#}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{1.7}$$

Equality-constrained vs. inequality-constrained norm sets

In the case where # represents norm constraints, inequality-constrained norm sets $\mathcal{IC}_\lambda^\alpha$ are most commonly used in practice[4]. However, in this chapter, we also present more demanding problem cases where the norm constraints are only satisfied with *equality* and, thus, are *non-convex* sets.

Definition 2 (Equality-norm sets). *We use $\mathcal{C}_\lambda^\alpha$ to denote an* equality-constrained *ℓ_α-norm set with parameter λ such that $\mathcal{C}_\lambda^\alpha := \{\mathbf{w} \ :\ \|\mathbf{w}\|_\alpha = \lambda\}$.*

In the next sections, we either use equality or inequality norm constraints, depending on the problem at hand.

Convergence of projected (sub)gradient descent methods

The optimization criterion studied and used in the applications part of this section is:

$$\mathbf{x}^\star \in \underset{\mathbf{x}:\ \mathbf{x} \in \Omega}{\arg\min}\, f(\mathbf{x}), \tag{1.8}$$

where $\Omega \subseteq \mathbb{R}^n$ and f is a convex function, bounded below on Ω and with non-empty domain, intersected with Ω. In our examples, we further often assume that f is a *continuously differentiable* function.[5]

Under this setting, (1.8) becomes:

$$\mathbf{x}_{i+1} := \mathcal{P}_\Omega \left(\mathbf{x}_i - \frac{\mu}{2} \nabla f(\mathbf{x}_i) \right), \tag{1.9}$$

where $\nabla f(\mathbf{x}_i)$ denotes the gradient of f at the putative point $\mathbf{x}_i$. For the case where Ω is convex, e.g., $\Omega \equiv \mathcal{IC}_\lambda^\alpha$ for $\alpha > 1$, [CM87] shows that, if $\{\mathbf{x}_i\}_{i \geq 0}$ is a sequence generated by (1.9), then any limit point $\{\mathbf{x}_i\}_{i \geq 0}$ (as $i \to \infty$) is a *stationary point* of (1.8), i.e., the sequence of projected gradients $\{\|\nabla f(\mathbf{x}_i)\|_2\}_{i \geq 0} \to \mathbf{0}$.[6]

Unfortunately, for the case where Ω is a *non-convex set*, similar convergence guarantees for the projected gradient descent framework are not generally known. However, our proposed solutions to non-convex projection operations can be also used in an *alternating projection* algorithm [ARS07], where, surprisingly, it is possible to obtain *stationary points* of general loss functions under very mild conditions [ABRS10].

[4] Due to their convexity, as long as $\mathcal{IC}_\lambda^\alpha$ is convex.
[5] Please, refer to the Appendix A for a rigorous definition of these notions.
[6] We underline that a stationary point or critical point $\mathbf{x}_i$ is an estimate where the gradient of f at $\mathbf{x}_i$ is zero.

1.2 Related work

To the best of our knowledge, there are only a few published works on sparse Euclidean projections onto general constraints. The work of Luss and Teboulle [LT13] on sparse Principal Component Analysis (sparse PCA) provides some preliminary results on sparse projections of the form:

$$\mathcal{P}_{\Sigma_k \cap \mathcal{C}_1^2}(\mathbf{y}) = \underset{\mathbf{x}:\ \mathbf{x} \in \Sigma_k \cap \mathcal{C}_1^2}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2,$$

Recently, we became aware of a recent work of Beck and Hallak [BH14] that concentrate on the minimization over sparse symmetric sets, a notion that generalizes the ideas presented in this chapter.

1.3 Sparse Euclidean projections onto norm constraints

A general description of the Euclidean projection we focus on is given next:

PROBLEM 1.1. *Given an anchor vector $\mathbf{y} \in \mathbb{R}^n$, a desired sparsity level k and a norm constraint $\mathcal{C}_\lambda^\alpha$ / $\mathcal{IC}_\lambda^\alpha$, we are interested in the optimization problems:*

$$\mathcal{P}_{\Sigma_k \cap \mathcal{C}_\lambda^\alpha}(\mathbf{y}) \in \underset{\mathbf{x}:\ \mathbf{x} \in \Sigma_k \cap \mathcal{C}_\lambda^\alpha}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2 \quad \text{or} \quad \mathcal{P}_{\Sigma_k \cap \mathcal{IC}_\lambda^\alpha}(\mathbf{y}) \in \underset{\mathbf{x}:\ \mathbf{x} \in \Sigma_k \cap \mathcal{IC}_\lambda^\alpha}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{1.10}$$

Here, we consider two well-known norm cases: for $\mathcal{C}_\lambda^\alpha$, we consider $\alpha = 2$, and for $\mathcal{IC}_\lambda^\alpha$, we consider $\alpha = \infty$. The selection of these norms is due to their presence in important real problem instances. To motivate our discussion, we describe next two applications:

1. **Sparse Principal Component Analysis (PCA):** Finding the *principal component* of a given data matrix $\mathbf{A}$ is an important task in data analysis: such component explains most of the variance in data, naturally leading to a compression / dimensionality reduction scheme. In particular, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix such that $\mathbf{A} \succeq 0$.[7] To find the principal component, we solve [Jol05]:

   $$\mathbf{x}^\star \in \underset{\mathbf{x}:\ \mathbf{x} \in \mathcal{C}_\lambda^2}{\arg\max}\, \mathbf{x}^T \mathbf{A} \mathbf{x},$$

   with optimal solution the eigenvector corresponding to the maximum eigenvalue of the matrix $\mathbf{A}$. However, such an $\mathbf{x}^\star$ makes hardly any space for data interpretation: $\mathbf{x}^\star$ is usually dense and thus all features contribute in defining the direction that explains most of the data.

   In sparse PCA [dEGJL04], we place a sparse prior on the support pattern of $\mathbf{x}^\star$. Sparse PCA looks for k-sparse linear combinations of variables (i.e., principal components) that correspond to directions of maximal variance in the data, according to:

   $$\mathbf{x}^\star \in \underset{\mathbf{x}:\ \mathbf{x} \in \Sigma_k \cap \mathcal{C}_\lambda^2}{\arg\max}\, \mathbf{x}^T \mathbf{A} \mathbf{x}.$$

   Here, using the projection onto $\Sigma_k \cap \mathcal{C}_\lambda^2$, we try to predispose the learning mechanism to return more interpretable sparse solutions. Other application examples of $\mathcal{C}_\lambda^2$ include 1-bit compressive sensing (see eq. (15) in [BB08] where a straightforward reformulation of the problem leads to (1.6)

[7] In most cases, $\mathbf{A}$ represents a covariance matrix.

where $\# := \mathcal{C}^2_\lambda$), Independent Component Analysis (ICA) (see eqs. (1)-(2) in [DAK00]), etc.

2. **Sparse feature selection in clustering:** Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a data matrix of m measurements and n features. In classical clustering methods, one desires to cluster the observations, based on the features. Currently, the dominating techniques for data clustering are based on hierarchical clustering [WJ63] and on flat/centroid-based clustering (e.g., K-Means based) [Ste56, Mac67]. However, due to noise or possible presence of outliers, one suspects that the true underlying clusters can be identified using only some of the features available.

   [WT10] propose a framework for *sparse* clustering where clusters are obtained using a sparse subset of the features. Abstractly, this leads to the following optimization criterion:

$$\underset{\mathbf{x},\boldsymbol{\Theta}}{\text{maximize}} \quad \sum_{i=1}^{n} x_i f_i(\mathbf{A}_i, \boldsymbol{\Theta}) \quad \text{subject to} \quad \boldsymbol{\Theta} \in \mathcal{D}, \quad \mathbf{x} \in \mathcal{C}^2_\lambda \cap \mathcal{IC}^1_\lambda, \quad x_i \geq 0.$$

   Here, $\boldsymbol{\Theta}$ is a parameter restricted to lie in a problem-dependent set $\mathcal{D}$ and $f_i(\cdot)$ is a function that involves the i-th feature of $\mathbf{A}$; e.g., $f_i(\cdot)$ can be the sum of squares distances between clusters for feature i. For more detailed description of the problem, we refer the reader to [WT10]. Recently, [CWLX14] identify that many "noise" features are still present in the final clustering results, jeopardizing the utility of $\mathcal{IC}^1_\lambda$ sparsity constraint. As an alternative, they propose the following criterion:

$$\underset{\mathbf{x},\boldsymbol{\Theta}}{\text{maximize}} \quad \sum_{i=1}^{n} x_i f_i(\mathbf{A}_i, \boldsymbol{\Theta}) \quad \text{subject to} \quad \boldsymbol{\Theta} \in \mathcal{D}, \quad \mathbf{x} \in \Sigma_k \cap \mathcal{IC}^\infty_\lambda, \quad x_i \geq 0,$$

   which forces the solution to be k-sparse. In our context and for fixed $\boldsymbol{\Theta}$, it turns out that the problem at hand is [CWLX14]:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{x}-\mathbf{a}\|_2^2 \\ \text{subject to} \quad & \mathbf{x} \in \Sigma_k \cap \mathcal{IC}^\infty_\lambda, \quad x_i \geq 0. \end{aligned}$$

   where $\mathbf{a}$ is a given vector, depending on the nature of $f_i(\cdot)$ and the current value $\boldsymbol{\Theta}$.[8]

Maintaining the constrained optimization criterion, as described in the problems above, the projection operations in (1.10) are indispensable tools towards solution for these tasks.

1.3.1 Sparse projection onto ℓ_2-norm constraints

In the case of $\alpha = 2$, one can easily observe the following for $\mathcal{C}^2_\lambda$ in (1.10):

$$\begin{aligned} \min_{\mathbf{x}:\ \mathbf{x}\in\Sigma_k\cap\mathcal{C}^2_\lambda} \|\mathbf{x}-\mathbf{y}\|_2^2 &= \min_{\mathbf{x}:\ \mathbf{x}\in\Sigma_k\cap\mathcal{C}^2_\lambda} \left\{\|\mathbf{x}\|_2^2 + \|\mathbf{y}\|_2^2 - 2\langle \mathbf{x},\mathbf{y}\rangle\right\} \\ &= \min_{\mathbf{x}:\ \mathbf{x}\in\Sigma_k\cap\mathcal{C}^2_\lambda} \left\{\lambda^2 + \|\mathbf{y}\|_2^2 - 2\langle \mathbf{x},\mathbf{y}\rangle\right\} \qquad \text{(By forcing the norm constraint)} \\ &\propto \max_{\mathbf{x}:\ \mathbf{x}\in\Sigma_k\cap\mathcal{C}^2_\lambda} \langle \mathbf{x},\mathbf{y}\rangle \end{aligned}$$

The solution to **PROBLEM 1.1** is given by the following useful lemma:

[8] Here, the positive constraints can easily be incorporated in the proposed solutions and, for simplicity, they are omitted in the next sections.

Lemma 1. *Let $\mathbf{y} \in \mathbb{R}^n$ be a given vector. Then:*

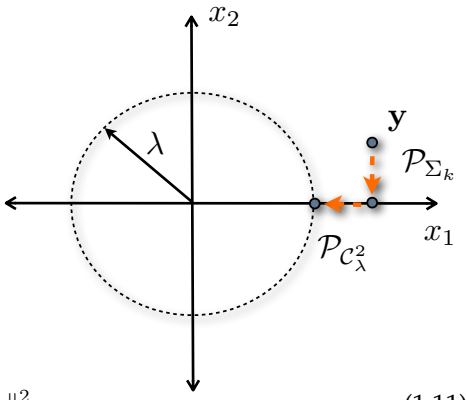
$$\widehat{\mathbf{x}} \in \mathcal{P}_{\Sigma_k \cap \mathcal{C}_\lambda^2}(\mathbf{y}) = \underset{\mathbf{x}:\, \mathbf{x} \in \Sigma_k \cap \mathcal{C}_\lambda^2}{\arg\max} \langle \mathbf{x}, \mathbf{y} \rangle,$$

such that, for $\widehat{\mathcal{S}} = supp\left(\mathcal{P}_{\Sigma_k}(\mathbf{y})\right)$ with $|\widehat{\mathcal{S}}| \leq k$, $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}^c} = \mathbf{0}$ and $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}} = \lambda \cdot \frac{\mathcal{P}_{\Sigma_k}(\mathbf{y})}{\|\mathcal{P}_{\Sigma_k}(\mathbf{y})\|_2}$.

Proof. By the Cauchy-Schwartz inequality and for fixed $\mathbf{y}$, we have: $\langle \mathbf{x}, \mathbf{y} \rangle \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$ where the equality is satisfied if and only if $\mathbf{x} = \alpha \mathbf{y}$, $\alpha \in \mathbb{R}$. Furthermore, by forcing the norm constraint $\mathbf{x} \in \mathcal{C}_\lambda^2$, we observe that this inequality is satisfied if and only if $\mathbf{x} = \lambda \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|_2}$. By these observations and using a *water-filling* argument, the optimal point $\widehat{\mathbf{x}}$ in Lemma 1 contains the k largest in magnitude elements of $\mathbf{y}$, normalized by the their total Euclidean norm and weighted with λ to satisfy $\mathcal{C}_\lambda^2$ constraint. □

See Figure 1.1 for a schematic representation.

Figure 1.1: Schematic representation of joint projection onto $\Sigma_1 \cap \mathcal{C}_\lambda^2$.



1.3.2 Sparse projection onto ℓ_∞-norm constraints

For this case, we first require the Euclidean projection onto convex ℓ_∞-norm sets:

$$\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y}) = \underset{\mathbf{x}:\, \mathbf{x} \in \mathcal{IC}_\lambda^\infty}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{1.11}$$

One can easily observe that the optimal Euclidean projection onto ℓ_∞-norm balls with radius λ is given by the *clipping operation* where:

$$\left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i = \begin{cases} y_i & \text{if } |y_i| \leq \lambda, \\ \lambda \cdot \text{sign}(y_i) & \text{if } |y_i| > \lambda \end{cases} = \text{sign}(y_i) \cdot \min\left\{\lambda,\ |y_i|\right\}. \tag{1.12}$$

Based on this, the solution to **PROBLEM 1.1** for $\mathcal{IC}_\lambda^\infty$ is given by the following useful lemma:

Lemma 2. *Let $\mathbf{y} \in \mathbb{R}^n$ be a given non-zero vector. Then:*

$$\widehat{\mathbf{x}} \in \mathcal{P}_{\Sigma_k \cap \mathcal{IC}_\lambda^\infty}(\mathbf{y}) = \underset{\mathbf{x}:\, \mathbf{x} \in \Sigma_k \cap \mathcal{IC}_\lambda^\infty}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2,$$

such that, for $\widehat{\mathcal{S}} := supp\left(\mathcal{P}_{\Sigma_k}(\mathbf{y})\right)$ with $|\widehat{\mathcal{S}}| \leq k$, $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}^c} = \mathbf{0}$ and $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}} = \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_{\widehat{\mathcal{S}}}$.

Proof. An important observation is given next:

Remark 1. *The problem in* (1.10) *for $\mathcal{IC}_\lambda^\infty$ can be equivalently transformed into the following nested minimization:*

$$\left\{\widehat{\mathcal{S}},\ \widehat{\mathbf{x}}_{\widehat{\mathcal{S}}}\right\} \leftarrow \underset{\mathcal{S}:\, |\mathcal{S}| \leq k}{\arg\min} \left\{ \min_{\mathbf{x}_{\mathcal{S}}:\, \mathbf{x}_{\mathcal{S}} \in \mathcal{IC}_\lambda^\infty} \|(\mathbf{x} - \mathbf{y})_{\mathcal{S}}\|_2^2 + \|\mathbf{y}_{\mathcal{S}^c}\|_2^2 \right\}. \tag{1.13}$$

Assume that the best support set $\widehat{\mathcal{S}}$ is known a priori; the inner minimization in (1.13) computes a solution vector, with support restricted in $\widehat{\mathcal{S}}$, that minimizes the distance to the input vector $\mathbf{y}$. In particular:

$$\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}} = \underset{\mathbf{x}_{\widehat{\mathcal{S}}}:\, \mathbf{x}_{\widehat{\mathcal{S}}} \in \mathcal{IC}_\lambda^\infty}{\arg\min} \|(\mathbf{x}-\mathbf{y})_{\widehat{\mathcal{S}}}\|_2^2 + \|\mathbf{y}_{\widehat{\mathcal{S}}^c}\|_2^2 = \underset{\mathbf{x}_{\widehat{\mathcal{S}}}:\, \mathbf{x}_{\widehat{\mathcal{S}}} \in \mathcal{IC}_\lambda^\infty}{\arg\min} \|(\mathbf{x}-\mathbf{y})_{\widehat{\mathcal{S}}}\|_2^2.$$

According to (1.12), $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}} = \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_{\widehat{\mathcal{S}}}$. Substituting in (1.13), $\widehat{\mathcal{S}}$ is computed as:

$$\begin{aligned}\widehat{\mathcal{S}} &= \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\min} \left[\| \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})-\mathbf{y}\right)_{\mathcal{S}} \|_2^2 + \|\mathbf{y}_{\mathcal{S}^c}\|_2^2\right] = \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \left[\|\mathbf{y}\|_2^2 - \| \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})-\mathbf{y}\right)_{\mathcal{S}^c} \|_2^2 - \|\mathbf{y}_{\mathcal{S}}\|_2^2\right] \\ &= \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \left[\|\mathbf{y}_{\mathcal{S}}\|_2^2 - \| \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})-\mathbf{y}\right)_{\mathcal{S}} \|_2^2\right] = \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \sum_{i\in\mathcal{S}} \left(2\left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i \cdot y_i - \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i^2\right) \end{aligned} \tag{1.14}$$

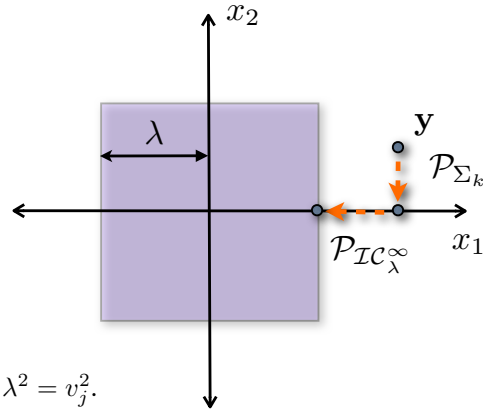
We observe that $\forall i \in \mathcal{S}$:

1. If $|y_i| > \lambda$, then $2\left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i \cdot y_i - \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i^2 = 2\lambda|y_i| - \lambda^2 = \lambda(2|y_i| - \lambda) > 0$.
2. If $|y_i| \le \lambda$, then $2\left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i \cdot y_i - \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i^2 = 2|y_i|^2 - |y_i|^2 = |y_i|^2 > 0$.

Let $v_i^2 := 2\left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i \cdot y_i - \left(\mathcal{P}_{\mathcal{IC}_\lambda^\infty}(\mathbf{y})\right)_i^2 > 0$, $\forall i \in \mathcal{S}$. We observe that (1.14) is a modular maximization problem:

$$\widehat{\mathcal{S}} = \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \|\mathbf{v}_{\mathcal{S}}\|_2^2 \tag{1.15}$$

Figure 1.2: Schematic representation of joint projection onto $\Sigma_1 \cap \mathcal{C}_\lambda^\infty$.



Moreover, for any $i, j \in \mathcal{N}$ with $|y_i| \ge |y_j|$, we observe:

1. If $|y_i| > \lambda$ and $|y_j| > \lambda$, then $v_i^2 = 2\lambda|y_i| - \lambda^2 \ge 2\lambda|y_j| - \lambda^2 = v_j^2$.
2. If $|y_i| > \lambda$ and $|y_j| \le \lambda$, then $v_i^2 = 2\lambda|y_i| - \lambda^2 \ge \lambda^2 \ge |y_j|^2 = 2|y_j|^2 - |y_j|^2 = v_j^2$.
3. If $|y_j| \le |y_i| < \lambda$, then $v_i^2 = 2|y_i|^2 - |y_i|^2 \ge 2|y_j|^2 - |y_j|^2 = v_j^2$.

Thus, there is equivalence between the index set $\widehat{\mathcal{S}}$ that maximizes $\|\mathbf{v}_{\mathcal{S}}\|_2^2$ and the index set that maximizes $\|\mathbf{y}_{\mathcal{S}}\|_2^2$, i.e., $\|\mathbf{y}_{\mathcal{S}_1}\|_2^2 \ge \|\mathbf{y}_{\mathcal{S}_2}\|_2^2 \Rightarrow \|\mathbf{v}_{\mathcal{S}_1}\|_2^2 \ge \|\mathbf{v}_{\mathcal{S}_2}\|_2^2$ for $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{N}$ with $|\mathcal{S}_1| = |\mathcal{S}_2|$. Therefore, we may conclude:

$$\mathcal{S}^* = \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \|\mathbf{v}_{\mathcal{S}}\|_2^2 = \underset{\mathcal{S}:|\mathcal{S}|\le k}{\arg\max} \|\mathbf{y}_{\mathcal{S}}\|_2^2 \tag{1.16}$$

As in the equality ℓ_2-norm case, $\widehat{\mathcal{S}}$ can be determined by sorting and keeping the k largest (in absolute value) elements of $\mathbf{y}$. □

1.4 Sparse Euclidean projections onto the simplex

While many learning methods with sparsity constraints can accommodate convex relaxations for solution in practice, i.e., one can still promote sparsity in (1.10) by relaxing the sparse set Σ_k into the ℓ_1-norm set:

$$\mathcal{P}_{\mathcal{IC}_\rho^1 \cap \mathcal{C}_\lambda^\alpha}(\mathbf{y}) \in \underset{\mathbf{x}:\, \mathbf{x}\in\mathcal{C}_\rho^1\cap\mathcal{IC}_\lambda^\alpha}{\arg\min} \|\mathbf{x}-\mathbf{y}\|_2^2 \quad \text{or} \quad \mathcal{P}_{\mathcal{IC}_\rho^1 \cap \mathcal{IC}_\lambda^\alpha}(\mathbf{y}) \in \underset{\mathbf{x}:\, \mathbf{x}\in\mathcal{IC}_\rho^1\cap\mathcal{IC}_\lambda^\alpha}{\arg\min} \|\mathbf{x}-\mathbf{y}\|_2^2,$$

there are important learning applications that cannot benefit from this approach. As a stylized example, consider the sparse portfolio optimization setting [Mar52]:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^T \widehat{\mathbf{\Sigma}} \mathbf{w} - \tau \widehat{\boldsymbol{\mu}}^T \mathbf{w} \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i = 1, \quad w_i \geq 0, \; i = 1, \ldots, n, \end{aligned} \tag{1.17}$$

where $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{n \times n}$ and $\widehat{\boldsymbol{\mu}} \in \mathbb{R}^n$ are the sample covariance matrix and the expected return, respectively, of n assets over a predefined period of time, and $\tau > 0$ is a regularizer parameter that trades-off risk and return. The solution $\mathbf{w}^\star$ in (1.17) is the *no-short-positions* distribution of investments over the n available assets due to the *simplex constraints*: In this case, we only propose asset allocations for the next stock market. Within the same framework, one can solve the *short-positions* case by dropping the non-negativity constraints, where both stock buys/sells can be predicted. In both cases though, it is clear that direct applications of the ℓ_1-norm (regularization, constraint, or otherwise) cannot achieve further sparsification, beyond what the simplex constraint obtains: the constraints in (1.17) are identical to $\mathcal{C}_1^1$.

But, why we require sparsity in such cases? In the context of sparse portfolio optimization, we typically need sparse solutions due to two reasons. The first reason is *robustness*: since empirical covariance and return estimates are rather noisy, sparse portfolios generalize better [DGNU09, BDDM$^+$09]. The second reason is *transaction cost*: a sparse portfolio with a few active assets is usually desired where cardinality constraints best model the total transaction costs. Other examples of learning problems with sparsity regularization and simplex constraints include sparse mixture/kernel density estimation [BTWB10], boosting/leveraging weak classifiers [RSS$^+$00].

Thus, a key step in these type of problems is the *sparse projection onto the simplex (type of) constraints*:

PROBLEM 1.2: Given $\mathbf{y} \in \mathbb{R}^n$, find a Euclidean projection of $\mathbf{y}$ onto the intersection of k-sparse vectors Σ_k and the simplex $\Delta_\lambda^+ = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0, \; \sum_i x_i = \lambda\}$:

$$\mathcal{P}_{\Sigma_k \cap \Delta_\lambda^+}(\mathbf{y}) \in \underset{\mathbf{x}: \; \mathbf{x} \in \Sigma_k \cap \Delta_\lambda^+}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{1.18}$$

PROBLEM 1.3: Replace Δ_λ^+ in (1.18) with the hyperplane constraint $\Delta_\lambda = \{\mathbf{x} \in \mathbb{R}^n \; : \; \sum_i \mathbf{x}_i = \lambda\}$.

Figure 1.3 provides a visual description of the above notions. *We prove that it is possible to compute such projections in quasilinear time via simple greedy algorithms.*

To the best of our knowledge, sparse Euclidean projections onto the simplex and hyperplane constraints have not been considered before, until very recently in [BH14]; there, the authors propose a general framework for sparse projections onto symmetric sets, inspired by this work. From a different perspective, [PEGC12] considers cardinality regularized loss function minimization subject to simplex constraints. Their convexified approach relies on solving a lower-bound to the objective function and has $\mathcal{O}(n^4)$ complexity, which is not scalable and practical. However, the proposed framework can also accommodate additional linear constraints, which partially justifies the increased computational time cost. Though, we also note that *regularizing* with the cardinality constraints is generally easier: e.g., our projectors become simpler.

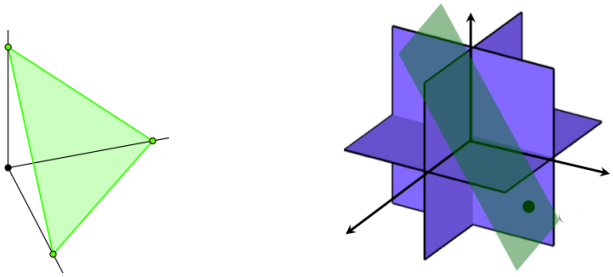


Figure 1.3: (**Left panel**): Toy example illustration of Δ_λ^+ in 3-dimensions. (**Right panel**): Visual representation of the set intersection $\Sigma_k \cap \Delta_\lambda$. Our algorithmic solutions give exact solutions in $\mathcal{O}(n \log n)$, as shown next.

1.4.1 Convex simplex projections and other definitions

Without loss of generality, assume $\mathbf{y} \in \mathbb{R}^n$ is sorted in descending order; so, y_1 is the largest element. We use $\mathcal{P}_{\Delta_\lambda^+}$ to denote the (convex) Euclidean projector onto Δ_λ^+, and $\mathcal{P}_{\Delta_\lambda}$ for its extension to Δ_λ.

Lemma 3 (Euclidean projection $\mathcal{P}_{\Delta_\lambda^+}$ [DSSSC08])**.** *The projector onto the simplex is given by*

$$(\mathcal{P}_{\Delta_\lambda^+}(\mathbf{y}))_i = [y_i - \tau]_+, \text{ where } \tau := \frac{1}{\rho}\left(\sum_{i=1}^{\rho} y_i - \lambda\right) \quad \text{for } \rho := \max\{j : y_j > \frac{1}{j}(\sum_{i=1}^{j} y_i - \lambda)\}.$$

Lemma 4 (Euclidean projection $\mathcal{P}_{\Delta_\lambda}$)**.** *The projector onto the extended simplex is given by*

$$(\mathcal{P}_{\Delta_\lambda}(\mathbf{y}))_i = y_i - \tau, \quad \text{where} \quad \tau = \frac{1}{n}\left(\sum_{i=1}^{n} y_i - \lambda\right).$$

For our analysis, we also define the following operator:

Definition 3 (Operator $\mathcal{P}_{L_k}$)**.** *We define $\mathcal{P}_{L_k}(\mathbf{y})$ as the operator that keeps the k-largest entries of $\mathbf{y}$ (not in magnitude) and sets the rest to zero. This operation can be computed in $\mathcal{O}(n \cdot \min(k, \log n))$-time.*

1.4.2 Greedy selectors for sparse simplex-type projections

Let $\widehat{\mathbf{x}} := \mathcal{P}_{\Sigma_k \cap \Delta_\lambda^+}(\mathbf{y})$ or $\widehat{\mathbf{x}} := \mathcal{P}_{\Sigma_k \cap \Delta_\lambda}(\mathbf{y})$ be the projection of $\mathbf{y}$ onto $\Sigma_k \cap \Delta_\lambda^+$ or $\Sigma_k \cap \Delta_\lambda$, respectively, with $\text{supp}(\widehat{\mathbf{x}}) = \widehat{\mathcal{S}}$. Similar to Remark 1, we make the following elementary observation:

Remark 2. **PROBLEM 1.2** *and* **PROBLEM 1.3** *can be equivalently transformed into:*

$$\{\widehat{\mathcal{S}}, \widehat{\mathbf{x}}\} \leftarrow \underset{\mathcal{S}:\mathcal{S}\in\Sigma_k}{\arg\min} \left\{ \min_{\substack{\mathbf{x}:\mathbf{x}_\mathcal{S}\in\Delta_\lambda^+ \text{ or } \Delta_\lambda, \\ \mathbf{x}_{\mathcal{S}^c}=0}} \|(\mathbf{x}-\mathbf{y})_{|\mathcal{S}}\|_2^2 + \|(\mathbf{y})_{|\mathcal{S}^c}\|_2^2 \right\}.$$

Therefore, given $\widehat{\mathcal{S}} = \text{supp}(\widehat{\mathbf{x}})$, we can find $\widehat{\mathbf{x}}$ by projecting $\mathbf{y}_{\widehat{\mathcal{S}}}$ onto Δ_λ^+ or Δ_λ within the k-dimensional

Algorithm 1 GSSP

1: **Input:** $\boldsymbol{y}$, k, λ
2: $\widehat{\mathcal{S}} = \text{supp}(\mathcal{P}_{L_k}(\mathbf{y}))$ *(Select support)*
3: $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}} = \mathcal{P}_{\Delta_\lambda^+}(\mathbf{y}_{|\widehat{\mathcal{S}}}), \widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}^c} = 0$ *(Final projection)*

space. Thus, the difficulty is in finding $\widehat{\mathcal{S}}$. Hence, we split the problem into the task of finding the support and then finding the values on the support.

Focusing on **PROBLEM 1.2** and given support $\widehat{\mathcal{S}}$, $\widehat{\mathbf{x}}$ satisfies $\widehat{\mathbf{x}}_{\widehat{\mathcal{S}}^c} = 0$ and $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}} = \mathcal{P}_{\Delta_\lambda^+}(\mathbf{y}_{|\widehat{\mathcal{S}}})$. Then, $\widehat{\mathcal{S}}$ is given by:

$$\widehat{\mathcal{S}} \in \underset{\mathcal{S}:\mathcal{S}\in\Sigma_k}{\arg\min} \left\{ \|\mathcal{P}_{\Delta_\lambda^+}(\mathbf{y}_{|\mathcal{S}}) - \mathbf{y}_{|\mathcal{S}}\|_2^2 + \|\mathbf{y}_{|\mathcal{S}^c}\|_2^2 \right\} = \underset{\mathcal{S}:\mathcal{S}\in\Sigma_k}{\arg\max} F_+(\mathcal{S}), \tag{1.19}$$

where $F_+(\mathcal{S}) := \sum_{i\in\mathcal{S}} \left(y_i^2 - ((\mathcal{P}_{\Delta_\lambda^+}(\mathbf{y}_{|\mathcal{S}}))_i - y_i)^2 \right)$.

This function can be simplified to

$$F_+(\mathcal{S}) = \sum_{i\in\mathcal{S}} (y_i^2 - \tau^2), \tag{1.20}$$

where $\tau := \frac{1}{|\mathcal{S}|}\left(\sum_{i\in\mathcal{S}} y_i - \lambda\right)$ (i.e., depends on S) is due to Lemma 5.

Lemma 5. *Let $\mathbf{q} = \mathcal{P}_{\Delta_\lambda^+}(\mathbf{y})$ where $q_i = [y_i - \tau]_+$, according to Definition 3. Then, $y_i \geq \tau$ for all $i \in \mathcal{S} = supp(\mathbf{q})$. Furthermore, $\tau = \frac{1}{|\mathcal{S}|}\left(\sum_{i\in\mathcal{S}} y_i - \lambda\right)$.*

Sketch of proof. For the first part of the lemma, the intuition is simple: the "threshold" τ should be smaller than the smallest entry in the selected support, or we unnecessarily shrink the coefficients that are larger without introducing any new support to the solution. Same arguments apply to inflating the coefficients to meet the simplex budget. Given this, the selection of τ comes directly from the definition of τ in Definition 3. □

Similarly for **PROBLEM 1.3**, we conclude that $\widehat{\mathbf{x}}$ satisfies $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}} = \mathcal{P}_{\Delta_\lambda}(\mathbf{y}_{|\widehat{\mathcal{S}}})$ and $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}^c} = 0$, where

$$\widehat{\mathcal{S}} \in \underset{\mathcal{S}:\mathcal{S}\in\Sigma_k}{\arg\max} F(\mathcal{S}), \tag{1.21}$$

for $F(\mathcal{S}) := \left(\sum_{i\in\mathcal{S}} y_i^2\right) - \frac{1}{|\mathcal{S}|}(\sum_{i\in\mathcal{S}} y_i - \lambda)^2$, using similar reasoning.

Sparse projections onto Δ_λ^+ and Δ_λ

Based on the above, we propose two algorithms for sparse projections onto simplex constraints. Algorithm 1 suggests a greedy approach for the projection onto $\Sigma_k \cap \Delta_\lambda^+$: In this case, we select the set $\widehat{\mathcal{S}}$ by projecting $\mathbf{y}$ using the $\mathcal{P}_{L_k}(\mathbf{y})$ operator. Remarkably, this gives the correct support set for **PROBLEM 1.2**, as we prove next. We call this algorithm the greedy selector and simplex projector (GSSP). The overall complexity of GSSP is dominated by the sort operation in n-dimensions.

Unfortunately, the GSSP fails for **PROBLEM 1.3**. As a result, we propose Algorithm 2 for the $\Sigma_k \cap \Delta_\lambda$

Algorithm 2 GSHP

1: **Input:** $\boldsymbol{y}$, k, λ
2: $\ell = 1$, $\mathcal{S} = j, \quad j \in \arg\max_i [\lambda y_i]$ *(Initialize)*
3: Repeat: $\ell \leftarrow \ell + 1$, $\mathcal{S} \leftarrow \mathcal{S} \cup j$, where
$j \in \arg\max_{i \in \mathcal{N} \setminus \mathcal{S}} \left| y_i - \frac{\sum_{j \in \mathcal{S}} y_j - \lambda}{\ell - 1} \right|$ *(Grow)*
4: Until $\ell = k$, set $\widehat{\mathcal{S}} \leftarrow \mathcal{S}$ *(Terminate)*
5: $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}} = \mathcal{P}_{\Delta_\lambda}(\mathbf{y}_{|\widehat{\mathcal{S}}})$, $\widehat{\mathbf{x}}_{|\widehat{\mathcal{S}}^c} = 0$ *(Final projection)*

case which is non-obvious. The algorithm first selects the index of the largest element that has the same sign as λ. It then grows the index set one at a time by finding the farthest element from the current mean, as adjusted by lambda. Surprisingly, the algorithm finds the correct support set, as we prove next. We call this algorithm the greedy selector and hyperplane projector (GSHP), whose overall complexity is similar to GSSP.

Correctness of GSSP/GSHP

Remark 3. *When the symbol $\mathcal{S}$ is used as $\mathcal{S} = supp(\bar{\mathbf{x}})$ where $\bar{\mathbf{x}} = \mathcal{P}_{L_k}(\bar{\mathbf{y}})$ for any $\bar{\mathbf{y}}$, then if $|\mathcal{S}| < k$, we enlarge $\mathcal{S}$ until it has k elements by taking the first $k - |\mathcal{S}|$ elements that are not already in $\mathcal{S}$, and setting $\bar{\mathbf{x}} = 0$ on these elements. The lexicographic approach is used to break ties when there are multiple solutions.*

Theorem 1. *Algorithm 1 exactly solves* **PROBLEM 1.2**.

Proof. Intuitively, the k-most positive coordinates should be in the solution. To see this, suppose that $\mathbf{u}$ is the projection of $\mathbf{y}$. Let y_i be one of the k-most-positive coordinates of $\mathbf{y}$ and $u_i = 0$. Also, let $y_j < y_i,\ i \neq j$ such that $u_j > 0$. We can then construct a new vector $\mathbf{u}'$ where $u'_j = u_i = 0$ and $u'_i = u_j$. Therefore, $\mathbf{u}'$ satisfies the constraints, and it is closer to $\mathbf{y}$, i.e., $\|\mathbf{y} - \mathbf{u}\|_2^2 - \|\mathbf{y} - \mathbf{u}'\|_2^2 = 2u_j(y_i - y_j) > 0$. Hence, $\mathbf{u}$ cannot be the optimal projection point.

To be complete in the proof, we also need to show that the cardinality k solutions are as good as any other solution with cardinality less than k. Suppose there exists a solution $\mathbf{u}$ with support $|\mathcal{S}| < k$. Now add *any* elements to $\mathcal{S}$ to form $\widetilde{\mathcal{S}}$ with size k. Then consider $\mathbf{y}$ restricted to $\widetilde{\mathcal{S}}$, and let $\hat{\mathbf{u}}$ be its projection onto the simplex. Because this is a projection, $\|\hat{\mathbf{u}}_{|\widetilde{\mathcal{S}}} - \mathbf{y}_{|\widetilde{\mathcal{S}}}\| \leq \|\mathbf{u}_{|\widetilde{\mathcal{S}}} - \mathbf{y}_{|\widetilde{\mathcal{S}}}\|$, hence $\|\hat{\mathbf{u}} - \mathbf{y}\| \leq \|\mathbf{u} - \mathbf{y}\|$. □

The proof of the next statement is given in the appendix of this chapter.

Theorem 2. *Algorithm 2 exactly solves* **PROBLEM 1.3**.

1.5 Applications

In this section, we address important learning problems where both standard sparsity and norm/simplex constraints are present. The application list includes problems such as kernel density learning and Markowitz portfolio design. In all cases, we compute solutions by minimizing a convex and differentiable

loss function $f(\mathbf{x})$, subject to sparse and norm/simplex constraints. Our minimization approach is based on the projected gradient descent algorithm:

$$\mathbf{x}_{i+1} = \mathcal{P}_{\Sigma_k \cap \#}\left(\mathbf{x}_i - \frac{\mu}{2}\nabla f(\mathbf{x}_i)\right). \tag{1.22}$$

1.5.1 Sparse portfolio optimization

No-short position portfolios

Given a sample covariance matrix $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{n\times n}$ and expected mean $\widehat{\boldsymbol{\mu}} \in \mathbb{R}^n$, the return-adjusted Markowitz mean-variance (MV) framework [Mar52] selects a portfolio $\widehat{\mathbf{x}} \in \mathbb{R}^n$ such that

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\Delta_1^+}{\arg\min}\left\{\mathbf{x}^T\widehat{\mathbf{\Sigma}}\mathbf{x} - \tau\widehat{\boldsymbol{\mu}}^T\mathbf{x}\right\}, \tag{1.23}$$

where Δ_1^+ encodes the normalized capital constraint, and τ trades-off risk and return [DGNU09, BDDM$^+$09]. The solution $\widehat{\mathbf{x}} \in \Delta_1^+$ is the distribution of investments over the n available assets.

In practice, the preferences of the investor may lead to further constraints in the optimization problem. Additional fees for asset trading (transaction costs) and costs of monitoring and portfolio re-weighting naturally lead to cardinality constraints in the optimization procedure [BS09]. Here, we are interested in the MV optimization with the added twist that the solution satisfies $\widehat{\mathbf{x}} \in \Sigma_k$:

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\Delta_1^+\cap\Sigma_k}{\arg\min}\left\{\mathbf{x}^T\widehat{\mathbf{\Sigma}}\mathbf{x} - \tau\widehat{\boldsymbol{\mu}}^T\mathbf{x}\right\}, \quad \text{for a given level of sparsity } k. \tag{1.24}$$

This additional flavor leads to mixed integer quadratic programming formulation which is difficult to solve by standard optimization techniques [ADF11]. Numerous approaches have been proposed in the literature to solve this problem: most of the works focus on finding solutions using greedy techniques, simulated annealing, evolution methods, genetic algorithms, and branch-and-bound ideas [CMBS00, BS09, GFO06, CST11].

Efficient frontier with cardinality constraints: For this numerical example, we require the following "informal" definition:

> **Definition 4** (Pareto-optimal MV portfolios [Mar52])**.** *Let $\widehat{\mathbf{x}}$ be the solution of* (1.23) *for given $\widehat{\mathbf{\Sigma}}$, $\widehat{\boldsymbol{\mu}}$ and $\tau > 0$. Then, $\widehat{\mathbf{x}}$ is called* Pareto-optimal portfolio *if there is no other portfolio that achieves greater gain (for fixed risk) and lesser risk (for fixed gain) than $\widehat{\mathbf{x}}$, i.e., $\widehat{\mathbf{x}}$ is the dominating portfolio.*

The set of Pareto-optimal portfolios is called the Pareto efficient frontier for the MV selection problem.

To show the empirical performance of using sparsity constraints in portfolio optimization, we study the Pareto efficient frontier for a synthetic case. We generate random expected returns $\widehat{\boldsymbol{\mu}}$ and covariance quantities $\widehat{\mathbf{\Sigma}}$ for $n = 100$ assets. We compare the following approaches: (i) the quadratic optimization as described in (1.23) using `quadprog` in MATLAB, (ii) the cardinality-constrained projected gradient descent algorithm that solves (1.24) using GSSP for various sparsity levels $s \equiv k$ and, (iii) the ℓ_1-norm regularized solver, described in eqs. (2)-(4) of [BDDM$^+$09].

In Figure 1.4, we depict the resulting portfolios by solving the optimization problems (1.23), (1.24) and eqs. (2)-(4) of [BDDM$^+$09]. The red curve denotes the `quadprog` solution of (1.23) with simplex constraints; numbers within brackets also show the sparsity of the resulting portfolio $\widehat{\mathbf{x}}$. As expected, since the simplex constraint Δ_1^+ is a subset and special case of ℓ_1-norm constraint, we can still obtain sparse portfolios on the Pareto curve, but with higher risk and with no control on the sparsity level.

Blue square markers represent the ℓ_1-norm regularized solution, obtained by solving an instance of eqs. (2)-(4) of [BDDM$^+$09]. Interestingly enough, it is easy to see that the proposed portfolios "live" on the `quadprog` Pareto curve, i.e., for our problem setting, the ℓ_1-norm regularizer has a fixed value since we force $\widehat{\mathbf{x}} \in \Delta_1^+$. Thus, further sparsity cannot be achieved using this method.

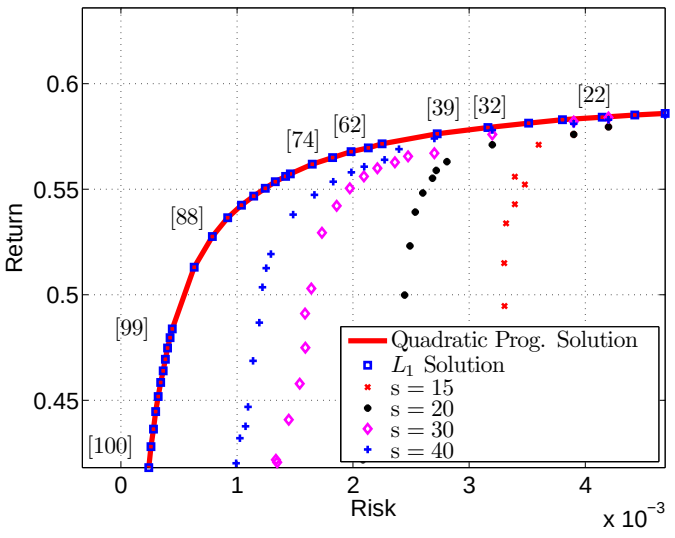


Figure 1.4: Pareto efficient frontier for $n = 100$ assets over a range of τ values. We perform 500 Monte-Carlo iterations and depict the median values. Red solid curve denotes the quadratic programming solution as obtained by (1.23) and blue squares represent a variation of ℓ_1-norm regularized solver in [BDDM$^+$09]. We present the solutions of our approach in (1.24) for $s \equiv k = 15, 20, 30, 40$.

We propose sparser portfolio strategies using a projected gradient descent solver for (1.24) where we use GSSP. The corresponding frontiers are depicted in Figure 1.4 for various $s \equiv k$. While our selections are not always "Pareto-efficient", we can guarantee the sparsity of the portfolio a priori, leading to predictable total transaction costs. Overall, without any cardinality constraints, the MV framework suggests dense portfolio solutions for low risk investments (additional selections lower the risk) while sparser solutions can be obtained for riskier investments. In practice, dense portfolios are difficult to administrate and have higher transactions costs.

Out-of-sample performance: We use a publicly available dataset compiled by Farma and French[9]. In this dataset, we monitor 49 diverse industry assets and consider only monthly recordings.

Procedure: We evaluate the out-of-sample performances of the estimated portfolios over various time periods. For instance, during each year from 1971 to 2011, we estimate expected monthly returns $\widehat{\boldsymbol{\mu}}$ of the stocks and their covariance values $\widehat{\boldsymbol{\Sigma}}$ using the available data from the preceding 5 years. Finally, we evaluate the estimated portfolio $\widehat{\mathbf{x}}$ by computing the monthly returns and risks for each upcoming year, keeping $\widehat{\mathbf{x}}$ fixed.

We compare the following approaches: (i) the constrained quadratic optimization as described in (1.23) using `quadprog` in MATLAB, (ii) the cardinality-constrained projected gradient descent algorithm that solves (1.24) using GSSP for $k = \{4, 10\}$ and, (iii) the naive $1/n$-strategy where we use the same weight over the portfolio, i.e., $x_i = 1/n$ for all i. In our experimental setting, the solver in [BDDM$^+$09] returns the same result as `quadprog` and thus is omitted.

[9] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Period	Quad. Prog.			GSSP, $k = 4$			GSSP, $k = 10$			$1/n$-strategy		
	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	μ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$
'71 - '11	11.82	0.423	**0.28**	**13.19**	0.489	0.27	11.78	0.418	**0.28**	10.1	0.49	0.21
'71 - '76	15.14	0.445	**0.34**	**15.58**	0.468	0.33	15.07	0.447	**0.34**	1.61	0.552	0.03
'76 - '81	7.61	0.473	0.16	**8.05**	0.543	0.15	7.83	0.467	**0.17**	2.49	0.608	0.04
'81 - '86	7.64	0.377	0.20	8.80	0.421	0.21	7.68	0.369	0.21	**11.48**	0.513	**0.22**
'86 - '91	21.25	0.449	0.43	**26.67**	0.538	0.49	25.13	0.475	**0.53**	20.93	0.477	0.44
'91 - '96	**11.15**	0.440	**0.25**	10.71	0.513	0.21	10.7	0.453	0.24	8.41	0.562	0.15
'96 - '01	19.42	0.329	**0.59**	**20.22**	0.402	0.50	17.85	0.321	0.56	14.22	0.274	0.52
'01 - '06	5.53	0.430	0.13	**6.71**	0.518	0.13	3.50	0.443	0.08	6.46	0.470	**0.14**
'06 - '11	6.42	0.329	0.20	8.83	0.329	0.27	6.36	0.315	0.2	**11.65**	0.356	**0.33**

Table 1.1: Portfolio performance evaluation with no-short positions for $\tau = 1$. In the table, $\hat{\mu}$ denotes the average monthly returns over all assets, $\hat{\sigma}$ is the standard deviation of these returns and, $\hat{\mu}/\hat{\sigma}$ is the Sharpe ratio. Numbers in magenta are the best for that row among all methods.

Results: We provide some return evaluations with $\tau = 1$ in Table 1.1.[10] Our approach with GSSP performs quite well, especially for smaller active portfolio sizes as constrained by k. We observe that as k decreases, the expected return $\hat{\mu}$ as well as the standard deviation $\hat{\sigma}$ of the returns increase. Surprisingly, the GSSP solutions exhibit competitive Sharpe ratios $\hat{\mu}/\hat{\sigma}$, which measures the risk adjusted return, as compared to the MV portfolio, and with much lower transactions costs. Overall, the quadratic programming approach has a median sparsity level of 14 and a mean sparsity level of 14.78. The $1/n$ baseline strategy has the worst returns and worst Sharpe ratios for most years.

Interestingly, the naive $1/n$-strategy does well in recession years like '81 to '86 and '06 to '11. In these years, presumably the model is less accurate, and hence the quadratic programming solution does much worse than the naive strategy. The sparsity-constrained solution does better than the quadratic programming solution, suggesting that sparsity helps against inaccurate models.

Short position portfolios

While the above proposed solutions construct portfolios from scratch, another scenario is to incrementally adjust an existing portfolio as the market changes. Due to costs per transaction, we can still naturally introduce cardinality constraints.

In mathematical terms, let $\widehat{\mathbf{x}} \in \mathbb{R}^n$ be the current portfolio selection. Given $\widehat{\mathbf{x}}$, we seek to adjust the current selection $\mathbf{x} = \widehat{\mathbf{x}} + \boldsymbol{\Delta}_{\mathbf{x}}$ such that $\|\boldsymbol{\Delta}_{\mathbf{x}}\|_0 \leq k$. According to (1.24), this leads to the following optimization problem:

$$\boldsymbol{\delta}_{\mathbf{x}}^{\star} \in \underset{\boldsymbol{\delta}_{\mathbf{x}} \in \Sigma_k \cap \Delta_\lambda}{\arg\min} \ (\widehat{\mathbf{x}} + \boldsymbol{\delta}_{\mathbf{x}})^T \widehat{\boldsymbol{\Sigma}} (\widehat{\mathbf{x}} + \boldsymbol{\delta}_{\mathbf{x}}) - \tau \widehat{\boldsymbol{\mu}}^T (\widehat{\mathbf{x}} + \boldsymbol{\delta}_{\mathbf{x}}),$$

where λ is the level of update such that $\sum_i (\boldsymbol{\delta}_{\mathbf{x}})_i = \lambda$, and k controls the transactions costs. E.g., during an update, $\lambda = 0$ would keep the portfolio value constant while $\lambda > 0$ would increase it.

To clearly highlight the impact of the non-convex projector, we create a synthetic portfolio update problem, where we know the solution. As in [BDDM+09], we can cast this problem as a regression problem and synthetically generate $\mathbb{1}^T \rho = \mathbf{A}\widehat{\mathbf{x}}$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an appropriately selected matrix such that the linear system is satisfied with $n = 1000$, $\widehat{\mathbf{x}} \in \Delta_\lambda$ (λ is chosen randomly) with $\|\widehat{\mathbf{x}}\|_0 = 100$ and, ρ is the

[10] Note that, as τ varies, the results qualitatively remain the same in comparison.

desired return of the selected portfolio to be satisfied, i.e., $\widehat{\mathbf{x}}^T\widehat{\boldsymbol{\mu}} = \rho$. Here, m represents the *time window* over which we observe and take snapshots of the assets.

Our goal here is to refine the sparse solution of a state-of-the-art convex solver [GB11] via (1.22) in order to accommodate the strict sparsity and budget constraints. Hence, we first consider the basis pursuit criterion [CDS98] and solve it using SPGL1 [VDBF08]:

$$\check{\mathbf{x}} \in \arg\min \|\mathbf{x}\|_1 \;\text{ subject to }\; \begin{bmatrix} \mathbf{A} \\ \mathbb{1}^T/\sqrt{n} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \boldsymbol{y} \\ \lambda/\sqrt{n} \end{bmatrix}. \tag{1.25}$$

The normalization by $1/\sqrt{n}$ in the last equality gives the constraint matrix a better condition number, since otherwise it is too ill-conditioned for a first-order solver.

Almost none of the solutions to (1.25) return a k-sparse solution. Hence, we initialize (1.22) with the SPGL1 solution to meet the constraints and using the GSHP algorithm.

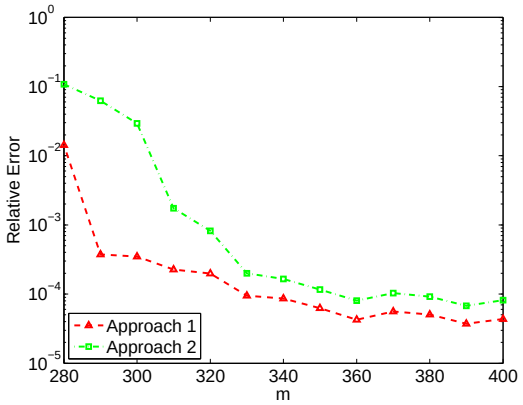


Figure 1.5: Relative error $\|\check{\mathbf{x}} - \widehat{\mathbf{x}}\|_2/\|\widehat{\mathbf{x}}\|_2$ comparison as a function of m: Approach 1 is the non-convex approach (1.22), and approach 2 is (1.25). Each point corresponds to the median value of 30 Monte-Carlo realizations.

Figure 1.5 shows the resulting relative errors $\|\check{\mathbf{x}} - \widehat{\mathbf{x}}\|_2/\|\widehat{\mathbf{x}}\|_2$. We see that not only does (1.22) return a k-sparse solution, but that this solution is also closer to $\widehat{\mathbf{x}}$, particularly when the sample size (i.e., the time window) is small. As the time window size increases, the knowledge that $\widehat{\mathbf{x}}$ is k-sparse makes up a smaller percentage of what we know about the signal, so the gap between (1.25) and (1.22) diminishes.

1.5.2 Sparse kernel density estimation

Here, we study the kernel density learning problem: Let $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(N)} \in \mathbb{R}^n$ be an N-size corpus of n-dimensional samples, drawn from an unknown probability density function (pdf) $f(\mathbf{w})$. Our purpose is to estimate $f(\mathbf{w})$ by forming an *kernel-based* estimator $\hat{f}(\mathbf{w}) := \sum_{i=1}^n x_i \kappa_\sigma(\mathbf{w}, \mathbf{w}^{(i)})$, where we choose $\kappa_\sigma(\mathbf{w}, \mathbf{z})$ to be a Gaussian kernel with parameter σ. In $\hat{f}(\mathbf{w})$, x_i denotes the *weight* on the corresponding kernel and thus, the contribution of $\kappa_\sigma(\mathbf{w}, \mathbf{w}^{(i)})$ in the final pdf estimate.

Given the sample corpus, classical non-parametric methods such as the Parzen window method [Par62] rely on the weighted interpolation of *all kernels* with mean point the corresponding sample from the set $\{\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(N)}\}$; see Figure 1.6. However, in many cases and especially when N is large, Parzen window method can be a quiet expensive routine, with hardly any information about the statistics of the underlying true $f(\cdot)$.

In this example, we follow a different path: Let us choose $\hat{f}(\mathbf{w})$ that minimizes the integrated squared error criterion: ISE $= \mathbb{E}\|\hat{f}(\mathbf{w}) - f(\mathbf{w})\|_2^2$. Within this context and using the same strategy of kernel superposition, we desire to identify only a subset of data points such that their weighted sum of kernels $\hat{f}(\cdot)$ result into a good approximation of $f(\cdot)$. As a result, we can introduce a density learning problem as estimating a weight vector $\widehat{\mathbf{x}} \in \Delta_1^+$. Following the work of [Kim95, BTWB10], the objective in this case

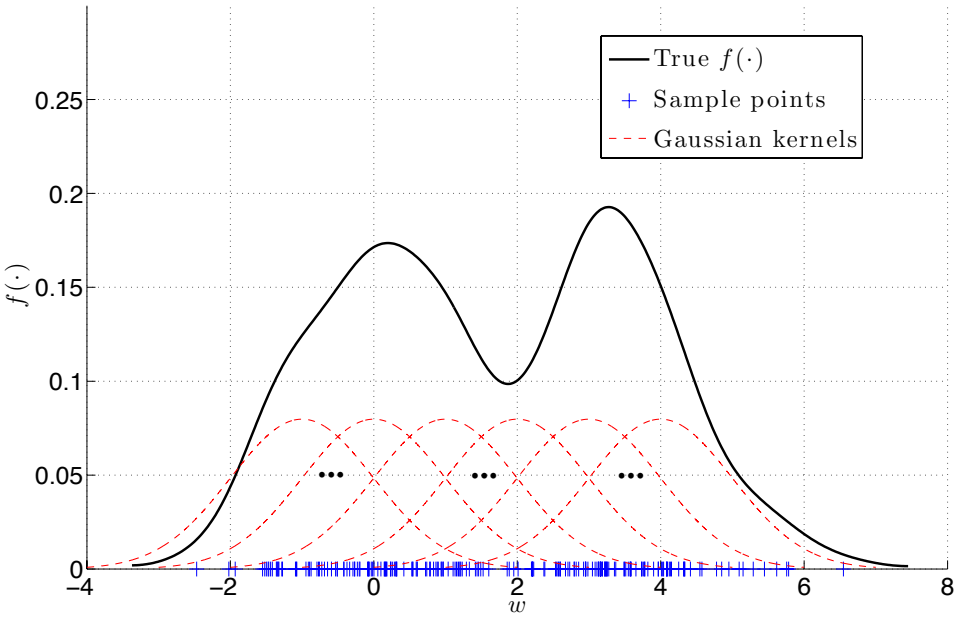


Figure 1.6: *Toy example using the Parzen method. Here, the black curve shows the true $f(\cdot)$, blue markers represent the 1-dimensional corpus $w^{(i)}$ for $i = 1, \cdots, 100$ and, red curves show some Gaussian kernels $\kappa_\sigma(w, w^{(i)})$.*

can be written as follows

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\Delta_1^+}{\arg\min} \left\{ \mathbf{x}^T \mathbf{\Sigma} \mathbf{x} - \mathbf{c}^T \mathbf{x} \right\}, \tag{1.26}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ with $\Sigma_{ij} = \kappa_{\sqrt{2}\sigma}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$, and

$$c_i = \frac{1}{N-1} \sum_{j\neq i} \kappa_\sigma(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}), \ \forall i, j.$$

To avoid overfitting or obtain interpretable results, one might control the level of solution sparsity [BTWB10]. In this context, we extend (1.26) to include cardinality constraints, i.e. $\widehat{\mathbf{x}} \in \Delta_1^+ \cap \Sigma_k$.

We consider the following Gaussian mixture: $f(w) = \frac{1}{5}\sum_{i=1}^{5} \kappa_{\sigma_i}(w_i, w)$, where $\sigma_i = (7/9)^i$ and $w_i = 14(\sigma_i - 1)$. A sample of 1000 points is drawn from $f(w)$. We compare the density estimation performance of: (i) the Parzen method [Par62], (ii) the quadratic programming formulation in (1.26), and (iii) our cardinality-constrained version of (1.26) using GSSP. While $f(w)$ is constructed by kernels with various widths, we assume a constant width during the kernel estimation. In practice, the width is not known *a priori* but can be found using cross-validation techniques [Rud82, Bow84]; for simplicity, we assume kernels with width $\sigma = 1$.

Figure 1.7(left) depicts the true pdf and the estimated densities using the Parzen method and the quadratic programming approach. Moreover, the figure also includes a scaled plot of $1/\sigma_i$, indicating the height of the individual Gaussian mixtures. By default, the Parzen window method estimation interpolates 1000 Gaussian kernels with centers around the sampled points to compute the estimate $\hat{f}(w)$; unfortunately, neither the quadratic programming approach (as Figure 1.7 (middle-top) illustrates) nor

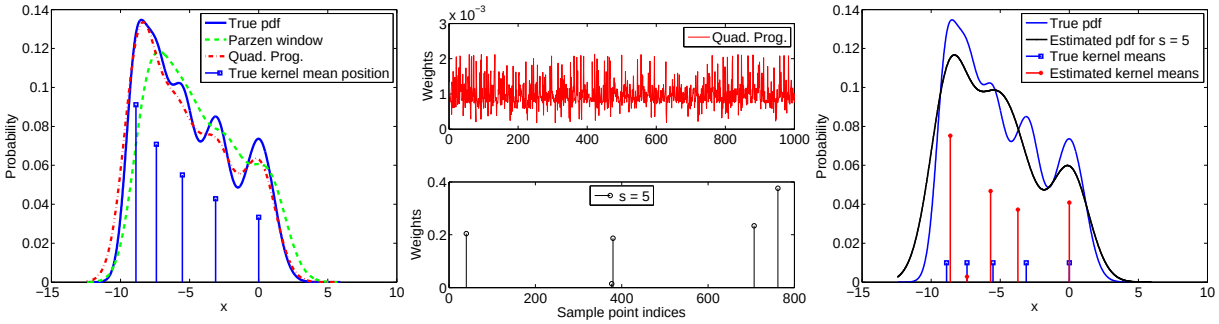


Figure 1.7: *Density estimation results using the Parzen method (left), the quadratic program* (1.26) *(left and middle-top), and our approach (middle-bottom and right).*

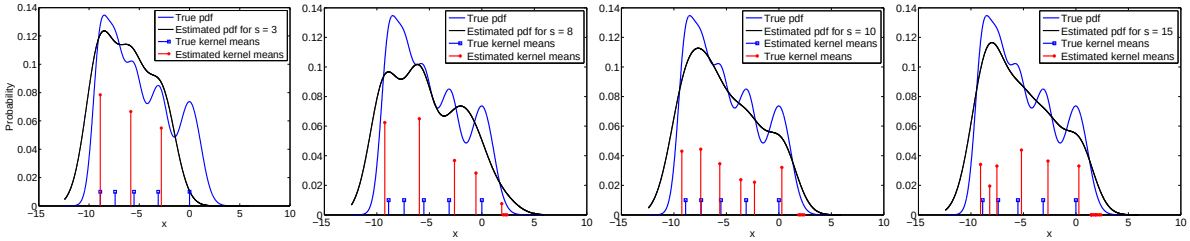


Figure 1.8: *Estimation results for different $k \equiv s$: Red spikes depict the estimated kernel means as well as the their relative contribution to the Gaussian mixture. As k increases, the additional nonzero coefficients in $\widehat{\mathbf{x}}$ tend to have small weights.*

the Parzen window estimator results are easily *interpretable*, even though both approaches provide a good approximation of the true pdf.

Using our cardinality-constrained approach, we can significantly enhance the interpretability. This is because, in the sparsity-constrained approach, we can control the number of estimated Gaussian components. Hence, if the model order is known *a priori*, the non-convex approach can be extremely useful.

To see this, we first show the coefficient profile of the sparsity based approach for $s \equiv k = 5$ in Figure 1.7 (middle-bottom). Figure 1.7 (right) shows the estimated pdf for 5-sparsity along with the positions of weight coefficients obtained by our sparsity enforcing approach. Note that most of the weights obtained concentrate around the true means, fully exploiting our prior information about the ingredients of $f(x)$—this happens with rather high frequency in the experiments we conducted. Figure 1.8 illustrates further estimated pdf's using our approach for various sparsity levels. Surprisingly, the resulting solutions are still approximately 5-sparse even if set the number of active compoentns > 5, as the over-estimated coefficients are extremely small, and hence the sparse estimator is reasonably robust to inaccurate estimates of the sparsity level.

1.6 Discussion

While non-convexity in learning algorithms is undesirable according to conventional wisdom, avoiding it might be "harmful" in many problems. In this chapter, we show how to efficiently obtain exact sparse projections onto norms and simplex type of constraints. We empirically demonstrate that our projectors provide substantial accuracy benefits in various problems such as kernel density estimation and portfolio optimization.

Using our projections in the kernel density estimation problem, one can approximate the true mean points of the true underlying probability mixture by sparsely selecting the dominant kernel data points. In the sparse portfolio optimization problem, our projections enable to update portfolios by enforcing non-convex constraints in the optimization procedure, while projections onto the standard simplex result in sparse portfolio designs and thus, lower transaction costs in practice.

The discussion in this chapter naturally leads to the following open problem:

Open question 1. *Let $\mathbf{y} \in \mathbb{R}^n$ be a given anchor vector, $k \leq n$ a sparsity level, $\mathbf{w} \in \mathbb{R}^n$ a weight vector and $\lambda \in \mathbb{R}$ a problem parameter. We are interested in finding the solution to the problem:*

$$\begin{aligned} \mathbf{x}^\star \in \arg\min_{\mathbf{x}} \quad & \|\mathbf{x}-\mathbf{y}\|_2^2, \\ \text{subject to} \quad & \|\mathbf{x}\|_0 \leq k, \\ & \mathbf{w}^T\mathbf{x} = \lambda. \end{aligned}$$

For the special case $\mathbf{w} \equiv \mathbb{1}_n$, we showed that this problem can be computed exactly in $\mathcal{O}(n \log n)$ computational cost. What can we say about the general case on $\mathbf{w}$?

Appendix

Proof of Theorem 2

To motivate the support selection of GSHP, we now identify a key relation that holds for any $\mathbf{b} \in \mathbb{R}^k$:

$$\sum_{i=1}^{k} b_i^2 - \frac{\left(\sum_{i=1}^{k} b_i - \lambda\right)^2}{k} = \lambda(2b_1 - \lambda) + \sum_{j=2}^{k} \frac{j-1}{j}\left(b_j - \frac{\sum_{i=1}^{j-1} b_i - \lambda}{j-1}\right)^2. \tag{1.27}$$

By its left-hand side, this relation is invariant under permutation of $\mathbf{b}$. Moreover, the summands in the sum over k are certainly non-negative for $k \geq 2$, so without loss of generality the solution sparsity of the original problem is $||\widehat{\mathbf{x}}||_0 = k$. For $k = 1$, F is maximized by picking an index $\mathcal{I}$ that maximizes λy_i, which is what the algorithm does.

For the sake of clarity (and space), we first describe the proof of the case $k \geq 2$ for $\lambda = 0$ and then explain how it generalizes for $\lambda \neq 0$. In the sequel, let us use the shortcut $\operatorname{avg}(S) = \frac{1}{|S|}\sum_{j\in S} y_j$.

Let $\mathcal{S}$ be an optimal solution index set and let $\mathcal{I}$ be the result computed by the algorithm. For a proof (of the case $k \geq 2, \lambda = 0$) by contradiction, assume that $\mathcal{I}$ and $\mathcal{S}$ differ. Let e be the first element of $I\backslash\mathcal{S}$ in the order of insertion into $\mathcal{I}$ by the algorithm. Let e' be the element of $\mathcal{S}\backslash\mathcal{I}_0$ that lies closest to e. Without loss of generality, we may assume that $y_e \neq y_{e'}$, otherwise we could have chosen $\mathcal{S}\backslash\{e'\} \cup \{e\}$ rather than $\mathcal{S}$ as solution in the first place. Let $\mathcal{I}_0 \subseteq \mathcal{I} \cap \mathcal{S}$ be the indices added to $\mathcal{I}$ by the algorithm before e. Assume that $\mathcal{I}_0$ is nonempty. We will later see how to ensure this.

Let $a := \operatorname{avg}(\mathcal{I}_0)$ and $a' := \operatorname{avg}(\mathcal{S}\backslash\{e'\})$. There are three ways in which y_e, $y_{e'}$ and a' can be ordered relative to each other:

1. e' lies between e and a', thus $|y_{e'} - a'| < |y_e - a'|$ since $y_e \neq y_{e'}$.
2. a' lies between e and e'. But then, since there are no elements of $\mathcal{S}$ between e and e', $\mathcal{S}\backslash\mathcal{I}_0$ moves the

average a' beyond a away from e towards e', so $|y_{e'} - a'| < |y_{e'} - a|$ and $|y_e - a| < |y_e - a'|$. But we know that $|y_{e'} - a| < |y_e - a|$ since $e = \operatorname{argmax}_{i \in \mathcal{I}_0} |y_i - a|$ by the choice of the greedy algorithm and $y_e \neq y_{e'}$. Thus $|y_{e'} - a'| < |y_e - a'|$.

3. $|y_e - a'| < |y_{e'} - a'|$, i.e., e lies between a' and e'. But this case is impossible: compared to a, a' averages over additional values that are closer to a than e, and e' is one of them. So a' must be on the same side as e' relative to e, not the opposite side.

So $|y_{e'} - a'| < |y_e - a'|$ is assured in all cases. Note in particular that if $|\mathcal{S}| \geq 1$, $|y_i - \operatorname{avg}(\mathcal{S})| \; \theta \; |y_j - \operatorname{avg}(\mathcal{S})|$, then

$$F(\mathcal{S} \cup \{i\}) = F(\mathcal{S}) + \frac{k-1}{k}\Big(y_i - \operatorname{avg}(\mathcal{S})\Big)^2 \underset{<}{\overset{=}{\text{or}}} F(\mathcal{S}) + \frac{k-1}{k}\Big(y_j - \operatorname{avg}(\mathcal{S})\Big)^2 = F(\mathcal{S} \cup \{j\}). \tag{1.28}$$

By inequality (1.28), $F(\mathcal{S}) < F((\mathcal{S} \backslash \{e'\}) \cup \{e\})$. But this means that $\mathcal{S}$ is not a solution: contradiction.

We have assumed that $\mathcal{I}_0$ is nonempty; this is ensured because any solution $\mathcal{S}$ must contain at least an index $i \in \arg\max_j y_j$. Otherwise, we could replace a maximal index w.r.t. $\mathbf{y}$ in $\mathcal{S}$ by this $\mathcal{I}$ and get, by (1.28), a larger F value. This would be a contradiction with our assumption that $\mathcal{S}$ is a solution. Note that this maximal index is also picked (first) by the algorithm. This completes the proof for the case $\lambda = 0$. Let us now consider the general case where λ is unrestricted.

We reduce the general problem to the case that $\lambda = 0$. Let us write $F_{\mathbf{y},\lambda}$ to make the parameters $\mathbf{y}$ and λ explicit when talking of F. Let $y'_{i^*} := y_{i^*} - \lambda$ for one i^* for which λy_{i^*} is maximal, and let $y'_i := y_i$ for all other $\mathcal{I}$. We use the fact that, by the definition of F,

$$F_{\mathbf{y},\lambda}(\mathcal{S}) = 2\lambda y'_{i^*} + \lambda^2 + F_{\mathbf{y}',0}(\mathcal{S})$$

when $\mathcal{S}$ contains such an element $i^* \in \operatorname{argmax}_j(\lambda y_j)$. Clearly, i^* is an extremal element w.r.t. $\mathbf{y}$ and y_{i^*} has maximum distance from $-\lambda$, so

$$i^* \in \arg\max_j \left| y_j - \frac{\sum_{i \neq j} y_i - \lambda}{j-1} \right|.$$

By (1.27), i^* must be in the optimal solution for $F_{\mathbf{y},\lambda}$. Also, $F_{\mathbf{y}',0}(\mathcal{S})$ and $2\lambda y'_{i^*} + \lambda^2 + F_{\mathbf{y}',0}(\mathcal{S})$ are maximized by the same index sets $\mathcal{S}$ when $i^* \in \mathcal{S}$ is required. Thus,

$$\arg\max_{\mathcal{S}} F_{\mathbf{y},\lambda}(\mathcal{S}) = \arg\max_{\mathcal{S}: j \in \mathcal{S}} F_{\mathbf{y}',0}(\mathcal{S}).$$

Now observe that our previous proof for the case $\lambda = 0$ also works if one adds a constraint that one or more indices be part of the solution: If the algorithm computes these elements as part of its result $\mathcal{I}$, they are in $\mathcal{I}_0 = I \cap \mathcal{S}$. But this is what the algorithm does on input $(\mathbf{y}, \lambda)$; it chooses i^* in its first step and then proceeds as if maximizing $F_{\mathbf{y}',0}$. Thus we have established the algorithm's correctness. □

2 Greedy methods for sparse linear regression

Introduction

Sampling, streaming, and storing any type of information usually creates a torrent of data that stretches to the limits the traditional analog-to-digital conversion, digital communication bandwidth and storage resources. Unfortunately, the traditional paradigm of capturing information *uncompressed* in order to compress it for subsequent processing becomes infeasible in many applications.

However, while the ambient data dimension is large in many problem cases, it turns out that the intrinsic information typically resides in a lower dimensional space. This observation has led to novel theoretical and algorithmic developments on data compression/decompression schemes, under different scientific communities. Resuming the discussion in the previous chapter, here we focus on linear "compression" strategies under the sparsity assumption; such development is known as *compressed sensing*.

The Compressed Sensing paradigm

Compressed Sensing (CS) is a data acquisition and recovery technique for finding sparse solutions to linear inverse problems from sub-Nyquist rate measurements. To describe the main idea, assume $\mathbf{x} \in \mathbb{R}^n$ is a dense vector of interest. According to the Shannon-Nyquist sampling theorem [Sha49], we can "perfectly" reconstruct $\mathbf{x}$ by uniformly taking Fourier transformed samples with frequency at least twice the highest frequency contained in $\mathbf{x}$. Unfortunately, as already mentioned, sampling in Shannon-Nyquist rate might not be possible without high-end specialized equipment [FH11]. Moreover, storing all this information creates storage bottlenecks [ME11]. To confront this problem, one has to either use high-end technology infrastructure, increasing the operational cost (higher cost per sample), or reduce the Quality of Service by subsampling.

The Fourier Transform (FT) is one of the many linear signal representations available: FT re-represents signals as a weighed sum of complex exponentials in various frequencies. By keeping only the nonzero coefficients, we can recover the signal by using the inverse FT procedure in an efficient manner.

Based on this premise, CS theory relies on the sparse transform coding technique [CDS98]: instead of processing $\mathbf{x}$ in its dense representation, the literature today offers signal basis transforms (other than FT)

that promote data compression and sparsity. Thus, using the appropriate basis matrix[1] $\mathbf{\Psi} \in \mathbb{R}^{n \times n}$, $\mathbf{x}$ can be described as a k-sparse ($k \ll n$) linear combination of atoms $\{\boldsymbol{\psi}_i\}_{i=1}^n$ that correspond to columns of $\mathbf{\Psi}$. This representation can be either exact, $\mathbf{x} = \mathbf{\Psi}\boldsymbol{\alpha}$, or approximate, $\mathbf{x} \approx \mathbf{\Psi}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^n$ denotes the set of coefficients with only k out of n entries being nonzero.Typical examples of sparse-inducing bases are wavelet transform for piecewise smooth signals [Huo99], Fourier transform for smooth and periodic signals [CDS98], curvelets for images with edges [CD00], etc. For clarity reasons, we assume for the rest of the chapter that $\mathbf{x}$ is k-sparse by nature, unless otherwise stated.

However, as long as the sensing mechanism remains the same, one cannot exploit such sparse signal representations; a more sophisticated sensing mechanism is required. CS proposes a new sampling scheme that compressively measures an n-dimensional k-sparse vector $\mathbf{x}$ through dimensionality reducing sensing matrices $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ where $c \cdot k \leq m \ll n$, for some $c > 0$. Here, $\mathbf{\Phi x}$ is called a "sketch" of $\mathbf{x}$. As we practically show in this chapter, the fascinating fact about $\mathbf{\Phi x}$ is that, under mild conditions on the number of samples m, the sparsity level k and the nature of $\mathbf{\Phi}$, it retains the essential properties of $\mathbf{x}$ for robust and efficient reconstruction from a limited set of samples.

Why use linear compression?

There are several reasons why linear sketching $\mathbf{\Phi x}$ is of great interest. From a computational perspective, linear transformations are the simplest mathematical models and are easier to maintain and update. That is, given $\mathbf{\Phi x}$, we can easily update a coordinate of $\mathbf{x}$ directly in the measurement domain. E.g., assume that we would like to update the i-th coordinate in $\mathbf{x}$ by the amount λ: in this case, we can simply update the sketch as $\mathbf{\Phi}\left(\mathbf{x} + \lambda \cdot \mathbf{e}_i\right) = \mathbf{\Phi x} + \lambda \mathbf{\Phi e}_i$, where only $\lambda \cdot \mathbf{\Phi e}_i$ is computed and added. Similarly, when we deal with superpositions of sparse signals, one can also easily obtain the sketch of their sum by measuring each signal individually and then combining the sketches, i.e., $\mathbf{\Phi}\left(\mathbf{x}_1 + \mathbf{x}_2\right) = \mathbf{\Phi x}_1 + \mathbf{\Phi x}_2$. Both properties are useful in several computational areas, notably computing over data streams [AMS96, Mut05, Ind07], network measurement [EV03], etc.

Furthermore, there are many practical settings where linear models appear naturally. In such cases, $\mathbf{x}$ models a physical event one wishes to sense and recover (e.g., neuronal data stream, image, etc.) through a natural physical measurement process. For example, a classic application of such linear compression schemes is where a digital camera captures images via their "projections" using pre-specified measurement vectors. The potential of such process is that an image can be almost perfectly reconstructed from the compressed samples via the physical process [DDT+08].

Problem statement

PROBLEM 2.1. *Let $\mathbf{x}^\star$ be a k-sparse n-dimensional vector of interest. We desire to reconstruct $\mathbf{x}^\star$ through a low-dimensional observation vector $\boldsymbol{y} \in \mathbb{R}^m$ $(m < n)$ where:*

$$\boldsymbol{y} = \mathbf{\Phi x}^\star + \boldsymbol{\varepsilon}; \tag{2.1}$$

here $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ is a fixed and known sensing matrix and $\boldsymbol{\varepsilon}$ is an additive noise term.

[1] Most of CS research papers assume the least common case where signals are sparse in an orthonormal basis. In practice, many signals of interest can only be expressed as a linear combination of a few atoms from an *overcomplete* dictionary $\mathbf{D} \in \mathbb{R}^{d \times n}$ $(d < n)$ where the columns are correlated [RSV08]. Here, we focus on the orthonormal basis case.

To recover $\mathbf{x}^\star$ given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$, *unconstrained* least-squares method is a classic approach to the solution of linear systems by minimizing the data error function $f(\mathbf{x}) := \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$; here, $\|\cdot\|_q$ denotes the ℓ_q-norm. Nevertheless, the reconstruction of $\mathbf{x}^\star$ from $\boldsymbol{y}$ is an ill-posed problem since $m < n$ and there is no hope in finding the *true vector* without ambiguity; there is an infinite number of possible solutions that satisfy the linear system of equations. Therefore, additional prior knowledge should be exploited by the optimization solver. Using the fact that $\mathbf{x}^\star$ is k-sparse, we concentrate on the following constrained minimization problem to recover $\mathbf{x}^\star$:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k, \tag{2.2}$$

where $\|\mathbf{x}\|_0$ is the "norm" that counts the non-zero entries in $\mathbf{x}$.

CS theory plays an important role in solving (2.2): assuming signal sparsity, the true solution $\mathbf{x}^\star$ can be found using $m \ll n$ measurements, as long as the geometry of sparse signals is preserved after projection on the column subspace defined by $\boldsymbol{\Phi}$.[2] To achieve this, CS also concentrates on developing polynomial-time algorithms for sparse signal recovery from a limited number of non-adaptive samples. Although the collection of works in this direction grows fast, the problem of constructing efficient methods both in execution time and signal recovery performance under various settings remains widely open.

Two main camps of compressed sensing algorithms

We briefly highlight two major classes of compressed sensing algorithms: (i) the convex optimization approach and (ii) the class of combinatorial-based greedy algorithms.

Convex optimization approach is one of the first algorithmic efforts for signal approximation in linear inverse problems. In [CDS98], Donoho et. al. demonstrate that, under basic incoherence properties of the sensing matrix $\boldsymbol{\Phi}$ and given $\mathbf{x}^\star$ is sufficiently sparse, we can substitute the combinatorial "norm" $\|\cdot\|_0$ by its sparsity-inducing convex envelope $\|\cdot\|_1$ with provable guarantees for unique signal recovery. Using this property, the authors proposed two reformulations of (2.2): the equality-constrained ℓ_1-norm minimization problem (Basis Pursuit (BP)):

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}, \tag{2.3}$$

for the noiseless case $\boldsymbol{\varepsilon} = \mathbf{0}$ and the ℓ_1-norm regularized least squares problem in the presence of noise (Basis Pursuit DeNoising (BPDN)):

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 + \tau\|\mathbf{x}\|_1, \tag{2.4}$$

where $\tau > 0$ balances the error norm and the sparsity of the solution. From a different perspective, Tibshirani [Tib96] proposes the Least Absolute Shrinkage Selection Operator algorithm, dubbed as LASSO:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_1 \leq \lambda, \tag{2.5}$$

where $\lambda > 0$ is a regularization parameter that governs the sparsity of the solution.

[2] This idea boils down to the so called restricted isometry property (RIP) that we explain next.

Once (2.2) is relaxed to a convex problem, decades of knowledge on convex analysis and optimization can be leveraged. To solve (2.3)-(2.5), interior point methods find a solution with fixed precision in polynomial time but their complexity might be prohibitive even for moderate-sized problems. More suitable for large-scale data analysis, fast first-order gradient algorithms constitute low-complexity alternatives to these methods; a non-exhaustive list of examples includes the optimal gradient methods proposed by Nesterov [Nes13, Nes83] and the iterative soft thresholding method (IST) [BT09b, WNF09].

In contrast to the conventional convex relaxation approaches, iterative greedy algorithms maintain the combinatorial nature of (2.2). Unfortunately, solving (2.2) with optimality is in general hard and exhaustive search over $\binom{n}{k}$ possible support configurations of the k-sparse solution is mandatory. Due to this computational intractability, the algorithms of this class greedily refine a k-sparse solution using only "local" information (i.e., gradient information) available at the current iteration. Representative examples of this class are hard thresholding methods [BD09a, BD10, NT09a, DM09, Fou11] and Matching Pursuit-based algorithms [MZ93, TG07].

The Restricted Isometry Property (RIP)

But how can we guarantee uniqueness in such ill-conditioned linear inverse problems? As it is obvious, signal sparsity does not guarantee successful recovery of the true vector for *any sensing matrix*; e.g., the all-zero sensing matrix $\mathbf{\Phi}$ does not "transfer" any knowledge of $\mathbf{x}^\star$ to $\boldsymbol{y}$. Many conditions on $\mathbf{\Phi}$ have been proposed in the literature to establish solution uniqueness and reconstruction stability such as the null space property [SXH08] and spark [DE03]. In this chapter, we focus on the so-called *restricted isometry property* (RIP).[3]

Definition 5 (RIP [CT06]). *A matrix $\mathbf{\Phi} \in \mathbb{R}^{m\times n}$ satisfies the k-RIP with isometry constant $\delta_k \in (0,1)$ if*

$$(1-\delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Phi}\mathbf{x}\|_2^2 \leq (1+\delta_k)\|\mathbf{x}\|_2^2, \quad \forall \mathbf{x} \in \Sigma_k. \tag{2.6}$$

In the above definition, $\|\mathbf{A}\|_{2\to 2} = \max_{\mathbf{x}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$, where $\mathbf{A}$ is a given matrix. If $\mathbf{\Phi}$ fulfils (2.6) for some $0 < \delta_k \ll 1$, then every set of k columns from $\mathbf{\Phi}$ is *nearly* orthogonal (i.e., well-conditioned). Roughly speaking, the Euclidean distance between k-sparse vectors is relatively preserved under linear projection using $\mathbf{\Phi}$.

While the majority of CS results assume (2.6) is satisfied with symmetry, we further consider the non-symmetric analog of the RIP:

$$\alpha_k\|\mathbf{x}\|_2^2 \leq \|\mathbf{\Phi}\mathbf{x}\|_2^2 \leq \beta_k\|\mathbf{x}\|_2^2, \forall \mathbf{x} \in \Sigma_k, \tag{2.7}$$

for positive constants α_k, β_k [BCT11].

In [CRT06], Candes et. al. prove the existence of random matrix ensembles that satisfy the RIP with overwhelming probability, provided that $m = \mathcal{O}\left(k\log(n/k)\right)$;[4] i.e., we can recover the unknown k-sparse signal $\mathbf{x}^\star$ using only $m \ll n$ linear non-adaptive samples using $\mathbf{\Phi}$. Representative examples include

[3] All the aforementioned conditions are unverifiable in polynomial time for deterministic matrices but hold with high probability for many random matrix ensembles, as explained next.

[4] The authors in [BCDH10] address the same problem using sparse vector approximation subject to a special structure, e.g. rooted connected tree sparsity model or non-overlapping group sparse model. They prove that under special structures, the number of measurements can be further reduced to $m = \mathcal{O}(k)$, independent of the ambient signal dimension, n. We elaborate more on these models in the next chapter.

random Gaussian matrices and random sparse binary (Bernoulli) matrices [Dir14].

A rule of thumb: In [Fou11], Foucart highlights the connection between the number of measurements m necessitated for exact signal recovery and the restricted isometry conditions of the form $\delta_{ck} \leq \delta$ where c is an integer and $\delta \in (0, 1)$. According to this remark, the following relation holds:

$$m \leq C \frac{ck}{\delta^2} \log(n/ck),$$

for some $C > 0$. Thus, fewer number of samples m implies smaller ratio ck/δ^2 and vice versa; a rule of thumb to be used in comparing the performance of sparse approximation algorithms.

Chapter roadmap

In this chapter, we study the compressed sensing problem, as described in **PROBLEM 2.1**, from a *non-convex* perspective. Our contributions are based on and inspired by the Iterative Hard Thresholding (IHT) framework [BD09a], characterized by the following two-step recursion:

$$\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu}{2} \nabla f(\mathbf{x}_i), \quad \mathbf{x}_{i+1} = \mathcal{P}_{\Sigma_k} \left(\bar{\mathbf{x}}_i \right). \tag{2.8}$$

Here, i is the iteration number, μ is the gradient descent step size, $\nabla f(\mathbf{x}) := -2\boldsymbol{\Phi}^T(\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x})$ denotes the gradient of the objective function $f(\mathbf{x})$, and $\mathcal{P}_{\Sigma_k}(\cdot)$ is the hard-thresholding combinatorial projection onto the subspace defined by Σ_k according to:

$$\mathcal{P}_{\Sigma_k}(\mathbf{y}) = \underset{\mathbf{x}:\mathbf{x}\in\Sigma_k}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2.$$

In Section 2.3, we analyze the behavior and performance of such hard thresholding methods from a global perspective. Three basic building blocks ("ingredients") are studied: $i)$ step size selection μ, $ii)$ memory exploitation, and $iii)$ gradient or least-squares updates over restricted support sets. We highlight the impact of these blocks on the convergence rate and signal reconstruction performance of iterative hard thresholding methods. We provide optimal and/or efficient strategies on how to set up these "ingredients" under different problem assumptions. As a by-product of this attempt, we propose the Algebraic Pursuit (ALPS) framework, an efficient solver for sparse linear regression problems.

In Section 2.4, we move a step further from simple sparsity and introduce our combinatorial selection and least absolute shrinkage (CLASH) operator. CLASH enhances the *model-based compressive sensing* (model-CS) framework [BCDH10] by additionally incorporating ℓ_1-norm constraints on the regression vector. This added twist significantly outperforms the model-CS approach, LASSO, or continuous structured sparsity approaches. Furthermore, CLASH characterizes the underlying tractability of approximation in combinatorial selection directly in the algorithm's estimation and convergence guarantees.

Inspired by CLASH, we propose an optimization paradigm, dubbed as NORMED PURSUITS, where both combinatorial and *generic* norm constraints are active in recovery: norm constraints restrict the candidate set of concise solutions to have a given norm, which leads to signal recovery improvements in practice.

This chapter is based on the joint work with Volkan Cevher and Gilles Puy [KC11, KC12a, KPC12].

2.1 Preliminaries

Notation: We use $(\mathbf{x})_j$ to denote the j-th element of $\mathbf{x}$, and let $\mathbf{x}_i$ represent the i-th iterate of the hard thresholding method. The index set of N dimensions is denoted as $\mathcal{N} = \{1, 2, \ldots, N\}$. Given $\mathcal{S} \subseteq \mathcal{N}$, we define the complement set $\mathcal{S}^c = \mathcal{N} \setminus \mathcal{S}$. Moreover, given a set $\mathcal{S} \subseteq \mathcal{N}$ and a vector $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{x}_\mathcal{S} \in \mathbb{R}^N$ denotes a vector with the following properties: $(\mathbf{x}_\mathcal{S})_\mathcal{S} = (\mathbf{x})_\mathcal{S}$ and $(\mathbf{x}_\mathcal{S})_{\mathcal{S}^c} = 0$. The notation $\nabla_\mathcal{S} f(\mathbf{x})$ is shorthand for $(\nabla f(\mathbf{x}))_\mathcal{S}$. $\mathbf{\Phi}_\mathcal{T}$ represents the restriction of the matrix $\mathbf{\Phi}$ to a column submatrix whose columns are listed in the set $\mathcal{T}$. The support set of $\mathbf{x}$ is defined as $\text{supp}(\mathbf{x}) = \{i : (\mathbf{x})_i \neq 0\}$. We use $|\mathcal{S}|$ to denote the cardinality of the set $\mathcal{S}$. The inner product between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ is denoted as $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y} = \sum_{i=1}^N (\mathbf{x})_i (\mathbf{y})_i$ where ^{T} is the transpose operation. $\|\cdot\|_2$ denotes the l_2-norm where $\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. $\mathbf{I}$ represents an identity matrix with dimensions apparent from the context.

Performance evaluation: To characterize the performance of iterative recovery processes such as the one in (2.8), both in terms of convergence rate and noise resilience, we use the following recursive expression:

$$\frac{\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \leq \rho \frac{\|\mathbf{x}_i - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} + \gamma \frac{1}{\text{SNR}}, \tag{2.9}$$

where $\text{SNR} \triangleq \frac{\|\mathbf{x}^\star\|_2}{\|\boldsymbol{\epsilon}\|_2}$ represents the signal-to-noise ratio metric; for our discussions we assume $\frac{1}{\text{SNR}}$ is bounded. In (2.9), γ denotes the approximation guarantee and provides insights of algorithm's reconstruction capabilities when noise is present; $|\rho| < 1$ expresses the contraction factor towards the true vector $\mathbf{x}^\star$.

Remark 4. *While the presented algorithms in this chapter guarantee convergence to $\mathbf{x}^\star$ in the noiseless case ($\rho < 1$), one can easily observe that, in the presence of heavy noise, the performance (with respect to finding $\mathbf{x}^\star$) degrades heavily. In this case, one can only guarantee that the computed solution "lives" within an error Euclidean ball around $\mathbf{x}^\star$.*

Remark 5. *In this chapter, we provide no convergence guarantees of the* iterates $\{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|_2\}_{i \geq 0}$*; such analysis can be conducted for our proposed schemes, following the results in [GK09a], but we retain it for future work.*

2.2 Related work

The compressed sensing problem has received intensive investigations from both theoretical and algorithmic aspects, since the publication of the seminal works [CT06, CDS98].[5] As already mentioned, there are both convex and non-convex algorithmic attempts.

In the convex case, ℓ_1-MAGIC [CRT06] is considered one of the first attempts to implement the convex formulations of the CS problem in practice. The proposed implementations are based on second-order methods, which makes them inappropriate for many large-scale applications. Figueiredo et al. [FNW07] reformulate the BPDN problem in (2.4) as a bounded-constrained quadratic program where gradient descent schemes are applied. [VDBF08] proposes SPGℓ_1, a gradient-projection method that approximately minimizes the least-squares objective subject to an explicit one-norm constraint. [HYZ08] proposes the Fixed-Point Continuation (FPC) algorithm for the BPDN problem (2.4) where fixed-point iterations are augmented with a *continuation approach* for better performance; it is a first-order gradient scheme followed by a soft-thresholding operation per iteration. Bioucas-Dias et al. [BDF07], Wright et al. [WNF09] and Teboulle et al. [BT09a] extend the ideas of basic first-order Iterative Soft-Thresholding (IST)

[5] The DSP online (http://dsp.rice.edu/cs) library on Compressed Sensing of Rice University counts more than 6000+ papers within the past decade.

method [DDDM04] using a two-step iterative scheme for acceleration; both works depend on the ideas of Nesterov's optimal methods [Nes83] and the Heavy Ball method [GR02]. The SALSA algorithm in [ABDF10] solves the BPDN formulation using augmented Lagrangian ideas where the acceleration in the convergence rate is due to the usage of quasi-Newton ideas in gradient descent direction definition.

In the non-convex case, the classes of hard thresholding methods [BD09a, BD10, NT09a, DM09, Fou11, BTW14] and Matching Pursuit-based algorithms [MZ93, TG07] are the most well-known and used schemes in practice.

Most of the aforementioned schemes are iterative in nature where the per iteration time-complexity is lower bounded by the computation of first-order (gradient) or second-order (Hessian) information.

2.3 Algebraic Pursuits (ALPS)

In this section, we present and analyze a class of sparse recovery algorithms, known as hard thresholding methods. The simplest form of such schemes satisfies (2.8). Thus, per iteration, resource requirements mainly depend on the total number of matrix-vector multiplication operations. To reduce the total computational complexity of IHT, we provide optimal strategies via basic "ingredients" for different configurations to achieve complexity vs. accuracy trade-offs. We describe several modular building blocks to derive IHT variants with faster convergence, reduced computational complexity and better reconstruction performance. Finally, we provide a general template that unifies the above ingredients into an algorithmic framework.

2.3.1 IHT: the ALPS backbone

For completeness, let us re-state the main iteration of hard thresholding gradient descent methods:

$$\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu_i}{2}\nabla f(\mathbf{x}_i), \quad \mathbf{x}_{i+1} = \mathcal{P}_{\Sigma_k}\left(\bar{\mathbf{x}}_i\right), \tag{2.10}$$

where we assume that $\mu_i > 0$ is an iteration dependent step size selection. According to **PROBLEM 2.1**, $\mathbf{x}^\star$ is the unknown signal to be estimated.

One can easily observe that (2.10) is the sparsity-constrained version of the projected gradient descent scheme in (6.5): per iteration, we compute a putative solution $\bar{\mathbf{x}}_i$ using gradient descent and then project $\bar{\mathbf{x}}_i$ onto the sparse scaffold Σ_k to satisfy the constraint $\|\mathbf{x}\|_0 \leq k$. While the performance analysis of such schemes is easier when we project onto a convex set, here we focus on the more demanding *non-convex* case. However, as we show next, in the context of CS, one can still obtain global convergence and strong recovery guarantees, as long as the sensing matrix $\mathbf{\Phi}$ and the signal of interest $\mathbf{x}^\star$ satisfy some conditions.

Before we provide these guarantees, let us gradually present our results. By the definition of the hard thresholding operation $\mathcal{P}_{\Sigma_k}(\cdot)$, at the i-th iteration, $\mathbf{x}_{i+1}$ is a better k-sparse approximation to $\bar{\mathbf{x}}_i$ than $\mathbf{x}^\star$.

This translates into:

$$\begin{aligned}
\|\mathbf{x}_{i+1} - \bar{\mathbf{x}}_i\|_2^2 \leq \|\mathbf{x}^\star - \bar{\mathbf{x}}_i\|_2^2 \Rightarrow \\
\| \left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right) + \left(\mathbf{x}^\star - \bar{\mathbf{x}}_i\right) \|_2^2 \leq \|\mathbf{x}^\star - \bar{\mathbf{x}}_i\|_2^2 \Rightarrow \\
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 + \|\mathbf{x}^\star - \bar{\mathbf{x}}_i\|_2^2 + 2\langle \mathbf{x}_{i+1} - \mathbf{x}^\star,\, \mathbf{x}^\star - \bar{\mathbf{x}}_i \rangle \leq \|\mathbf{x}^\star - \bar{\mathbf{x}}_i\|_2^2 \Rightarrow \\
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 \leq 2\langle \mathbf{x}_{i+1} - \mathbf{x}^\star,\, \bar{\mathbf{x}}_i - \mathbf{x}^\star \rangle
\end{aligned} \tag{2.11}$$

However, we observe that:

$$\begin{aligned}
\bar{\mathbf{x}}_i := \mathbf{x}_i - \frac{\mu_i}{2}\nabla f(\mathbf{x}_i) &= \mathbf{x}_i + \mu_i \mathbf{\Phi}^T \left(\boldsymbol{y} - \mathbf{\Phi}\mathbf{x}_i\right) && \text{(By definition of } \nabla f(\mathbf{x}_i)) \\
&= \mathbf{x}_i + \mu_i \mathbf{\Phi}^T \left(\mathbf{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon} - \mathbf{\Phi}\mathbf{x}_i\right) && \text{(by } \boldsymbol{y} = \mathbf{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon}) \\
&= \mathbf{x}_i + \mu_i \mathbf{\Phi}^T \mathbf{\Phi} \left(\mathbf{x}^\star - \mathbf{x}_i\right) + \mu_i \mathbf{\Phi}^T \boldsymbol{\varepsilon}
\end{aligned} \tag{2.12}$$

Combining (2.11) and (2.12), we obtain:

$$\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 \leq 2\langle \mathbf{x}_{i+1} - \mathbf{x}^\star,\, \mathbf{x}_i + \mu_i \mathbf{\Phi}^T \mathbf{\Phi} \left(\mathbf{x}^\star - \mathbf{x}_i\right) + \mu_i \mathbf{\Phi}^T \boldsymbol{\varepsilon} - \mathbf{x}^\star \rangle \tag{2.13}$$

"Massaging" the right hand side of (2.13) further, observe that the following two applications of the linear map are present in the inequality above:

$$\langle \mathbf{\Phi}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \mu_i \mathbf{\Phi}\left(\mathbf{x}_i - \mathbf{x}^\star\right) \rangle + \mu_i \langle \mathbf{\Phi}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \boldsymbol{\varepsilon} \rangle \tag{2.14}$$

Let $\mathcal{S}^\star := \text{supp}(\mathbf{x}^\star)$, $\mathcal{S}_{i+1} := \text{supp}(\mathbf{x}_{i+1})$ and $\mathcal{S}_i := \text{supp}(\mathbf{x}_i)$; in all cases, $|\mathcal{S}^\star| \leq k$, $|\mathcal{S}_{i+1}| \leq k$ and, $|\mathcal{S}_i| \leq k$. Thus, the above can be equivalently written as:

$$\langle \mathbf{\Phi}_{\mathcal{S}_{i+1} \cup \mathcal{S}^\star}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \mu_i \mathbf{\Phi}_{\mathcal{S}_i \cup \mathcal{S}^\star}\left(\mathbf{x}_i - \mathbf{x}^\star\right) \rangle + \mu_i \langle \mathbf{\Phi}_{\mathcal{S}_{i+1} \cup \mathcal{S}^\star}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \boldsymbol{\varepsilon} \rangle$$

where $\mathbf{\Phi}_{\mathcal{S}}$ is the submatrix in $\mathbf{\Phi}$, restricted in the columns indexed in $\mathcal{S}$. Let $\mathcal{A} := \mathcal{S}^\star \cup \mathcal{S}_{i+1} \cup \mathcal{S}_i$ which satisfies $|\mathcal{A}| \leq 3k$. Then, one can easily observe that in the inequality above, we can restrict the "active" columns in $\mathbf{\Phi}$ to those indexed by $\mathcal{A}$, such that (2.14) is equal to:

$$\langle \mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \mu_i \mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_i - \mathbf{x}^\star\right) \rangle + \mu_i \langle \mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \boldsymbol{\varepsilon} \rangle$$

Combining the above with (2.13) and applying the Cauchy-Schwarz inequality iteratively, we obtain:

$$\begin{aligned}
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 &\leq 2\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, \left(\mathbf{I} - \mu_i \mathbf{\Phi}_{\mathcal{A}}^T \mathbf{\Phi}_{\mathcal{A}}\right)\left(\mathbf{x}_i - \mathbf{x}^\star\right) \rangle + 2\mu_i \langle \mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right), \boldsymbol{\varepsilon} \rangle \\
&\leq 2\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 \cdot \| \left(\mathbf{I} - \mu_i \mathbf{\Phi}_{\mathcal{A}}^T \mathbf{\Phi}_{\mathcal{A}}\right)\left(\mathbf{x}_i - \mathbf{x}^\star\right) \|_2 + 2\mu_i \|\mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right) \|_2 \|\boldsymbol{\varepsilon}\|_2 \\
&\leq 2\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 \cdot \|\mathbf{I} - \mu_i \mathbf{\Phi}_{\mathcal{A}}^T \mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2} \|\mathbf{x}_i - \mathbf{x}^\star\|_2 + 2\mu_i \|\mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right) \|_2 \|\boldsymbol{\varepsilon}\|_2
\end{aligned} \tag{2.15}$$

Using the *non-symmetric* RIP definition in (2.7), we observe:

$$\|\mathbf{\Phi}_{\mathcal{A}}\left(\mathbf{x}_{i+1} - \mathbf{x}^\star\right) \|_2 \leq \sqrt{\beta_{2k}} \cdot \|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2$$

and, thus, (2.15) becomes:

$$\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 \leq 2\|\mathbf{I} - \mu_i \mathbf{\Phi}_{\mathcal{A}}^T \mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2} \|\mathbf{x}_i - \mathbf{x}^\star\|_2 + 2\mu_i \sqrt{\beta_{2k}} \|\boldsymbol{\varepsilon}\|_2.$$

By dividing with $\|\mathbf{x}^\star\|_2$, we obtain the recursion described in (2.9):

$$\frac{\|\mathbf{x}_{i+1}-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \leq \underbrace{2\|\mathbf{I}-\mu_i\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2}}_{:=\rho}\frac{\|\mathbf{x}_i-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} + \underbrace{2\mu_i\sqrt{\beta_{2k}}}_{:=\gamma}\frac{1}{\text{SNR}}, \tag{2.16}$$

where

$$\|\mathbf{I}-\mu_i\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2} \leq \max\left\{\mu_i\lambda_{\max}(\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}})-1,\ \ 1-\mu_i\lambda_{\min}(\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}})\right\}. \tag{2.17}$$

2.3.2 Step size selection strategies

Recent works on the performance of IHT algorithm provide strong convergence rate guarantees in terms of RIP constants; c.f., [BD09a] and [Fou11] to name a few. However, as a prerequisite to achieve these strong isometry constant bounds, the step size is set $\mu_i = 1, \forall i$, under the strong assumption that $\|\mathbf{\Phi}\|_2^2 < 1$. Unfortunately, such assumptions are not naturally met; the authors in [BD10] provide an intuitive example where IHT algorithm diverges under various scalings of the sensing matrix $\mathbf{\Phi}$. Therefore, more sophisticated step size selection procedures should be devised to tackle these issues during actual recovery. Existing approaches broadly fall into two categories: constant and adaptive step size selection. In the following subsections, we describe different step size selection strategies for various problem assumptions.

Constant step size selection

[GK09a] proposes a constant step size[6] $\mu_i = 1/\beta_{2k},\ \forall i$, as a by-product of a simple convergence analysis of the gradient descent method. Based on this idea, we propose *optimal* constant step size selections, both for symmetric and asymmetric RIP conditions, where μ_i is a function of the RIP constants.

As a first scenario, assume $\mathbf{\Phi}$ satisfies the non-symmetric RIP with *known* $\alpha_{ck}, \beta_{ck}, (c \in \{2,\ 3\})$ constants. In this case, the eigenvalues of $\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}}$, restricted in the set $\mathcal{A}$, satisfy:

$$\lambda_i(\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}}) \in [\alpha_{3k}, \beta_{3k}],\ \forall i.$$

Given this observation and in order to optimize the convergence rate ρ in (2.16), we can pick μ_i as the minimizer of the expression in (2.17):

$$\min_{\mu_i}\|\mathbf{I}-\mu_i\mathbf{\Phi}_{\mathcal{A}}^*\mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2} \leq \min_{\mu_i}\max\left\{\mu_i\beta_{3k}-1,\ \ 1-\mu_i\alpha_{3k}\right\}, \tag{2.18}$$

which leads to the following result, inspired by convex optimization constant step size strategies [Nes04].

Lemma 6 (Non-symmetric RIP constant step size strategy)**.** *Assume $\mathbf{\Phi}$ satisfies the non-symmetric RIP with* known *upper/lower bounds $\alpha_{cK}, \beta_{cK}, (c \in \{2,\ 3\})$. The step size $\mu_i^\star$ that implies optimal convergence rate in* (2.16) *is $\mu_i^\star = \frac{2}{\alpha_{3k}+\beta_{3k}}, \forall i = \{1, 2, \ldots, \}$, where, for $\beta_{3k} < 3\alpha_{3k}$, we have $\rho = \frac{2(\beta_{3k}-\alpha_{3k})}{\alpha_{3k}+\beta_{3k}} < 1$ and $\gamma = \frac{4\sqrt{\beta_{2k}}}{\alpha_{3k}+\beta_{3k}}$.*

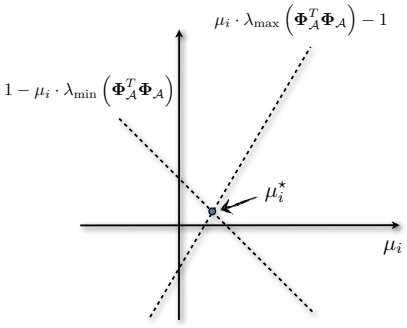
[6] In the case of *symmetric* RIP conditions, this step size becomes $\mu_i = 1/(1+\delta_{2k})$.

Proof. It is obvious that the step size μ_i that minimizes (2.18) lies at the intersection of the linear functions $\psi_1(\mu_i) := \mu_i\beta_{3k} - 1,\ \psi_2(\mu_i) := 1 - \mu_i\alpha_{3k}$. Hence, the minimum occurs when

$$\psi_1(\mu_i^\star) = \psi_2(\mu_i^\star) \Rightarrow \mu_i^\star = \frac{2}{\alpha_{3k} + \beta_{3k}}; \tag{2.19}$$

see Figure 2.1. By substituting $\mu_i^\star$ in (2.18), we obtain $2\|\mathbf{I} - \mu_i\mathbf{\Phi}_{\mathcal{A}}^*\mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2} \leq 2\max\left\{\frac{2\beta_{3k}}{\alpha_{3k}+\beta_{3k}} - 1,\ 1 - \frac{2\alpha_{3k}}{\alpha_{3k}+\beta_{3k}}\right\} = \frac{2(\beta_{3k}-\alpha_{3k})}{\alpha_{3k}+\beta_{3k}}$ which is less than 1 for $\beta_{3k} < 3\alpha_{3k}$. Using similar tools, we can easily prove that $\gamma = \frac{4\sqrt{\beta_{2k}}}{\alpha_{3k}+\beta_{3k}}$ in (2.16). □

Figure 2.1: Schematic representation of optimal constant step size selection $\mu_i^\star$.



In the special case where $\mathbf{\Phi}$ satisfies the symmetric RIP condition in (2.6) for some constant δ_{3k}, we can conclude to the same convergence rate achieved in [Fou11]; the proof is trivially implied by the above lemma.

Corollary 1 (RIP constant step size strategy). *Given $\mathbf{\Phi}$ satisfies the RIP for some δ_{3k}, the step size $\mu_i^\star$ that implies the fastest convergence rate in* (2.16) *amounts to $\mu_i^\star = 1, \forall i = \{1, 2, \dots, \}$, with $\rho = 2\delta_{3k}$ and $\gamma = 2\sqrt{1+\delta_{2k}}$. Moreover, the iterations are contractive iff $\delta_{3k} < 1/2 \Rightarrow |\rho| < 1$.*

Adaptive step size selection

Our discussion so far revolves around defining step size strategies when RIP constants are *known* a priori. However, since the computation of the exact RIP bounds is NP-hard, these strategies are impractical even for moderate-sized random matrices; an adaptive RIP-less scheme is mandatory.

There is limited work on the adaptive step size selection for hard thresholding methods. To the best of our knowledge, [BD10]-[Blu12] are the only studies that attempt this via line searching: given current estimate $\mathbf{x}_i$ and its support $\mathcal{S}_i$, the optimality of the proposed step size μ_i in these works is not guaranteed and a binary search over the range of μ_i is required to guarantee stability; c.f., Section III in [BD10].

Proposed step size selection strategy: According to (2.10), let $\mathbf{x}_i \in \Sigma_k$ be the k-sparse signal estimate with known support $\mathcal{S}_i := \text{supp}(\mathbf{x}_i)$ at the i-th iteration. It then holds that the non-zero elements $(\bar{\mathbf{x}}_i)_j,\ \forall j \in \bar{\mathcal{S}}_i := \text{supp}(\bar{\mathbf{x}}_i)$ satisfy:

$$(\bar{\mathbf{x}}_i)_j = \begin{cases} -\frac{\mu_i}{2}\left(\nabla f(\mathbf{x}_i)\right)_j & \text{if } (\mathbf{x}_i)_j = 0, \\ (\mathbf{x}_i)_j - \frac{\mu_i}{2}\left(\nabla f(\mathbf{x}_i)\right)_j & \text{otherwise.} \end{cases}$$

for any step size μ_i. Since $\text{supp}\,(\mathbf{x}_{i+1}) := |\mathcal{S}_{i+1}| \leq k$, we easily deduce the following key observation:

Remark 6. *Let $\mathcal{T}_i$ be a $2k$-sparse support set defined as:*

$$\mathcal{T}_i = \mathcal{S}_i \cup \text{supp}\left(\mathcal{P}_{\Sigma_k}\left(\nabla_{\mathcal{S}_i^c} f(\mathbf{x}_i)\right)\right).$$

Given $\mathcal{S}_{i+1}$ is unknown at the i-th iteration, $\mathcal{T}_i$ is the smallest index set that contains it such that

$$\mathcal{P}_{\Sigma_k}\left(\mathbf{x}_i - \frac{\mu_i}{2}\nabla f(\mathbf{x}_i)\right) = \mathcal{P}_{\Sigma_k}\left(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\right)$$

Algorithm 3 ALPS algorithm with zero memory (0-ALPS(0))

Input: $\boldsymbol{y}$, $\boldsymbol{\Phi}$, k, Tolerance η, MaxIterations
Initialize: $\mathbf{x}_0 \leftarrow 0$, $\mathcal{S}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  $\mathcal{T}_i \leftarrow \text{supp}(\mathcal{P}_{\Sigma_k}(\nabla_{\mathcal{S}_i^c} f(\mathbf{x}_i))) \cup \mathcal{S}_i$ *(Active support expansion)*
  $\mu_i = \frac{\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}{\|\boldsymbol{\Phi}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}$ *(Step size selection)*
  $\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)$ *(Gradient descent step)*
  $\mathbf{x}_{i+1} = \mathcal{P}_{\Sigma_k}(\bar{\mathbf{x}}_i)$ *(Hard-thresholding projection)*
  $\mathcal{S}_{i+1} \leftarrow \text{supp}(\mathbf{x}_{i+1})$
$i \leftarrow i + 1$.
until $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \leq \eta\|\mathbf{x}_i\|_2$ or MaxIterations.

necessarily holds.

Using Remark 1, IHT can be equivalently written as

$$\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i), \quad \mathbf{x}_{i+1} = \mathcal{P}_{\Sigma_k}(\bar{\mathbf{x}}_i).$$

where $\bar{\mathbf{x}}_i \in \Sigma_{2k}$ with $\text{supp}(\bar{\mathbf{x}}_i) \subseteq \mathcal{T}_i$. To compute the step size μ_i, we propose:

$$\mu_i = \arg\min_{\mu} \|\boldsymbol{y} - \boldsymbol{\Phi}\left(\mathbf{x}_i - \frac{\mu}{2}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\right)\|_2^2 = \frac{\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}{\|\boldsymbol{\Phi}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}, \tag{2.20}$$

i.e., μ_i is such that minimizes the objective function f, evaluated at the current putative point $\bar{\mathbf{x}}_i$. Note that $\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)$ is a $2k$-sparse vector and thus, by RIP assumptions on $\boldsymbol{\Phi}$, we have: $\alpha_{2k}\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2 \leq \|\boldsymbol{\Phi}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2 \leq \beta_{2k}\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2$ and $(1-\delta_{2k})\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2 \leq \|\boldsymbol{\Phi}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2 \leq (1+\delta_{2k})\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2$ for non-symmetric and symmetric RIP, respectively.

Using the definition of RIP, this implies

$$\frac{1}{1+\delta_{2k}} \leq \mu_i \leq \frac{1}{1-\delta_{2k}} \quad \text{and} \quad \frac{1}{\beta_{2k}} \leq \mu_i \leq \frac{1}{\alpha_{2k}}.$$

The discussion above leads to the Algebraic Pursuits (ALPS) algorithm, with zero memory; see Algorithm 3. The added twist to the regular IHT algorithm is the iteration-dependent, adaptive step step size selection μ_i, which is given in closed-form: μ_i "complies" with the sparse scaffold Σ_{ck} and thus, guarantees global convergence in estimates, as shown next.

Theorem 3 (Iteration Invariant)**.** *Assume $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ satisfies* (2.7) *with $\alpha_{ck}, \beta_{ck}, (c \in \{2,\ 3\})$ unknown. In the worst case scenario, 0-ALPS(0) with adaptive step size selection* (2.20) *satisfies the following recursion:*

$$\frac{\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \leq \rho\frac{\|\mathbf{x}_i - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} + \gamma\frac{1}{\text{SNR}}, \ \textit{where}\ \ \rho = 2\max\left\{\frac{\beta_{3k}}{\alpha_{2k}} - 1, 1 - \frac{\alpha_{3k}}{\beta_{2k}}\right\} \textit{ and } \gamma = \frac{2\sqrt{\beta_{2k}}}{\alpha_{2k}}.$$

Proof. 0-ALPS(0) satisfies the recursion described in (2.16). By construction, μ_i satisfies $\frac{1}{\beta_{2k}} \leq \mu_i \leq \frac{1}{\alpha_{2k}}$.

Thus, in (2.17) we have:

$$\|\mathbf{I}-\mu_i\mathbf{\Phi}_{\mathcal{A}}^T\mathbf{\Phi}_{\mathcal{A}}\|_{2\to 2}\leq \max\left\{\frac{\beta_{3k}}{\alpha_{2k}}-1,\ \ 1-\frac{\alpha_{3k}}{\beta_{2k}}\right\}.$$

Moreover, $\gamma:=2\mu_i\sqrt{\beta_{3k}}\leq 2\frac{\sqrt{\beta_{3k}}}{\alpha_{2k}}$, which completes the proof. □

Corollary 2. *Assuming* symmetric *RIP* (2.6) *with constants* $\delta_{ck},(c\in\{2,3\})$*, 0-ALPS(0) satisfies:*

$$\frac{\|\mathbf{x}_{i+1}-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2}\leq 2\frac{\delta_{3k}+\delta_{2k}}{1-\delta_{2k}}\frac{\|\mathbf{x}_i-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2}+\frac{2\sqrt{1+\delta_{2k}}}{1-\delta_{2k}}\frac{1}{\text{SNR}},\ \ \textit{where}\ \ \delta_{3k}<1/5\Rightarrow 2\frac{\delta_{3k}+\delta_{2k}}{1-\delta_{2k}}<1.$$

Proof. The proof easily follows by Theorem 3, where we substitute $\alpha_{ck}=(1-\delta_{ck})$ and $\beta_{ck}=(1+\delta_{ck})$. Furthermore, since $\delta_{c_1k}\leq\delta_{c_2k}$ for $c_1<c_2$ [NT09a], we observe:

$$2\frac{\delta_{3k}+\delta_{2k}}{1-\delta_{2k}}\leq\frac{4\delta_{3k}}{1-\delta_{3k}}.$$

By forcing it to be less that 1, we require $\delta_{3k}<1/5$. □

As a generic comment, we observe that adaptive μ_i scheme results in more restrictive "worst-case" isometry constants compared to [Fou12, BD09a], but faster convergence and better stability are empirically observed, as shown in the experiments at the end of the chapter.

2.3.3 Updates over restricted support sets in ALPS

Per iteration in 0-ALPS(0), the new estimate $\mathbf{x}_{i+1}$ can be further refined by applying a single or multiple gradient descent updates, restricted on $\mathcal{S}_{i+1}$ [Fou11], as described in Algorithm 4. In particular, let $\hat{\mathbf{x}}_{i+1}=\mathcal{P}_{\Sigma_k}(\bar{\mathbf{x}}_i)$ be the new k-sparse estimate with $\mathcal{S}_{i+1}:=\text{supp}(\hat{\mathbf{x}}_{i+1})$. Then, we can further update the estimate over the set $\mathcal{S}_{i+1}$ by performing the following motions:

$$\mathbf{x}_{i+1}=\hat{\mathbf{x}}_{i+1}-\frac{\bar{\mu}_i}{2}\nabla_{\mathcal{S}_{i+1}}f(\hat{\mathbf{x}}_{i+1}),\ \text{where}\ \ \bar{\mu}_i=\frac{\|\nabla_{\mathcal{S}_{i+1}}f(\hat{\mathbf{x}}_{i+1})\|_2^2}{\|\mathbf{\Phi}\nabla_{\mathcal{S}_{i+1}}f(\hat{\mathbf{x}}_{i+1})\|_2^2},$$

or solving the minimization problem over $\mathcal{S}_{i+1}$ [DM09]-[Fou11]:

$$\mathbf{x}_{i+1}=\underset{\mathbf{x}:\text{supp}(\mathbf{x})\subseteq\mathcal{S}_{i+1}}{\arg\min}\ \|\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}\|_2^2. \tag{2.21}$$

For k small compared to n, solving (2.21) can be efficient using off-the-self conjugate gradient implementations [HS52]. Based on the above, we propose a variation of 0-ALPS(0), called 0-ALPS(2), as described in Algorithm 4, with the following guarantees; the proof is provided in the Appendix.

Algorithm 4 ALPS algorithm with zero memory and updates on restricted sets (0-ALPS(2))

Input: $\boldsymbol{y}$, $\boldsymbol{\Phi}$, k, Tolerance η, MaxIterations
Initialize: $\mathbf{x}_0 \leftarrow 0$, $\mathcal{S}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  $\mathcal{T}_i \leftarrow \text{supp}(\mathcal{P}_{\Sigma_k}(\nabla_{\mathcal{S}_i^c} f(\mathbf{x}_i))) \cup \mathcal{S}_i$ *(Active support expansion)*
  $\mu_i = \frac{\|\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}{\|\boldsymbol{\Phi}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)\|_2^2}$ *(Step size selection)*
  $\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{T}_i} f(\mathbf{x}_i)$ *(Gradient descent step)*
  $\hat{\mathbf{x}}_{i+1} = \mathcal{P}_{\Sigma_k}(\bar{\mathbf{x}}_i)$ *(Hard-thresholding projection)*
  $\mathcal{S}_{i+1} \leftarrow \text{supp}(\hat{\mathbf{x}}_{i+1})$
  $\mathbf{x}_{i+1} = \arg\min_{x:\text{supp}(\mathbf{x})\in\mathcal{S}_{i+1}} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$ *(Least-squares update over $\mathcal{S}_{i+1}$)*
  $i \leftarrow i + 1$.
until $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \leq \eta\|\mathbf{x}_i\|_2$ or MaxIterations.

Theorem 4. *Assuming* symmetric *RIP* (2.6) *with constants* δ_{ck}, $(c \in \{2, 3\})$, *0-ALPS(2) satisfies:*

$$\frac{\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \leq \rho\frac{\|\mathbf{x}_i - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} + \gamma\frac{1}{\text{SNR}}, \quad \textit{where } \gamma := \frac{\sqrt{2(1+\delta_{2k})}}{(1-\delta_{2k})\sqrt{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}} + \frac{\frac{2\delta_{2k}\sqrt{1+\delta_k}}{(1-\delta_{2k})^2} + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}}{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}},$$

and $\rho := \frac{\sqrt{2}\delta_{3k}}{\sqrt{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}}$. *Moreover, the iterations are contractive, i.e.,* $\rho < 1$, *if and only if* $\delta_{3k} < 0.31$.

To compare Corollary 2 and Theorem 4, one can use the rule-of-thumb, described earlier in the chapter: according to this rule and assuming $\delta_{3k} < \delta$ where $\delta \in (0, 1)$, the number of samples m for exact recovery are proportional to $C\frac{3k}{\delta^2}\log(n/3k)$ for $C > 0$. Thus, the larger δ, the lesser number of measurements required.

Connection to CoSaMP [NT09a]/Subspace Pursuit [DM09] algorithms

Instead of performing a gradient descent step with adaptive μ_i selection per iteration, a more accurate but computationally intensive alternative is the objective minimization problem restricted on the extended support set $\mathcal{T}_i$, similar to (2.21). The ALPS variant for this case is given in Algorithm 5, with obvious similarities with the well-known CoSaMP/Subspace Pursuit algorithms [DM09, NT09a]. The only difference lies on the fact that we perform a $2k$-sparse support detection in the first step, while CoSaMP/Subspace Pursuit algorithms construct a bigger $3k$-sparse set $\mathcal{T}_i$. The convergence proof for this case can be found in [Fou12].

2.3.4 Memory in ALPS

Iterative algorithms can use memory to provide momentum in convergence. The success of the memory-based approaches depends on the iteration dependent momentum step size term that combines the previous estimates; see [BDF07] for a convex approach in image restoration. In particular, based on Nesterov's optimal gradient method [Nes04], we propose the following hard thresholding variant of the

Algorithm 5 ALPS algorithm with zero memory and least-squares on restricted sets (0-ALPS(4))

Input: $\boldsymbol{y}$, $\boldsymbol{\Phi}$, k, Tolerance η, MaxIterations
Initialize: $\mathbf{x}_0 \leftarrow 0$, $\mathcal{S}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  $\mathcal{T}_i \leftarrow \text{supp}(\mathcal{P}_{\Sigma_k}(\nabla_{\mathcal{S}_i^c} f(\mathbf{x}_i))) \cup \mathcal{S}_i$ *(Active support expansion)*
  $\hat{\mathbf{x}}_{i+1} = \arg\min_{x:\text{supp}(\mathbf{x})\in\mathcal{T}_i} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$ *(Least-squares update over $\mathcal{T}_i$)*
  $\mathcal{S}_{i+1} \leftarrow \text{supp}(\hat{\mathbf{x}}_{i+1})$ *(Hard-thresholding projection)*
  $\mathbf{x}_{i+1} = \arg\min_{x:\text{supp}(\mathbf{x})\in\mathcal{S}_{i+1}} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$ *(Least-squares update over $\mathcal{S}_{i+1}$)*
  $i \leftarrow i + 1$.
until $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \leq \eta\|\mathbf{x}_i\|_2$ or MaxIterations.

basic IHT algorithm:

$$\mathbf{x}_i = \mathcal{P}_{\Sigma_k}\left(\mathbf{u}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{u}_i)\right),\ \mathbf{u}_{i+1} = \mathbf{x}_i + \tau_i(\mathbf{x}_i - \mathbf{x}_{i-1}), \tag{2.22}$$

where $\mathcal{U}_i = \text{supp}(\mathbf{u}_i)$, $\mathcal{S}_i = \mathcal{U}_i \cup \text{supp}(\mathcal{P}_{\Sigma_k}(\nabla_{\mathcal{U}_i^c} f(\mathbf{u}_i)))$ with $|\mathcal{S}_i| \leq 3k$ and τ_i represents the momentum step size. For the determination of μ_i, either constant or adaptive step size selection strategy can be applied, depending on the problem assumptions.

Similarly to μ_i strategies, τ_i can be preset as constant or adaptively computed at each iteration. Constant momentum step size selection has no additional computational cost per iteration, but convergence rate acceleration is not guaranteed for a wide range of problem settings. On the other hand, empirical evidence has shown that adaptive τ_i selection strategies result to faster convergence with (almost) *equivalent complexity* to zero-memory methods.

For the case of strongly convex objective functions f, Nesterov [Nes04] proposed the following constant momentum step size selection scheme for (2.22):

$$\tau_i = \frac{\alpha_i(1-\alpha_i)}{\alpha_i^2 + \alpha_{i+1}},$$

where $\alpha_0 \in (0,1)$ and $\alpha_{i+1} \in (0,1)$ is computed as the root of $\alpha_{i+1}^2 = (1-\alpha_{i+1})\alpha_i^2 + q\alpha_{i+1}$, for $q := \frac{\lambda_{\min}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})}{\lambda_{\max}(\boldsymbol{\Phi}^*\boldsymbol{\Phi})} := \frac{\mu}{L}$. Here, μ is the strong convexity parameter and L is the Lipschitz constant of f. In this scheme, exact calculation of q parameter is computationally expensive for large-scale data problems and approximation schemes are leveraged to compensate this complexity bottleneck.

Based upon the same ideas as μ_i selection, we propose to select τ_i as the minimizer of the objective function[7]:

$$\tau_i = \arg\min_{\tau} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{u}_{i+1}\|_2^2 = \frac{\langle \boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}_i, \boldsymbol{\Phi}\mathbf{x}_i - \boldsymbol{\Phi}\mathbf{x}_{i-1}\rangle}{\|\boldsymbol{\Phi}\mathbf{x}_i - \boldsymbol{\Phi}\mathbf{x}_{i-1}\|_2^2}, \tag{2.23}$$

where $\boldsymbol{\Phi}\mathbf{x}_i, \boldsymbol{\Phi}\mathbf{x}_{i-1}$ are *previously computed and stored*. According to (2.23), τ_i requires only vector-vector inner product operations, a computationally cheaper operation than q calculation. Convergence rate performance of the above schemes is depicted in Figure 2.2.

Memory schemes can naturally be applied in the ALPS framework. Algorithm 6 describes the memory-based version of 0-ALPS(2), called 1-ALPS(2). Similarly, one can define 1-ALPS(0) and 1-ALPS(4).

[7] Similar ideas were simultaneously proposed in [Blu12].

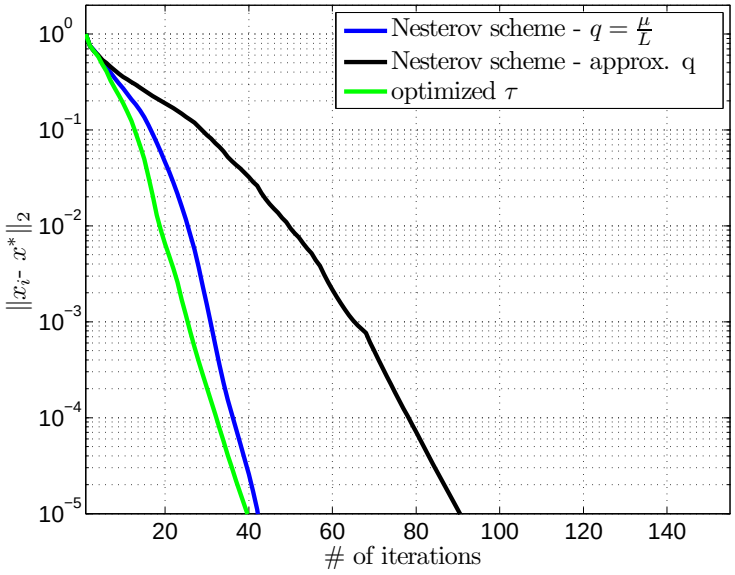


Figure 2.2: Hard thresholding scheme convergence rate example using memory. Here, $n = 2000, m = 600, k = 120$. Blue and black lines represent Nesterov's τ_i selection scheme with $q = \frac{\lambda_{\min}(\mathbf{\Phi}^*\mathbf{\Phi})}{\lambda_{\max}(\mathbf{\Phi}^*\mathbf{\Phi})}$ and approximate q, respectively; green line represents the proposed momentum step size selection.

Algorithm 6 ALPS algorithm with memory and updates on restricted sets (1-ALPS(2))

Input: $\boldsymbol{y}$, $\mathbf{\Phi}$, k, Tolerance η, MaxIterations
Initialize: $\mathbf{x}_0 \leftarrow 0, \mathbf{u}_0 \leftarrow 0, \mathcal{U}_0 \leftarrow \{\emptyset\}, i \leftarrow 0$
repeat
  $\mathcal{T}_i \leftarrow \text{supp}(\mathcal{P}_{\Sigma_k}(\nabla_{\mathcal{U}_i^c} f(\mathbf{u}_i))) \cup \mathcal{U}_i$ *(Active support expansion)*
  $\bar{\mathbf{u}}_i = \mathbf{u}_i - \frac{\mu_i}{2}\nabla_{\mathcal{T}_i} f(\mathbf{u}_i)$ where $\mu_i = \frac{\|\nabla_{\mathcal{U}_i} f(\mathbf{u}_i)\|_2^2}{\|\mathbf{\Phi}\nabla_{\mathcal{U}_i} f(\mathbf{u}_i)\|_2^2}$ *(Gradient descent step)*
  $\hat{\mathbf{x}}_{i+1} = \mathcal{P}_{\Sigma_k}(\bar{\mathbf{u}}_i)$ where $\mathcal{S}_{i+1} \leftarrow \text{supp}(\hat{\mathbf{x}}_{i+1})$ *(Hard-thresholding projection)*
  $\mathbf{x}_{i+1} = \arg\min_{x:\text{supp}(\mathbf{x})\in\mathcal{S}_{i+1}} \|\boldsymbol{y} - \mathbf{\Phi}\mathbf{x}\|_2^2$ *(Least-squares update over $\mathcal{S}_{i+1}$)*
  $\tau_{i+1} = \frac{\langle \boldsymbol{y} - \mathbf{\Phi}\mathbf{x}_{i+1}, \mathbf{\Phi}\mathbf{x}_{i+1} - \mathbf{\Phi}\mathbf{x}_i\rangle}{\|\mathbf{\Phi}\mathbf{x}_{i+1} - \mathbf{\Phi}\mathbf{x}_i\|_2^2}$ *(Memory step size selection)*
  $\mathbf{u}_{i+1} = \mathbf{x}_{i+1} + \tau_{i+1}(\mathbf{x}_{i+1} - \mathbf{x}_i)$ where $\mathcal{U}_{i+1} \leftarrow \text{supp}(\mathbf{u}_{i+1})$ *(Memory momentum update)*
  $i \leftarrow i + 1$.
until $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \leq \eta\|\mathbf{x}_i\|_2$ or MaxIterations.

2.4 Combinatorial selection and least absolute shrinkage via the CLASH algorithm

As already mentioned, classic hard thresholding algorithms for the compressed sensing setting [BD09a, Fou11, KC11] solve the following ℓ_0-constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{argmin}} \quad & \|\boldsymbol{y} - \mathbf{\Phi}\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \|\mathbf{x}\|_0 \leq k, \end{aligned} \tag{2.24}$$

where a putative k-sparse solution is iteratively refined using locally greedy decision rules. From a convex perspective, the *least absolute shrinkage and selection operator* (LASSO) [Tib96] can be recognized as

a relaxation of (2.24):

$$\begin{aligned} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{argmin}} \quad & \|\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}\|_2^2 \\ \text{subject to} \quad & \|\mathbf{x}\|_1 \leq \lambda, \end{aligned} \tag{2.25}$$

where $\lambda > 0$ is a parameter that governs the sparsity of the solution.

While both ℓ_0 and ℓ_1 sparse recovery formulations above have similar theoretical guarantees, it is *incorrect* to view the convex ℓ_1-norm as a convex relaxation of the Σ_k set, which extends to infinity. For instance, ℓ_1-norm in (2.25) not only acts as a geometrical proxy to k-sparse signals, but also provides a *scale* that the hard thresholding methods in (2.24) cannot exploit. Moreover, along with many efficient algorithms for its solution, it is now backed with a rather mature theory for the generalization of its solutions as well as its variable selection consistency [DSSSC08, BRT09, Wai09, ZY06].

However, while this geometric interplay of the ℓ_2-data error objective and the ℓ_1-norm constraint inherently promotes sparsity, in many cases it leads to arbitrariness in subset selection via shrinkage that best explains the responses. In fact, this *uninformed* selection process not only prevents interpretability of results in many problems, but also fails to exploit key prior information that could radically improve learning performance.

Along this line, the majority of modern convex approaches try to encapsulate any discrete constraints, known a priori, into their inherent continuous selection process. For instance, a prevalent approach is to tailor a *sparsity inducing* norm to the constraints on the support set (c.f., [JAB11, BJMO11, Bac10]). That is, we create a structured convex norm by *mixing* basic norms with weights over pre-defined groups [YLY11, JOV09, KX10] or using the Lovász extension of non-decreasing submodular set functions of the support [Bac10].

While such structure inducing, convex norm-based approaches on the LASSO are impressive, our contention is that, in order to truly make an impact in structured sparsity problems, one could also leverage explicitly combinatorial approaches to guide LASSO's subset selection process. To this end, we introduce our combinatorial selection and least absolute shrinkage (CLASH) operator and theoretically characterize its estimation guarantees. CLASH enhances the *model-based compressive sensing* (model-CS) framework [BCDH10] by additionally incorporating ℓ_1-norm constraints on the regression vector: CLASH uses a combination of shrinkage and hard thresholding operations to outperform the model-CS approach, LASSO, or continuous structured sparsity approaches in learning performance of sparse linear models. As a by-product, CLASH establishes a regression framework where the underlying tractability of approximation in combinatorial selection is directly reflected in the algorithm's estimation and convergence guarantees.

2.4.1 Intuition behind CLASH

Let us consider the following toy example in two dimensions: Assume $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}^\star$ be the set of measurements, according to **PROBLEM 2.1** for $\boldsymbol{\varepsilon} = \mathbf{0}$. Since $\ker(\boldsymbol{\Phi})$ is non-trivial, there are infinite solutions $\mathbf{x}$ that satisfy $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x}$; we depict this affine subspace of solutions with a green line in Figure 2.3.

In this toy example, we assume $\mathbf{x}^\star$ is 1-sparse, i.e., $\|\mathbf{x}^\star\|_0 = 1$; two points satisfy this condition, annotated as points (A) and (B) in Figure 2.3. By construction, greedy methods for (2.24) look for "high-energy" solutions via hard-thresholding operations and, thus, we might safely assume that (2.24) returns solution (A), as shown in Figure 2.3(Left). In stark contrast and for completeness, the Basis Pursuit solution in

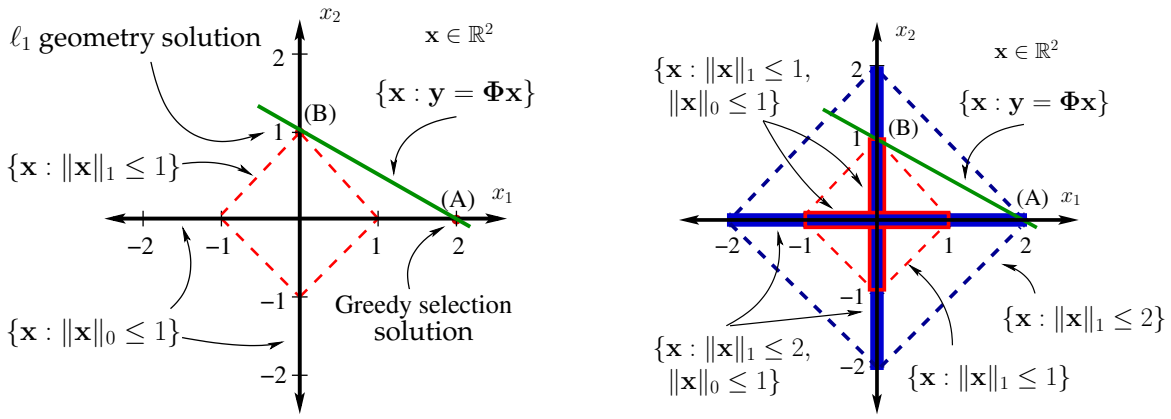


Figure 2.3: *Geometric interpretation of the selection process of convex- and combinatorial-based methods, as well as the proposed framework for a simple test case $\boldsymbol{y} = \mathbf{\Phi x}^\star$ where $\|x^*\|_0 = 1$. The admissible set of greedy solutions with the norm constraint lie on the segments inside the boxes.*

(2.3) (i.e., the solution with the *minimum* ℓ_1-norm) would choose (B) in Figure 2.3(Left), as it has smaller ℓ_1-norm than (A).

In the LASSO case (2.25), the performance of the solver depends on the selection of λ. If $\lambda = 1$, then the only 1-sparse solution satisfying the observations, i.e., minimizing the objective $\|\boldsymbol{y} - \mathbf{\Phi x}\|_2^2$ in (2.25), is vector (B). On the other hand and as shown in Figure 2.3(Right), as λ increases, LASSO might return as solution any point on the $\ker(\mathbf{\Phi})$, i.e., on the green line, that "lives" in the ℓ_1-norm ball with radius λ. For example, in order to include solution (A) in the feasible set of (2.25), one sets $\lambda = 2$; however, LASSO might return any of the solutions on the green line, as shown in Figure 2.3(Right).

Interestingly, we can exploit further combinatorial prior information on the support of the sparse signals. Using both ℓ_1-norm and ℓ_0 "norm" constraints and depending on the selection of λ, the admissible set of solutions lie on scaled segments on the canonical axes. For example, for constraints $\|\mathbf{x}\|_1 \leq 1$ and $\|\mathbf{x}\|_0 \leq 1$, we operate over the set inside the red boxes, along the canonical axes, in Figure 2.3(Right); similarly, for $\|\mathbf{x}\|_1 \leq 2$ and $\|\mathbf{x}\|_0 \leq 1$ the admissible set lies in the blue boxes. Thus, we can capture both 1-sparse solutions (A) and (B) with less ambiguity than the rest of the methods in our toy example of Figure 2.3 via, say, an ℓ_1-norm constraint (i.e., (B) with any constraint $\|\mathbf{x}\|_1 \leq \lambda$, $\lambda \in [1, 2)$)—in case of solution ambiguity, combinatorial selection rules dictate the sparse solution.

2.4.2 From simple sparsity to structured sparsity

As described above, by using combinatorial constraints in convex solvers, one can further restrict the cardinality of the solution set. In this subsection, we introduce the notion of structured sparsity, that goes beyond simple sparsity, and helps towards this direction. Section 3 elaborates more on this subject; the purpose of this subsection is to highlight the universality of the proposed method when more complicated structured models are assumed in practice.

Definition 6 (Combinatorial sparsity models (CSMs))**.** *We define a combinatorial sparsity model $\mathcal{M}_k = \{\mathcal{S}_q : \forall q,\ \mathcal{S}_q \subseteq \mathcal{N},\ |\mathcal{S}_q| \leq k\}$ with the sparsity parameter k as a collection of distinct index subsets $\mathcal{S}_q$.*

The workhorse in using CSMs in regression is the following non-convex projection problem, as defined by $\mathcal{M}_k$:

$$\mathcal{P}_{\mathcal{M}_k}(\mathbf{x}) = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \left\{ \|\mathbf{w}-\mathbf{x}\|_2^2 : \text{supp}(\mathbf{w}) \in \mathcal{M}_k \right\}, \tag{2.26}$$

where $\mathcal{P}_{\mathcal{M}_k}(\mathbf{x})$ is the projection operator. In particular, [BCDH10] shows that, as long as $\mathcal{P}_{\mathcal{M}_k}(\cdot)$ is exact for a CSM, their proposed sparse recovery algorithms inherit strong approximation guarantees for that CSM. Moreover, [BCDH10] shows that, under specific signal structures, the number of measurements for successful recovery can be as few as $m = \mathcal{O}(k)$, independent of the ambient signal dimension n. Unfortunately, only a special set of discrete constraints can be incorporated to their framework, such as tree structures and block sparsity, as long as the projection can be obtained exactly. Here, we extend this list based on matroids, totally unimodular systems, and knapsack constraints.

For many CSMs, the computation of the exact k-sparse projection is NP-hard. Moreover, in many cases, there is no analytical model but an algorithm that explains the structure in the sparse coefficients. To handle both cases, we identify tractable sparsity models as follows:

Definition 7 (Polynomial time modular ϵ-approximation property (PMAP$_\epsilon$))**.** *A CSM has the PMAP$_\epsilon$ with constant $\epsilon \in (0,1)$, if the following variance reduction problem admits an ϵ-approximation scheme with polynomial or pseudo-polynomial time complexity as a function of n for all $\mathbf{x} \in \mathbb{R}^n$:*

$$\max_{\mathcal{S}\in\mathcal{M}_k} F(\mathcal{S};\mathbf{x}), \text{ where } F(\mathcal{S};\mathbf{x}) := \|\mathbf{x}\|_2^2 - \|\mathbf{x}_{\mathcal{S}} - \mathbf{x}\|_2^2 = \sum_{i\in\mathcal{S}} |(\mathbf{x})_i|^2 . \tag{2.27}$$

Denoting the ϵ-approximate solution of (2.27) *as $\widehat{\mathcal{S}}_\epsilon$, this implies $F(\widehat{\mathcal{S}}_\epsilon;\mathbf{x}) \geq (1-\epsilon)\max_{\mathcal{S}\in\mathcal{M}_k} F(\mathcal{S};\mathbf{x})$.*

To connect the above, we state the following key observation.

Lemma 7 (Euclidean projections onto CSMs)**.** *The support of the Euclidean projection onto $\mathcal{M}_k$ in* (2.26) *can be obtained as a solution to the following discrete optimization problem:*

$$\text{supp}\left(\mathcal{P}_{\mathcal{M}_k}(\mathbf{x})\right) = \arg\max_{\mathcal{S}:\mathcal{S}\in\mathcal{M}_k} F(\mathcal{S};\mathbf{x}) \tag{2.28}$$

Moreover, let $\widehat{\mathcal{S}} \in \mathcal{M}_k$ be the minimizer of the discrete problem. Then, it holds that $\mathcal{P}_{\mathcal{M}_k}(\mathbf{x}) = \mathbf{x}_{\widehat{\mathcal{S}}}$, which corresponds to hard thresholding.

Proof. It is clear that the best Euclidean projection onto $\mathcal{M}_k$ in (2.26) is an index selection problem:

$$\text{supp}\Big(\arg\min_{\mathbf{w}:\text{supp}(\mathbf{w})\in\mathcal{M}_k} \|\mathbf{w}-\mathbf{x}\|_2^2\Big) = \arg\min_{\mathcal{S}:\mathcal{S}\in\mathcal{M}_k} \|\mathbf{x}_{\mathcal{S}}-\mathbf{x}\|_2^2 = \arg\max_{\mathcal{S}:\mathcal{S}\in\mathcal{M}_k} \|\mathbf{x}\|_2^2 - \|\mathbf{x}_{\mathcal{S}}-\mathbf{x}\|_2^2 = \arg\max_{\mathcal{S}:\mathcal{S}\in\mathcal{M}_k} \|\mathbf{x}_{\mathcal{S}}\|_2^2.$$

□

Lemma 8 (CSM projections via ILP's)**.** *The problem* (2.28) *is equivalent to the following integer linear program (ILP):*

$$\text{supp} \arg\min_{\substack{\mathbf{z}:[\mathbf{z}]_i\in\{0,1\},\\ \textit{supp}(\mathbf{z})\in\mathcal{M}_k}} \left\{ \mathbf{w}^T\mathbf{z} : (\mathbf{w})_i = -|(\mathbf{x})_i|^2 \right\}, \tag{2.29}$$

where $(\mathbf{z})_i$, $(i = 1,\ldots,n)$, are support indicator variables.

The proof of Lemma 8 is straightforward and is omitted.

Example CSMs with PMAP$_0$

Matroids: When $\mathcal{M}_k$ forms a matroid, the greedy basis algorithm can efficiently obtain the exact projection (2.26) by solving (2.28) [NW88]. By matroid, we mean that $\mathcal{M}_k$ is a finite collection of subsets of $\mathcal{N}$ that satisfies three conditions: (i) $\mathcal{M}_k$ includes the empty set, (ii) if $\mathcal{S}$ is in $\mathcal{M}_k$, then any subset of $\mathcal{S}$ is also in $\mathcal{M}_k$, and (iii) for $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{M}_k$ and $|\mathcal{S}_1| > |\mathcal{S}_2|$, there is an element $s \in \mathcal{S}_1 \setminus \mathcal{S}_2$ such that $\mathcal{S}_2 \cup \{s\}$ is in $\mathcal{M}_k$. As a simple example, the unstructured sparsity model (i.e., $\mathbf{x}$ is k-sparse) forms a *uniform matroid* as it is defined as the union of all subsets of $\mathcal{N}$ with cardinality k or less.

In turns out that this particular perspective provides a principled and tractable approach to encode an interesting class of *matroid-structured sparsity models*. The recipe is quite simple: we seek the intersection of a structure provider matroid (e.g., partition, cographic/graphic, disjoint path, or matching matroid) with the sparsity provider uniform matroid. While the intersection of two matroids is not a matroid in general, one can prove that the intersection of the uniform matroid with any other matroid satisfies the three conditions above.

Linear support constraints: Many interesting CSMs can be encoded using *linear support constraints* of the form:

$$\mathcal{M}_k = \bigcup_{\mathbf{z} \in \mathfrak{Z}} \operatorname{supp}(\mathbf{z}), \ \mathfrak{Z} := \{\mathbf{z} \in \{0,1\}^n : \mathbf{A}\mathbf{z} \leq \mathbf{b}\},$$

where $[\mathbf{A}, \mathbf{b}]$ is an integral matrix, and the first row of $\mathbf{A}$ is all 1's and $(\mathbf{b})_1 = k$. As a basic example, the neuronal spike model of [HDC09] is based on linear support constraints where each spike respects a minimum refractory distance to each other.

A key observation is that if each of the nonempty faces of $\mathfrak{Z}$ contains an integral point (i.e., forming an integral polyhedra), then convex optimization algorithms can exactly obtain the correct integer solutions in polynomial time. In general, checking the integrality of $\mathfrak{Z}$ is NP-Hard. However, if $\mathfrak{Z}$ is integral and non-empty for all integral $\mathbf{b}$, then a necessary condition is that $\mathbf{A}$ be a totally unimodular (TU) matrix [NW88]. A matrix is totally unimodular if the determinant of each square submatrix is equal to 0,1, or -1. Example TU matrices include interval, perfect, and network matrices [NW88].

How about PMAP$_\epsilon$?

For completeness, we only mention PMAP$_\epsilon$, which extends the breath of the model-CS approach. As a representative example for this type of approximation, we identify the multi-knapsack CSMs as a concrete examples. Moreover, for many of the PMAP$_0$ examples above, we can employ ϵ-approximate—randomized—algorithms to reduce computational complexity. However, using PMAP$_\epsilon$ in practice creates open questions for future work, as we declare at the end of this chapter.

2.4.3 The CLASH algorithm

We propose the *combinatorial selection and least absolute shrinkage* algorithm (CLASH) that obtains approximate solutions to the LASSO problem in (2.25) with the added twist that the solution must live within

Algorithm 7 CLASH Algorithm

1: **Input:** $\boldsymbol{y}$, $\boldsymbol{\Phi}$, λ, $\mathcal{P}_{\mathcal{M}_k}$, Tolerance η, MaxIterations
2: **Initialize:** $\mathbf{x}_0 \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
3: **repeat**
4: $\quad \mathcal{S}_i \leftarrow \text{supp}(\mathcal{P}_{\mathcal{M}_k}(\nabla_{\mathcal{X}_i^c} f(\mathbf{x}_i))) \cup \mathcal{X}_i$ *(Active set expansion)*
5: $\quad \mathbf{v}_i \leftarrow \arg\min_{\mathbf{v}:\|\mathbf{v}\|_1 \le \lambda,\, \text{supp}(\mathbf{v}) \in \mathcal{S}_i} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{v}\|_2^2$ *(Greedy descent with least absolute shrinkage)*
6: $\quad \boldsymbol{\gamma}_i \leftarrow \mathcal{P}_{\mathcal{M}_k}(\mathbf{v}_i)$ with $\Gamma_i \leftarrow \text{supp}(\boldsymbol{\gamma}_i)$ *(Combinatorial selection)*
7: $\quad \mathbf{x}_{i+1} \leftarrow \arg\min_{\mathbf{x}:\|\mathbf{x}\|_1 \le \lambda,\, \text{supp}(\mathbf{x}) \in \Gamma_i} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$ *(De-bias)*
8: $\quad \mathcal{X}_{i+1} \leftarrow \text{supp}(\mathbf{x}_{i+1})$
9: $i \leftarrow i + 1$.
10: **until** $\|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2 \le \eta \|\mathbf{x}_i\|_2$ or MaxIterations.

the CSM, as defined by $\mathcal{M}_k$:

$$\widehat{\mathbf{x}}_{\text{CLASH}} = \arg\min \left\{ f(\mathbf{x}) : \|\mathbf{x}\|_1 \le \lambda, \text{supp}(\mathbf{x}) \in \mathcal{M}_k \right\}. \tag{2.30}$$

Using the CSM constraint $\mathcal{M}_k$ in addition to the ℓ_1-norm constraint enhances learning in two important ways. First, the combinatorial constraints restrict the LASSO solution to exhibit interpretable and model-based supports. Second, it empirically requires much fewer number of samples to obtain the true solution than both the LASSO and the model-CS approaches.[8]

We provide a pseudo-code of an example implementation of CLASH in Algorithm 7. Steps 5 and 7 can be solved using a convex projected gradient descent approach. Of courase, one can think of alternative ways of implementing CLASH, such as single gradient updates in Step 5, or removing Step 7 altogether—for variations of Algorithm 7, one can use the ingredients described in the previous section of this chapter. While such changes may lead to different—possibly better—approximation guarantees for the solution of (2.30), we observe degradation in the empirical performance of the algorithm as compared to this implementation, whose guarantees are as follows:

Theorem 5 (Iteration invariant). *Let $\mathbf{x}^\star \in \mathbb{R}^n$ be the true vector that satisfies the constraints of* (2.30). *Then, the i-th iterate $\mathbf{x}_i$ of* CLASH *satisfies the following recursion:*

$$\frac{\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \le \rho \frac{\|\mathbf{x}_i - \mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} + \gamma \cdot \frac{1}{\text{SNR}}$$

where $\rho := \frac{\delta_{3k} + \delta_{2k}}{\sqrt{1-\delta_{2k}^2}} \sqrt{\frac{1+3\delta_{3k}^2}{1-\delta_{3k}^2}}$ and γ is a constant. The iterations contract when $\delta_{3k} < 0.3658$.

Theorem 5 shows that the isometry requirements of CLASH are competitive with the mainstream hard thresholding methods, such as CoSaMP [NT09a] and Subspace Pursuit [DM09], even though it incorporates the ℓ_1-norm constraints, which, as Section 2.6 illustrates, improves learning performance. In the absence of information on k and λ, we automate the parameter selection by using the Donoho-Tanner phase transition [DT05, BT14] to choose the maximum k allowed for a given (m, n)-pair, and then cross-validate to pick λ [War09].

Remark 7. *[Model mismatch and selection] Let us assume a generative model $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x} + \tilde{\varepsilon}$. Let $\mathbf{x}^\star$ be the best approximation of $\mathbf{x}$ in $\mathcal{M}_k$ within ℓ_1-ball of radius λ. Then, we can show that Theorem 5 still holds with*

[8] Unfortunately, the RIP sampling bound characterization does not change even if we have a norm-constraint—we believe that there is room for some new analysis.

SNR $= \frac{\|\mathbf{x}^\star\|_2}{\|\boldsymbol{\varepsilon}\|_2}$, *where* $\|\boldsymbol{\varepsilon}\|_2 \leq \|\tilde{\boldsymbol{\varepsilon}}\|_2 + \|\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^\star)\|_2$, *where the latter quantity (the impact of mismatch) can be analyzed using the restricted amplification property of* $\boldsymbol{\Phi}$ *[BCDH10]. For instance, when* $\mathcal{M}_k$ *is the uniform sparsity model, then* $\|\boldsymbol{\Phi}(\mathbf{x}-\mathbf{x}^\star)\|_2 \leq \sqrt{1+\delta_k}\left(\|\mathbf{x}-\mathbf{x}^\star\|_2 + \frac{\|\mathbf{x}-\mathbf{x}^\star\|_1}{\sqrt{k}}\right)$, *which should presumably be small if the model is selected correctly.*

Sketch of proof of Theorem 5

We sketch the proof of Theorem 5 a lá [NT09a] and [Fou12], assuming the general case of PMAP$_\epsilon$. For simplicity, we assume symmetric RIP as in (2.6). The details of the proof can be found in the Appendix.

Lemma 9 (Active set expansion). *The support set* $\mathcal{S}_i$, *where* $|\mathcal{S}_i| \leq 2k$, *identifies a subspace in* $\mathcal{M}_{2k}$ *such that:*

$$\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2 \leq (\delta_{3k}+\delta_{2k}+\sqrt{\epsilon}(1+\delta_{2k}))\|\mathbf{x}_i-\mathbf{x}^\star\|_2 + \left(\sqrt{2(1+\delta_{3k})}+\sqrt{\epsilon(1+\delta_{2k})}\right)\|\boldsymbol{\varepsilon}\|_2$$

Lemma 9 states that, at each iteration, this step identifies a $2k$ support set such that the unrecovered energy of $\mathbf{x}^\star$ is bounded. For $\epsilon = 0$, CLASH exactly identifies the support where the projected gradient onto $\mathcal{M}_k$ can make most impact on the loading vector in the support complement of its current solution, which are subsequently merged together.

Lemma 10 (Greedy descent with least absolute shrinkage). *Let* $\mathcal{S}_i$ *be a* $2k$*-sparse support set. Then, the least squares solution* $\mathbf{v}_i$ *Algorithm 1 satisfies*

$$\|\mathbf{v}_i-\mathbf{x}^\star\|_2 \leq \frac{1}{\sqrt{1-\delta_{3k}^2}}\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2 + \frac{\sqrt{1+\delta_{2k}}}{1-\delta_{3k}}\|\boldsymbol{\varepsilon}\|_2.$$

This step improves the objective function $f(\mathbf{x})$ as much as possible on the active set in order to arbitrate the active set. The solution simultaneously satisfies the ℓ_1-norm constraint.

Next, we project the solution onto $\mathcal{M}_k$, whose action is characterized by the following lemma. Here, we show the ϵ-approximate projection explicitly:

Lemma 11 (Combinatorial selection). *Let* $\mathbf{v}_i$ *be a* $2k$*-sparse proxy vector with indices in support set* $\mathcal{S}_i$, $\mathcal{M}_k$ *be a CSM and* $\boldsymbol{\gamma}_i$ *the projection of* $\mathbf{v}_i$ *under* $\mathcal{M}_k$. *Then:*

$$\|\boldsymbol{\gamma}_i-\mathbf{v}_i\|_2^2 \leq (1-\epsilon)\|(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2.$$

Finally, we require the following Corollary, which proof is similar to the proof of Lemma 10:

Corollary 3 (De-bias). *Let* Γ_i *be the support set of a proxy vector* γ_i *where* $|\Gamma_i| \leq k$. *Then, the least squares solution* $\mathbf{x}_{i+1}$ *in Step 4 satisfies*

$$\|\mathbf{x}_{i+1}-\mathbf{x}^\star\|_2 \leq \frac{1}{\sqrt{1-\delta_{2k}^2}}\|\boldsymbol{\gamma}_i-\mathbf{x}^\star\|_2 + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2.$$

This step de-biases the current result on the putative solution support. The next lemma connects Lemmas

10 and 11:

Lemma 12. *Let $\mathbf{v}_i$ be the least squares solution of the greedy descent step and $\boldsymbol{\gamma}_i$ be a proxy vector to $\mathbf{v}_i$ after applying Combinatorial selection step. Then, $\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2$ can be expressed in terms of the distance from $\mathbf{v}_i$ to $\mathbf{x}^\star$ as follows:*

$$\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2 \leq \sqrt{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon} \cdot \|\mathbf{v}_i - \mathbf{x}^\star\|_2 + D_1\|\boldsymbol{\varepsilon}\|_2 + D_2\|\mathbf{x}^\star\|_2 + D_3\sqrt{\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2},$$

where D_1, D_2, D_3 are constants depending on $\epsilon, \delta_{2k}, \delta_{3k}$.

Finally, the proof of Theorem 5 follows by concatenating Corollary 3 with Lemmas 9, 10, and 12.

2.5 Beyond ℓ_1-norm: NORMED-PURSUITS

Inspired by CLASH, we propose an alternative optimization paradigm, dubbed as NORMED PURSUITS, where both combinatorial (hard-thesholding) and generic norm constraints are active in sparse recovery: apart from the ℓ_1-norm constraint, a non-exhaustive list of norm candidates include ℓ_2, ℓ_∞ and total variation (TV) constraints.

NORMED PURSUITS is based on the CLASH algorithm, where ℓ_1-norm is replaced by other convex norms. NORMED PURSUITS has identical theoretical approximation guarantees with CLASH; see Theorem 5.

A key strength of NORMED PURSUITS is the ability to explicitly enforce sparsity using efficient combinatorial projections, while using the norm constraints to regularize the sparse coefficient values. The current sparse recovery literature offers a variety of convex optimization formulations that attempt to capture the strength of combinatorial models via *exclusively* norm information [ZH05, TSR+05, YL06]. In Table 2.1, we present the corresponding NORMED PURSUITS formulations of the corresponding convex formulations [ZH05, TSR+05, YL06].

Method	NORMED PURSUITS Formulation
Elastic net [ZH05]	$\hat{\mathbf{x}} = \arg\min \{f(\mathbf{x}) : \|\mathbf{x}\|_0 \leq k, \|\mathbf{x}\|_2 \leq \lambda\}$
Fused LASSO [TSR+05]	$\hat{\mathbf{x}} = \arg\min \{f(\mathbf{x}) : \|\mathbf{x}\|_0 \leq k, \|\mathbf{x}\|_{\text{TV}} \leq \lambda\}$
Group LASSO [YL06]	$\hat{\mathbf{x}} = \arg\min \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{M}_k, \|\mathbf{x}\|_{1,2} \leq \lambda\}$

Table 2.1: NORMED PURSUITS problem formulation of [ZH05, TSR+05, YL06]. Here, $\|\mathbf{x}\|_{1,2} \leq \lambda$ represents the group $\ell_{1,2}$-norm constraint over groups such that $\|\mathbf{x}\|_{1,2} := \sum_{\mathcal{G}} \|\mathbf{x}_{\mathcal{G}}\|_2$.

2.6 Experiments

2.6.1 Performance evaluation of ALPS

In this section we conduct a series of experiments to demonstrate the efficiency of the ALPS framework with respect to the convergence rate, the computational complexity (execution time) and the reconstruction performance.

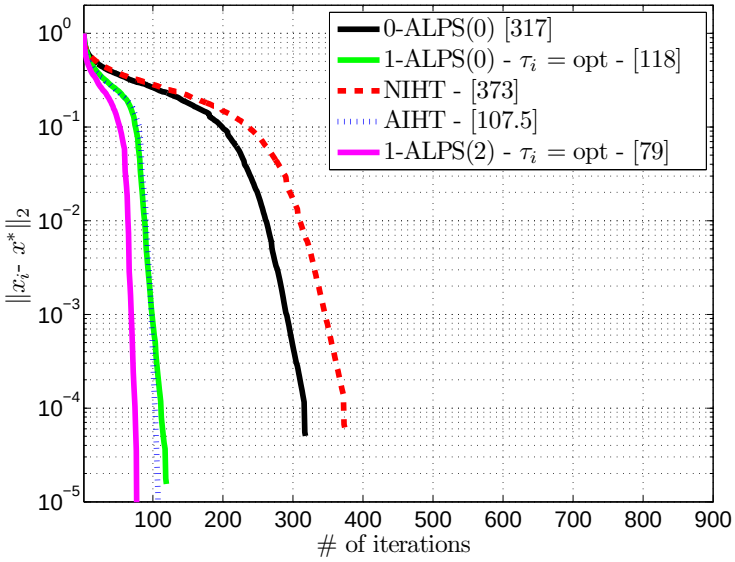


Figure 2.4: Median error per iteration; in brackets we show the [median # of iterations]. The list of algorithms includes: #-ALPS(0): adaptive μ_i with # memory, NIHT: Normalized IHT [BD10], AIHT: NIHT with Double Relaxation [Blu12], 1-ALPS(2): adaptive μ_i and additional gradient update.

Experiment 1: Computational complexity and convergence rate

We generate 100 random Monte-Carlo realizations according to **PROBLEM 2.1** where $n = 5000, m = 2000$ and $k = 700$. $\mathbf{\Phi}$ is a dense random matrix with independent entries, sampled from zero-mean Gaussian distribution with variance $1/m$. The sparse signal $\mathbf{x}^\star$ follows the simple sparsity model with k nonzero elements, acquired according to standard normal distribution with $\|\mathbf{x}^\star\|_2 = 1$. In Figure 2.4, we compare five state-of-the-art hard thresholding methods in terms of convergence rate.

We also provide in Table 2.2 the matrix-vector multiplication complexity per iteration (in Big-Oh notation), along with the total number of projections $\mathcal{P}_{\Sigma_k}(\cdot)$. While for the simple sparsity case $\mathcal{P}_{\Sigma_k}(\cdot)$ is "cheap" to compute, when structured sparsity $\mathcal{M}_k$ is known a priori, each hard thresholding operation over the model can be the most expensive task per iteration.

Table 2.2: (Left) Comparison of complexity per iteration. (Right) Number of hard thresholding operations per iteration.

Algorithm	Complexity
0-ALPS(0)	$\mathcal{O}(MN) + 3\mathcal{O}(MK)$
NIHT [BD10] [8]	$\mathcal{O}(MN) + 2\mathcal{O}(MK)$
AIHT [Blu12] [8]	$\mathcal{O}(MN) + 3\mathcal{O}(MK)$
1-ALPS(0)	$\mathcal{O}(MN) + 3\mathcal{O}(MK)$
1-ALPS(2)	$2\mathcal{O}(MN) + 5\mathcal{O}(MK)$

Algorithm	Number of projections
0-ALPS(0)	2
NIHT [BD10] [8]	2
AIHT [Blu12] [8]	3
1-ALPS(0)	2
1-ALPS(2)	2

[8] Best case scenario where no additional binary line search over μ_i is needed.

Experiment 2: Memory does not hurt

Figure 2.5 illustrates the phase transition diagrams of 0-ALPS(0) and 1-ALPS(0) algorithms over a range of problem dimensions. To visualize the phase transition diagram, we perform 100 Monte Carlo random realizations and take the expected number of successful recovery. A signal recovery with solution $\hat{\mathbf{x}}$ is considered successful provided that $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_2 < 10^{-6}$. The ambient dimension of the true signal is $n = 1000$. We observe that memory acceleration does not degrade the signal reconstruction performance compared to equivalent zero-memory schemes. As a side remark, we note that 1-ALPS(0) behaves better than AIHT [Blu12] and NIHT [BD10] algorithms in terms of phase transition performance.

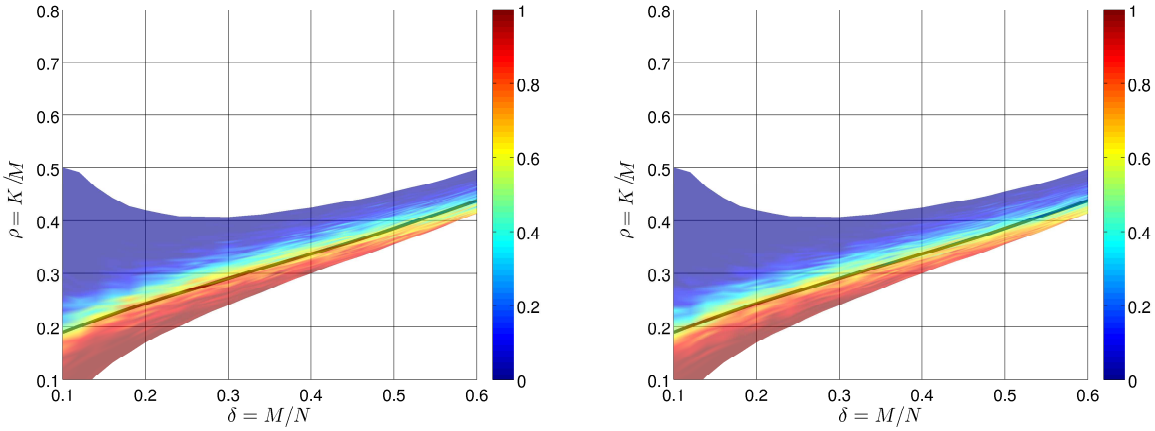


Figure 2.5: Empirical phase transition performance of 0-ALPS(0) (left column) and 1-ALPS(0) (right column) algorithms. A signal recovery with solution $\hat{\mathbf{x}}$ is considered successful provided that $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_2 < 10^{-6}$. Solid black line denotes the theoretical ℓ_1-norm minimization phase transition curve. Red to blue color denotes successful to unsuccessful signal recovery probability.

Experiment 3: Phase transition performance

In this experiment, we compare the signal recovery behaviour of 0-ALPS(4) algorithm using our adaptive step size selection and HTP algorithm [Fou11] with NIHT adaptive μ_i selection [BD10]. Here, we assume $n = 1000$. The empirical phase transition results are depicted in Figure 2.6.

2.6.2 Sparsity and ℓ_1-norm

In the following experiments, we compare algorithms from the following list: (i) the LASSO algorithm [Tib96], (ii) the Basis Pursuit DeNoising (BPDN) [CDS98], (iii) the sparse-CLASH algorithm, where $\mathcal{M}_k \equiv \Sigma_k$, (iv) the model-CLASH algorithm, which explicitly carries a model $\mathcal{M}_k$, defined later in the text, and (v) Subspace Pursuit (SP) algorithm [DM09], as a state-of-the-art hard thresholding method (also integrated with $\mathcal{M}_k$ if present). We emphasize here that when $\lambda \to \infty$ in (2.26), CLASH must converge to the SP solution.

Implementation details: The LASSO algorithm finds a solution to the problem defined in (2.25), where we use a Nesterov accelerated projected gradient algorithm. The BPDN algorithm in turn solves the following optimization problem:

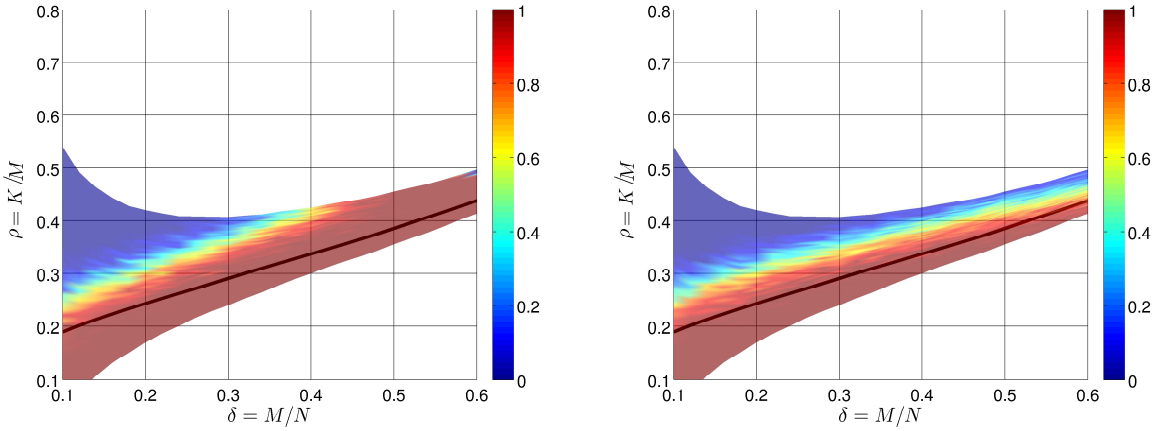


Figure 2.6: Empirical phase transition performance of 0-ALPS(4) with the proposed step size selection (left column) and HTP with NIHT step size selection (right column). A signal recovery with solution $\hat{\mathbf{x}}$ is considered successful provided that $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_2 < 10^{-6}$. Solid black line denotes the theoretical ℓ_1-norm minimization phase transition curve. Red to blue color denotes successful to unsuccessful signal recovery probability.

$$\widehat{\mathbf{x}}_{\text{BPDN}} = \arg\min \left\{ \|\mathbf{x}\|_1 : \quad \|\boldsymbol{\Phi}\mathbf{x} - \boldsymbol{y}\|_2 \leq \sigma \right\}, \tag{2.31}$$

where σ represents prior knowledge on the energy of the additive noise term. To solve (2.31), we use the spectral projected gradient method SPGL1 algorithm [VDBF08].

Settings: In the experiments below, the nonzero coefficients of $\mathbf{x}^\star$ are generated iid according to the standard normal distribution with $\|\mathbf{x}^\star\|_2 = 1$. The BPDN algorithm is provided with the true σ value. While CLASH is given the true value of k for the experiments below, we empirically observe that our phase transition heuristics is quite good and the mismatch is graceful as indicated in Remark 1. All the algorithms use a high precision stopping tolerance $\eta = 10^{-5}$.

Experiment 1: Improving simple sparse recovery: In this experiment, we generate random realizations of the model $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon}$ for $n = 800$. Here, $\boldsymbol{\Phi}$ is a dense random matrix whose entries are iid Gaussian with zero mean and variance $1/m$. We consider two distinct (and extreme) generative model settings: (i) with additive Gaussian white noise with $\|\boldsymbol{\varepsilon}\|_2 = 0.05$, $m = 240$ and $k = 89$, and (ii) the noiseless model ($\|\boldsymbol{\varepsilon}\|_2 = 0$), $m = 250$ and sparsity parameter $k = 93$. For this experiment, we perform 500 Monte Carlo model realizations.

We sweep λ and illustrate the recovery performance of CLASH . The top row of Figure 2.7 illustrates that the combination of hard thresholding with norm constraints can *improve* the signal recovery performance over convex-only and hard thresholding-only methods—both in noisy and noiseless problem settings. For $\|\boldsymbol{\varepsilon}\| = 0$, CLASH perfectly recovers the signal when λ is close to the true value. When $\lambda \ll \|\mathbf{x}^\star\|_1$, the performance degrades due to the large norm mismatch.

Experiment 2: Improving structured sparse recovery: We consider two signal CSMs: in the first model, we assume k-sparse signals that admit clustered sparsity with coefficients in C-contiguous blocks on an undirected, acyclic chain graph [CIHB09]. Without loss of generality, we use $C = 5$ (Figure 2.7, bottom row, left). The second model corresponds to a TU system [HDC09] where we partition the k-sparse signals into uniform blocks and force sparsity constraints on individual blocks; in this case, we solve the

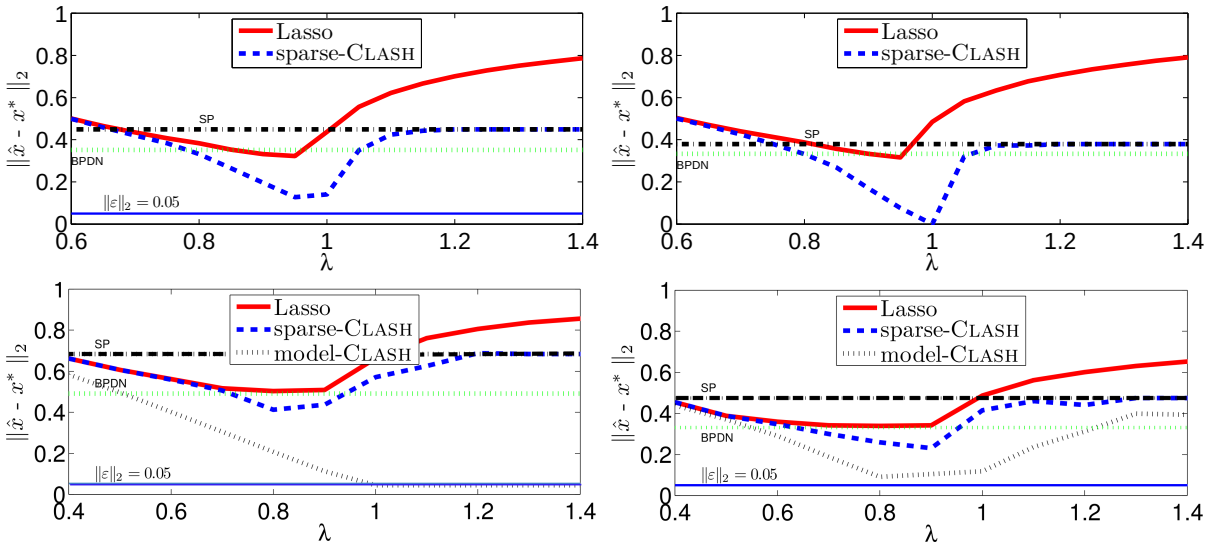


Figure 2.7: Median values of signal error $\|\hat{\mathbf{x}} - \mathbf{x}^\star\|_2$. Top row: simple sparsity model under noisy $\|\boldsymbol{\varepsilon}\|_2 = 0.05$ (left column) and noiseless $\|\boldsymbol{\varepsilon}\|_2 = 0$ (right column) settings. Bottom row: the (k, C)-clustered sparsity model (left column) and the TU model (right column).

set optimization problem optimally via linear programming relaxation (Figure 2.7, bottom row, right). Here, the noise energy level satisfies $\|\boldsymbol{\varepsilon}\|_2 = 0.05$, and $n = 500$, $m = 125$, and $k = 50$. In both cases, we conduct 100 Monte Carlo iterations and perform sparse estimation for a range of λ values.

In Figure 2.7, bottom row, left panel, we observe that clustered sparsity structure provides a distinct advantage in reconstruction compared to LASSO formulation and the sparse-CLASH algorithm. Furthermore, note that when λ is large, norm constraints have *no effect* and the model-CLASH provides essentially the same results as the model-CS approach [BCDH10]. On the other hand, the sparse-CLASH improves beyond the LASSO solution, thanks to the ℓ_1-norm constraint.

In Figure 2.7, bottom row, right panel, we show a case where model-CLASH and ℓ_1-norm can be successfully combined: while sparse-CLASH by itself improves over SP, BPDN and LASSO, model-CLASH further leads to better reconstruction performance for a wide range of λ values. Moreover, one can easily observe the impact of ℓ_1-norm in the recovery process since the model $\mathcal{M}_k$ itself does not "hit" the error lower bound, as in Figure 2.7, bottom row, left panel.

2.6.3 Sparsity and other norms

± 1-signal recovery with NORMED PURSUITS: We instantiate $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ satisfies $\|\boldsymbol{\varepsilon}\|_2 = 0.1$, and $n = 125$, $m = 65$, $k = 25$. Here, the coefficients of $\mathbf{x}^\star$ are randomly assigned to ± 1 values. To reconstruct $\mathbf{x}^\star$ from $\boldsymbol{y}$, we test NORMED PURSUITS with (i) ℓ_1-norm, (ii) ℓ_2-norm and, (iii) ℓ_∞-norm.

We illustrate the signal recovery results in Figure 2.9(Left). We notice that the reconstruction performance varies by using different norm constraints; in the case of signed signals, we deduce that ℓ_∞ norm provides the best results as compared to ℓ_1 and ℓ_2-norm constraints.

$\|\cdot\|_{\text{TV}}$ provides a strong norm constraint: Here, $\mathbf{x}^\star$ follows the (k, C)-clustered sparsity model where

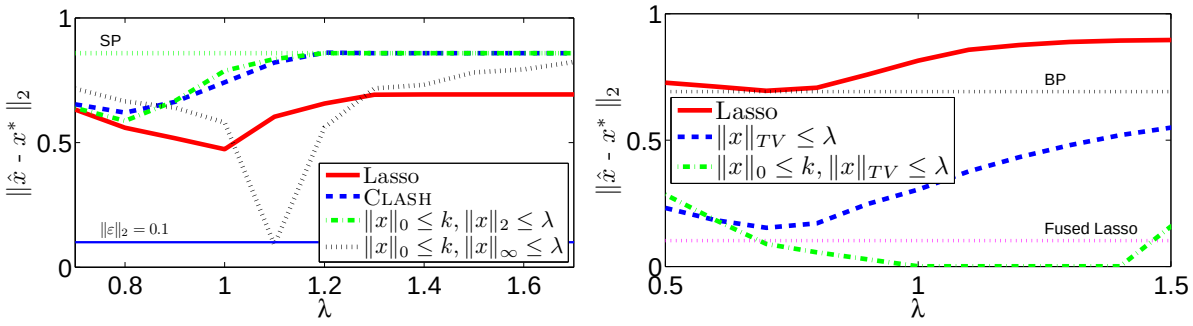


Figure 2.8: For each λ, we run 100 Monte-Carlo iterations and pick the median value of signal data error $\|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_2$.

the clustered nonzero elements have approximately flat values. We consider the noiseless case $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star$ for $n = 500$, $m = 100$ and $k = 50$.

We now compare NORMED PURSUITS with $\|\mathbf{x}\|_{\text{TV}} \leq \lambda$, $\|\mathbf{x}\|_0 \leq k$ with: i) the LASSO method, ii) Basis Pursuit using the SPGL1 implementation [VDBF08], iii) the TV-constrained version of LASSO where $\|\mathbf{x}\|_1 \leq \lambda$ is replaced with $\|\mathbf{x}\|_{\text{TV}} \leq \lambda$ and, iv) Fused LASSO [TSR+05] with TV and ℓ_1-norm constraints, where the *true regularization parameters are assumed known*. Figure 2.9(Right) provides empirical evidence that hard thresholding with the TV-norm constraint *outperforms* the other algorithms in terms of signal reconstruction, where sparsity constraints assist norm-constrained optimization in the estimation performance.

2.6.4 Image processing

In this subsection, we evaluate the following optimization formulations:

$$\widehat{\mathbf{x}}_{\text{TVDN}} = \arg\min \left\{ \|\boldsymbol{\Psi}\mathbf{x}\|_{\text{TV}} : \ \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2 \leq \sigma \right\}, \tag{2.32}$$

$$\widehat{\mathbf{x}}_{\text{TV-LASSO}} = \arg\min \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 : \ \|\boldsymbol{\Psi}\mathbf{x}\|_{\text{TV}} \leq \lambda \right\}, \tag{2.33}$$

$$\widehat{\mathbf{x}}_{\text{TV-PURSUIT}} = \arg\min \left\{ \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2 : \ \|\mathbf{x}\|_{\text{TV}} \leq \lambda, \ \|\boldsymbol{\Psi}\mathbf{x}\|_0 \leq k \right\}, \tag{2.34}$$

where $\boldsymbol{\Psi}$ represents an orthonormal wavelet transform.

Compressive sensing with TV-NORMED PURSUIT: To study the compressive sensing recovery performance of NORMED PURSUITS, we use the classical cameraman and brain[9] images of $n = 256 \times 256$ pixels. We compressively measure both images with the spread spectrum technique [PMG+12]. That is, the sensing matrix $\boldsymbol{\Phi}$ consists of a random pre-modulation followed by a random selection of $m = 0.25\, n$ complex Fourier coefficients. The performance of NORMED PURSUIT (2.34) is compared with: (i) the TV version of Basis Pursuit where the TV norm is substituted for the ℓ_1-norm as in (2.32), (ii) the TV-constrained version of LASSO as in (2.33). We choose the Daubechies-4 wavelet for $\boldsymbol{\Psi}$ and $k = m$. The parameter λ in (2.34) and (2.33) was chosen to obtain the best reconstruction for each method. Figure 2.9 shows the reconstruction obtained with the three methods. NORMED PURSUIT (2.34) *outperforms* all the other methods with an improvement of at least $0.8\,\text{dB}$ on the signal-to-noise ratio.

[9] BRAINIX database: http://pubimage.hcuge.ch:8080/.

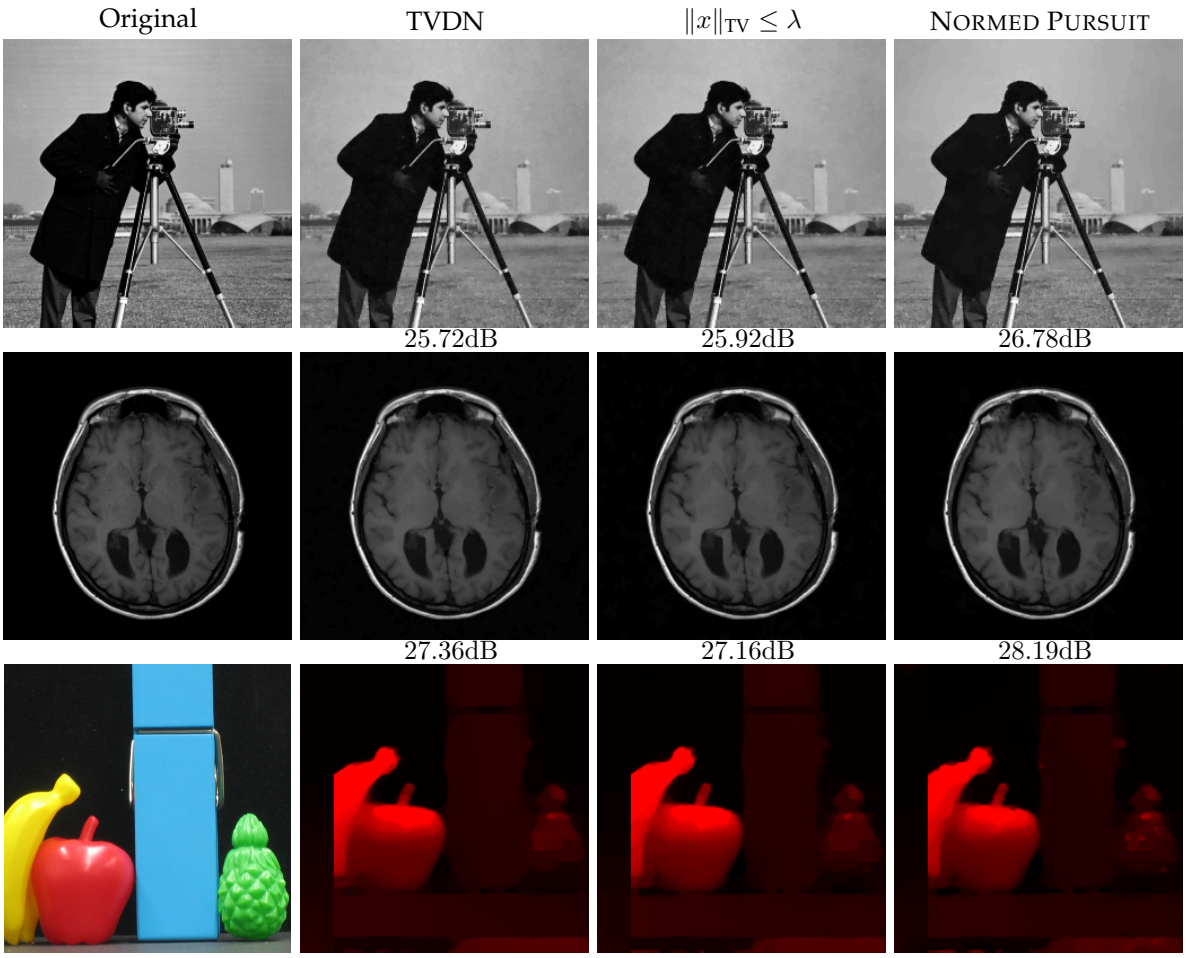


Figure 2.9: Results from real data.

CASSI recovery with TV-NORMED PURSUIT: We test the performance of our approach using the Coded Aperture Snapshot Spectral Imager (CASSI) data [WPSB08].[10] We reconstruct three-dimensional spatio-spectral data cube from measurements. While we obtain the full set of images, we only provide the result at the wavelength $549\,\text{nm}$ in Figure 2.9(Bottom); the full spectrum of results is provided in the Appendix of this chapter. In general, the contours are better resolved with NORMED PURSUIT compared to the other methods studied. Moreover, in our subjective evaluation, the contrast is improved overall across all the wavelengths.

2.7 Discussion

We presented variants of hard thresholding methods along with optimal/efficient strategies for their usage. While convergence derivations of the proposed schemes might be characterized by weaker bounds as compared to state-of-the-art approaches, the performance gained by such choices, both in terms of convergence rate and data recovery, is quite significant. A highlight of our discussion is that memory-based methods lead to convergence speed with (almost) no extra cost on the complexity of

[10] http://www.disp.duke.edu/projects/CASSI

baseline hard thresholding methods but more theoretical justification is needed.

Along this line of work, we proposed CLASH and NORMED PURSUIT schemes which establish a regression framework where efficient algorithms from combinatorial and convex optimization can interface for interpretable sparse solutions. Overall, our experiments demonstrate that, while the model-based combinatorial selection by itself can greatly improve sparse recovery over the approaches based on uniform sparsity alone (see next Chapter for more information), the shrinkage operations due to the norm constraints has an undeniable, positive impact on the learning performance. Understanding the tradeoffs between the complexity of approximation and the recovery guarantees is a promising theoretical as well as practical direction.

The discussion in this chapter naturally leads to the following open problems:

Open question 2. *Within the CS framework, memory-based methods show impressive empirical performance, both in terms of their convergence rate guarantees and their signal recovery performance. However, assuming RIP, there are no known results with stronger recovery conditions and/or provably faster convergence rate as compared to memoryless approaches.*

Open question 3. *By using norm constraints with discrete sparsity models in regression, we show efficient algorithms with strong recovery guarantees. An important empirical observation is the ability of such schemes to recover the unknown sparse signal $\mathbf{x}^\star$ from fewer number of measurements m than dictated by state-of-the-art schemes. Unfortunately though, the RIP sampling bound characterization in our analysis does not change, even if we have a norm-constraint. We believe that along this research direction, there is room for some new theoretical developments.*

Open question 4. *Inspired by recent theoretical computer science developments [AGM12], there are many discrete structures naturally emerging in practice that come with $(1-\epsilon)$-approximation guarantees when projections are needed, similar to the PMAP_ϵ definition. As a stylized example, consider the case of* graph sketching *where one is interested in finding minimum spanning trees and sparsifiers from a limited number of linear measurements, assuming that $\mathbf{x}^\star$ follows an underlying graph structure. We believe that the above constitutes a good research direction for the future.*

Appendix

This section contains all the proofs not reported in the main text.

Proof of Corollary 2

In what follows, assume $\mathcal{S}^\star := \text{supp}(\mathbf{x}^\star)$. Starting with the orthogonality principle for the operation:

$$\mathbf{x}_{i+1} = \underset{x:\text{supp}(\mathbf{x})\in\mathcal{S}_{i+1}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$$

the following holds true: $\langle \mathbf{\Phi}\mathbf{x}_{i+1} - \boldsymbol{y}, \mathbf{\Phi}\mathbf{w}\rangle = 0$ for any $\mathbf{w}$ with $\text{supp}(\mathbf{w}) \subseteq \mathcal{S}_{i+1}$. Given that $\boldsymbol{y} = \mathbf{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon}$ and $\text{supp}\big((\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\big) \subseteq \mathcal{S}_{i+1}$, by the orthogonality principle we have:

$$
\begin{aligned}
\big\langle \mathbf{\Phi}\mathbf{x}_{i+1} - \boldsymbol{y}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle = 0 \Rightarrow \\
\big\langle \mathbf{\Phi}\mathbf{x}_{i+1} - \mathbf{\Phi}\mathbf{x}^\star - \boldsymbol{\varepsilon}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle = 0 \Rightarrow \\
\big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, \mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle - \big\langle \mathbf{\Phi}^*\boldsymbol{\varepsilon}, (\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle = 0 \Rightarrow \\
\mu_i \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, \mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle - \mu_i \big\langle \mathbf{\Phi}^*\boldsymbol{\varepsilon}, (\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle = 0 \Rightarrow \\
\big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, \mu_i\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle - \mu_i \big\langle \boldsymbol{\varepsilon}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle = 0
\end{aligned}
$$

One can easily observe the following equation:

$$
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 = \|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2^2 + \|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}^c}\|_2^2 \tag{2.35}
$$

Beginning with the $\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2^2$ term, we observe that:

$$
\begin{aligned}
&\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2^2 = \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, (\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle \\
&= \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, (\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle - \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, \mu_i\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle + \mu_i \big\langle \boldsymbol{\varepsilon}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle \\
&= \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, (\mathbf{I} - \mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle + \mu_i \big\langle \boldsymbol{\varepsilon}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle \\
&= \big\langle \mathbf{x}_{i+1} - \mathbf{x}^\star, (\mathbf{I} - \mu_i\mathbf{\Phi}_T^*\mathbf{\Phi}_T)(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle + \mu_i \big\langle \boldsymbol{\varepsilon}, \mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}} \big\rangle \\
&\overset{(i)}{\leq} \|\mathbf{I} - \mu_i\mathbf{\Phi}_T^*\mathbf{\Phi}_T\|_{2\to 2}\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2 + \mu_i\|\mathbf{\Phi}(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2\|\boldsymbol{\varepsilon}\|_2 \\
&\overset{(ii)}{\leq} \|\mathbf{I} - \mu_i\mathbf{\Phi}_T^*\mathbf{\Phi}_T\|_{2\to 2}\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2 + \mu_i\sqrt{1+\delta_k}\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2\|\boldsymbol{\varepsilon}\|_2
\end{aligned}
$$

where $T = \text{supp}(\mathbf{x}_{i+1}) \cup \text{supp}(\mathbf{x}^\star)$ with $|T| \leq 2k$, (i) is due to Cauchy-Schwarz inequality and (ii) comes from RIP property. Eliminating $\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2$ from each side, we get the following inequality:

$$
\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2 \leq \|\mathbf{I} - \mu_i\mathbf{\Phi}_T^*\mathbf{\Phi}_T\|_{2\to 2}\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 + \mu_i\sqrt{1+\delta_k}\|\boldsymbol{\varepsilon}\|_2
$$

Given that $\frac{1}{1+\delta_{2k}} \leq \mu_i \leq \frac{1}{1-\delta_{2k}}$ and $\|\mathbf{I} - \mu_i\mathbf{\Phi}_T^*\mathbf{\Phi}_T\|_{2\to 2} \leq \frac{2\delta_{2k}}{1-\delta_{2k}}$, we obtain:

$$
\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}}\|_2 \leq \frac{2\delta_{2k}}{1-\delta_{2k}}\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}\|e\|_2
$$

Substituting in (2.35), we get the following quadratic polynomial:

$$
\begin{aligned}
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 \leq \Big(\frac{2\delta_{2k}}{1-\delta_{2k}}\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2\Big)^2 + \|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}^c}\|_2^2 \Rightarrow \\
\Big(1 - \frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}\Big)\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2^2 - \frac{4\delta_{2k}\sqrt{1+\delta_k}}{(1-\delta_{2k})^2}\|\boldsymbol{\varepsilon}\|_2\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 - \Big(\frac{1+\delta_k}{(1-\delta_{2k})^2}\|\boldsymbol{\varepsilon}\|_2^2 + \|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}^c}\|_2^2\Big) \leq 0
\end{aligned}
$$

Considering $\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2$ as variable in the above polynomial, we compute the root solutions and obtain the following bound:

$$
\|\mathbf{x}_{i+1} - \mathbf{x}^\star\|_2 \leq \frac{1}{\sqrt{1 - \frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}}\|(\mathbf{x}_{i+1} - \mathbf{x}^\star)_{\mathcal{S}_{i+1}^c}\|_2 + c_1\|\boldsymbol{\varepsilon}\|_2, \quad \text{where } c_1 := \frac{\frac{2\delta_{2k}\sqrt{1+\delta_k}}{(1-\delta_{2k})^2} + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}}{1 - \frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}.
$$

It is obvious that:

$$\|(\mathbf{x}_{i+1}-\mathbf{x}^\star)_{\mathcal{S}^c_{i+1}}\|_2^2 = \|(\mathbf{x}_{i+1}-\mathbf{x}^\star)_{\mathcal{S}^*\setminus\mathcal{S}_{i+1}}\|_2^2 = \|(\hat{\mathbf{x}}_{i+1}-\mathbf{x}^\star)_{\mathcal{S}^*\setminus\mathcal{S}_{i+1}}\|_2^2$$

We focus on the hard-thresholding projection $\hat{\mathbf{x}}_{i+1} = \mathcal{P}_{\Sigma_k}(\bar{\mathbf{x}}_i)$ where $\bar{\mathbf{x}}_i = \mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i)$. Let $\widehat{\mathcal{X}}_{i+1} := \text{supp}(\hat{\mathbf{x}}_{i+1})$. Given support set $\mathcal{S}_i$, we know that $\text{supp}(\bar{\mathbf{x}}_i) \subseteq \mathcal{S}_i$ and, futhermore, $\widehat{\mathcal{X}}_{i+1} \subseteq \mathcal{S}_i$. Since $\hat{\mathbf{x}}_{i+1}$ contains the k largest elements of $\bar{\mathbf{x}}_i$, we deduce that:

$$\|(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i))_{\mathcal{S}^*}\|_2^2 \leq \|(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i))_{\widehat{\mathcal{X}}_{i+1}}\|_2^2$$

Eliminating the common contribution $\|(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i))_{\mathcal{S}^*\cap\widehat{\mathcal{X}}_{i+1}}\|_2^2$ on both sides, we get:

$$\begin{aligned}
&\|(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \leq \|(\mathbf{x}_i - \frac{\mu_i}{2}\nabla_{\mathcal{S}_i} f(\mathbf{x}_i))_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 \Rightarrow \\
&\|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \leq \|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i))_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2
\end{aligned}$$

The right hand side of the above inequality satisfies:

$$\begin{aligned}
\|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i))_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 &= \|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\mathbf{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon} - \mathbf{\Phi}\mathbf{x}_i))_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 \\
&= \|(\mathbf{x}_i - \mathbf{x}^\star + \mu_i\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star - \mathbf{x}_i) + \mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 \\
&\overset{(i)}{\leq} \|\big((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}_i-\mathbf{x}^\star)\big)_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 + \|(\mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2
\end{aligned}$$

where (i) is due to triangle inequality. On the other hand, the left hand side can be rewritten as:

$$\begin{aligned}
\|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 &= \|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\mathbf{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon} - \mathbf{\Phi}\mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \\
&= \|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star - \mathbf{x}_i) + \mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \\
&= \|(\mathbf{x}_i - \mathbf{x}^\star + \mathbf{x}^\star + \mu_i\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star - \mathbf{x}_i) + \mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \\
&= \|((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}^\star - \mathbf{x}_i) + \mathbf{x}^\star + \mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \\
&= \|((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}^\star - \mathbf{x}_i) + (\mathbf{x}^\star - \mathbf{x}_{i+1}) + \mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2
\end{aligned}$$

$$\geq \|(\mathbf{x}^\star - \mathbf{x}_{i+1})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 - \|((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}^\star - \mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 - \|(\mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2$$

Combining the above two expressions of $\|(\mathbf{x}_i + \mu_i\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2$, we obtain:

$$\begin{aligned}
&\|(\mathbf{x}^\star - \mathbf{x}_{i+1})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \leq \|((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}^\star - \mathbf{x}_i))_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 + \|(\mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}^*\setminus\widehat{\mathcal{X}}_{i+1}}\|_2 \\
&+ \|\big((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}_i-\mathbf{x}^\star)\big)_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 + \|(\mu_i\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\widehat{\mathcal{X}}_{i+1}\setminus\mathcal{S}^*}\|_2 \\
&\leq \sqrt{2}\|\big((\mathbf{I}-\mu_i\mathbf{\Phi}^*\mathbf{\Phi})(\mathbf{x}_i-\mathbf{x}^\star)\big)_{(\widehat{\mathcal{X}}_{i+1}\cup\mathcal{S}^*)\setminus(\widehat{\mathcal{X}}_{i+1}\cap\mathcal{S}^*)}\|_2 + \sqrt{2}\mu_i\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{(\widehat{\mathcal{X}}_{i+1}\cup\mathcal{S}^*)\setminus(\widehat{\mathcal{X}}_{i+1}\cap\mathcal{S}^*)}\|_2 \\
&\leq \sqrt{2}\delta_{3k}\|\mathbf{x}_i - \mathbf{x}^\star\|_2 + \frac{\sqrt{2(1+\delta_{2k})}}{1-\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2
\end{aligned}$$

In conclusion, we combine the above results to get:

$$\|\mathbf{x}_{i+1}-\mathbf{x}^\star\|_2 \leq \frac{1}{\sqrt{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}}\Big(\sqrt{2}\delta_{3k}\|\mathbf{x}_i-\mathbf{x}^\star\|_2+\frac{\sqrt{2(1+\delta_{2k})}}{1-\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2\Big)+c_1\|\boldsymbol{\varepsilon}\|_2 \leq \rho\|\mathbf{x}_i-\mathbf{x}^\star\|_2+c_2\|\boldsymbol{\varepsilon}\|_2 \tag{2.36}$$

where $\rho := \frac{\sqrt{2}\delta_{3k}}{\sqrt{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}}$ and $c_2 := \frac{\sqrt{2(1+\delta_{2k})}}{1-\delta_{2k}}\frac{1}{\sqrt{1-\frac{4\delta_{2k}^2}{(1-\delta_{2k})^2}}}+c_1$.

Moreover, the iterations are constractive iff $\rho<1$. This happens for δ_{3k} such that

$$2\delta_{3k}^4-4\delta_{3k}^3+5\delta_{3k}^2+2\delta_{3k}-1<0, \quad \text{i.e. } \delta_{3k}<0.31. \tag{2.37}$$

□

Proof of Theorem 5

A well-known lemma used in the convergence guarantee proof of CLASH is defined next.

Lemma 13 (Optimality condition). *Let $\Theta\subseteq\mathbb{R}^n$ be a convex set and $f:\Theta\to\mathbb{R}$ be a smooth objective function defined over Θ. Let $\boldsymbol{\psi}^\star\in\Theta$ be a local minimum of the objective function f over the set Θ. Then*

$$\langle\nabla f(\boldsymbol{\psi}^*),\boldsymbol{\psi}-\boldsymbol{\psi}^\star\rangle\geq 0, \quad \forall\boldsymbol{\psi}\in\Theta. \tag{2.38}$$

For clarity reasons, we present the proof of Theorem 1 as a collection of lemmas to help readability.

Lemma 14 (Active set expansion). *The support set $\mathcal{S}_i$, where $|\mathcal{S}_i|\leq 2k$, identifies a subspace in $\mathcal{M}_{2k}$ such that:*

$$\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2 \leq (\delta_{3k}+\delta_{2k}+\sqrt{\epsilon}(1+\delta_{2k}))\|\mathbf{x}_i-\mathbf{x}^\star\|_2+\big(\sqrt{2(1+\delta_{3k})}+\sqrt{\epsilon(1+\delta_{2k})}\big)\|\boldsymbol{\varepsilon}\|_2. \tag{2.39}$$

Proof. Let $\mathcal{X}_i\cup\mathcal{X}^*$ denote the union of the support sets of the current estimate $\mathbf{x}_i$ and the signal of interest $\mathbf{x}^\star$. Then, the following sequence of inequalities hold true:

$$F(\mathcal{X}_i\cup\mathcal{X}^*;\nabla f(\mathbf{x}_i))\leq F(\mathcal{X}_i\cup\text{supp}(\mathcal{P}_{\mathcal{M}_k}(\nabla_{\mathcal{X}_i^c}f(\mathbf{x}_i)));\nabla f(\mathbf{x}_i)) \Rightarrow$$
$$(1-\epsilon)F(\mathcal{X}_i\cup\mathcal{X}^*;\nabla f(\mathbf{x}_i))\leq(1-\epsilon)F(\mathcal{X}_i\cup\text{supp}(\mathcal{P}_{\mathcal{M}_k}(\nabla_{\mathcal{X}_i^c}f(\mathbf{x}_i)));\nabla f(\mathbf{x}_i))$$

Given $\mathcal{S}_i$ is an ϵ-approximate support set, from the definition of PMAP, we further havea:

$$(1-\epsilon)F(\mathcal{X}_i\cup\mathcal{X}^*;\nabla f(\mathbf{x}_i))\leq F(\mathcal{S}_i;\nabla f(\mathbf{x}_i)).$$

Substituting the variance reduction modular function $F(\mathcal{S};\mathbf{x})\triangleq\|\mathbf{x}\|_2^2-\|(\mathbf{x})_\mathcal{S}-\mathbf{x}\|_2^2=\|(\mathbf{x})_\mathcal{S}\|_2^2$, we get:

$$(1-\epsilon)\big\|\nabla_{\mathcal{X}_i\cup\mathcal{X}^*}f(\mathbf{x}_i)\big\|_2^2\leq\big\|\nabla_{\mathcal{S}_i}f(\mathbf{x}_i)\big\|_2^2\Rightarrow(1-\epsilon)\Big\|\Big(\boldsymbol{\Phi}^*(\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\Big\|_2^2\leq\Big\|\Big(\boldsymbol{\Phi}^*(\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}_i)\Big)_{\mathcal{S}_i}\Big\|_2^2\Rightarrow$$
$$\Big\|\Big(\boldsymbol{\Phi}^*(\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\Big\|_2^2\leq\Big\|\Big(\boldsymbol{\Phi}^*(\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}_i)\Big)_{\mathcal{S}_i}\Big\|_2^2+\epsilon\Big\|\Big(\boldsymbol{\Phi}^*(\boldsymbol{y}-\boldsymbol{\Phi}\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\Big\|_2^2.$$

Using the subadditivity property of the square root function and excluding the common distribution $\big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\cap\mathcal{S}_i}$, we have:

$$\begin{aligned}
\|\Big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\Big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2 &\leq \|\Big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\Big)_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\sqrt{\epsilon}\|\Big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2\\
&\overset{(i)}{\leq} \|\Big(\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\sqrt{\epsilon}\|\Big(\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2+\sqrt{\epsilon}\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2\\
&\overset{(ii)}{=} \|\Big((\mathbf{\Phi}^*\mathbf{\Phi}-\mathbb{I})(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\sqrt{\epsilon}\|\Big(\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2+\sqrt{\epsilon}\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2\\
&\overset{(iii)}{\leq} (\delta_{3k}+\sqrt{\epsilon}(1+\delta_{2k}))\|\mathbf{x}_i-\mathbf{x}^\star\|_2+\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}\|_2+\sqrt{\epsilon}\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{\mathcal{X}_i\cup\mathcal{X}^*}\|_2. \qquad (2.40)
\end{aligned}$$

where (i) is obtained by applying the triangle inequality, (ii) holds since $(\mathbf{x}^\star-\mathbf{x}_i)_{\mathcal{S}_i\setminus(\mathcal{X}_i\cup\mathcal{X}^*)}=0$ and (iii) is due to Cauchy-Swartz inequality and isometry constant definition.

In addition, we can obtain a lower bound for $\|\big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2$:

$$\begin{aligned}
\|\Big(\mathbf{\Phi}^*(\boldsymbol{y}-\mathbf{\Phi}\mathbf{x}_i)\Big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2 &= \|\Big(\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}+(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2\\
&= \|\Big(\mathbf{\Phi}^*\mathbf{\Phi}(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}+(\mathbf{x}^\star-\mathbf{x}_i)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}-(\mathbf{x}^\star-\mathbf{x}_i)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}+(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2\\
&\geq \|(\mathbf{x}^\star-\mathbf{x}_i)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2-\|\Big((\mathbf{\Phi}^*\mathbf{\Phi}-\mathbb{I})(\mathbf{x}^\star-\mathbf{x}_i)\Big)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2-\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2\\
&\overset{(i)}{\geq} \|(\mathbf{x}^\star-\mathbf{x}_i)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2-\delta_{2k}\|\mathbf{x}^\star-\mathbf{x}_i\|_2-\|(\mathbf{\Phi}^*\boldsymbol{\varepsilon})_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2. \qquad (2.41)
\end{aligned}$$

where (i) is obtained by using Cauchy-Swartz inequality and isometry constant definition.

Since $\|(\mathbf{x}_i-\mathbf{x}^\star)_{(\mathcal{X}_i\cup\mathcal{X}^*)\setminus\mathcal{S}_i}\|_2=\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2$, combining (2.40) and (2.41), we get:

$$\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2\leq(\delta_{3k}+\delta_{2k}+\sqrt{\epsilon}(1+\delta_{2k}))\|\mathbf{x}_i-\mathbf{x}^\star\|_2+\big(\sqrt{2(1+\delta_{3k})}+\sqrt{\epsilon(1+\delta_{2k})}\big)\|\boldsymbol{\varepsilon}\|_2.$$

as a consequence of the RIP inequality. □

Lemma 15 (Greedy descent with least absolute shrinkage)**.** *Let $\mathcal{S}_i$ be a $2k$-sparse support set. Then, the least squares solution $\mathbf{v}_i$ in step 2 of Algorithm 1 satisfies*

$$\|\mathbf{v}_i-\mathbf{x}^\star\|_2\leq\frac{1}{\sqrt{1-\delta_{3k}^2}}\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2+\frac{\sqrt{1+\delta_{2k}}}{1-\delta_{3k}}\|\boldsymbol{\varepsilon}\|_2.$$

Proof. We know that $\text{supp}(\mathbf{v}_i)\in\mathcal{S}_i$. Starting from $\|\mathbf{v}_i-\mathbf{x}^\star\|_2^2$, the following holds true:

$$\|\mathbf{v}_i-\mathbf{x}^\star\|_2^2=\|(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2+\|(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2^2.$$

Using the optimality condition, $\mathbf{v}_i$ is the minimizer of $\|\boldsymbol{y}-\mathbf{\Phi}v\|_2^2$ over the convex set $\Theta=\{v:\|v\|_1\leq\lambda,\text{supp}(v)\in\mathcal{S}_i\}$ and therefore:

$$\langle\nabla f(\mathbf{v}_i),(\mathbf{x}^\star-\mathbf{v}_i)_{\mathcal{S}_i}\rangle\geq0\Rightarrow\langle\mathbf{\Phi}\mathbf{v}_i-\boldsymbol{y},\mathbf{\Phi}(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i}\rangle\leq0.$$

We calculate the following:

$$\begin{aligned}
\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2 &= \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&\leq \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle - \langle \boldsymbol{\Phi}\mathbf{v}_i - \boldsymbol{y}, \boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&= \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle - \langle \boldsymbol{\Phi}\mathbf{v}_i - \boldsymbol{\Phi}\mathbf{x}^\star - \boldsymbol{\varepsilon}, \boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&= \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle - \langle \mathbf{v}_i - \mathbf{x}^\star, \boldsymbol{\Phi}^*\boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle + \langle \boldsymbol{\varepsilon}, \boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&= \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbb{I} - \boldsymbol{\Phi}^*\boldsymbol{\Phi})(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle + \langle \boldsymbol{\varepsilon}, \boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&\leq |\langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbb{I} - \boldsymbol{\Phi}^*\boldsymbol{\Phi})(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle| + \langle \boldsymbol{\varepsilon}, \boldsymbol{\Phi}(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i} \rangle \\
&\overset{(i)}{\leq} \delta_{3k}\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2\|\boldsymbol{\varepsilon}\|_2, \qquad (2.42)
\end{aligned}$$

where (i) comes from Cauchy-Swartz inequality and isometry constant definition. Simplifying the above quadratic expression, we obtain:

$$\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2 \leq \delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2. \qquad (2.43)$$

As a consequence, (2.42) can be upper bounded by:

$$\|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 \leq (\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2)^2 + \|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2^2.$$

We form the quadratic polynomial for this inequality assuming as unknown variable the quantity $\|\mathbf{v}_i - \mathbf{x}^\star\|_2$. Bounding by the largest root of the resulting polynomial, we get:

$$\|\mathbf{v}_i - \mathbf{x}^\star\|_2 \leq \frac{1}{\sqrt{1-\delta_{3k}^2}}\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2 + \frac{\sqrt{1+\delta_{2k}}}{1-\delta_{3k}}\|\boldsymbol{\varepsilon}\|_2.$$

□

Lemma 16 (Combinatorial selection). *Let $\mathbf{v}_i$ be a $2k$-sparse proxy vector with indices in support set $\mathcal{S}_i$, $\mathcal{M}_k$ be a CSM and $\boldsymbol{\gamma}_i$ the projection of $\mathbf{v}_i$ under $\mathcal{M}_k$. Then:*

$$\|\boldsymbol{\gamma}_i - \mathbf{v}_i\|_2^2 \leq (1-\epsilon)\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2.$$

Proof. Let $\boldsymbol{\gamma}_i^{\text{opt}}$ denote the optimal combinatorial projection of $\mathbf{v}_i$ under $\mathcal{M}_k$, i.e.

$$\boldsymbol{\gamma}_i^{\text{opt}} = \mathcal{P}_{\mathcal{M}_k}(\mathbf{v}_i) = \underset{(\mathbf{v}_i)_{\mathcal{S}}:\mathcal{S}\in\mathcal{N},\mathcal{S}\in\mathcal{M}_k}{\arg\max} F(\mathcal{S};\mathbf{v}_i).$$

By the definition of the non-convex projection onto CSMs, it is apparent that:

$$\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2 \leq \|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2, \qquad (2.44)$$

over $\mathcal{M}_k$ since $\boldsymbol{\gamma}_i^{\text{opt}}$ is the best approximation to $\mathbf{v}_i$ for that particular CSM. In the general case, this step is performed approximately and we get $\boldsymbol{\gamma}_i$ as $\boldsymbol{\gamma}_i = \mathcal{P}_{\mathcal{M}_k}^{\epsilon}(\mathbf{v}_i)$, an ϵ-approximate projection of $\mathbf{v}_i$ with

corresponding variance reduction $F(\widehat{\mathcal{S}}_\epsilon; \mathbf{v}_i)$. According to the definition of PMAP$_\epsilon$, we calculate:

$$\begin{aligned} F(\widehat{\mathcal{S}}_\epsilon; \mathbf{v}_i) \geq (1-\epsilon) \max_{\mathcal{S}\in\mathcal{M}_k} F(\mathcal{S}; \mathbf{v}_i) &\Rightarrow \|\mathbf{v}_i\|_2^2 - \|\boldsymbol{\gamma}_i - \mathbf{v}_i\|_2^2 \geq (1-\epsilon)\Big[\|\mathbf{v}_i\|_2^2 - \|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2\Big] \\ &\Rightarrow \|\boldsymbol{\gamma}_i - \mathbf{v}_i\|_2^2 \leq (1-\epsilon)\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2 \\ &\Rightarrow \|\boldsymbol{\gamma}_i - \mathbf{v}_i\|_2^2 \overset{eq:ch2_s4_05}{\leq} (1-\epsilon)\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2. \end{aligned}$$

□

Lemma 17. *Let $\mathbf{v}_i$ be the least squares solution of the greedy descent step and $\boldsymbol{\gamma}_i$ be a proxy vector to $\mathbf{v}_i$ after applying Combinatorial selection step. Then, $\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2$ can be expressed in terms of the distance from $\mathbf{v}_i$ to $\mathbf{x}^\star$ as follows:*

$$\begin{aligned} \|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2 \leq \sqrt{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon} \cdot \|\mathbf{v}_i - \mathbf{x}^\star\|_2 \\ + D_1\|\boldsymbol{\varepsilon}\|_2 + D_2\|\mathbf{x}^\star\|_2 + D_3\sqrt{\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2}, \end{aligned} \tag{2.45}$$

where D_1, D_2, D_3 are constants depending on $\epsilon, \delta_{2k}, \delta_{3k}$.

Proof. We observe the following

$$\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2^2 = \|\boldsymbol{\gamma}_i - \mathbf{v}_i + \mathbf{v}_i - \mathbf{x}^\star\|_2^2 = \|(\mathbf{v}_i - \mathbf{x}^\star) - (\mathbf{v}_i - \boldsymbol{\gamma}_i)\|_2^2 \tag{2.46}$$

$$= \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + \|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2^2 - 2\langle \mathbf{v}_i - \mathbf{x}^\star, \mathbf{v}_i - \boldsymbol{\gamma}_i\rangle. \tag{2.47}$$

Focusing on the right hand side of expression (4.71), $\langle \mathbf{v}_i - \mathbf{x}^\star, \mathbf{v}_i - \boldsymbol{\gamma}_i\rangle = \langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \boldsymbol{\gamma}_i)_{\mathcal{S}_i}\rangle$ can be similarly analysed as (2.42) where we obtain the following expression:

$$|\langle \mathbf{v}_i - \mathbf{x}^\star, (\mathbf{v}_i - \boldsymbol{\gamma}_i)_{\mathcal{S}_i}\rangle| \leq \delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2 + \sqrt{1+\delta_{2k}}\|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2\|\boldsymbol{\varepsilon}\|_2. \tag{2.48}$$

Now, expression (4.71) can be further transformed as:

$$\begin{aligned} \|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2^2 &= \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + \|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2^2 - 2\langle \mathbf{v}_i - \mathbf{x}^\star, \mathbf{v}_i - \boldsymbol{\gamma}_i\rangle \\ &\leq \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + \|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2^2 + 2|\langle \mathbf{v}_i - \mathbf{x}^\star, \mathbf{v}_i - \boldsymbol{\gamma}_i\rangle| \\ &\overset{(i)}{\leq} \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + \|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2^2 + 2(\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2 + \sqrt{1+\delta_{2k}}\|\mathbf{v}_i - \boldsymbol{\gamma}_i\|_2\|\boldsymbol{\varepsilon}\|_2) \\ &\overset{(ii)}{\leq} \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + (1-\epsilon)\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2 + 2\Big(\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\sqrt{(1-\epsilon)\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2} \\ &+ \sqrt{1+\delta_{2k}}\sqrt{(1-\epsilon)\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2}\|\boldsymbol{\varepsilon}\|_2\Big), \end{aligned} \tag{2.49}$$

where (i) is due to (4.72) and (ii) is due to Lemma 11. Given that $\sqrt{a^2+b^2} \leq a+b$ for $a, b \geq 0$, we further

have:

$$\begin{aligned}
&\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2^2 \leq \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + (1-\epsilon)\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2 + 2\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\big(\sqrt{1-\epsilon}\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2 + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big) \\
&+ 2\sqrt{1+\delta_{2k}}\big(\sqrt{1-\epsilon}\|\boldsymbol{\gamma}_i^{\text{opt}} - \mathbf{v}_i\|_2 + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big)\|\boldsymbol{\varepsilon}\|_2 \\
&\overset{(i)}{\leq} \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + (1-\epsilon)\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2^2 + \epsilon\|\mathbf{v}_i\|_2^2 + 2\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\big(\sqrt{1-\epsilon}\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2 + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big) \\
&+ 2\sqrt{1+\delta_{2k}}\big(\sqrt{1-\epsilon}\|(\mathbf{v}_i - \mathbf{x}^\star)_{\mathcal{S}_i}\|_2 + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big)\|\boldsymbol{\varepsilon}\|_2 \\
&\overset{(ii)}{\leq} \|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + (1-\epsilon)(\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2)^2 + \epsilon\|\mathbf{v}_i\|_2^2 \\
&+ 2\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\big(\sqrt{1-\epsilon}(\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2) + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big) \\
&+ 2\sqrt{1+\delta_{2k}}\big(\sqrt{1-\epsilon}(\delta_{3k}\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{1+\delta_{2k}}\|\boldsymbol{\varepsilon}\|_2) + \sqrt{\epsilon}\|\mathbf{v}_i\|_2\big)\|\boldsymbol{\varepsilon}\|_2, \qquad (2.50)
\end{aligned}$$

where (i) is due to (2.44) and (ii) is due to (2.43). Applying basic algebra on the right hand side of (2.50), we get:

$$\begin{aligned}
\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2^2 &= \big(1 + (1-\epsilon)\delta_{3k}^2 + 2\delta_{3k}^2\sqrt{1-\epsilon}\big)\|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 \\
&\quad + \big(2(1-\epsilon)\delta_{3k}\sqrt{1+\delta_{2k}} + 4\delta_{3k}\sqrt{1-\epsilon}\sqrt{1+\delta_{2k}}\big)\|\mathbf{v}_i - \mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2 \\
&\quad + \big((1-\epsilon)(1+\delta_{2k}) + 2(1+\delta_{2k})\sqrt{1-\epsilon}\big)\|\boldsymbol{\varepsilon}\|_2^2 \\
&\quad + 2\delta_{3k}\sqrt{\epsilon}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\|\mathbf{v}_i\|_2 + 2\sqrt{\epsilon(1+\delta_{2k})}\|\mathbf{v}_i\|_2\|\boldsymbol{\varepsilon}\|_2 + \epsilon\|\mathbf{v}_i\|_2^2 \qquad (2.51) \\
&\overset{(i)}{\leq} \Big(1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2\Big)\left(\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{\frac{\big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)(1+\delta_{2k})}{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2}}\|\boldsymbol{\varepsilon}\|\right)^2 \\
&\quad + 2\delta_{3k}\sqrt{\epsilon}\|\mathbf{v}_i - \mathbf{x}^\star\|_2\|\mathbf{v}_i\|_2 + 2\sqrt{\epsilon(1+\delta_{2k})}\|\mathbf{v}_i\|_2\|\boldsymbol{\varepsilon}\|_2 + \epsilon\|\mathbf{v}_i\|_2^2. \qquad (2.52)
\end{aligned}$$

where (i) is obtained by completing the squares and eliminating negative terms in (2.51).

Using triangle inequality, we know that:

$$\|\mathbf{v}_i\|_2 \leq \|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \|\mathbf{x}^\star\|_2, \qquad (2.53)$$

and, thus, (4.70) can be further analyzed as:

$$\begin{aligned}
\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2^2 &\leq \Big(1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2\Big)\left(\|\mathbf{v}_i - \mathbf{x}^\star\|_2 + \sqrt{\frac{\big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)(1+\delta_{2k})}{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2}}\|\boldsymbol{\varepsilon}\|\right)^2 \\
&\quad + (2\delta_{3k}\sqrt{\epsilon} + \epsilon)\|\mathbf{v}_i - \mathbf{x}^\star\|_2^2 + (2\delta_{3k}\sqrt{\epsilon}\|\mathbf{x}^\star\|_2 + 2\sqrt{\epsilon(1+\delta_{2k})}\|\boldsymbol{\varepsilon}\|_2 + 2\epsilon\|\mathbf{x}^\star\|_2)\|\mathbf{v}_i - \mathbf{x}^\star\|_2 \\
&\quad + 2\sqrt{\epsilon(1+\delta_{2k})}\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2 + \epsilon\|\mathbf{x}^\star\|_2^2.
\end{aligned}$$

After tedious computations, we end up with the following inequality:

$$\begin{aligned}
\|\boldsymbol{\gamma}_i - \mathbf{x}^\star\|_2 &\leq \sqrt{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon} \cdot \|\mathbf{v}_i - \mathbf{x}^\star\|_2 \\
&\quad + D_1\|\boldsymbol{\varepsilon}\|_2 + D_2\|\mathbf{x}^\star\|_2 + D_3\sqrt{\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2},
\end{aligned}$$

where

$$D_1 \triangleq \frac{\sqrt{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2}\sqrt{\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)(1+\delta_{2k})}+\sqrt{\epsilon(1+\delta_{2k})}}{\sqrt{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}},$$

$$D_2 \triangleq \frac{\delta_{3k}\sqrt{\epsilon}+\epsilon}{\sqrt{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}}+\sqrt{\epsilon-\frac{(\epsilon+\delta_{3k}\sqrt{\epsilon})^2}{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}},$$

$$D_3 \triangleq \sqrt{2\sqrt{\epsilon(1+\delta_{2k})}}.$$

□

Using the above lemmas, we now complete the proof of Theorem 1.

Proof. Combining Lemma 10 with Lemma 12, we get:

$$\begin{aligned}\|\boldsymbol{\gamma}_i-\mathbf{x}^\star\|_2 \le \sqrt{\frac{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}{1-\delta_{3k}^2}}\cdot\|(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2\\ +D_4\|\boldsymbol{\varepsilon}\|_2+D_2\|\mathbf{x}^\star\|_2+D_3\sqrt{\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2},\end{aligned}$$

where

$$D_4 \triangleq D_1+\frac{\sqrt{1+\delta_{2k}}}{1-\delta_{3k}}\sqrt{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}.$$

We know that $\mathcal{X}_i \subseteq \mathcal{S}_i$. Thus, $(\mathbf{v}_i)_{\mathcal{S}_i^c}=0$ iff $(\mathbf{x}_i)_{\mathcal{S}_i^c}=0$. Therefore,

$$\|(\mathbf{v}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2=\|(\mathbf{v}_i)_{\mathcal{S}_i^c}-(\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2=\|(\mathbf{x}_i)_{\mathcal{S}_i^c}-(\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2=\|(\mathbf{x}_i-\mathbf{x}^\star)_{\mathcal{S}_i^c}\|_2.$$

Now, using Lemma 9, we form the following recursion:

$$\begin{aligned}\|\boldsymbol{\gamma}_i-\mathbf{x}^\star\|_2 \le \sqrt{\frac{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}{1-\delta_{3k}^2}}(\delta_{3k}+\delta_{2k}+\sqrt{\epsilon}(1+\delta_{2k}))\|\mathbf{x}_i-\mathbf{x}^\star\|_2\\ +D_5\|\boldsymbol{\varepsilon}\|_2+D_2\|\mathbf{x}^\star\|_2+D_3\sqrt{\|\mathbf{x}^\star\|_2\|\boldsymbol{\varepsilon}\|_2},\end{aligned} \tag{2.54}$$

where

$$D_5 \triangleq \sqrt{\frac{1+\big((1-\epsilon)+2\sqrt{1-\epsilon}\big)\delta_{3k}^2+2\delta_{3k}\sqrt{\epsilon}+\epsilon}{1-\delta_{3k}^2}}\big(\sqrt{2(1+\delta_{3k})}+\sqrt{\epsilon(1+\delta_{2k})}\big)+D_4.$$

Finally, substituting (2.54) in Corollary 3, we compute the desired recursive formula:

$$\frac{\|\mathbf{x}_{i+1}-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2} \le \rho\frac{\|\mathbf{x}_i-\mathbf{x}^\star\|_2}{\|\mathbf{x}^\star\|_2}+\frac{c_1(\delta_{2k},\delta_{3k},\epsilon)}{SNR}+c_2(\delta_{2k},\delta_{3k},\epsilon)+c_3(\delta_{2k},\delta_{3k},\epsilon)\sqrt{\frac{1}{SNR}},$$

where $SNR = \frac{\|\mathbf{x}^\star\|_2}{\|\boldsymbol{\varepsilon}\|_2} = \frac{\|\mathbf{x}^\star\|_2}{\sqrt{f(\mathbf{x}^\star)}}$ and

$$\rho \triangleq \frac{\delta_{3k} + \delta_{2k} + \sqrt{\epsilon}(1+\delta_{2k})}{\sqrt{1-\delta_{2k}^2}} \sqrt{\frac{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon}{1-\delta_{3k}^2}},$$

$$\begin{aligned}
c_1(\delta_{2k}, \delta_{3k}, \epsilon) &\triangleq \frac{D_5}{\sqrt{1-\delta_{2k}^2}} + \frac{\sqrt{1+\delta_k}}{1-\delta_{2k}}, \\
c_2(\delta_{2k}, \delta_{3k}, \epsilon) &\triangleq \frac{1}{\sqrt{1-\delta_{2k}^2}} \Bigg(\frac{\delta_{3k}\sqrt{\epsilon} + \epsilon}{\sqrt{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon}} \\
&\qquad + \sqrt{\epsilon - \frac{(\epsilon + \delta_{3k}\sqrt{\epsilon})^2}{1 + \big((1-\epsilon) + 2\sqrt{1-\epsilon}\big)\delta_{3k}^2 + 2\delta_{3k}\sqrt{\epsilon} + \epsilon}} \Bigg), \\
c_3(\delta_{2k}, \delta_{3k}, \epsilon) &\triangleq \frac{D_3}{\sqrt{1-\delta_{2k}^2}}.
\end{aligned}$$

□

CASSI Results

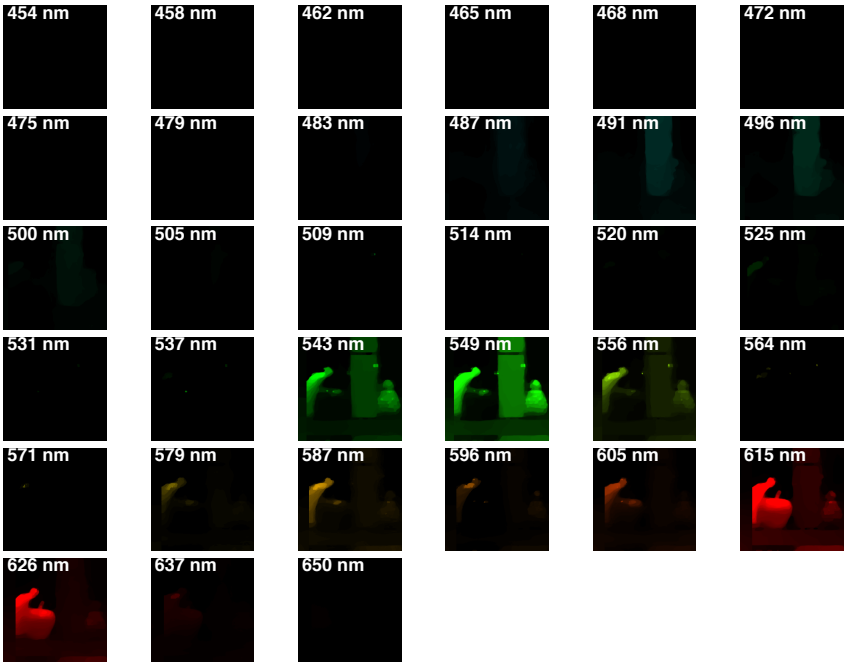


Figure 2.10: Full wavelength CASSI results for the TV version of Basis Pursuit where the TV norm is substituted for the ℓ_1-norm as in (2.32)

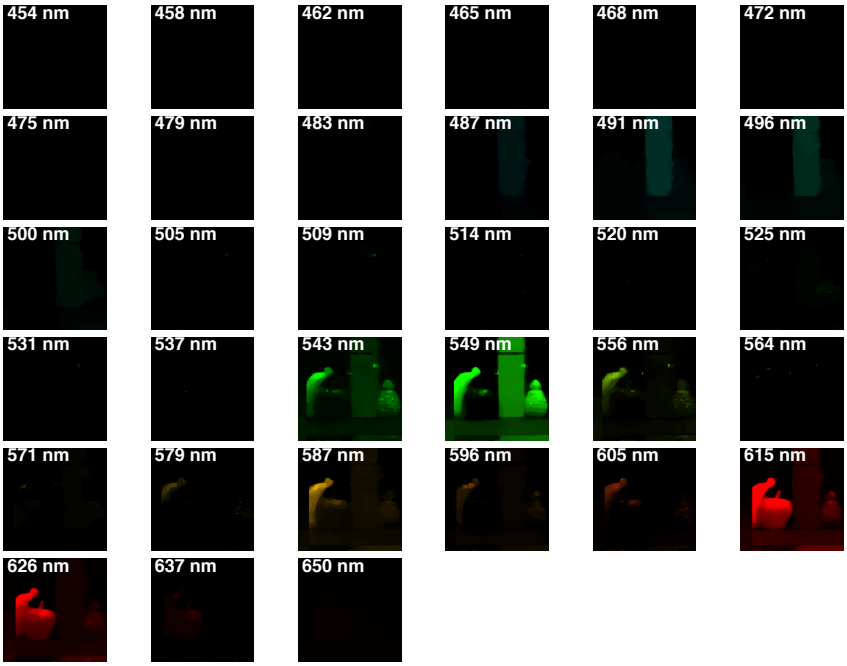


Figure 2.11: Full wavelength CASSI results for the TV-constrained version of LASSO as in (2.33)

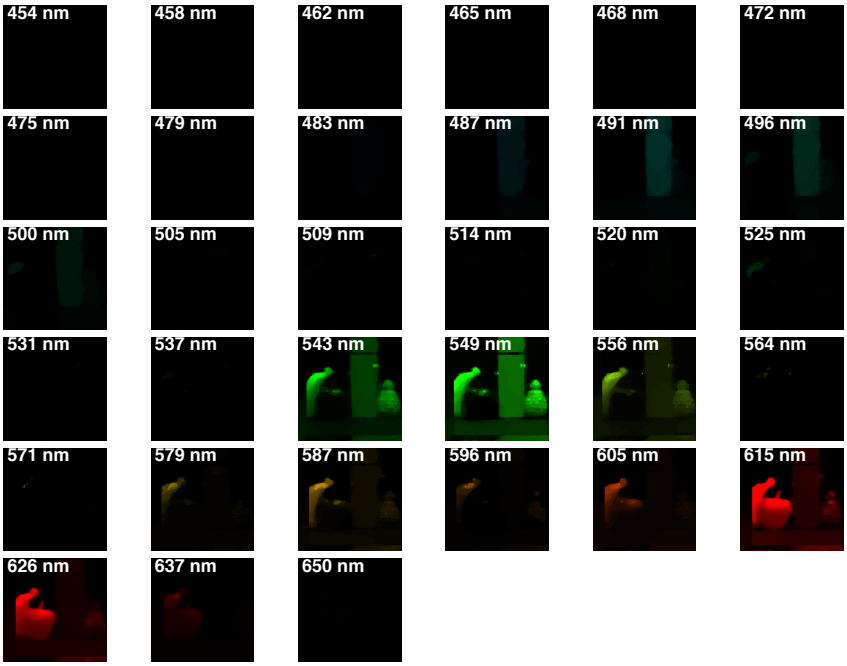


Figure 2.12: Full wavelength CASSI results for the NORMED PURSUIT

3 Beyond simple sparsity

Introduction

While many of the optimization solutions proposed so far result in *sparse* model selections, in many cases they do not capture the true underlying structure to best explain the observations [BCDH10]. Recent results in CS consider more sophisticated *structured* sparsity models, which describe the interdependency between the nonzero coefficients, increase the interpretability of the results and lead to better recovery performance [EM09, BD09b, BCDH10, RRN12]. Nowadays, we have witnessed aplenty elaborate approaches that guide the selection process: (overlapping) group LASSO, fused LASSO, greedy approaches for signal approximation under tree-structure assumptions just to name a few; see [YL06, FHT10, TSR$^+$05, JAB11]. To show the merits of such approaches, consider the problem of image recovery from a *limited* set of measurements using the tree-structured group sparse model [Bar99]. Figure 3.1 shows the performance of structured sparsity models, compared to simple ones.

Moreover, such a priori model-based assumptions result into more robust solutions and allow recovery with far fewer samples, e.g. $\mathcal{O}(k)$ instead of $\mathcal{O}\left(k\log(n/k)\right)$ samples for k sparse signals whose nonzero coefficients are arranged into few blocks or form a rooted connected subtree over the coefficients [BCDH10]. To highlight the importance of this property, in the case of Magnetic Resonance Imaging (MRI), reducing the total number of measurements is highly desirable for both capturing functional activities within small time periods and rendering the whole procedure less "painful" for the patient [LDP07].

In order to use such structures in practice, one needs efficient optimization solutions for structured sparsity problems that scale up in high-dimensional settings. From our discussions below, it will be apparent that the key actors for this purpose are *projection and proximity operations over structured sets* that go beyond simple selection heuristics and towards provable quality as well as runtime/space bounds.

Overall, projection operations faithfully follow the underlying combinatorial model but, in most cases, result in hard-to-solve or even combinatorial optimization problems. Furthermore, model misspecification often results in wildly inaccurate solutions. Proximity operators of convex sparsity-inducing norms often can only partially describe the underlying discrete model and might lead to "rules-of-thumb" in problem solving (e.g., how to set up the regularization parameter). However, such approaches work quite well in practice and are more robust to deviations from the model, leading to satisfactory solutions.

To this end, in this chapter we study two problem cases within the structured sparsity realm:

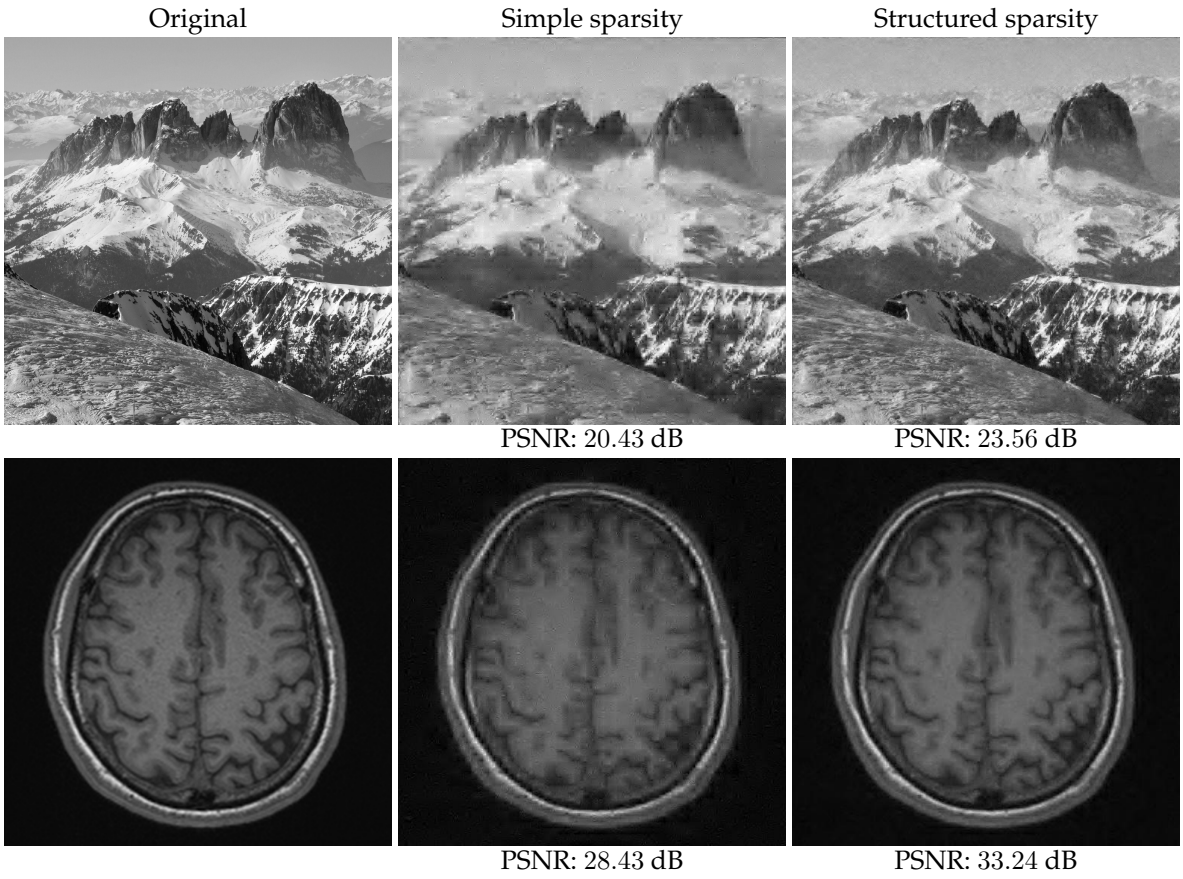


Figure 3.1: Empirical performance of simple and structured sparsity recovery on natural images. In all cases, the number of linear measurements are 5% (top row) and 10% (bottom row) of the actual image dimensions. **Left panel:** Original images of dimension: (Top row) 2048×2048, (Bottom row) 512×512. **Middle panel:** Conventional recovery using simple sparsity model. **Right panel:** Tree-structured sparse recovery.

PROBLEM 3.1. *Let $\mathbf{y} \in \mathbb{R}^n$ be a given anchor point. For an a priori known* discrete *model $\mathcal{M}_k$ with sparsity level k, we are interested in finding the best Euclidean projection on $\mathcal{M}_k$, i.e.,*

$$\mathcal{P}_{\mathcal{M}_k}(\mathbf{y}) \in \underset{\mathbf{x}:\ \mathbf{x} \in \mathcal{M}_k}{\arg\min} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{3.1}$$

PROBLEM 3.2. *Let $\mathbf{y} \in \mathbb{R}^n$ be a given anchor point. For an a priori known* discrete *model $\mathcal{M}_k$ with sparsity level k, let $g : \mathbb{R}^n \to \mathbb{R}$ be a (usually convex) function that well-approximates the behavior of $\mathcal{M}_k$. Under this setting, we are interested in finding the proximity operator*

$$\text{prox}_\lambda^g(\mathbf{y}) := \underset{\mathbf{x} \in \mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \cdot g(\mathbf{x}) \right\}, \tag{3.2}$$

where $\lambda > 0$.

Chapter roadmap

In order to better understand the impact of structured sparsity, in this chapter we analyze the connections between the discrete models and their convex relaxations, highlighting their relative advantages. We start with the general group sparse model (Section 3.2) and then elaborate on two important special cases: the dispersive (Section 3.3) and the hierarchical models (Section 3.4). For each, we present the models in their discrete nature, discuss how to solve the ensuing discrete problems and then describe convex relaxations. Further, we discuss efficient optimization solutions for structured sparsity problems and illustrate structured sparsity in action via two applications (Section 3.5).

This chapter is based on joint work with Volkan Cevher, Luca Baldassarre, Nirav Bhan, Quoc Tran-Dinh and Marwa El-Halabi [BBCK13, KC12a].

3.1 Preliminaries

We use $\mathbb{B}^n$ to represent the space of n-dimensional binary vectors and define $\iota : \mathbb{R}^n \to \mathbb{B}^n$ to be the indicator function of the nonzero components of a vector in $\mathbb{R}^n$, i.e., $\iota(\mathbf{x})_i = 1$ if $x_i \neq 0$ and $\iota(\mathbf{x})_i = 0$, otherwise. We let $\mathbb{1}_n$ to be the n-dimensional vector of all ones, $\mathbb{1}_{n,\mathcal{S}}$ the n-dimensional vector of all ones projected onto $\mathcal{S}$ and $\mathbf{I}_n$ the $n \times n$ identity matrix; we often use $\mathbf{I}$ when dimension is clear from the context. We also refer the reader to Section 6.2.

3.2 Sparse group models

We start our discussion with the *group sparse* models, i.e., models where groups of variables are either selected or discarded together [BCW10, JAB11, OJV11, RRN12, RNWK11, HZ10]. These structures naturally arise in applications such as neuroimaging [GK09b, JGM+11], gene expression data [STM+05, OJV11], bioinformatics [RBV08, ZSSL10] and computer vision [CHDB09, BCDH10]. For example, in cancer research, the groups might represent genetic pathways that constitute cellular processes. Identifying which processes lead to the development of a tumor can allow biologists to directly target certain groups of genes instead of others [STM+05]. Incorrect identification of the active/inactive groups can thus have a rather dramatic effect on the speed at which cancer therapies are developed. Figures 3.2-3.3 illustrate some more applications of group sparse models used in practice.

Such group sparsity models—denoted as $\mathfrak{G}$—feature collections of groups of variables that could overlap arbitrarily; that is $\mathfrak{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_M\}$ is a collection of M groups where each $\mathcal{G}_j$ is a subset of the index set $\mathcal{N} := \{1, \dots, n\}$. Arbitrary overlaps means that we do not restrict the intersection between any two sets $\mathcal{G}_j$ and $\mathcal{G}_\ell$ from $\mathfrak{G}$, $j \neq \ell$.

The *group-support* of $\widehat{\mathbf{x}}$ allows us to "interpret" the original signal and discover its properties so that we can, for example, target specific groups of genes instead of others [STM+05] or focus more precise imaging techniques on certain brain regions only [MGV+11]. As a result, we study under which circumstances we can correctly and tractably identify the group-support of a given signal.

We can represent a group structure $\mathfrak{G}$ as a bipartite graph, where on one side we have the n variables nodes and on the other the M group nodes. An edge connects a variable node i to a group node j if

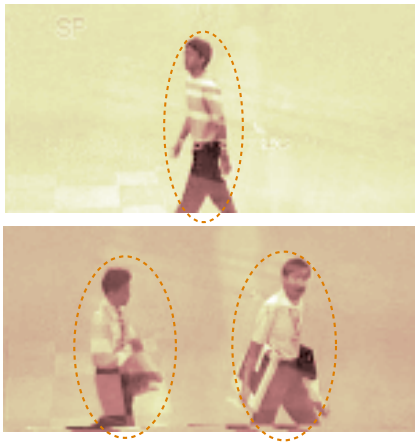


Figure 3.2: Image segmentation application: the signal of interest includes human activity, expressed in groups.

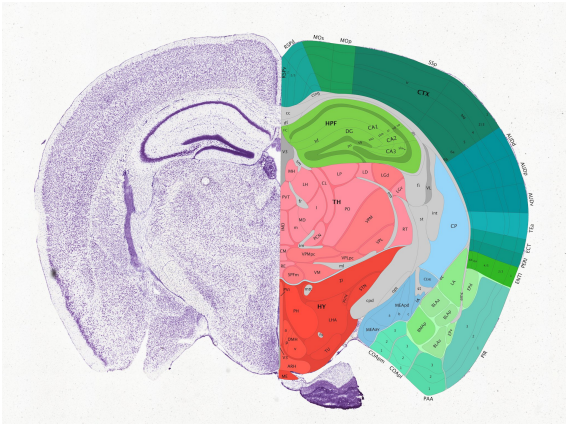


Figure 3.3: Example of a mouse brain where brain regions are represented as groups of voxels [LHA+07] (©2014 Allen Institute for Brain Science).

$i \in \mathcal{G}_j$. The bi-adjacency matrix $\mathbf{A} \in \mathbb{B}^{n \times M}$ of the bipartite graph encodes the group structure,

$$A_{ij} = \begin{cases} 1, & \text{if } i \in \mathcal{G}_j; \\ 0, & \text{otherwise.} \end{cases}$$

Given the above and for a user-defined group budget $G \in \mathbb{Z}_+$, we define the group model $\mathcal{M}_G$ as $\mathcal{M}_G := \{\bigcup_{\mathcal{G}_\ell \in \mathcal{I}} \mathcal{G}_\ell, \mathcal{I} \subseteq \mathfrak{G}, |\mathcal{I}| \leq G\}$, that is all sets of indexes that are the union of at most G groups from the collection $\mathfrak{G}$. Then, the corresponding projection operation becomes:

$$\widehat{\mathbf{x}} =: \mathcal{P}_{\mathcal{M}_G}(\mathbf{x}) \in \underset{\mathbf{z} \in \mathbb{R}^n}{\arg\min} \left\{ \|\mathbf{x} - \mathbf{z}\|_2^2 \;\middle|\; \operatorname{supp}(\mathbf{z}) \in \mathcal{M}_G \right\}. \tag{3.3}$$

Moreover, one might be only interested in identifying the *group-support* of the approximation $\widehat{\mathbf{x}}$, that is the G groups that constitute its support. We call this the group-sparse *model selection* problem.

3.2.1 The discrete model

According to (3.3), we search for $\widehat{\mathbf{x}} \in \mathbb{R}^n$ such that $\|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2$ is minimized, while $\widehat{\mathbf{x}}$ does not exceed a given group budget G. A useful notion in the group sparse model is that of the **group ℓ_0-"norm"**:

$$\|\mathbf{w}\|_{\mathfrak{G},0} := \min_{\boldsymbol{\omega} \in \mathbb{B}^M} \left\{ \sum_{j=1}^{M} \omega_j \;\middle|\; \mathbf{A}\boldsymbol{\omega} \geq \iota(\mathbf{w}) \right\}; \tag{3.4}$$

here, $\mathbf{A}$ denotes the adjacency matrix as defined in the previous subsection, the binary vector $\boldsymbol{\omega}$ indicates which groups are active and the constraint $\mathbf{A}\boldsymbol{\omega} \geq \iota(\mathbf{w})$ makes sure that, for every non-zero component of $\mathbf{w}$, there is at least one active group that covers it. Given the above definitions, the group-based signal approximation problem (3.3) can be reformulated as

$$\widehat{\mathbf{x}} \in \underset{\mathbf{w} \in \mathbb{R}^n}{\arg\min} \left\{ \|\mathbf{w} - \mathbf{x}\|_2^2 \;\middle|\; \|\mathbf{w}\|_{\mathfrak{G},0} \leq G \right\}. \tag{3.5}$$

One can easily observe that, in the case where we already know the group cover of the approximation $\widehat{\mathbf{x}}$, we can obtain $\widehat{\mathbf{x}}$ as $\widehat{\mathbf{x}}_{\mathcal{I}} = \mathbf{x}_{\mathcal{I}}$ and $\widehat{\mathbf{x}}_{\mathcal{I}^c} = 0$, where $\mathcal{I} = \bigcup_{\mathcal{G}\in\mathcal{S}^G(\widehat{\mathbf{x}})} \mathcal{G}$, with $\mathcal{S}^G(\widehat{\mathbf{x}})$ denoting the group support of $\widehat{x}$ and $\mathcal{I}^c = \mathcal{N} \setminus \mathcal{I}$. I.e., if we know the group support of the solution, the entries' values are naturally given by the anchor point $\mathbf{x}$.

To show this, we observe that [KC12a]

$$\min_{\mathbf{z}\in\mathbb{R}^n} \left\{ \|\mathbf{x}-\mathbf{z}\|_2^2 : \operatorname{supp}(\mathbf{z}) = \mathcal{I}, \mathcal{I} = \bigcup_{\mathcal{G}\in\mathcal{S}} \mathcal{G}, \mathcal{S} \subseteq \mathfrak{G}, |\mathcal{S}| \leq G \right\},$$

which can be rewritten as

$$\min_{\substack{\mathcal{S}\subseteq\mathfrak{G} \\ |\mathcal{S}|\leq G \\ \mathcal{I}=\bigcup_{\mathcal{G}\in\mathcal{S}}\mathcal{G}}} \; \min_{\substack{\mathbf{z}\in\mathbb{R}^n \\ \operatorname{supp}(\mathbf{z})=\mathcal{I}}} \|\mathbf{x}-\mathbf{z}\|_2^2 \,.$$

The optimal solution is not changed if we introduce a constant, change sign of the objective and consider maximization instead of minimization

$$\max_{\substack{\mathcal{S}\subseteq\mathfrak{G} \\ |\mathcal{S}|\leq G \\ \mathcal{I}=\bigcup_{\mathcal{G}\in\mathcal{S}}\mathcal{G}}} \; \max_{\substack{\mathbf{z}\in\mathbb{R}^n \\ \operatorname{supp}(\mathbf{z})=\mathcal{I}}} \left\{ \|\mathbf{x}\|_2^2 - \|\mathbf{x}-\mathbf{z}\|_2^2 \right\}.$$

As mentioned above, the internal maximization is achieved for $\widehat{\mathbf{x}}$ as $\widehat{\mathbf{x}}_{\mathcal{I}} = \mathbf{x}_{\mathcal{I}}$ and $\widehat{\mathbf{x}}_{\mathcal{I}^c} = 0$, so that we have, as desired,

$$\mathcal{S}^G(\hat{\mathbf{x}}) \in \operatorname*{arg\,max}_{\substack{\mathcal{S}\subseteq\mathfrak{G} \\ |\mathcal{S}|\leq G \\ \mathcal{I}=\bigcup_{\mathcal{G}\in\mathcal{S}}\mathcal{G}}} \|\mathbf{x}_{\mathcal{I}}\|_2^2 \,.$$

The following Lemma connects the group support selection as a binary problem and allows us to characterize its tractability; the proof can be found in [BBCK13]:

Lemma 18. *[BBCK13] Given $\mathbf{x} \in \mathbb{R}^n$ and a group structure $\mathfrak{G}$, the group support of the solution $\widehat{\mathbf{x}}$—denoted as $\mathcal{S}^G(\widehat{\mathbf{x}}) = \{\mathcal{G}_j \in \mathfrak{G} : \omega_j^G = 1\}$—is given by the solution $(\boldsymbol{\omega}^G, \mathbf{y}^G)$ of the following binary maximization problem:*

$$\max_{\boldsymbol{\omega}\in\mathbb{B}^M,\, \mathbf{y}\in\mathbb{B}^n} \left\{ \sum_{i=1}^n y_i x_i^2 : \mathbf{A}\boldsymbol{\omega} \geq \mathbf{y}, \sum_{j=1}^M \omega_j \leq G \right\}. \tag{3.6}$$

Moreover, the above problem is NP-hard. However, can be approximated using the greedy WMC algorithm [NWF78].

3.2.2 Convex approaches

Recent works in compressive sensing and machine learning with group sparsity have mainly focused on leveraging the group structures for lowering the number of samples required for recovering signals [SPH09, EM09, BD09b, BCDH10, RRN12, HZM11, JOV09, OJV11].

For the special case of non-overlapping groups, dubbed as the block-sparsity model, the problem of model selection does not present computational difficulties and features a well-understood theory [SPH09]. The

first convex relaxations for group-sparse approximation [YL06] considered only non-overlapping groups: the authors proposed the `Group LARS` (**L**east **A**ngle **R**egre**S**sion) algorithm to solve this problem, a natural extension of simple sparsity LARS algorithm [EHJT04]. Using the same algorithmic principles, its extension to overlapping groups [ZRY09] has the drawback of selecting supports defined as the complement of a union of groups, even though it is possible to engineer the groups in order to favor certain sparsity patterns over others [JAB11]. Eldar et al. [EM09] consider the union of subspaces framework and cast the model selection problem as a block-sparse model selection one by duplicating the variables that belong to overlaps between the groups, which is the optimization approach proposed also in [JOV09]. Moreover, [EM09] considers a model-based pursuit approach [CDS98] as potential solver for this problem, based on a predefined model $\mathcal{M}_k$. For these cases, one uses the group LASSO norm

$$\sum_{\mathcal{G}\in\mathfrak{G}} \|\mathbf{x}_{|_\mathcal{G}}\|_p \ . \tag{3.7}$$

In addition, convex proxies to the group ℓ_0-norm (3.4) have been proposed (e.g., [JOV09]) for finding group-sparse approximations of signals. Given a group structure $\mathfrak{G}$, an example generalization is defined as the *latent group LASSO*

$$\|\mathbf{x}\|_{\mathfrak{G},\{1,p\}} := \inf_{\substack{\mathbf{v}_1,\ldots,\mathbf{v}_M\in\mathbb{R}^n \\ \forall i,\,\mathrm{supp}(\mathbf{v}_i)=\mathcal{G}_i}} \left\{ \sum_{i=1}^M d_i \|\mathbf{v}_i\|_p \;\Big|\; \sum_{i=1}^M \mathbf{v}_i = \mathbf{x} \right\}, \tag{3.8}$$

where $\|\mathbf{x}\|_p = \left(\sum_{i=1}^n x_i^p\right)^{1/p}$ is the ℓ_p-norm, and d_j are positive weights that can be designed to favor certain groups over others [OJV11]. This norm can be seen as a weighted instance of the atomic norm described in [CRPW12, RRN12]; see Chapter 6.

Lemma 19. *[CRPW12, RRN12] If in* (3.8) *the weights are all equal to 1 ($d_i = 1, \forall\, i$), we have*

$$\|\mathbf{x}\|_\mathcal{A} = \|\mathbf{x}\|_{\mathfrak{G},\{1,p\}} \ .$$

The group-norm (3.8) can also be viewed as the tightest convex relaxation of a particular set function related to the *weighted set-cover* (see [OB12]). One can in general use (3.8) to find a group-sparse approximation under the chosen group norm

$$\widehat{\mathbf{x}} \in \operatorname*{arg\,min}_{\mathbf{w}\in\mathbb{R}^n} \left\{ \|\mathbf{w}-\mathbf{x}\|_2^2 : \|\mathbf{w}\|_{\mathfrak{G},\{1,p\}} \le \lambda \right\}, \tag{3.9}$$

where $\lambda > 0$ controls the trade-off between approximation accuracy and group-sparsity. However, solving (3.9) does not necessarily yield a group-support for $\widehat{\mathbf{x}}$: even though we can recover one through the decomposition $\{\mathbf{v}^j\}$ used to compute $\|\widehat{\mathbf{x}}\|_{\mathfrak{G},\{1,p\}}$, it may not be unique and when it is unique it may not capture the minimal group-cover of $\mathbf{x}$ [OJV11]. Therefore, the equivalence of ℓ_0 and ℓ_1 minimization [Don06, Can06] does not generally hold in the overlapping group-based setting.

The regularized version of problem (3.9) is equivalent to the proximity operator of $\|\mathbf{x}\|_{\mathfrak{G},\{1,p\}}$. Recently, [MVVR10, VRMV12] proposed an efficient algorithm for this proximity operator in large scale settings with extended overlap among groups. In this case, the proximity operator involves: (i) an active set preprocessing step [WN99] that restricts the proximity operations on a subset of the model—i.e., "active" groups and, (ii) a dual optimization step based on Bertsekas' projected Newton method [Ber82]; however, its convergence requires the strong regularity of the Hessian of the objective near the optimal solution.

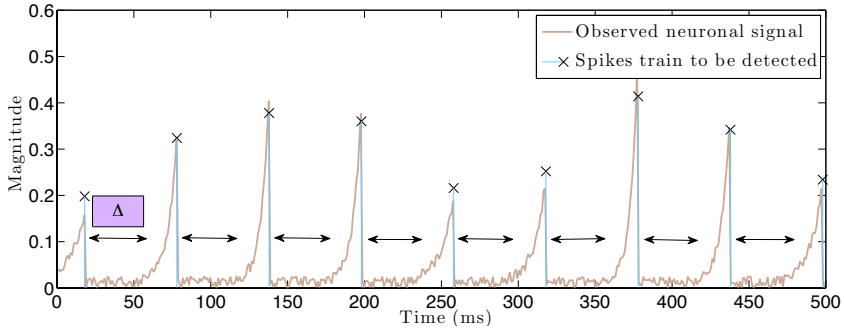


Figure 3.4: Neuronal spike train example.

3.3 Sparse dispersive models

To describe the *dispersive* structure, we motivate our discussion with an application from neurobiology. Living beings behave and function via transmission of electrical signals between electrically excitable neuronal brain cells. Such chemical "information" causes a swift change in the electrical potential of a possibly discharged neuron cell, which results in its electrical excitation. Such activity has not been fully decoded by biologists and neuroscientists, rendering the full understanding of neuronal systems as one of the important problems of our current century. However, neuronal system measurements that extract information about neuron states and how they behave under different circumstances are being performed now. As a result, in the quest for understanding the human brain, we require extensive experimental studies on animal or human brains, through signal acquisition and further signal processing. Currently, we are far from understanding the grid of neurons in its *entirety*: large-scale brain models are difficult to handle while complex neuronal signal models lead to non-interpretable results. To this end, "...we must find compromises between two seemingly mutually exclusive requirements: The model for a single neuron must be (i) computationally simple and, (ii) capable of producing rich interpretable patterns, exhibited by real biological neurons..." [Izh03].

Inspired by the statistical analysis in [GK02], the authors in [HDC09] consider a simple one-dimensional model, where the neuronal signal behaves as a train of spike signals with some *refractoriness* period $\Delta > 0$: there is a minimum nonzero time period Δ where a neuron remains inactive between two consecutive electrical excitations. In statistical terms, neuronal signals are defined by a inter-spike interval distribution that characterizes the probability a new spike to be generated as a function of the inter-arrival time. Figure 3.4 illustrates how a collection of noisy neuronal spike signals with $\Delta > 0$ might appear in practice.

3.3.1 The discrete model

Definition 8 (Dispersive model). *We define the dispersive model $\mathcal{D}_k$ in n-dimensions with sparsity level k and refractory parameter $\Delta \in \mathbb{Z}_+$ as:*

$$\mathcal{D}_k = \left\{ \mathcal{S}_q \,\middle|\, \forall q,\ \mathcal{S}_q \subseteq \mathcal{N},\ |\mathcal{S}_q| \leq k \quad \textit{and} \quad |i-j| > \Delta, \forall i \neq j \in \mathcal{S}_q \right\}, \tag{3.10}$$

i.e., $\mathcal{D}_k$ is a collection of k-sparse index subsets in $\mathcal{N}$ with distance between the indices greater than the interval Δ.

We note that if there are no constraints on the interval of consecutive spikes, the dispersive model naturally boils down to the simple sparsity model Σ_k.

Given the definition above, the projection operation is:

$$\mathcal{P}_{\mathcal{D}_k}(\mathbf{x}) \in \underset{\mathbf{w}\in\mathbb{R}^n}{\arg\min} \left\{ \|\mathbf{w}-\mathbf{x}\|_2^2 \;\middle|\; \text{supp}(\mathbf{w}) \in \mathcal{D}_k \right\}. \tag{3.11}$$

Let $\boldsymbol{\omega} \in \mathbb{B}^n$ be a *support indicator* binary vector, i.e., $\boldsymbol{\omega}$ represents the support set of a sparse vector $\mathbf{x}$ such that $\text{supp}(\boldsymbol{\omega}) = \text{supp}(\mathbf{x})$. Moreover, let $\mathbf{D} \in \mathbb{B}^{(n-\Delta+1)\times n}$ such that:

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 1 & 1 & 0 & \cdots & 0 \\ & & & & \ddots & & & & \\ 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}_{(n-\Delta+1)\times n} \tag{3.12}$$

Here, per row, there are Δ consecutive ones that denote the time interval between two potential consecutive spikes. Finally, let $\mathbf{b} \in \mathbb{R}^{n-\Delta+2}$ such that $\mathbf{b} := \begin{bmatrix} k & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}^T$.

According to [HDC09], the following linear support constraints encodes the definition of the dispersive model $\mathcal{D}_k$:

$$\mathbf{A} := \begin{bmatrix} \mathbf{1} \\ \mathbf{D} \end{bmatrix} \boldsymbol{\omega} \leq \mathbf{b}. \tag{3.13}$$

One can observe that $\mathcal{D}_k \equiv \bigcup_{\forall \boldsymbol{\omega}\in\mathfrak{Z}} \text{supp}(\boldsymbol{\omega})$, where $\mathfrak{Z} := \{\boldsymbol{\omega} \in \mathbb{B}^n : \mathbf{A}\boldsymbol{\omega} \leq \mathbf{b}\}$. Consequently, (3.11) becomes:

$$\mathcal{P}_{\mathcal{D}_k}(\mathbf{x}) \in \underset{\mathbf{w}\in\mathbb{R}^n}{\arg\min} \left\{ \|\mathbf{w}-\mathbf{x}\|_2^2 \;\middle|\; \mathbf{A}\cdot\text{supp}(\mathbf{w}) \leq \mathbf{b} \right\}. \tag{3.14}$$

A key observation is given in the next lemma.

Lemma 20. *[HDC09] Given the problem setting above, it is easy to observe that* (3.14) *has solution* $\mathcal{P}_{\mathcal{D}_k}(\mathbf{x})$ *such that* $\mathcal{S} := supp\left(\mathcal{P}_{\mathcal{D}_k}(\mathbf{x})\right)$ *and* $\left(\mathcal{P}_{\mathcal{D}_k}(\mathbf{x})\right)_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}$ *where:*

$$\mathcal{S} \in supp\left(\underset{\boldsymbol{\omega}\in\mathbb{B}^n:\ \mathbf{A}\boldsymbol{\omega}\leq\mathbf{b}}{\arg\max} \left\{ \mathbf{c}^T\boldsymbol{\omega} \right\} \right), \quad \textit{where } \mathbf{c} := \begin{bmatrix} x_1^2 & x_2^2 & \cdots & x_n^2 \end{bmatrix}^T, \tag{3.15}$$

i.e., we target to capture most of the signal's $\mathbf{x}$ *energy, given structure* $\mathcal{D}_k$*. To solve* (3.15)*, the authors in [HDC09] identify that the binary integer program* (3.15) *is identical to the solution of the linear program, obtained by relaxing the integer constraints into continuous constraints.*

Lemma 20 indicates that (3.14) can be efficiently performed using linear programming tools [BV04]. Once (3.14) is relaxed to a convex problem, decades of knowledge on convex analysis and optimization can be leveraged. Interior point methods find a solution with fixed precision in polynomial time but their complexity might be prohibitive even for moderate-sized problems.

3.3.2 Convex approaches

The constraint matrix $\mathbf{D}$ describes a *collection of groups*, where each group is assumed to have at most one non-zero entry to model the refractoriness property.[1] Moreover, these groups are *overlapping* which aggrandizes the "clash" between neighboring groups: a non-zero entry in a group discourages every other overlapping group to have a distinct non-zero entry.

In mathematical terms, each row i of $\mathbf{D}$ defines a group $\mathcal{G}_i$ such that $\mathcal{G}_i = \text{supp}(\mathbf{d}_i) \subseteq \mathcal{N}$ where $\mathbf{d}_i$ denotes the i-th row of $\mathbf{D}$, $\forall i \in \{1, \dots, M := n - \Delta + 1\}$:

$$\mathbf{D} = \begin{bmatrix} \overbrace{\quad}^{\Delta} & & & \\ \mathcal{G}_1 & & & \\ & \mathcal{G}_2 & & \\ & & \ddots & \\ & & & \mathcal{G}_M \end{bmatrix}$$

Given such group structure, the dispersive model is characterized both by *inter-group* and *intra-group* properties:

- *Intra-group sparity*: we desire $\|\mathbf{D}\boldsymbol{\omega}\|_\infty \leq 1$, i.e., per refractoriness period of length Δ, we require only one "active" spike.
- *Inter-group exclusion*: due to the refractoriness property, the activation of a group implies the deactivation of its closely neighboring groups.

While the sparsity level within a group can be easily "convexified" using standard ℓ_1-norm regularization, the dispersive model further introduces the notion of *inter-group exclusion*, which is highly *combinatorial*. However, one can relax it by introducing *competitions* among variables in overlapping groups: variables that have a "large" neighbor should be penalized more than variables with "smaller" neighbors.

In this premise and based on [ZJH10], we identify the following family of norms[2]:

$$\Omega_{\text{exclusive}}(\mathbf{x}) = \sum_{\mathcal{G}_i} \left(\sum_{j \in \mathcal{G}_i} |x_j| \right)^p, \quad p = 2, 3, \dots, \tag{3.17}$$

as convex regularizers that imitate the dispersive model. In (3.17), $\left(\sum_{j\in\mathcal{G}_i} |x_j|\right) := \|\mathbf{x}_{\mathcal{G}_i}\|_1$ promotes sparsity within each group $\mathcal{G}_i$, while the outer sum over groups $\sum_{\mathcal{G}_i} \|\mathbf{x}_{\mathcal{G}_i}\|_1^p$ imposes sparsity over the number of groups that are activated. Observe that for $p = 1$, (3.17) becomes the standard ℓ_1-norm over $\mathcal{N}$. Notice that the definition of the overlapping groups (instead of non-overlapping) is a key property for capturing the discrete structure: variables belonging to overlapping groups are weighted differently

[1] Other convex structured models that can be described as the composition of a simple function over a linear transformation $\mathbf{D}$ can be found in [AMP+11].

[2] The proposed norm originates from the composite absolute penalties (CAP) convex norm, proposed in [ZRY09], according to which:

$$g(\mathbf{x}) = \sum_{\mathcal{G}_i} \left(\sum_{j \in \mathcal{G}_i} |x_j|^\gamma \right)^p, \tag{3.16}$$

for various values of γ and p. Observe that this model also includes the famous group sparse model where $g(\mathbf{x}) = \sum_{\mathcal{G}_i} \|\mathbf{x}_{\mathcal{G}_i}\|_2$, described in Section 3.2, for $p = 1/2$ and $\gamma = 2$.

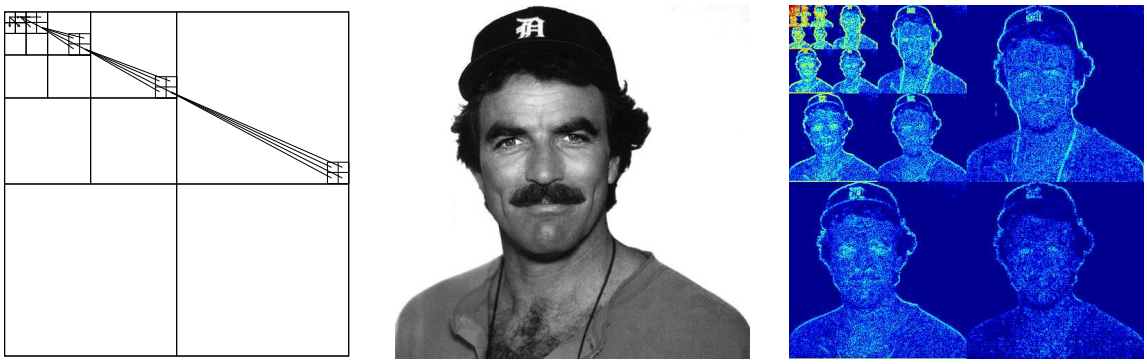


Figure 3.5: Wavelet coefficients naturally cluster along a rooted connected subtree and tend to decay towards the leaves. (Left) Example of wavelet tree for a 32 x 32 image. The root of the tree is the top-left corner and there are three regular subtrees related to horizontal, vertical and diagonal details. Each node is connected to four children representing detail at a finer scale. (Centre) Grayscale 512 x 512 image. (Right) Wavelet coefficients for the image at center. Best viewed in color, dark blue represents values closer to zero.

when considered parts of different groups. This leads to variable "suppression" (i.e., thresholding) of elements, depending on the "weight" of their neighborhood within the groups they belong to.

3.4 Hierarchical sparse models

Hierarchical structures are found in many signals and applications. For example, the wavelet coefficients of images are naturally organized on regular quad-trees to reflect their multi-scale structure; see Figure 3.5 and [Sha93, CNB98, Mal99, Bar99, BDKY02, HC09, ZRY09, BCDH10, HZM11]; gene networks are described by a hierarchical structure that can be leveraged for multi-task regression [KX10]; hierarchies of latent variables are typically used for deep learning [Ben09].

In essence, a hierarchical structure defines an ordering of importance among the elements (either individual variables or groups of them) of a signal with the rule that an element can be selected only after its ancestors. Such structured models result into more robust solutions and allow recovery with far fewer samples. In compressive sensing, assuming that the signal possesses a hierarchical structure with sparsity k leads to improved sample complexity bounds of the order of $O(k)$ for dense measurement matrices [BCDH10], compared to the bound of $O(k \log(n/k))$ for standard sparsity. Also in the case of sparse measurement matrices, e.g. expanders, hierarchical structures yield improved sample complexity bounds [IR13, BBC14].

3.4.1 The discrete model

Definition 9 (Hierarchical model)**.** *Let $\mathcal{T}$ denote an arbitrary* tree *or* forest *representation over the variables in a set $\mathcal{N}$. We define a k rooted connected (RC) subtree $\mathcal{S}$ with respect to $\mathcal{T}$ as a collection of k variables in $\mathcal{N}$ such that $v \in \mathcal{S}$ implies $\mathcal{A}(v) \in \mathcal{S}$, where $\mathcal{A}(v)$ is the set that contains all the ancestors of the node v. The hierarchical model of budget k, $\mathcal{T}_k$ is the set of all k rooted-connected subtrees of $\mathcal{T}$.*

Given a tree $\mathcal{T}$, the rooted connected approximation is the solution of the following discrete problem

$$\mathcal{P}_{\mathcal{T}_k}(\mathbf{x}) \in \operatorname*{arg\,min}_{\mathbf{z}\in\mathbb{R}^n} \left\{ \|\mathbf{x}-\mathbf{z}\|_2^2 \;\middle|\; \operatorname{supp}(\mathbf{z}) \in \mathcal{T}_k \right\}. \tag{3.18}$$

which can be reformulated as follows

$$\widehat{\mathbf{y}} \in \operatorname*{arg\,max}_{\mathbf{y}\in\mathbb{B}^n} \left\{ \sum_{i=1}^{N} y_i x_i^2 : \mathbf{y} \in \mathcal{T}_k \right\}, \tag{3.19}$$

where $\mathbf{y}$ is a binary vector with k non-zero components that indicates which components of $\mathbf{x}$ are selected. Given a solution $\widehat{\mathbf{y}}$ of the above problem, a solution $\widehat{\mathbf{z}}$ of (3.18) is then obtained as $\widehat{\mathbf{z}}_{|\mathcal{S}} = \mathbf{x}_{|\mathcal{S}}$ and $\widehat{\mathbf{z}}_{|\mathcal{S}^c} = 0$, where $\mathcal{S} = \operatorname{supp}(\widehat{\mathbf{y}})$.

This type of constraint can be represented by a group structure with an overall sparsity constraint k, where for each node in the tree we define a group consisting of that node and all its ancestors. When a group is selected, we require that all its elements are selected as well. Problem (3.19) can then be cast as a special case of the Weighted Maximum Coverage problem (3.6). Fortunately, this particular group structure leads to tractable solutions.

Indeed (3.19) can be solved exactly via a dynamic program that runs in polynomial time [CT13, BBCK13]. For d-regular trees, that is trees for which each node has d children, the algorithm in [BBCK13] has complexity $\mathcal{O}(nkd)$.

3.4.2 Convex approaches

The hierarchical structure can also be enforced by convex penalties, based on groups of variables. Given a tree structure $\mathcal{T}$, define groups consisting of a node and all its descendants and let $\mathfrak{G}_{\mathcal{T}}$ represent the set of all these groups. Based on this construction, the hierarchical group LASSO penalty [ZRY09, KX10, JMOB11] imitates the hierarchical sparse model and is defined as follows

$$\Omega(\mathbf{x})_{\mathrm{HGL}} = \sum_{\mathcal{G}\in\mathfrak{G}_{\mathcal{T}}} w_{\mathcal{G}} \|\mathbf{x}_{|\mathcal{G}}\|_p \tag{3.20}$$

where $p \geq 1$, $w_{\mathcal{G}}$ are positive weights and $\mathbf{x}_{|\mathcal{G}}$ is the restriction of $\mathbf{x}$ to the elements contained in $\mathcal{G}$. Since the nodes lower down in the tree appear in more groups than their ancestors, they will contribute more to $\Omega(\mathbf{x})_{\mathrm{HGL}}$ and therefore will be more easily encouraged to be zero. The proximity operator of Ω_{HGL} can be computed exactly for $p = 2$ and $p = \infty$ via an active set algorithm [JMOB11].

Other convex penalties have been recently proposed in order to favor hierarchical structures, but also allowing for a certain degree of flexibility in deviations from the discrete model. One approach considers groups consisting of all parent-child pairs and uses the latent group LASSO penalty (see Section 3.2.2) in order to obtain solutions whose support is the union of few such pairs [RNWK11], see Figure 3.6 (left).

An interesting extension is given by the *family* model [BBC13, ZSY13], where the groups consist of a node and all its children, see Figure 3.6 (right). Again the latent groups LASSO penalty is used. This model is better suited for wavelet decomposition of images because it better reflects the fact that a large coefficient value implies large coefficients values for all its children at a finer scale.

For both these cases, one can use the duplication strategy to transform the overlapping proximity problem

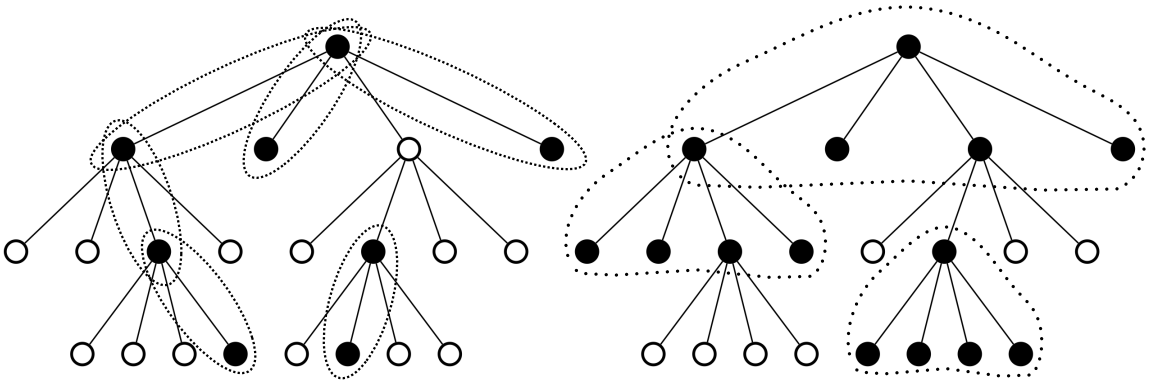


Figure 3.6: Examples of parent-child and family models. Active groups are indicated by dotted ellipses. The support (black nodes) is given by the union of the active groups. (Left) Parent-child model. (Right) Family model.

into a block one, which can be efficiently solved in closed-form [JOV09].

3.5 Applications

Here, our intention is to present an overview of the dominant approaches is structured sparse recovery followed in practice. We consider the following three general optimization formulations:[3]

- *Discrete projection formulation:* Given a signal model $\mathcal{M}_k$, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed convex data fidelity/loss function. Here, we focus on the *projected* non-convex minimization problem:
$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{M}_k. \tag{3.21}$$

- *Convex proximity formulation:* Given a signal model $\mathcal{M}_k$, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed convex data fidelity/loss function, $g : \mathbb{R}^n \to \mathbb{R}$ a closed convex regularization term, possibly non-smooth, that faithfully models $\mathcal{M}_k$ and $\lambda > 0$. In this chapter, we focus on the convex composite minimization problem:
$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}). \tag{3.22}$$

- *Convex structured-norm minimization:* Given a signal model $\mathcal{M}_k$, let $g : \mathbb{R}^n \to \mathbb{R}$ be a closed convex regularization term, possibly non-smooth, that faithfully models $\mathcal{M}_k$. Moreover, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed convex data fidelity/loss function and $\sigma > 0$. We consider the following minimization problem:
$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\mathbf{x}) \quad \text{subject to} \quad f(\mathbf{x}) \leq \sigma. \tag{3.23}$$

[3] We acknowledge that there are other criteria that can be considered in practice; for completeness, in the simple sparsity case, we refer the reader to the ℓ_1-norm constrained linear regression (a.k.a. LASSO [Tib96])—similarly, there are alternative optimization approaches for the discrete case [WNF09]. However, our intention in this chapter is to use the most prevalent structured-sparsity formulations used in practice.

3.5.1 Compressive Imaging

Natural images are usually sparse in wavelet basis. In this experiment, we study the image reconstruction problem from compressive measurements, where structured sparsity ideas are applied in practice.

For this purpose and given a $p \times p$ natural grayscale image $\mathbf{x} \in \mathbb{R}^{p^2}$, we use the Discrete Wavelet Transform (DWT) with $\log_2(p)$ levels, based on the Daubechies 4 wavelet, to represent $\mathbf{x}$; see the Wavelet representation of two images in Figures 3.7-3.8. In math terms, the DWT can be represented by an operator matrix $\mathbf{W}^\top$, so that $\mathbf{x}$ can be sparsely represented (or well-approximated) as $\mathbf{x} = \mathbf{W}\mathbf{c}$, where $\mathbf{c} \in \mathbb{R}^n$, $n := p^2$ are the wavelet coefficients for $\mathbf{x}$.

To exploit this fact in practice, we consider the problem of recovering $\mathbf{x} \in \mathbb{R}^n$ from m compressive measurements $\boldsymbol{y} \in \mathbb{R}^m$. The measurements are obtained by applying a sparse matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ to the vectorized image such that:

$$\boldsymbol{y} = \mathbf{A}\mathbf{x}.$$

Here, $\mathbf{A}$ is the adjacency matrix of an expander graph of degree $d = 8$, so that $\|\mathbf{A}\|_0 = dn$; c.f., [BGI+08]. Thus, the overall measurement operator on the wavelet coefficients is then given by the concatenation of the expander matrix with the DWT: $\boldsymbol{y} = \mathbf{A}\mathbf{W}\mathbf{c}$, with $\mathbf{c} \approx \widehat{\mathbf{c}}$ with $\|\widehat{\mathbf{c}}\|_0 \ll n$, i.e., $\mathbf{x}$ can be well-approximated by using only a limited number of wavelet coefficients.

We use the following methods for recovering $\mathbf{c}$ from the measurements $\boldsymbol{y}$:

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\boldsymbol{y} - \mathbf{A}\mathbf{W}\mathbf{c}\|_2^2 \\ \text{subject to} \quad & \operatorname{supp}(\mathbf{c}) \in \mathcal{T}_k. \end{aligned} \qquad \text{(Rooted Connected Tree model (RC))}$$

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\mathbf{c}\|_1 \\ \text{subject to} \quad & \boldsymbol{y} = \mathbf{A}\mathbf{W}\mathbf{c}. \end{aligned} \qquad \text{(Basis Pursuit (BP))}$$

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\mathbf{c}\|_{\text{HGL}} \\ \text{subject to} \quad & \boldsymbol{y} = \mathbf{A}\mathbf{W}\mathbf{c}. \end{aligned} \qquad \text{(Hierarchical Group LASSO (HGL) pursuit)}$$

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\mathbf{c}\|_{\text{PC}} \\ \text{subject to} \quad & \boldsymbol{y} = \mathbf{A}\mathbf{W}\mathbf{c}. \end{aligned} \qquad \text{(Parent-Child Latent Group LASSO (PC) pursuit)}$$

$$\begin{aligned} \underset{\mathbf{c}\in\mathbb{R}^n}{\text{minimize}} \quad & \|\mathbf{c}\|_{\text{FAM}} \\ \text{subject to} \quad & \boldsymbol{y} = \mathbf{A}\mathbf{W}\mathbf{c}. \end{aligned} \qquad \text{(Family Latent Group LASSO (FAM) pursuit)}$$

The RC model is solved via the improved projected gradient descent given in [KC11] with the projections computed via the dynamic program proposed in [BBCK13]. All the remaining methods are solved using the primal-dual method described in [TDC14] which relies on the proximity operator of the associated structure-sparsity inducing penalties. For BP the proximity operator is given by the standard soft-thresholding function. For HGL, we use the algorithm and code given by [JMOB11]. For the latent group LASSO approaches, PC and FAM, we adopt the duplication strategy proposed in [JOV09, OJV11], for which the proximity operator reduces to the standard block-wise soft-thresholding on the duplicated

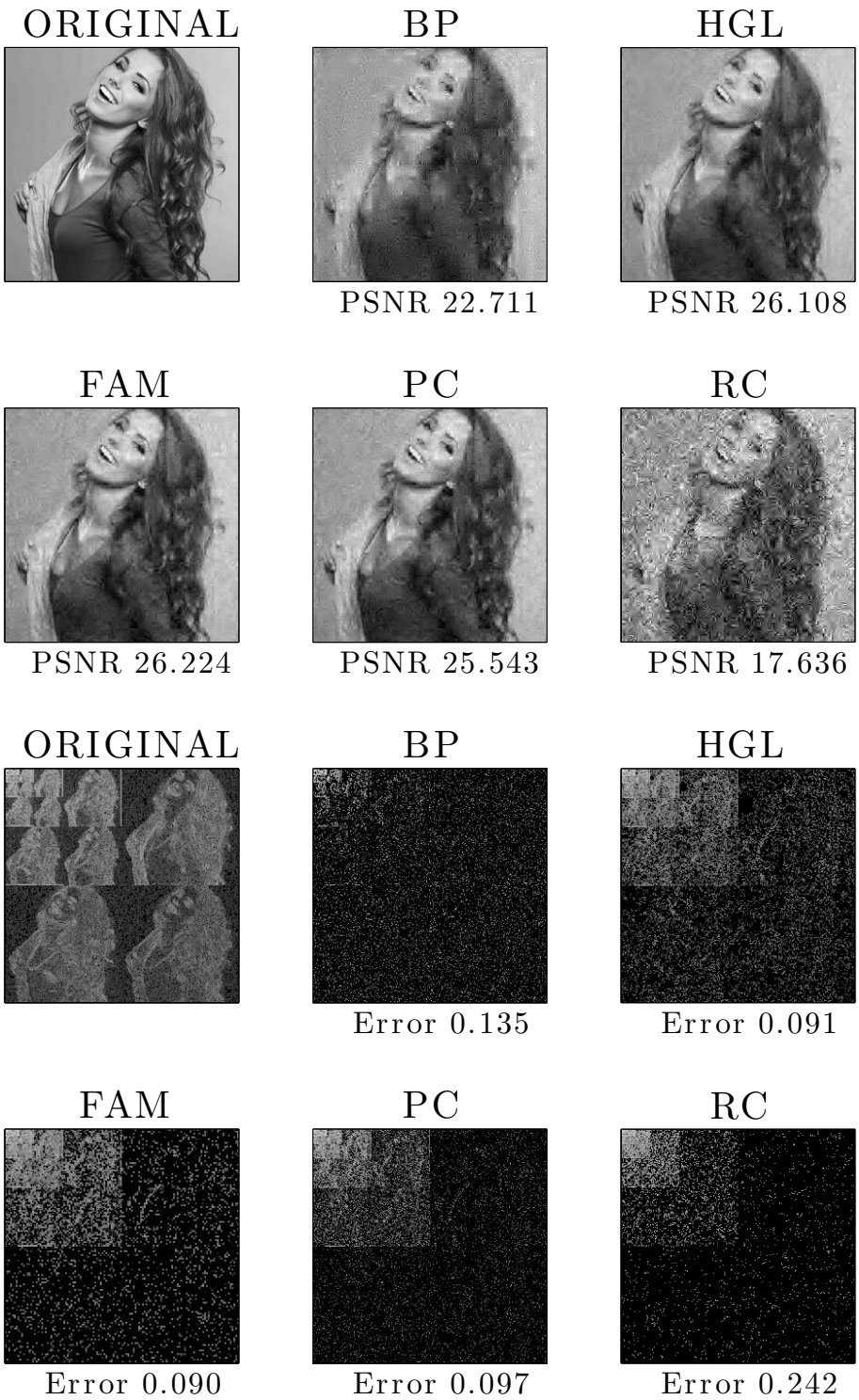


Figure 3.7: Woman image recovery performance from compressive measurements. Here, $p = 256$. The top two rows show the reconstruction performance in the original domain, along with the PSNR levels achieved. The bottom two rows show the corresponding representations into the wavelet domain.

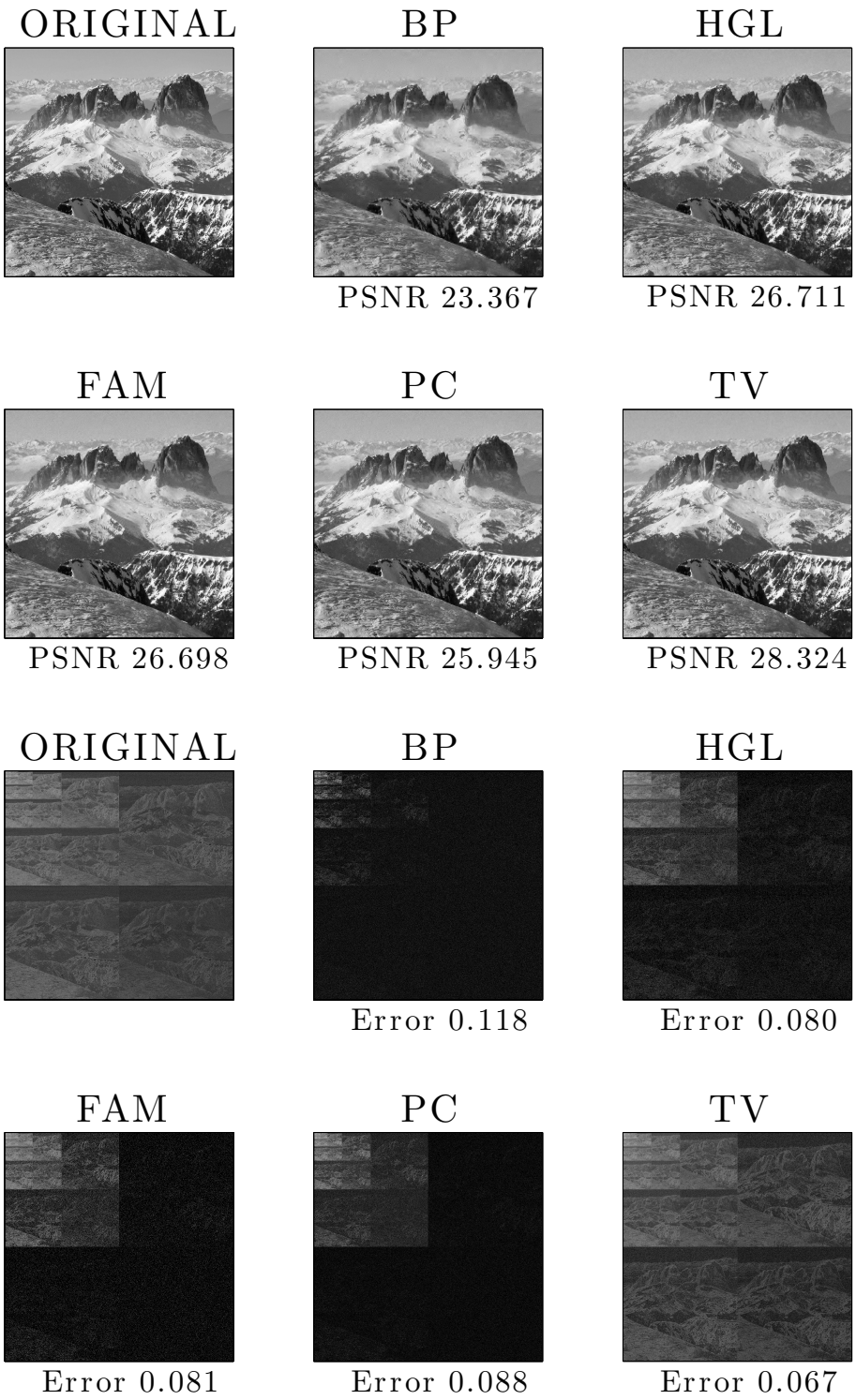


Figure 3.8: Mountains image recovery performance from compressive measurements. Here, $p = 2048$. The top two rows show the reconstruction performance in the original domain, along with the PSNR levels achieved. The bottom two rows show the corresponding representations into the wavelet domain.

variables. All algorithms are written in `Matlab`, except for the HGL proximity operator and the RC projection that are in `C`.

The duplication approach consists in creating a latent vector that contains copies of the original variables. The number of copies is determined by the number of groups that a given variable belongs to. For the Parent-Child model, each node belongs to the four groups that contain each of its children and the group that contains its father. The root has only three children, corresponding to the roots of the horizontal, vertical and diagonal wavelet trees. The leaves belong only to the group that contains their fathers. A simple calculation shows that the latent vector for the PC model contains $2(n-1)$ variables.

For the Family model, instead, each node belongs to only two groups: the group containing its children and the group containing its siblings and its father. Each leaf belongs to only the group that contains its siblings and its father. Overall, the number of variables in the latent vector for the Family models is equal to $\frac{5n}{4} - 1$.

The duplication approach does not significantly increase the problem size, while it allows an efficient implementation of the proximity operator. Indeed, given that the proximity operator can be computed in closed form over the duplicated variables, this approach is as fast as the hierarchical group LASSO one, where the proximity operator is computed via `C` code.

In order to obtain a good performance, both the parent-child and the family model require a proper weighting scheme to penalize groups lower down in the tree, where smaller wavelet coefficients are expected, compared to nodes closer to the root, which normally carry most of the energy of the signals and should be penalized less. We have observed that setting the group weights proportional to the level L of the node of the group closest to the root gives good results. In particular, we set the weights equal to L^2, with 0 being the root level.

Results: We performed the compressive imaging experiments on both a 256×256 portrait of a woman and a 2048×2048 mountain landscape. Apart from conversion to grayscale and resizing through the Matlab function `imresize`, no preprocessing has been carried out. The primal-dual algorithm of [TDC14] has been run up to precision 10^{-5}. We measure the recovery performance in terms of Power Signal to Noise Ratio (PSNR) and relative recovery error ℓ_2 norm as $\frac{\|\widehat{\mathbf{x}}-\mathbf{x}\|_2}{\|\mathbf{x}\|}$, where $\widehat{\mathbf{x}}$ is the estimated image and $\mathbf{x}$ is the true image.

Figures 3.7-3.8 report the recovery results, using $m = \frac{n}{8}$, that is using only 12.5% samples compared to the ambient dimension. The estimated images are on the top two rows, while the third and fourth rows show the estimated wavelet coefficients.

The effect of imposing structured sparsity can be clearly seen for the HGL, PC and FAM models, where the high values of the coefficients tend to cluster around the root of the wavelet tree (i.e., top-left corner of the image) and their intensity decreases descending the tree. The family model shows the grouping among the siblings, where four leaves are either all zero or all non-zero. For the 256×256 image, despite being coded in `C`, the discrete model is approximately 160 times slower than the other methods, which are computationally equivalent: e.g., in our tests, the family model took around 60 seconds, while the RC one required almost 2 hours. We therefore did not use the RC model on the larger 2048×2048 mountain image, but we compared also against Total Variation (TV) pursuit, which obtains the best performance on this image.

3.5.2 Neuronal spike detection from compressed data

In the experiments that follow, we compare the performance of the following three optimization criteria, assuming the dispersive model $\mathcal{D}_k$.

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{D}_k. \tag{Discrete dispersive}$$

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + \lambda \cdot \Omega_{\text{exclusive}}(\mathbf{x}). \tag{Exclusive norm regularization}$$

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad \Omega_{\text{exclusive}}(\mathbf{x}) \quad \text{subject to} \quad f(\mathbf{x}) \leq \sigma^2. \tag{Exclusive norm pursuit}$$

Empirical performance on synthetic data: Figures 3.9-3.10 illustrate the utility of each approach in the compressed sensing setting where $f(\mathbf{x}) := \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$. That is, we observe $\mathbf{x}^\star \in \mathbb{R}^n$ through a limited set of linear sketches $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star + \boldsymbol{\varepsilon} \in \mathbb{R}^m$ where $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ is a known linear sketch matrix. Here, we assume $n = 500$ and $m = 70$ for $\|\mathbf{x}^\star\|_0 = 25$. Without loss of generality, we assume $(\mathbf{x}^\star)_i \geq 0,\ \forall i$ and $\Delta^\star = 20$.

In the discrete case, we relax the refractory period Δ to model signal structure deviations; here, we assume $\Delta = 15$. The discrete exclusive model [BCDH10, NT09a] clearly outperforms the rest of the approaches under comparison; such behavior is also observed on average over the set of experiments conducted (Figure 3.9). This also implies that the discrete model usually requires fewer measurements for accurate recovery compared to conventional sparse approximation, as long as the underlying signal approximately follows $\mathcal{D}_k$.

On the other hand, due to convex relaxations, convex approaches introduce unnecessary nonzero coefficients that do not comply with the underlying model. However, both approaches show good performance in recovering $\mathbf{x}^\star$ from limited measurements; see Figure 3.10.

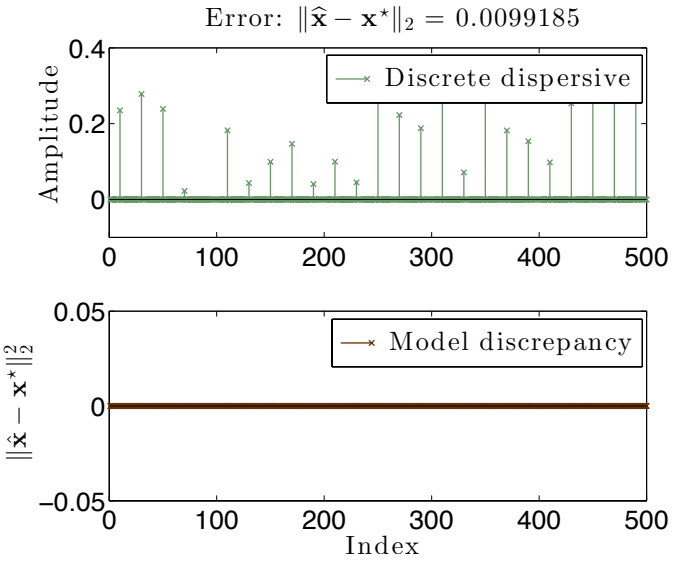
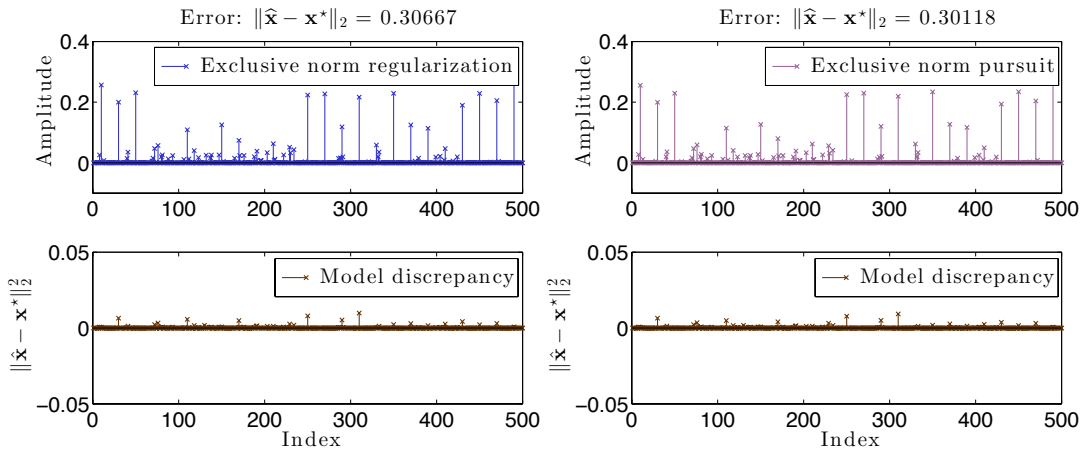


Figure 3.9: Performance of the discrete dispersive approach for the problem spike train recovery from a limited set of linear measurements.

Real neuronal spike data: In order to understand the functioning of the human brain, it is necessary to identify and study the behavior of neuronal cell membranes under rapid change in the electric potential. However, to observe such phenomena, electrical activities on neurons need to be recorded using specialized equipment. In this experiment, we perform somatic spike detection of a tufted L5 pyramidal cell responding to in-vivo-like current injected in the apical dendrites and the soma simultaneously (see [Fac09] for the experimental details).

A snapshot of the neuronal spikes waveforms is shown in Figure 3.11. In order to accurately detect the neuronal spikes, a high-frequency sample acquisition equipment is required. Within this context, we apply CS ideas to decrease the number of samples needed to approximately detect the *positions* of the spike train. Let $\mathbf{x}^\star \in \mathbb{R}^n$ with $n = 832$ represent the signal in Figure 3.11a; furthermore, let $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ be the sensing matrix where $m = 0.25 \cdot n$, *i.e., we perform a* 75% *compression.* We use the proposed schemes to



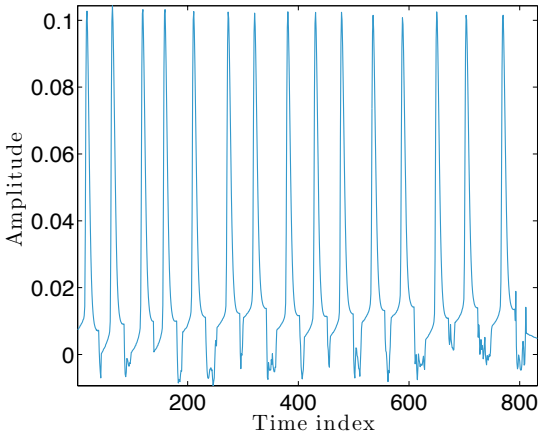
(a) Exclusive norm regularization (b) Exclusive norm pursuit

Figure 3.10: Performance of dispersive convex relaxations for the problem of spike train recovery from a limited set of linear measurements. (**Left panel:**) Exclusive norm regularization approach. (**Right panel:**) Exclusive norm pursuit approach.

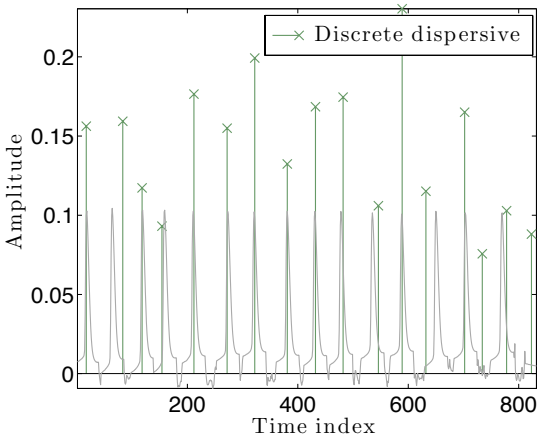
recover the locations of the neuronal spikes under the assumption of the dispersive model with refractory period Δ. Here, Δ is set equal to the *average* period between two consecutive spikes.

Figure 3.11b represents the recovered k-sparse approximation using the discrete dispersive model $\mathcal{D}_k$. Here, k is set to the number of spikes expected to appear for a given time period—such number can be easily deduced by observing the behavior of a specific neuron type. From Figure 3.11b, we observe that the discrete model approximates the locations of the spikes quite accurately: most of the spike locations are exactly recovered. However, due to the "strictness" of the discrete model, we observe that small deviations from $\mathcal{D}_k$ lead to imprecise estimations; e.g., between the 12^{th} and 13^{th} spike of the sequence, a larger (than usual) refractory period is observed that leads to mis-location of the next spike estimation.

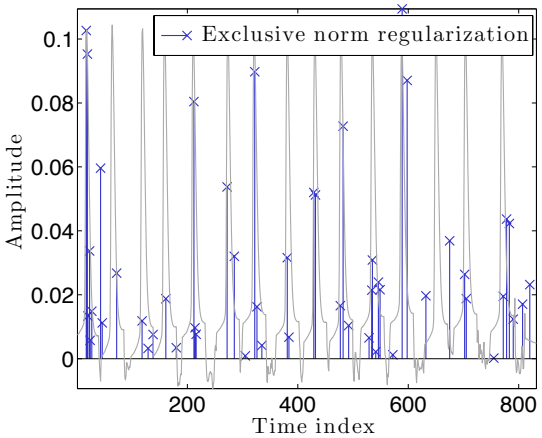
Figures 3.11c-3.11d depict the performance of convex solvers using the exclusive norm as (i) regularizer and (ii) objective function. Tweaking the λ parameter in the (i) case, one can achieve *sparse* solutions that approximate the underlying model (Figure 3.11c); however, one can observe multiple detected spikes with separation less than Δ, violating the assumed model. In the model-based Basis pursuit case, the solver tries to *fit* the solution to the data, which usually leads to less sparse solutions (Figure 3.11d). One can further sparsify the convex solutions to obtain a k-sparse answer as in Figures 3.11e-3.11f: however, in most cases, further processing of the returned signal is required to maintain a $\mathcal{D}_k$-modeled solution. E.g., in this case, due to the fact that convex norms force the solution to fully *explain* the observations, the sparsified solution includes more than one spike per true spike location.



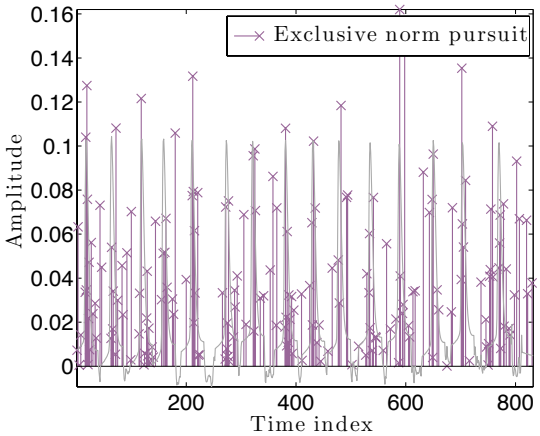
(a) Original neuronal spike train signal.



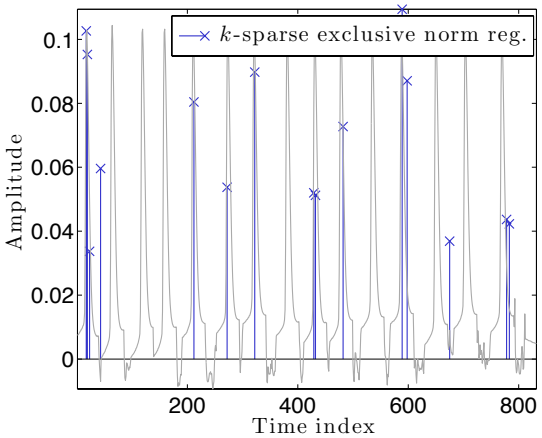
(b) Spike detection using the discrete model.



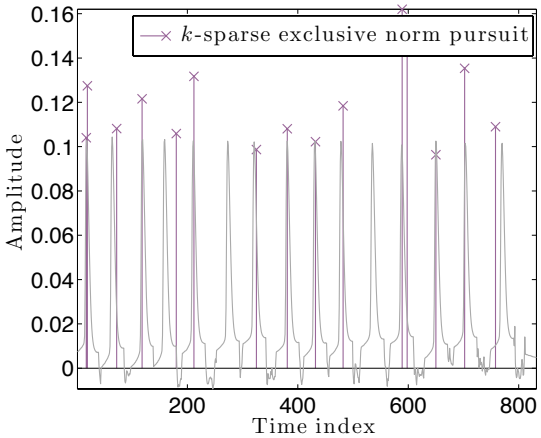
(c) Spike detection using the exclusive norm-regularized convex approach.



(d) Spike detection using the Model-based Basis pursuit.



(e) k-sparse approximation of the exclusive norm-regularized convex solution.



(f) k-sparse approximation of the model-based Basis pursuit solution.

Figure 3.11: Spike detection in real neuronal data using the dispersive model. Figure 3.11a depicts the original signal $\mathbf{x}^\star$. We observe $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star$ using only 25% measurements through a linear sketch $\boldsymbol{\Phi}$. Figures 3.11b-3.11d illustrate the performance of the three approaches under comparison. Figures 3.11e-3.11f show the convex solutions, sparsified to be k-sparse.

3.6 Discussion

To summarize, recent results in CS extend the simple sparsity idea to more sophisticated *structured* sparsity models, which describe the interdependency between the nonzero components of a signal, allowing to increase the interpretability of the results and lead to better recovery performance. In order to better understand the impact of structured sparsity, in this chapter we analyze the connections between the discrete models and their convex relaxations, highlighting their relative advantages. We start with the general group sparse model and then elaborate on two important special cases: the dispersive and the hierarchical models. For each, we present the models in their discrete nature, discuss how to solve the ensuing discrete problems and then describe convex relaxations.

For most of our discussions in the CS context, we make *no assumption* about the sensing matrix and what are the consequences of such selection in the recovery performance under structured signal assumptions. An important class of matrices that have both strong theoretical guarantees and practical importance is the class of *expander matrices*, i.e., binary matrices of specific construction; for more information, see [BGI+08]. However, most structured sparsity approaches proposed for signal recovery using expander matrices are non-convex; c.f., [Pri11, IR13, BBC14]. Moreover, the majority of these approaches come with recovery guarantees quantified using either the ℓ_2-norm or the ℓ_1-norm, i.e., for arbitrary $\mathbf{x}^\star \in \mathbb{R}^n$

$$\|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_\# \leq \rho \|\mathbf{x}_k - \mathbf{x}^\star\|_\# + \gamma \|\boldsymbol{\varepsilon}\|_2, \qquad \# = 1 \text{ or } 2,$$

where $\widehat{\mathbf{x}}$ is the approximation for $\mathbf{x}^\star$ and $\mathbf{x}_k$ represents the best k-sparse approximation of $\mathbf{x}^\star$.

Normally though, in the convex case, the most natural norm to express the error in is the corresponding structured norm $\|\cdot\|_{\mathcal{A}}$ that well-approximates the underlying signal model; here, e.g., $\mathcal{A}$ may represent the atomic or generic structured norm. In the case of *Group ℓ_1-norm*, we have $\|\cdot\|_{\mathcal{A}} \equiv \|\cdot\|_{2,1}$.

The discussion in this chapter leads to the following open problem:

Open question 5. *Using expander matrices in the structured sparsity CS framework, an open question is to provide approximation guarantees of the form:*

$$\|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_{\mathcal{A}} \leq \rho \|\mathbf{x}_k - \mathbf{x}^\star\|_{\mathcal{A}} + \gamma \|\boldsymbol{\varepsilon}\|_2,$$

where $\mathcal{A}$ represents the structured atomic norm used in the convex criterion:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{x}\|_{\mathcal{A}} \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}. \end{aligned}$$

Can we obtain such recovery conditions using specialized sensing matrices, such as expanders?

4 Greedy methods for affine rank minimization

Introduction

In (2.2), we easily observe that the minimization problem can be equivalently rewritten as:

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X})\|_2^2 \\ \text{subject to} \quad & \operatorname{rank}(\mathbf{X}) \leq k, \\ & \mathbf{X} \in \mathbb{D}^n, \end{aligned} \tag{4.1}$$

where $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n \times n} \to \mathbb{R}^m$ is a generic linear map such that, given $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, $\boldsymbol{\mathcal{A}}(\mathbf{X}) = \boldsymbol{\Phi}\mathbf{x}$ for $\mathbf{X} \in \mathbb{D}^n$ a diagonal matrix with $\mathbf{x} \in \mathbb{R}^n$ on the main diagonal. Here, the solution $\mathbf{X}^\star$ is forced to be rank-k, due to the implicit sparsity constraints on its diagonal. In other words, the solution of (4.1) contains on its diagonal the vector solution of (2.2). However, (4.1) constitutes only a special case of the *affine rank minimization* (ARM) problem as described below, appearing in many applications; low-dimensional Euclidean embedding [BCW10], matrix completion [CR09], image compression [JMD10] just to name a few.

PROBLEM 4.1: *Let $\mathbf{X}^\star$ be a rank-r, $p \times n$ matrix of interest, where $r \ll min\{p, n\}$. We desire to reconstruct $\mathbf{X}^\star$ through a low-dimensional observation vector $\boldsymbol{y} \in \mathbb{R}^m$ $(m < pn)$ where:*

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}(\mathbf{X}^\star) + \boldsymbol{\varepsilon}; \tag{4.2}$$

here $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p \times n} \to \mathbb{R}^m$ is a fixed and known linear map and $\boldsymbol{\varepsilon}$ is an additive noise term.

The challenge in **PROBLEM 4.1** is to recover the true low-rank matrix in subsampled settings where $m \ll p \cdot n$. In such cases, we typically exploit the prior information that $\mathbf{X}^\star$ is low-rank and thus, we are interested in finding a matrix $\mathbf{X}$ of rank at most r that minimizes the data error $f(\mathbf{X}) := \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{X}\|_2^2$ as follows:

$$\begin{aligned} \underset{\mathbf{X}}{\text{minimize}} \quad & \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X})\|_2^2 \\ \text{subject to} \quad & \operatorname{rank}(\mathbf{X}) \leq k, \end{aligned} \tag{4.3}$$

We present below important ARM problem cases, as characterized by the nature of the linear operator $\boldsymbol{\mathcal{A}}$.

(i) **General linear maps:** In many ARM problem cases, $\boldsymbol{\mathcal{A}}$ or $\boldsymbol{\mathcal{A}}^*$ has a dense range, satisfying specific incoherence or restricted isometry properties (discussed later in this chapter); here, $\boldsymbol{\mathcal{A}}^*$ is the adjoint operator of $\boldsymbol{\mathcal{A}}$. In Quantum Tomography, [Liu11] studies the Pauli operator, a *compressive* linear map $\boldsymbol{\mathcal{A}}$ that consists of the Kronecker product of 2×2 matrices and obeys restricted isometry properties, defined next. Furthermore, recent developments indicate connections of ridge function learning [TC12b, TC12a] and phase retrieval [CL12] with the ARM problem where $\boldsymbol{\mathcal{A}}$ is a Bernoulli and a Fourier operator, respectively.

(ii) **Matrix Completion (MC):** Let Ω be the set of ordered pairs that represent the coordinates of the observable entries in $\mathbf{X}^\star$. Then, the set of observations satisfy $\boldsymbol{y} = \boldsymbol{\mathcal{A}}_\Omega \mathbf{X}^\star + \boldsymbol{\varepsilon}$ where $\boldsymbol{\mathcal{A}}_\Omega$ defines a linear mask over the observable entries Ω. To solve the MC problem, a potential criterion is given by (4.3) [CR09]. As a motivating example, consider the famous Netflix problem [BL07], a recommender system problem where users' movie preferences are inferred by a limited subset of entries in a database.

(iii) **Principal Component Analysis:** In Principal Component Analysis (PCA), we are interested in identifying a low rank subspace that best explains the data in the Euclidean sense from the observations $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star$ where $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p \times n} \to \mathbb{R}^m$ is an identity linear map that stacks the columns of the matrix $\mathbf{X}^\star$ into a single column vector with $m = p \cdot n$. We observe that the PCA problem falls under the ARM criterion in (4.3). While (4.3) is generally NP-hard to solve optimally, PCA can be solved in polynomial time using the truncated Singular Value Decomposition (SVD) of $\boldsymbol{\mathcal{A}}^*\boldsymbol{y}$. As an extension to the PCA setting, [CLMW11] considers the Robust PCA problem where $\boldsymbol{y}$ is further corrupted by gross sparse noise. We extend the framework proposed for low rank recovery to the RPCA case and its generalizations in [KC12b].

As running test cases to support our claims, we consider the MC setting as well as the general ARM setting where $\boldsymbol{\mathcal{A}}$ is constituted by permuted subsampled noiselets [WSB11].

Restricted Isometry Property for low-rank matrices

Similarly to the vector case, one cannot guarantee exact and unique matrix recovery in **PROBLEM 4.1** for any linear map $\boldsymbol{\mathcal{A}}$. Many conditions have been proposed in the literature to establish solution uniqueness and recovery stability for the matrix case. [FRP10] proposed the *restricted isometry property* (RIP) for the ARM problem.

Definition 10. *[Rank Restricted Isometry Property (R-RIP) for matrix linear operators [FRP10]] A linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p \times n} \to \mathbb{R}^m$ satisfies the R-RIP with constant $\delta_r(\boldsymbol{\mathcal{A}}) \in (0, 1)$ if and only if:*

$$(1 - \delta_r(\boldsymbol{\mathcal{A}}))\|\mathbf{X}\|_F^2 \leq \|\boldsymbol{\mathcal{A}}\mathbf{X}\|_2^2 \leq (1 + \delta_r(\boldsymbol{\mathcal{A}}))\|\mathbf{X}\|_F^2, \tag{4.4}$$

$\forall \mathbf{X} \in \mathbb{R}^{p \times n}$ such that rank$(\mathbf{X}) \leq r$. We write δ_r to mean $\delta_r(\boldsymbol{\mathcal{A}})$, unless otherwise stated.

[Liu11] shows that Pauli operators satisfy the rank-RIP in compressive settings while, in function learning, the linear map $\boldsymbol{\mathcal{A}}$ is designed specifically to satisfy the rank-RIP [TC12a].

Chapter roadmap

Based on the greedy hard thresholding methods presented in the previous chapters, we present and analyze a new set of low-rank recovery algorithms for linear inverse problems. In Section 4.3, we provide strategies on how to set up these algorithms via basic ingredients for different configurations to achieve complexity vs. accuracy trade-offs. Moreover, we study acceleration schemes via memory-based techniques and randomized, ϵ-approximate matrix projections to decrease the computational costs in the recovery process. For most of the configurations, we present theoretical analysis that guarantees convergence under mild problem conditions. The above lead to the definition of the MATRIX ALPS framework.

We further improve the performance of the proposed schemes using randomized linear algebra and parallelization in practice. Affine rank minimization algorithms typically rely on calculating the gradient of a data error followed by a singular value decomposition at every iteration. Because these two steps are expensive, heuristic approximations are often used to reduce computational burden. In Section 4.4, we propose a recovery scheme that merges the two steps with randomized approximations, and as a result, operates on space proportional to the degrees of freedom in the problem. We theoretically establish the estimation guarantees of the algorithm as a function of approximation tolerance. While the theoretical approximation requirements are overly pessimistic, we demonstrate that in practice the algorithm performs well on the quantum tomography recovery problem.

As an extension, in Section 4.5 we propose MATRIX ALPS for recovering a sparse plus low-rank decomposition of a matrix given its corrupted and incomplete linear measurements. Our approach is a first-order projected gradient method over non-convex sets, and it exploits a well-known memory-based acceleration technique. We theoretically characterize the convergence properties of MATRIX ALPS using the stable embedding properties of the linear measurement operator $\mathcal{A}$.

Simulation results in Section 4.6 demonstrate notable performance improvements as compared to state-of-the-art algorithms both in terms of reconstruction accuracy and computational complexity.

This chapter is based on the joint work with Volkan Cevher and Stephen Becker [KC14, KC12b, BCK13].

4.1 Preliminaries

Let $\mathcal{S}$ be a set of orthonormal, rank-1 matrices that span an arbitrary subspace in $\mathbb{R}^{p\times n}$. We reserve $\text{span}(\mathcal{S})$ to denote the subspace spanned by $\mathcal{S}$. With slight abuse of notation, we use:

$$\text{rank}(\text{span}(\mathcal{S})) \equiv \max_{\mathbf{X}} \{\text{rank}(\mathbf{X}) : \ \mathbf{X} \in \text{span}(\mathcal{S})\}, \tag{4.5}$$

to denote the *maximum* rank a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ can have such that $\mathbf{X}$ lies in the subspace spanned by the set $\mathcal{S}$. For any matrix $\mathbf{X}$, we use $R(\mathbf{X})$ to denote its range.

We define a *minimum cardinality* set of orthonormal, rank-1 matrices that span the subspace induced by a set of rank-1 (and possibly non-orthogonal) matrices $\mathcal{S}$ as:

$$\text{ortho}(\mathcal{S}) \in \arg\min_{\mathcal{T}} \{|\mathcal{T}| : \mathcal{T} \subseteq \mathcal{U} \text{ s.t. } \text{span}(\mathcal{T}) = \text{span}(\mathcal{S})\},$$

where $\mathcal{U}$ denotes the superset that includes all the sets of *orthonormal*, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that

$\langle \boldsymbol{T}_i, \boldsymbol{T}_j \rangle = 0,\ i \neq j, \forall \boldsymbol{T}_i, \boldsymbol{T}_j \in \mathcal{T}$ and, $\left\|\boldsymbol{T}_i\right\|_F = 1,\ \forall i$. In general, ortho($\mathcal{S}$) is not unique.

Singular Value Decomposition (SVD) and its properties

Definition 11. *[SVD] Let $\mathbf{X} \in \mathbb{R}^{p\times n}$ be a rank-l ($l < \min\{p, n\}$) matrix. Then, the SVD of $\mathbf{X}$ is given by:*

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \begin{bmatrix}\mathbf{U}_\alpha & \mathbf{U}_\beta\end{bmatrix} \begin{bmatrix}\widetilde{\boldsymbol{\Sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix} \begin{bmatrix}\mathbf{V}_\alpha^T \\ \mathbf{V}_\beta^T\end{bmatrix}, \tag{4.6}$$

where $\mathbf{U}_\alpha \in \mathbb{R}^{p\times l}, \mathbf{U}_\beta \in \mathbb{R}^{p\times(p-l)}, \mathbf{V}_\alpha \in \mathbb{R}^{n\times l}, \mathbf{V}_\beta \in \mathbb{R}^{n\times(n-l)}$ and $\widetilde{\boldsymbol{\Sigma}} = diag(\sigma_1, \ldots, \sigma_l) \in \mathbb{R}^{l\times l}$ for $\sigma_1, \ldots, \sigma_l \in \mathbb{R}_+$. Here, the columns of $\mathbf{U}, \mathbf{V}$ represent the set of left and right singular vectors, respectively, and $\sigma_1, \ldots, \sigma_l$ denote the singular values.

For any matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ with arbitrary rank($\mathbf{X}$) $\leq \min\{p, n\}$, its best orthogonal projection $\mathcal{P}_r(\mathbf{X})$ onto the set of rank-r ($r <$ rank($\mathbf{X}$)) matrices $\mathcal{C}_r := \{\mathbf{A} \in \mathbb{R}^{p\times n} : \text{rank}(\mathbf{A}) \leq r\}$ defines the optimization problem:

$$\mathcal{P}_r(\mathbf{X}) \in \underset{\mathbf{Y}\in\mathcal{C}_r}{\arg\min} \left\|\mathbf{Y} - \mathbf{X}\right\|_F. \tag{4.7}$$

According to [HJ90], the best rank-r approximation of a matrix $\mathbf{X}$ corresponds to its truncated SVD: if $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, then $\mathcal{P}_r(\mathbf{X}) := \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^T$ where $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$ is a diagonal matrix that contains the first r diagonal entries of $\boldsymbol{\Sigma}$ and $\mathbf{U}_r,\ \mathbf{V}_r$ contain the corresponding left and right singular vectors, respectively. Moreover, this projection is not always unique. In the case of multiple identical singular values, the lexicographic approach is used to break ties. In any case, $\left\|\mathcal{P}_r(\mathbf{X}) - \mathbf{X}\right\|_F \leq \left\|\mathbf{W} - \mathbf{X}\right\|_F$ for any rank-r $\mathbf{W} \in \mathbb{R}^{p\times n}$.

Subspace projections

Given a set of orthonormal, rank-1 matrices $\mathcal{S}$, we denote the orthogonal projection operator onto the subspace induced by $\mathcal{S}$ as $\mathcal{P}_\mathcal{S}$[1] which is an idempotent linear transformation; furthermore, we denote the orthogonal projection operator onto the orthogonal subspace of $\mathcal{S}$ as $\mathcal{P}_{\mathcal{S}^\perp}$. We can always decompose a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ into two matrix components, as follows:

$$\mathbf{X} := \mathcal{P}_\mathcal{S}\mathbf{X} + \mathcal{P}_{\mathcal{S}^\perp}\mathbf{X}, \text{ such that } \langle \mathcal{P}_\mathcal{S}\mathbf{X}, \mathcal{P}_{\mathcal{S}^\perp}\mathbf{X}\rangle = 0.$$

If $\mathbf{X} \in$ span($\mathcal{S}$), the best projection of $\mathbf{X}$ onto the subspace induced by $\mathcal{S}$ is the matrix $\mathbf{X}$ itself. Moreover, $\left\|\mathcal{P}_\mathcal{S}\mathbf{X}\right\|_F \leq \left\|\mathbf{X}\right\|_F$ for any $\mathcal{S}$ and $\mathbf{X}$.

[1] The distinction between $\mathcal{P}_\mathcal{S}$ and $\mathcal{P}_r$ for r positive integer is apparent from context.

Definition 12. *[Orthogonal projections using SVD] Let $\mathbf{X} \in \mathbb{R}^{p\times n}$ be a matrix with arbitrary rank and SVD decomposition given by (4.6). Then, $\mathcal{S} := \{\boldsymbol{u}_i\boldsymbol{v}_i^T : i = 1,\ldots,r\}$ ($r \leq rank(\mathbf{X})$) constitutes a set of orthonormal, rank-1 matrices that spans the best k-rank subspace in $R(\mathbf{X})$ and $R(\mathbf{X}^T)$; here, $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ denote the i-th left and right singular vectors, respectively. The orthogonal projection onto this subspace is given by [CR09]:*

$$\mathcal{P}_{\mathcal{S}}\mathbf{X} = \mathcal{P}_{\mathcal{U}}\mathbf{X} + \mathbf{X}\mathcal{P}_{\mathcal{V}} - \mathcal{P}_{\mathcal{U}}\mathbf{X}\mathcal{P}_{\mathcal{V}} \tag{4.8}$$

where $\mathcal{P}_{\mathcal{U}} = \mathbf{U}_{:,1:r}\mathbf{U}_{:,1:r}^T$ and $\mathcal{P}_{\mathcal{V}} = \mathbf{V}_{:,1:r}\mathbf{V}_{:,1:r}^T$ in MATLAB *notation.*

Moreover, the orthogonal projection onto the $\mathcal{S}^{\perp}$ is given by:

$$\mathcal{P}_{\mathcal{S}^{\perp}}\mathbf{X} = \mathbf{X} - \mathcal{P}_{\mathcal{S}}\mathbf{X}. \tag{4.9}$$

We use $\mathcal{S} \leftarrow \mathcal{P}_r(\mathbf{X})$ to denote the set of rank-1, orthonormal matrices as outer products of the r left $\boldsymbol{u}_i$ and right $\boldsymbol{v}_i$ principal singular vectors of $\mathbf{X}$ that span the best rank-r subspace of $\mathbf{X}$; e.g. $\mathcal{S} = \{\boldsymbol{u}_i\boldsymbol{v}_i^T,\ i = 1,\ldots,r\}$. Moreover, $\widehat{\mathbf{X}} \leftarrow \mathcal{P}_r(\mathbf{X})$ denotes a/the best rank-r projection matrix of $\mathbf{X}$. In some cases, we use $\{\mathcal{S},\ \widehat{\mathbf{X}}\} \leftarrow \mathcal{P}_r(\mathbf{X})$ when we compute both. The distinction between these cases is apparent from the context.

4.2 Related work

The problem of recovering a low rank matrix from a limited set of measurements—as well as its generalization to the case of low rank and sparse signal recovery (see Section 4.5)—is found in a wide variety of practical context. Such problems have received intensive investigations recently, both from theoretical and algorithmic aspects.

From the convex perspective, the rank constraint in (4.3) is substituted by its convex envelope *nuclear norm*—i.e., given a matrix $\mathbf{X}$ of rank r, its nuclear norm is is defined as:

$$\|\mathbf{X}\|_{\star} = \sum_{i=1}^{r}\sigma_i,$$

where σ_i represents the i-th singular value of $\mathbf{X}$.[2] Within this context, one can use both first- and second-order gradient methods [CCS10, BCG11], as well as ideas originating from Lagrange duality and the method of Alternating Direction Method of Multipliers (ADMM) (e.g. [LCM10])—the latter is also believed to be one of the best performing convex approaches in practice. The main drawback of convex approaches is the requirement of a partial singular value decomposition (SVD) per iteration: this is usually problematic at least for the first few iterations of convex recovery algorithms, where they may have to perform *full SVD's*.

From the non-convex aspect, the Singular Value Projection (SVP) algorithm [MJD10] is the closest to the IHT algorithm for the vector case, a non-convex first-order projected gradient descent algorithm with *constant* step size selection. The CoSaMP/SP analog for the matrix case is developed by Lee et al. in

[2]One can easily observe that the nuclear norm represents the ℓ_1-norm on the singular values of a matrix, in accordance with the ℓ_1-norm in the vector case.

[LB10] with the acronym ADMiRA. Finally, there are algorithms that avoid explicit SVD calculations, such as [WYZ12, RR13, LRS+10], and are typically based on the Burer-Monteiro splitting [BM03]. The main idea in Burer-Monteiro splitting is to remove the non-convex rank constraint by directly embedding into the objective: as opposed to optimizing over a rank-r matrix $\mathbf{X}$, splitting algorithms directly work with its fixed factors $\mathbf{U}\mathbf{V}^T = \mathbf{X}$ in an alternating fashion, where $\mathbf{U} \in \mathbb{R}^{m\times\hat{r}}$ and $\mathbf{V} \in \mathbb{R}^{n\times\hat{r}}$ for some $\hat{r} \geq r$. Unfortunately, rigorous guarantees are difficult.

Finally, a different approach to follow is that of *manifold* methods that better "sense" the geometrical representation of low rank matrices in space. In such case, there is also a long list of available algorithms for the ARM problem such as: (i) the OptSpace algorithm [KMO10], a gradient descent algorithm on the Grassmann manifold, (ii) the Grassmannian Rank-One Update Subspace Estimation (GROUSE) and the Grassmannian Robust Adaptive Subspace Tracking methods (GRASTA) [BNR10, HBL11], two stochastic gradient descent algorithms that operate on the Grassmannian—moreover, to allay the impact of outliers in the subspace selection step, GRASTA incorporates the augmented Lagrangian of ℓ_1-norm loss function into the Grassmannian optimization framework and, (iii) the Riemannian Trust Region Matrix Completion algorithm (RTRMC) [BA11], a matrix completion method using first- and second-order Riemannian trust-region approaches,

4.3 Matrix Algebraic Pursuits

Here, we study a special class of iterative greedy algorithms known as hard thresholding methods. Similar results have been derived for the vector case in Chapter 2 [KC11]. Note that the transition from sparse vector approximation to ARM is *non-trivial*; while k-sparse signals "live" in the union of finite number of subspaces, the set of rank-r matrices expands to infinitely many subspaces. Thus, the selection rules do not generalize in a straightforward way.

Ingredients of hard thresholding methods: Similarly to Chapter 2, we analyze the behaviour and performance of hard thresholding methods from a global perspective. Five building blocks are studied: $i)$ step size selection μ_i, $ii)$ gradient or least-squares updates over restricted low-rank subspaces (e.g., adaptive block coordinate descent), $iii)$ memory exploitation, $iv)$ active low-rank subspace tracking and, $v)$ low-rank matrix approximations (described next). We highlight the impact of these key pieces on the convergence rate and signal reconstruction performance and provide optimal and/or efficient strategies on how to set up these ingredients under different problem conditions.

Low-rank matrix approximations in hard thresholding methods: In [KC12a], we show that the solution efficiency can be significantly improved by ϵ-approximation algorithms. Based on similar ideas, we analyze the impact of ϵ-approximate low rank-revealing schemes in the proposed algorithms with well-characterized time and space complexities. Moreover, we provide extensive analysis to prove convergence using ϵ-approximate low-rank projections.

MATRIX ALPS in a nutshell

Explicit descriptions of the proposed algorithms are provided in Algorithms 8 and 9. Algorithm 8 follows from the ALgrebraic PursuitS (ALPS) scheme for the vector case [KC11]. MATRIX ALPS I provides efficient strategies for adaptive step size selection and additional signal estimate updates at each iteration (these motions are explained in detail in the next subsection). Algorithm 9 (ADMiRA) [LB10] further improves the performance of Algorithm 8 by introducing least squares optimization steps on restricted

Algorithm 8 MATRIX ALPS I

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, r, Tolerance η, MaxIterations
Initialize: $\mathbf{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  $\mathcal{D}_i \leftarrow \mathcal{P}_r\big(\mathcal{P}_{\mathcal{X}_i^{\perp}} \nabla f(\mathbf{X}(i))\big)$ *(Best rank-r subspace orthogonal to $\mathcal{X}_i$)*
  $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{X}_i$ *(Active subspace expansion)*
  $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\mathbf{X}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\|_2^2}$ *(Step size selection)*
  $\mathbf{V}(i) \leftarrow \mathbf{X}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))$ *(Error norm reduction via gradient descent)*
  $\{\mathcal{W}_i,\ \mathbf{W}(i)\} \leftarrow \mathcal{P}_r(\mathbf{V}(i))$ *(Best rank-r subspace selection)*
  $\xi_i \leftarrow \arg\min_{\xi} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\mathbf{W}(i) - \frac{\xi}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))\|_2^2}$ *(Step size selection)*
  $\mathbf{X}(i+1) \leftarrow \mathbf{W}(i) - \frac{\xi_i}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))$ with $\mathcal{X}_{i+1} \leftarrow \mathcal{P}_k(\mathbf{X}(i+1))$ *(De-bias using gradient descent)*
  $i \leftarrow i+1$
until $\|\mathbf{X}(i) - \mathbf{X}(i-1)\|_2 \leq \eta\|\mathbf{X}(i)\|_2$ or MaxIterations.

Algorithm 9 ADMiRA Instance

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, r, Tolerance η, MaxIterations
Initialize: $\mathbf{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  $\mathcal{D}_i \leftarrow \mathcal{P}_r\big(\mathcal{P}_{\mathcal{X}_i^{\perp}} \nabla f(\mathbf{X}(i))\big)$ *(Best rank-r subspace orthogonal to $\mathcal{X}_i$)*
  $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{X}_i$ *(Active subspace expansion)*
  $\mathbf{V}(i) \leftarrow \arg\min_{\mathbf{V}:\mathbf{V}\in\mathrm{span}(\mathcal{S}_i)} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{V}\big\|_2^2$ *(Error norm reduction via least-squares optimization)*
  $\{\mathcal{X}_{i+1},\ \mathbf{X}(i+1)\} \leftarrow \mathcal{P}_r(\mathbf{V}(i))$ *(Best rank-r subspace selection)*
  $i \leftarrow i+1$
until $\|\mathbf{X}(i) - \mathbf{X}(i-1)\|_2 \leq \eta\|\mathbf{X}(i)\|_2$ or MaxIterations.

subspaces—this technique borrows from a series of vector reconstruction algorithms such as CoSaMP [NT09a], Subspace Pursuit (SP) [DM09] and Hard Thresholding Pursuit (HTP) [Fou11].

In a nutshell, both algorithms simply seek to improve the subspace selection by iteratively collecting an extended subspace $\mathcal{S}_i$ with $\mathrm{rank}(\mathrm{span}(\mathcal{S}_i)) \leq 2r$ and then finding the rank-r matrix that fits the measurements in this restricted subspace using least squares or gradient descent motions.

At each iteration, the Algorithms 8 and 9 perform motions from the following list:

1) *Best rank-r subspace orthogonal to $\mathcal{X}_i$ and active subspace expansion:* We identify the best rank-r subspace of the current gradient $\nabla f(\mathbf{X}(i))$, orthogonal to $\mathcal{X}_i$ and then merge this low-rank subspace with $\mathcal{X}_i$. This motion guarantees that, at each iteration, we expand the current rank-r subspace estimate with r new, rank-1 orthogonal subspaces to explore.

2a) *Error norm reduction via greedy descent with adaptive step size selection (Algorithm 8):* We decrease the data error by performing a single gradient descent step. This scheme is based on a one-shot step size selection procedure (Step size selection step)—detailed description of this approach is given in Section 4.3.1.

2b) *Error norm reduction via least squares optimization (Algorithm 9):* We decrease the data error $f(\mathbf{X})$ on the active $\mathcal{O}(r)$-low rank subspace. Assuming $\boldsymbol{\mathcal{A}}$ is well-conditioned over low-rank subspaces, the main complexity of this operation is dominated by the solution of a symmetric linear system of

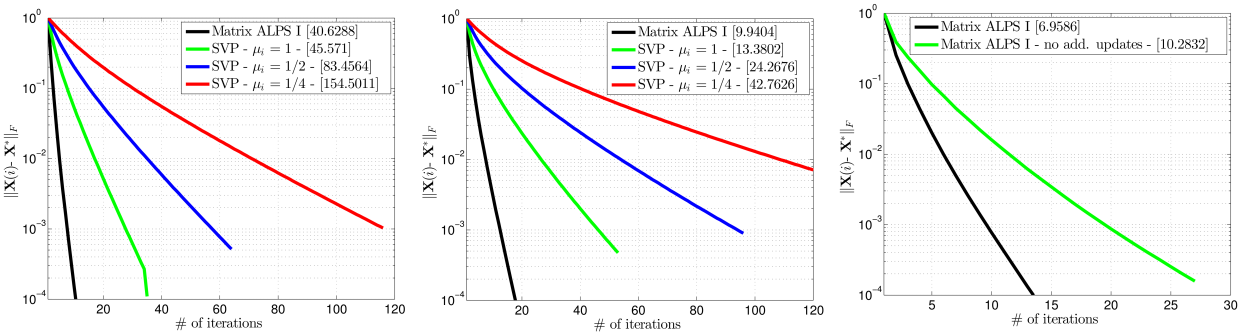


Figure 4.1: *Median error per iteration for various step size policies and 20 Monte-Carlo repetitions. In brackets, we present the median time consumed for convergene in seconds. (a)* $p = n = 2048$, $m = 0.4n^2$, *and rank* $r = 70$*—*$\boldsymbol{\mathcal{A}}$ *is formed by permuted and subsampled noiselets [CGM01]. (b)* $n = 2048$, $p = 512$, $m = 0.4n^2$, *and rank* $r = 50$*—we use underdetermined linear map* $\boldsymbol{\mathcal{A}}$ *according to the MC problem (c)* $n = 2048$, $p = 512$, $m = 0.4n^2$, *and rank* $r = 40$*—we use underdetermined linear map* $\boldsymbol{\mathcal{A}}$ *according to the MC problem.*

equations.

3) *Best rank-r subspace selection:* We project the constrained solution onto the set of rank-r matrices $\mathcal{C}_r := \{\boldsymbol{A} \in \mathbb{R}^{p\times n} : \text{rank}(\boldsymbol{A}) \leq r\}$ to arbitrate the active support set. This step is calculated in polynomial time complexity as a function of $p \times n$ using SVD or other matrix rank-revealing decomposition algorithms—further discussions about this step and its approximations can be found in Sections 4.3.5 and 4.3.6.

4) *De-bias using gradient descent (Algorithm 8):* We de-bias the current estimate $\mathbf{W}(i)$ by performing an additional gradient descent step, decreasing the data error. The step size selection procedure follows the same motions as in 2a).

4.3.1 Hard thresholding ingredients in the matrix case

Step size selection

There is limited work on the adaptive step size selection for matrix hard thresholding methods. To the best of our knowledge, only the work of [TW13] implements ideas presented in [BD10] for the matrix case.

According to Algorithm 8, let $\mathbf{X}(i)$ be the current rank-r matrix estimate spanned by the set of orthonormal, rank-1 matrices in $\mathcal{X}_i$. Using regular gradient descent motions, the new rank-r estimate $\mathbf{W}(i)$ can be calculated through:

$$\mathbf{V}_i = \mathbf{X}(i) - \frac{\mu}{2}\nabla f(\mathbf{X}(i)), \quad \{\mathcal{W}_i,\ \mathbf{W}(i)\} \leftarrow \mathcal{P}_r(\mathbf{V}(i)).$$

We highlight that the rank-r approximate matrix may not be unique. It then holds that the subspace spanned by $\mathcal{W}_i$ originates: i) either from the subspace of $\mathcal{X}_i$, ii) or from the best subspace (in terms of the Frobenius norm metric) of the current gradient $\nabla f(\mathbf{X}(i))$, *orthogonal to* $\mathcal{X}_i$, iii) or from the combination of orthonormal, rank-1 matrices lying on the union of the above two subspaces. The statements above can

be summarized in the following expression:

$$\text{span}(\mathcal{W}_i) \in \text{span}\left(\mathcal{D}_i \cup \mathcal{X}_i\right) \tag{4.10}$$

for any step size μ_i and $\mathcal{D}_i \leftarrow \mathcal{P}_r\big(\mathcal{P}_{\mathcal{X}_i^\perp} \nabla f(\mathbf{X}(i))\big)$. Since $\text{rank}(\text{span}(\mathcal{W}_i)) \leq r$, we easily deduce the following key observation: let $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{X}_i$ be a set of rank-1, orthonormal matrices where $\text{rank}(\text{span}(\mathcal{S}_i)) \leq 2r$. Given $\mathcal{W}_i$ is unknown before the i-th iteration, $\mathcal{S}_i$ spans the smallest subspace that contains $\mathcal{W}_i$ such that the following equality

$$\mathcal{P}_r\left(\mathbf{X}(i) - \frac{\mu_i}{2}\nabla f(\mathbf{X}(i))\right) = \mathcal{P}_r\left(\mathbf{X}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\right) \tag{4.11}$$

necessarily holds.[3]

To compute step-size μ_i, we use:

$$\mu_i = \arg\min_{\mu} \left\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\left(\mathbf{X}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\right)\right\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\|_2^2}, \tag{4.12}$$

i.e., μ_i is the minimizer of the objective function, given the current gradient $\nabla f(\mathbf{X}(i))$. Note that:

$$1 - \delta_{2r}(\boldsymbol{\mathcal{A}}) \leq \frac{1}{\mu_i} \leq 1 + \delta_{2r}(\boldsymbol{\mathcal{A}}), \tag{4.13}$$

due to R-RIP—i.e., we select $2r$ subspaces such that μ_i satisfies (4.13). We can derive similar arguments for the additional step size selection ξ_i in Step 6 of Algorithm 8.

Adaptive μ_i scheme results in more restrictive worst-case isometry constants compared to [JMD10], but faster convergence and better stability are empirically observed in general. In [JMD10], the authors present the Singular Value Projection (SVP) algorithm, an iterative hard thresholding algorithm for the ARM problem. According to [JMD10], both constant and iteration dependent (but user-defined) step sizes are considered. Adaptive strategies presented in [JMD10] require the computation of R-RIP constants which has exponential time complexity. Figures 4.1(a)-(b) illustrate some characteristic examples. The performance varies for different problem configurations. For $\mu > 1$, SVP *diverges* for various test cases. We note that, for large fixed matrix dimensions p, n, adaptive step size selection becomes computationally expensive compared to constant step size selection strategies, as the rank of $\mathbf{X}^\star$ increases.

Updates on restricted subspaces

In Algorithm 8, at each iteration, the new estimate $\mathbf{W}(i) \leftarrow \mathcal{P}_r\left(\mathbf{V}(i)\right)$ can be further refined by applying a single or multiple gradient descent updates with line search restricted on $\mathcal{W}_i$ [Fou11] (Step 7 in Algorithm 8):

$$\mathbf{X}(i+1) \leftarrow \mathbf{W}(i) - \frac{\xi_i}{2}\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i)),$$

where $\xi_i = \frac{\|\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{W}_i}\nabla f(\mathbf{W}(i))\|_2^2}$. In spirit, the gradient step above is the same as block coordinate descent in convex optimization where we find the subspaces adaptively. Figure 4.1(c) depicts the acceleration achieved by using additional gradient updates over restricted low-rank subspaces for a test case.

[3] In the case of multiple identical singular values, any ties are lexicographically dissolved.

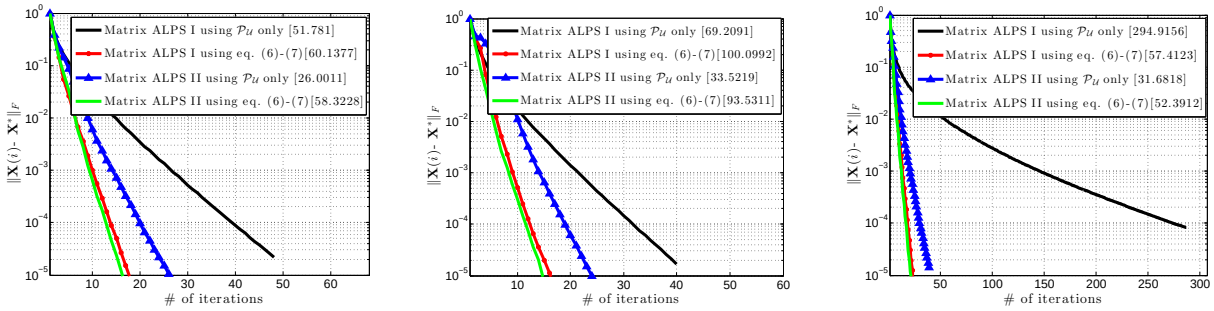


Figure 4.2: *Median error per iteration for* MATRIX ALPS I *and* MATRIX ALPS II *variants over 10 Monte-Carlo repetitions. In brackets, we present the median time consumed for convergene in seconds. (a)* $n = 2048, p = 512$, $m = 0.25n^2$, *and rank* $r = 40$. *(b)* $n = 2000, p = 1000$, $m = 0.25n^2$, *and rank* $r = 50$. *(c)* $n = p = 1000$, $m = 0.25n^2$, *and rank* $r = 50$.

Acceleration via memory-based schemes and low-rank matrix approximations

Memory-based techniques can be used to improve convergence speed. Furthermore, low-rank matrix approximation tools overcome the computational overhead of computing the best low-rank projection by inexactly solving (4.7). We keep the discussion on memory utilization for Section 4.3.4 and low-rank matrix approximations for Sections 4.3.5 and 4.3.6 where we present new algorithmic frameworks for low-rank matrix recovery.

Active low-rank subspace tracking

Per iteration of Algorithms 8 and 9, we perform projection operations $\mathcal{P}_{\mathcal{S}}\mathbf{X}$ and $\mathcal{P}_{\mathcal{S}^\perp}\mathbf{X}$ where $\mathbf{X} \in \mathbb{R}^{p\times n}$, as described by (4.8) and (4.9), respectively. Since $\mathcal{S}$ is constituted by outer products of left and right singular vectors as in Definition 12, $\mathcal{P}_{\mathcal{S}}\mathbf{X}$ (resp. $\mathcal{P}_{\mathcal{S}^\perp}\mathbf{X}$) projects onto the (resp. complement of the) best low-rank subspace in $R(\mathbf{X})$ and $R(\mathbf{X}^T)$. These operations are highly connected with the adaptive step size selection and the updates on restricted subspaces. Unfortunately, the time-complexity to compute $\mathcal{P}_{\mathcal{S}}\mathbf{X}$ is dominated by three matrix-matrix multiplications which decelerates the convergence of the proposed schemes in high-dimensional settings. To accelerate the convergence in many test cases, it turns out that we do not have to use the best projection $\mathcal{P}_{\mathcal{S}}$ in practice.[4] Rather, employing *inexact* projections is sufficient to converge to the optimal solution: either $i)$ $\mathcal{P}_{\mathcal{U}}\mathbf{X}$ onto the best low-rank subspace in $R(\mathbf{X})$ only (if $p \ll n$) or $ii)$ $\mathbf{X}\mathcal{P}_{\mathcal{V}}$ onto the best low-rank subspace in $R(\mathbf{X}^T)$ only (if $p \gg n$)[5]; $\mathcal{P}_{\mathcal{U}}$ and $\mathcal{P}_{\mathcal{V}}$ are defined in Definition 12 and require only one matrix-matrix multiplication.

Figure 4.2 shows the time overhead due to the exact projection application $\mathcal{P}_{\mathcal{S}}$ compared to $\mathcal{P}_{\mathcal{U}}$ for $p \leq n$. In Figure 4.2(a), we use subsampled and permuted noiselets for linear map $\boldsymbol{\mathcal{A}}$ and in Figures 4.2(b)-(c), we test the MC problem. While in the case $p = n$ the use of (4.8)-(4.9) has a clear advantage (in terms of required number of iterations for convergence) over inexact projections using only $\mathcal{P}_{\mathcal{U}}$, the latter case converges faster to the desired accuracy $5 \cdot 10^{-4}$ when $p \ll n$ as shown in Figures 4.2(a)-(b). In our derivations, we assume $\mathcal{P}_{\mathcal{S}}$ and $\mathcal{P}_{\mathcal{S}^\perp}$ as defined in (4.8) and (4.9).

[4] From a different perspective and for a different problem case, similar ideas have been used in [LCM10].
[5] We can move between these two cases by a simple transpose of the problem.

4.3.2 Convergence guarantees for matrix ALPS

In this section, we present the theoretical convergence guarantees of Algorithms 8 and 9 as functions of R-RIP constants.

MATRIX ALPS I

An important lemma for our derivations below is given next:

Lemma 21. *[Active subspace expansion] Let $\mathbf{X}(i)$ be the matrix estimate at the i-th iteration and let $\mathcal{X}_i$ be a set of orthonormal, rank-1 matrices such that $\mathcal{X}_i \leftarrow \mathcal{P}_r(\mathbf{X}(i))$. Then, at each iteration, the Active Subspace Expansion step in Algorithms 8 and 9 identifies information in $\mathbf{X}^\star$, such that:*

$$\left\|\mathcal{P}_{\mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i^\perp}\mathbf{X}^\star\right\|_F \leq (2\delta_{2r}+2\delta_{3r})\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F + \sqrt{2(1+\delta_{2r})}\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{4.14}$$

where $\mathcal{S}_i \leftarrow \mathcal{X}_i \cup \mathcal{D}_i$ and $\mathcal{X}^* \leftarrow \mathcal{P}_r(\mathbf{X}^\star)$.

Lemma 21 states that, at each iteration, the active subspace expansion step identifies a $2r$ rank subspace such that the amount of unrecovered energy of $\mathbf{X}^\star$—i.e., the projection of $\mathbf{X}^\star$ onto the orthogonal subspace of span$(\mathcal{S}_i)$—is bounded by (4.14).

Then, Theorem 6 characterizes the iteration invariant of Algorithm 8 for the matrix case:

Theorem 6. *[Iteration invariant for* MATRIX ALPS I*] The $(i+1)$-th matrix estimate $\mathbf{X}(i+1)$ of* MATRIX ALPS I *satisfies the following recursion:*

$$\left\|\mathbf{X}(i+1)-\mathbf{X}^\star\right\|_F \leq \rho\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{4.15}$$

where $\rho := \left(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\right)\left(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r}+2\delta_{3r})\frac{2\delta_{3r}}{1-\delta_{2r}}\right)$ and $\gamma := \left(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\right)\left(\frac{2\sqrt{1+\delta_{2r}}}{1-\delta_{2r}} + \frac{2\delta_{3r}}{1-\delta_{2r}}\sqrt{2(1+\delta_{2r})}\right) + \frac{\sqrt{1+\delta_r}}{1-\delta_r}$. Moreover, when $\delta_{3r} < 0.1235$, the iterations are contractive.

To provide some intuition behind this result, assume that $\mathbf{X}^\star$ is a rank-r matrix. Then, according to Theorem 6, for $\rho < 1$, the approximation parameter γ in (4.15) satisfies:

$$\gamma < 5.7624, \quad \text{for} \quad \delta_{3r} < 0.1235.$$

Moreover, we derive the following:

$$\rho < \frac{1+2\delta_{3r}}{(1-\delta_{3r})^2}\left(4\delta_{3r}+8\delta_{3r}^2\right) < \frac{1}{2} \Rightarrow \delta_{3r} < 0.079,$$

which is *a stronger* R-RIP condition assumption compared to state-of-the-art approaches [LB10]. In the next section, we further improve this guarantee using Algorithm 9.

Unfolding the recursive formula (4.15), we obtain the following upper bound for $\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F$ at the

i-th iteration:

$$\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F \leq \rho^i \left\|\mathbf{X}(0)-\mathbf{X}^\star\right\|_F + \frac{\gamma}{1-\rho}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{4.16}$$

Then, given $\mathbf{X}(0)=\mathbf{0}$, MATRIX ALPS I finds a rank-r solution $\widehat{\mathbf{X}} \in \mathbb{R}^{p\times n}$ such that $\left\|\widehat{\mathbf{X}}-\mathbf{X}^\star\right\|_F \leq \frac{\gamma+1-\rho}{1-\rho}\left\|\boldsymbol{\varepsilon}\right\|_2$ after $i := \left\lceil \frac{\log(\|\mathbf{X}^\star\|_F/\|\boldsymbol{\varepsilon}\|_2)}{\log(1/\rho)} \right\rceil$ iterations.

If we ignore steps 5 and 6 in Algorithm 8, we obtain another projected gradient descent variant for the affine rank minimization problem, for which we obtain the following performance guarantees—the proof follows from the proof of Theorem 6.

Corollary 4. *[MATRIX ALPS I Instance] In Algorithm 8, we ignore steps 5 and 6 and let* $\{\mathcal{X}_{i+1},\ \mathbf{X}(i+1)\} \leftarrow \mathcal{P}_r(\mathbf{V}_i)$. *Then, by the same analysis, we observe that the following recursion is satisfied:*

$$\left\|\mathbf{X}(i+1)-\mathbf{X}^\star\right\|_F \leq \rho\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_2, \tag{4.17}$$

for $\rho := \left(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r}+2\delta_{3r})\frac{2\delta_{3r}}{1-\delta_{2r}}\right)$ *and* $\gamma := \left(\frac{2\sqrt{1+\delta_{2r}}}{1-\delta_{2r}} + \frac{2\delta_{3r}}{1-\delta_{2r}}\sqrt{2(1+\delta_{2r})}\right)$. *Moreover,* $\rho<1$ *when* $\delta_{3r}<0.1594$.

We observe that the absence of the additional estimate update over restricted support sets results in less restrictive isometry constants compared to Theorem 6. In practice, additional updates result in faster convergence, as shown in Figure 4.1(c).

ADMiRA Instance

In MATRIX ALPS I, the gradient descent steps constitute a first-order approximation to least-squares minimization problems. Replacing Step 4 in Algorithm 8 with the following optimization problem:

$$\mathbf{V}(i) \leftarrow \underset{\mathbf{V}:\mathbf{V}\in\text{span}(\mathcal{S}_i)}{\arg\min} \left\|\boldsymbol{y}-\boldsymbol{\mathcal{A}}\mathbf{V}\right\|_2^2, \tag{4.18}$$

we obtain ADMiRA (furthermore, we remove the de-bias step in Algorithm 8). Assuming that the linear operator $\boldsymbol{\mathcal{A}}$, restricted on sufficiently low-rank subspaces, is well conditioned in terms of the R-RIP assumption, the optimization problem (4.18) has a unique optimal minimizer. ADMiRA instance in Algorithm 9 features the following guarantee:

Theorem 7. *[Iteration invariant for ADMiRA instance] The* $(i+1)$*-th matrix estimate* $\mathbf{X}(i+1)$ *of ADMiRA answers the following recursive expression:*

$$\left\|\mathbf{X}(i+1)-\mathbf{X}^\star\right\|_F \leq \rho\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_F,$$

$\rho := (2\delta_{2r}+2\delta_{3r})\sqrt{\frac{1+3\delta_{3r}^2}{1-\delta_{3r}^2}}$, *and* $\gamma := \sqrt{\frac{1+3\delta_{3r}^2}{1-\delta_{3r}^2}}\sqrt{2(1+\delta_{3r})} + \left(\frac{\sqrt{1+3\delta_{3r}^2}}{1-\delta_{3r}}+\sqrt{3}\right)\sqrt{1+\delta_{2r}}$. *Moreover, when* $\delta_{3r}<0.2267$, *the iterations are contractive.*

Similarly to MATRIX ALPS I analysis, the parameter γ in Theorem 7 satisfies:

$$\gamma < 5.1848, \text{ for } \delta_{3r} < 0.2267.$$

Furthermore, to compare the approximation guarantees of Theorem 7 with [LB10], we further observe:

$$\delta_{3r} < 0.1214, \text{ for } \rho < 1/2.$$

We remind that [LB10] provides convergence guarantees for ADMiRA with $\delta_{4r} < 0.04$ for $\rho = 1/2$.

4.3.3 Complexity Analysis

A non-exhaustive list of linear map examples includes the identity operator (Principal component analysis (PCA) problem), Fourier/Wavelets/Noiselets tranformations and the famous Matrix Completion problem where $\boldsymbol{\mathcal{A}}$ is a mask operator such that only a fraction of elements in $\mathbf{X}$ is observed. Assuming the most demanding case where $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^*$ are dense linear maps with no structure, the computation of the gradient $\nabla f(\mathbf{X}(i))$ at each iteration requires $\mathcal{O}(mrpn)$ arithmetic operations.

Given a set $\mathcal{S}$ of orthonormal, rank-1 matrices, the projection $\mathcal{P}_{\mathcal{S}}\mathbf{X}$ for any matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ requires time complexity $\mathcal{O}(\max\{p^2n, pn^2\})$ as a sequence of matrix-matrix multiplication operations.[6] In MATRIX ALPS I, the adaptive step size selection steps require $\mathcal{O}(\max\{mrpn, p^2n\})$ time complexity for the calculation of μ_i and ξ_i quantities. In ADMiRA solving a least-squares system restricted on rank-$2r$ and rank-r subspaces requires $\mathcal{O}(mr^2)$ complexity; according to [NT09a], [LB10], the complexity of this step can be further reduced using iterative techniques such as the Richardson method or conjugate gradients algorithm.

Using the Lanczos method, we require $\mathcal{O}(rpn)$ arithmetic operations to compute a rank-r matrix approximation for a given constant accuracy; a prohibitive time-complexity that does not scale well for many practical applications. Sections 4.3.5 and 4.3.6 describe approximate low rank matrix projections and how they affect the convergence guarantees of the proposed algorithms.

Overall, the operation that dominates per iteration requires $\mathcal{O}(\max\{mrpn, p^2n, pn^2\})$ time complexity.

4.3.4 Memory-based Acceleration

Algorithm 10 MATRIX ALPS II

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, r, Tolerance η, MaxIterations
Initialize: $\mathbf{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $\mathbf{Q}(0) \leftarrow 0$, $\mathcal{Q}_0 \leftarrow \{\emptyset\}$, $\tau_i\ \forall i$, $i \leftarrow 0$
repeat
  $\mathcal{D}_i \leftarrow \mathcal{P}_r\big(\mathcal{P}_{\mathcal{Q}_i^{\perp}} \nabla f(\mathbf{Q}(i))\big)$ *(Best rank-r subspace orthogonal to $\mathcal{Q}_i$)*
  $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{Q}_i$ *(Active subspace expansion)*
  $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\mathbf{Q}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\|_2^2}$ *(Step size selection)*
  $\mathbf{V}(i) \leftarrow \mathbf{Q}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))$ *(Error norm reduction via gradient descent)*
  $\{\mathcal{X}_{i+1},\ \mathbf{X}(i+1)\} \leftarrow \mathcal{P}_r(\mathbf{V}(i))$ *(Best rank-r subspace selection)*
  $\mathbf{Q}(i+1) \leftarrow \mathbf{X}(i+1) + \tau_i(\mathbf{X}(i+1) - \mathbf{X}(i))$ *(Momentum update)*
  $\mathcal{Q}_{i+1} \leftarrow \text{ortho}(\mathcal{X}_i \cup \mathcal{X}_{i+1})$
  $i \leftarrow i + 1$
until $\|\mathbf{X}(i) - \mathbf{X}(i-1)\|_2 \leq \eta\|\mathbf{X}(i)\|_2$ or MaxIterations.

[6] While such operation has $\mathcal{O}(\max\{p^2n, pn^2\})$ complexity, each application of $\mathcal{P}_{\mathcal{S}}\mathbf{X}$ requires three matrix-matrix multiplications. To reduce such computational cost, we *relax* this operation in Section 4.6 where in practice we use only $\mathcal{P}_{\mathcal{U}}$ that needs one matrix-matrix multiplication.

Similar to the vector case, we propose to select τ_i as the minimizer of the objective function:

$$\tau_i = \arg\min_{\tau} \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{Q}(i+1)\|_2^2 = \frac{\langle \boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{X}(i), \boldsymbol{\mathcal{A}}\mathbf{X}(i) - \boldsymbol{\mathcal{A}}\mathbf{X}(i-1)\rangle}{\|\boldsymbol{\mathcal{A}}\mathbf{X}(i) - \boldsymbol{\mathcal{A}}\mathbf{X}(i-1)\|_2^2}, \tag{4.19}$$

where $\boldsymbol{\mathcal{A}}\mathbf{X}(i), \boldsymbol{\mathcal{A}}\mathbf{X}(i-1)$ are already *pre-computed* at each iteration. According to (4.19), τ_i is dominated by the calculation of a vector inner product, a computationally cheaper process than q calculation.

Theorem 8 characterizes Algorithm 10 for *constant* momentum step size selection. To keep the main ideas simple, we ignore the additional gradient updates in Algorithm 10. In addition, we only consider the noiseless case for clarity. The convergence rate proof for these cases is provided in the appendix.

Theorem 8. *[Iteration invariant for* MATRIX ALPS II*] Let $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star$ be a noiseless set of observations. To recover $\mathbf{X}^\star$ from $\boldsymbol{y}$ and $\boldsymbol{\mathcal{A}}$, the $(i+1)$-th matrix estimate $\mathbf{X}(i+1)$ of* MATRIX ALPS II *satisfies the following recursion:*

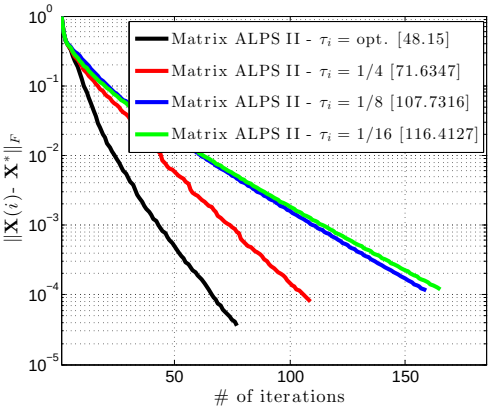
$$\left\|\mathbf{X}(i+1) - \mathbf{X}^\star\right\|_F \leq \alpha(1+\tau_i)\left\|\mathbf{X}(i) - \mathbf{X}^\star\right\|_F + \alpha\tau_i\left\|\mathbf{X}(i-1) - \mathbf{X}^\star\right\|_F, \tag{4.20}$$

where $\alpha := \frac{4\delta_{3r}}{1-\delta_{3r}} + (2\delta_{3r} + 2\delta_{4r})\frac{2\delta_{3r}}{1-\delta_{3r}}$. Moreover, the following inequality holds true:

$$\left\|\mathbf{X}(i+1) - \mathbf{X}^\star\right\|_F \leq \rho^{i+1}\left\|\mathbf{X}(0) - \mathbf{X}^\star\right\|_F, \text{ for } \rho := \frac{\alpha(1+\tau_i) + \sqrt{\alpha^2(1+\tau_i)^2 + 4\alpha\tau_i}}{2}. \tag{4.21}$$

Theorem 8 provides convergence rate behaviour proof for the case where τ_i is constant $\forall i$. The more elaborate case where τ_i follows the policy described in (4.19) is left as an open question for future work. To provide some insight for (4.21), for $\tau_i = 1/4,\ \forall i$ and $\tau_i = 1/2,\ \forall i$, $\delta_{4r} < 0.1187$ and $\delta_{4r} < 0.095$ guarantee convergence in Algorithm 10, respectively. While the RIP requirements for memory-based MATRIX ALPS II are more stringent than the schemes proposed in the previous section, it outperforms Algorithms 8 and 9. Figure 4.3 shows the acceleration achieved in MATRIX ALPS II by using inexact projections $\mathcal{P}_{\mathcal{U}}$. Using the proper projections (4.8)-(4.9), Figure 4.3 shows acceleration in practice when using the adaptive momentum step size strategy: while a wide range of constant momentum step sizes leads to convergence, providing flexibility to select an appropriate τ_i, adaptive τ_i avoids this arbitrary τ_i selection while further decreases the number of iterations needed for convergence in most cases.

Figure 4.3: *Median error per iteration for various momentum step size policies and 10 Monte-Carlo repetitions. Here, $n = 1024$, $p = 256$, $m = 0.25n^2$, and rank $r = 40$. We use permuted and subsampled noiselets for the linear map $\boldsymbol{\mathcal{A}}$. In brackets, we present the median time for convergence in seconds.*



4.3.5 Accelerating MATRIX ALPS: ϵ-Approximation of SVD via Column Subset Selection

A time-complexity bottleneck in the proposed schemes is the computation of the singular value decomposition to find subspaces that describe the unexplored information in matrix $\mathbf{X}^\star$. Unfortunately, the computational cost of regular SVD for best subspace tracking is prohibitive for many applications.

Based on [DFK+04, DKM06], we can obtain randomized SVD approximations of a matrix $\mathbf{X}$ using *column subset selection* ideas: we compute a score for each column that represents its "significance". In particular, we define a probability distribution that weights each column depending on the amount of information they contain; usually, the distribution is related to the ℓ_2-norm of the columns. The main idea of this approach is to compute a surrogate rank-r matrix $\mathcal{P}_r^\epsilon(\mathbf{X})$ by subsampling the columns according to this distribution. It turns out that the total number of sampled columns is a function of the parameter ϵ. Moreover, [DRVW06, DV06] proved that, given a target rank r and an approximation parameter ϵ, we can compute an ϵ-approximate rank-r matrix $\mathcal{P}_r^\epsilon(\mathbf{X})$, i.e.,

Definition 13. *[ϵ-approximate low-rank projection] Let $\mathbf{X}$ be an arbitrary matrix. Then, $\mathcal{P}_r^\epsilon(\mathbf{X})$ projection provides a rank-r matrix approximation to $\mathbf{X}$ such that:*

$$\left\|\mathcal{P}_r^\epsilon(\mathbf{X}) - \boldsymbol{X}\right\|_F^2 \leq (1+\epsilon)\left\|\mathcal{P}_r(\mathbf{X}) - \boldsymbol{X}\right\|_F^2, \text{ where } \mathcal{P}_r(\mathbf{X}) \in \underset{\boldsymbol{Y}:\mathit{rank}(\boldsymbol{Y})\leq r}{\arg\min} \|\mathbf{X} - \boldsymbol{Y}\|_F. \tag{4.22}$$

For the following theoretical results, we assume the following condition on the sensing operator $\boldsymbol{\mathcal{A}}$: $\left\|\boldsymbol{\mathcal{A}}^*\boldsymbol{\beta}\right\|_F \leq \lambda,\ \forall \boldsymbol{\beta} \in \mathbb{R}^p$, where $\lambda > 0$. Using ϵ-approximation schemes to perform the Active subspace selection step, the following upper bound holds. The proof is provided in [KC14]:

Lemma 22. *[ϵ-approximate active subspace expansion] Let $\mathbf{X}(i)$ be the matrix estimate at the i-th iteration and let $\mathcal{X}_i$ be a set of orthonormal, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that $\mathcal{X}_i \leftarrow \mathcal{P}_r(\mathbf{X}(i))$. Furthermore, let*

$$\mathcal{D}_i^\epsilon \leftarrow \mathcal{P}_r^\epsilon\big(\mathcal{P}_{\mathcal{X}_i^\perp}\nabla f(\mathbf{X}(i))\big),$$

be a set of orthonormal, rank-1 matrices that span rank-r subspace such that (4.22) is satisfied for $\boldsymbol{X} := \mathcal{P}_{\mathcal{X}_i^\perp}\nabla f(\mathbf{X}(i))$. Then, at each iteration, the Active Subspace Expansion step in Algorithms 8 and 9 captures information contained in the true matrix $\mathbf{X}^\star$, such that:

$$\left\|\mathcal{P}_{\mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i^\perp}\mathbf{X}^\star\right\|_F \leq \left(2\delta_{2r} + 2\delta_{3r}\right)\left\|\mathbf{X}(i) - \mathbf{X}^\star\right\|_F + \sqrt{2(1+\delta_{2r})}\left\|\boldsymbol{\varepsilon}\right\|_2 + 2\lambda\sqrt{\epsilon}, \tag{4.23}$$

where $\mathcal{S}_i \leftarrow \mathcal{X}_i \cup \mathcal{D}_i^\epsilon$ and $\mathcal{X}^* \leftarrow \mathcal{P}_r(\mathbf{X}^\star)$.

Furthermore, to prove the following theorems, we require the following lemma; the proof is provided in [KC14].

Lemma 23. *[ϵ-approximation rank-r subspace selection] Let $\mathbf{V}(i)$ be a rank-$2r$ proxy matrix in the subspace spanned by $\mathcal{S}_i$ and let $\widehat{\mathbf{W}}(i) \leftarrow \mathcal{P}_r^\epsilon(\mathbf{V}(i))$ denote the rank-r ϵ-approximation to $\mathbf{V}(i)$. Then:*

$$\left\|\widehat{\mathbf{W}}(i) - \mathbf{V}(i)\right\|_F^2 \leq (1+\epsilon)\left\|\mathcal{P}_r(\mathbf{V}(i)) - \mathbf{V}(i)\right\|_F \leq (1+\epsilon)\left\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\right\|_F \leq (1+\epsilon)\left\|\mathbf{V}(i) - \mathbf{X}^\star\right\|_F$$

MATRIX ALPS I using ϵ-approximate low-rank projection via column subset selection

Using ϵ-approximate SVD in MATRIX ALPS I, the following iteration invariant theorem holds:

Theorem 9. *[Iteration invariant with ϵ-approximate projections for* MATRIX ALPS I*] The $(i+1)$-th matrix estimate $\mathbf{X}(i+1)$ of* MATRIX ALPS I *with ϵ-approximate projections $\mathcal{D}_i^\epsilon \leftarrow \mathcal{P}_r^\epsilon\big(\mathcal{P}_{\mathcal{X}_i^\perp}\nabla f(\mathbf{X}(i))\big)$ and $\widehat{\mathbf{W}}(i) \leftarrow \mathcal{P}_r^\epsilon(\mathbf{V}(i))$ in Algorithm 8 satisfies the following recursion:*

$$\big\|\mathbf{X}(i+1)-\mathbf{X}^\star\big\|_F \leq \rho\big\|\mathbf{X}(i)-\mathbf{X}^\star\big\|_F + \gamma\|\boldsymbol{\varepsilon}\|_2 + \beta\lambda, \tag{4.24}$$

where $\rho := \left(1+\frac{3\delta_r}{1-\delta_r}\right)(2+\epsilon)\left[(1+\frac{\delta_{3r}}{1-\delta_{2r}})4\delta_{3r}+\frac{2\delta_{2r}}{1-\delta_{2r}}\right]$, $\beta := \left(1+\frac{3\delta_r}{1-\delta_r}\right)(2+\epsilon)\left(1+\frac{\delta_{3r}}{1-\delta_{2r}}\right)2\sqrt{\epsilon}$, and $\gamma := \left(1+\frac{3\delta_r}{1-\delta_r}\right)(2+\epsilon)\left[(1+\frac{\delta_{3r}}{1-\delta_{2r}})\sqrt{2(1+\delta_{2r})}+2\frac{\sqrt{1+\delta_{2r}}}{1-\delta_{2r}}\right]$.

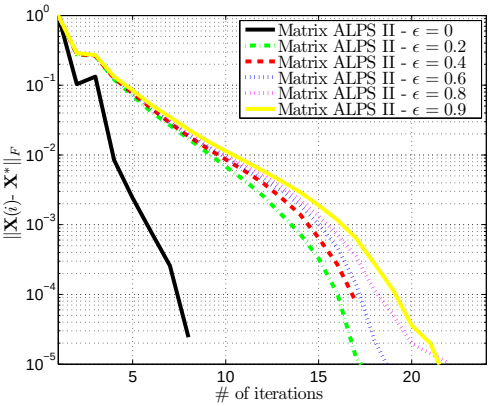
Similar analysis can be conducted for the ADMiRA algorithm.

To illustrate the impact of SVD ϵ-approximation on the signal reconstruction performance of the proposed methods, we replace the *best* rank-r projections in Algorithm 8 by the ϵ-approximation SVD algorithm, presented in [DV06]. In our discussions, the column subset selection algorithm satisfies the following theorem:

Theorem 10. *Let $\mathbf{X} \in \mathbb{R}^{p\times n}$ be a signal of interest with arbitrary rank $< \min\{p,n\}$ and let $\mathbf{X}_r$ represent the best rank-r approximation of $\mathbf{X}$. After $2(r+1)(\log(r+1)+1)$ passes over the data, the Linear Time Low-Rank Matrix Approximation algorithm in [DV06] computes a rank-r approximation $\mathcal{P}_r^\epsilon(\mathbf{X}) \in \mathbb{R}^{p\times n}$ such that Definition 14 is satisfied with probability at least 3/4.*

The proof is provided in [DV06]. In total, Linear Time Low-Rank Matrix Approximation algorithm [DV06] requires $\mathcal{O}(pn(r/\epsilon + r^2\log r) + (p+n)(r^2/\epsilon^2 + r^3\log r/\epsilon + r^4\log^2 r))$ and $\mathcal{O}(\min\{p,n\}(r/\epsilon + r^2\log r))$ time and space complexity, respectively. However, while column subset selection methods such as [DV06] reduce the overall complexity of low-rank projections in theory, in practice this applies only in very high-dimensional settings. To strengthen this argument, in Figure 4.4 we compare SVD-based MATRIX ALPS II with MATRIX ALPS II using the ϵ-approximate column subset selection method in [DV06]. We observe that the total number of iterations for convergence increases due to ϵ-approximate low-rank projections, as expected. Nevertheless, we observe that, on average, the column subset selection process [DV06] is computationally prohibitive compared to regular SVD due to the time overhead in the column selection procedure—fewer passes over the data are desirable in practice to trade-off the increased number of iterations for convergence. In the next section, we present alternatives based on recent trends in randomized matrix decompositions and how we can use them in low-rank recovery.

Figure 4.4: *Performance comparison using ϵ-approximation SVD [DV06] in* MATRIX ALPS II. *$p = n = 256$, $m = 0.4n^2$, rank of $\mathbf{X}^\star$ equals 2 and $\boldsymbol{\mathcal{A}}$ constituted by permuted noiselets. The non-smoothness in the error curves is due to the extreme low rankness of $\mathbf{X}^\star$ for this setting.*



4.3.6 Accelerating MATRIX ALPS: SVD Approximation using Randomized Matrix Decompositions

Algorithm 11 Randomized MATRIX ALPS II with QR Factorization

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, r, q, Tolerance η, MaxIterations
Initialize: $\mathbf{X}(0) \leftarrow 0$, $\mathcal{X}_0 \leftarrow \{\emptyset\}$, $\mathbf{Q}(0) \leftarrow 0$, $\mathcal{Q}_0 \leftarrow \{\emptyset\}$, $\tau_i \; \forall i$, $i \leftarrow 0$
repeat
  $\mathcal{D}_i \leftarrow \text{RANDOMIZEDPOWERITERATION}\big(\mathcal{P}_{\mathcal{Q}_i^{\perp}} \nabla f(\mathbf{Q}(i)),\; r,\; q\big)$ *(Rank-r subspace via Power Iteration)*
  $\mathcal{S}_i \leftarrow \mathcal{D}_i \cup \mathcal{Q}_i$ *(Active subspace expansion)*
  $\mu_i \leftarrow \arg\min_{\mu} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\big(\mathbf{Q}(i) - \frac{\mu}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\big)\big\|_2^2 = \frac{\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\|_F^2}{\|\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))\|_2^2}$ *(Step size selection)*
  $\mathbf{V}(i) \leftarrow \mathbf{Q}(i) - \frac{\mu_i}{2}\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{Q}(i))$ *(Error norm reduction via gradient descent)*
  $\mathcal{W} \leftarrow \text{RANDOMIZEDPOWERITERATION}\big(\mathbf{V}(i),\; r,\; q\big)$ *(Rank-r subspace via Power Iteration)*
  $\mathbf{X}(i+1) \leftarrow \mathcal{P}_{\mathcal{W}}\mathbf{V}(i)$ *(Best rank-r subspace selection)*
  $\mathbf{Q}(i+1) \leftarrow \mathbf{X}(i+1) + \tau_i(\mathbf{X}(i+1) - \mathbf{X}(i))$ *(Momentum update)*
  $\mathcal{Q}_{i+1} \leftarrow \text{ortho}(\mathcal{X}_i \cup \mathcal{X}_{i+1})$
  $i \leftarrow i+1$
until $\|\mathbf{X}(i) - \mathbf{X}(i-1)\|_2 \leq \eta\|\mathbf{X}(i)\|_2$ or MaxIterations.

Finding low-cost SVD approximations to tackle the above complexity issues is a challenging task. Recent works on probabilistic methods for matrix approximation [HMT11] provide a family of efficient approximate projections on the set of rank-deficient matrices with clear computational advantages over regular SVD computation in practice and attractive theoretical guarantees [PKB14]. In this work, we build on the low-cost, power-iteration *subspace tracking* scheme, described in Algorithms 4.3 and 4.4 in [HMT11]. Our proposed algorithm is described in Algorithm 11.

The convergence guarantees of Algorithm 11 follow the same motions described in Section 4.3.5, where ϵ is a function of $p,\; n,\; r$ and q.

4.4 Randomized Low-Memory Singular Value Projection

Our discussions so far evolve around *greedy, first-order methods* for the affine rank minimization problem (4.3): (i) by "greedy method" we refer to using SVD computations to estimate the *current* best low-rank approximation of a given putative anchor matrix point and, (ii) by "first-order method" we refer to the calculation of the gradient of f per iteration Virtually all recovery algorithms require calculating the gradient $\nabla f(\mathbf{X}) = 2\boldsymbol{\mathcal{A}}^*(\boldsymbol{\mathcal{A}}(\mathbf{X}) - \boldsymbol{y})$ at an intermediate iterate $\mathbf{X}$, where $\boldsymbol{\mathcal{A}}^*$ is the adjoint of $\boldsymbol{\mathcal{A}}$. When the range of $\boldsymbol{\mathcal{A}}^*$ is dense, this forces algorithms to use memory proportional to $\mathcal{O}(pn)$, where $\mathbf{X} \in \mathbb{R}^{p\times n}$. Second, after the iterate is updated with the gradient, projecting onto the low-rank space requires a partial singular value decomposition (SVD); see the proposed algorithms in the previous section. Also, this is usually problematic for the first few iterations of convex recovery algorithms, where they may have to perform full SVD's. In our case, greedy algorithms [KC14] fend off the complexity of full SVD's, since they need fixed rank projections, which can be approximated via Lanczos or randomized SVD's [HMT11].

Algorithms that avoid these two issues do exist, such as [WYZ12, RR13, LRS$^+$10], and are typically based on the Burer-Monteiro splitting [BM03]. The main idea in Burer-Monteiro splitting is to remove the non-convex rank constraint by directly embedding into the objective: as opposed to optimizing $\mathbf{X}$, splitting algorithms directly work with its fixed factors $\mathbf{U}\mathbf{V}^T = \mathbf{X}$ in an alternating fashion, where $\mathbf{U} \in \mathbb{R}^{m\times\hat{r}}$ and $\mathbf{V} \in \mathbb{R}^{n\times\hat{r}}$ for some $\hat{r} \geq r$. Unfortunately, rigorous guarantees are difficult.[7] Recent

[7] If $\hat{r} \gtrsim \sqrt{m}$, then [BM03] shows their method obtains a global solution, but this is impractical for large m. Moreover, it is shown

work [JNS13] has shown approximation guarantees if $\mathcal{A}$ satisfies the rank restricted isometry property with constant $\delta_{2r} \leq \kappa^2/(100r)$ (in the noiseless case), where $\kappa = \sigma_1(\mathbf{X}^\star)/\sigma_r(\mathbf{X}^\star)$, or $\delta_{2r} \leq 1/(3200r^2)$ for a bound independent of κ. The authors suggest that these bounds may be tightened, and that practical performance is better than the bound suggests.

In this section, we merge the gradient calculation and the singular value projection steps into one and show that this not only removes a huge computational burden, but suffers only a minor convergence speed drawback in practice. Our contribution is a natural but non-trivial fusion of the Singular Value Projection (SVP) algorithm in [JMD10] and the approximate projection ideas presented in the previous section. The SVP algorithm is a hard-thresholding algorithm that has been considered in [JMD10, GM11]. Inexact steps in SVP have been considered as a heuristic [GM11] but have not been incorporated into a convergence result.[8] In the previous section, we propose a non-convex framework for affine rank minimization (including variants of the SVP algorithm) that utilizes inexact projection operations with provable signal approximation and convergence guarantees. Both [JMD10, KC14] do not consider splitting techniques in the proposed schemes.

Contrary to [JMD10, KC14], we engineer the SVP algorithm to operate like splitting algorithms that *directly work with the factors*; this added twist decreases the per iteration requirements in terms of storage and computational complexity. Using this new formulation, each iteration is nearly as fast as in the splitting method, hence removing a drawback to SVP in relation to splitting methods. Furthermore, we prove that, under some conditions, it is still possible to obtain perfect recovery even if the projections are inexact. In particular, our assumption is that the linear map $\mathcal{A}$ satisfies the rank restricted isometry property, and we give an application that satisfies this assumption, allowing perfect recovery (in the noiseless case) or stable recovery (in the presence of noise) from measurements $m \ll pn$. For example, in the noiseless case, we require approximately $\delta_{2r} \leq 0.0037$. This approach has been used for convex [RFP10] and non-convex [JMD10, KC14] algorithms to obtain approximation guarantees.

Approximate singular value computations

The standard method to compute a partial SVD is the Lanczos method. By itself it is not numerically stable and requires re-orthogonalization and implicit restarts. Excellent implementations are available, but it is a sequential algorithm that calls matrix-vector products. This makes it more difficult to parallelize, which is an issue on modern multi-processor computers. The matrix-vector multiplies are also slower than grouping into matrix-matrix multiplies since it is harder to predict memory usage and this will lead to cache misses; it also precludes the use of theoretically faster algorithms such as Strassen's. Theoretically, there are no known relative error bounds in norm (à la Theorem 11).

As an alternative, we turn to randomized linear algebra. On this front, we restrict ourselves to algorithms that require only multiplications, as opposed to sub-sampling entries/rows/columns, as it is not efficient in practice; see previous section. The randomized approach presented in Algorithm 12 has been rediscovered many times, but has seen a recent resurgence of interest due to theoretical analysis [HMT11]:

that the explicit rank $\hat{r}$ splitting method solves a non-convex problem that has the same local minima as (4.3) (if $\hat{r} = r$). However, the non-convex problems are not *equivalent* (e.g. $\mathbf{U} = \mathbf{0}$, $\mathbf{V} = \mathbf{0}$ is a stationary point for the splitting problem whereas $\mathbf{X} = \mathbf{0}$ is generally not a stationary point for (4.3)). Furthermore, recovery bounds for non-convex algorithms, as in [GK09a] and the present section, are statements about a sequence of iterates of the algorithm, and say nothing about the local minima.

[8] Inexact steps are often incorporated into analysis of algorithms for convex problems. Of particular note, [Lau12] allows inexact eigenvalue computations in a modified Frank-Wolfe algorithm that has applications to (4.3).

Algorithm 12 `RandomizedSVD`

Finds $\mathbf{Q}$ *such that* $X \approx \mathcal{P}_{\mathbf{Q}} X$ *where* $\mathcal{P}_{\mathbf{Q}} = \mathbf{Q}\mathbf{Q}^H$.

Require: Function $\mathtt{h} : \widetilde{\mathbf{Z}} \mapsto \mathbf{X}\widetilde{\mathbf{Z}}$, $\mathtt{h}^{\mathrm{H}} : \widetilde{\mathbf{Q}} \mapsto \mathbf{X}^{\mathrm{H}}\widetilde{\mathbf{Q}}$, $r, q \in \mathbb{N}$ // r: Rank of output, q: # of power iterations

$\ell = r + \rho$ // Typical value of ρ is 5

$\mathbf{\Omega}$ a $n \times \ell$ standard Gaussian matrix

$\mathbf{Q} \leftarrow \mathtt{QR}(\mathtt{h}(\mathbf{\Omega}))$ // The QR algorithm to orthogonalize $\mathbf{W}$

for $j = 1, 2, \ldots, q$ **do**

  $\mathbf{Z} \leftarrow \mathtt{QR}(\mathtt{h}^{\mathrm{H}}(\mathbf{Q}))$

  $\mathbf{Q} \leftarrow \mathtt{QR}(\mathtt{h}(\mathbf{Z}))$

end for

$\mathbf{Z} \leftarrow \mathtt{h}^{\mathrm{H}}(\mathbf{Q})$

$(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow \mathtt{factoredSVD}(\mathbf{Q}, \mathbf{I}_\ell, \mathbf{Z})$ // $\widetilde{\mathbf{X}}_{i+1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{H}}$ in the appendix

Let $\mathbf{\Sigma}_r$ be the best rank r approximation of $\mathbf{\Sigma}$

return $(\mathbf{U}, \mathbf{\Sigma}_r, \mathbf{V})$ // $\mathbf{X}_{i+1} = \mathbf{U}\mathbf{\Sigma}_r\mathbf{V}^{\mathrm{H}}$ in the appendix

Algorithm 13 $\mathtt{factoredSVD}(\widetilde{\mathbf{U}}, \widetilde{\mathbf{D}}, \widetilde{\mathbf{V}})$

Computes the SVD $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ *of the matrix* $\mathbf{X}$ *implicitly given by* $\mathbf{X} = \widetilde{\mathbf{U}}\widetilde{\mathbf{D}}\widetilde{\mathbf{V}}^H$

$(\mathbf{U}, \mathbf{R}_{\mathbf{U}}) \leftarrow \mathtt{QR}(\widetilde{\mathbf{U}})$ and $(\mathbf{V}, \mathbf{R}_{\mathbf{V}}) \leftarrow \mathtt{QR}(\widetilde{\mathbf{V}})$

$(\widehat{\mathbf{U}}, \mathbf{\Sigma}, \widehat{\mathbf{V}}) \leftarrow \mathtt{DenseSVD}(\mathbf{R}_{\mathbf{U}}\widetilde{\mathbf{D}}\mathbf{R}_{\mathbf{V}}^{\mathrm{H}})$

return $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow (\mathbf{U}\widehat{\mathbf{U}}, \mathbf{\Sigma}, \mathbf{V}\widehat{\mathbf{V}})$

Theorem 11 (Average Frobenius error). *Suppose* $\mathbf{X} \in \mathbb{R}^{p \times n}$*, and choose a target rank* r *and oversampling parameter* $\rho \geq 2$ *where* $\ell := r + \rho \leq \min\{m, n\}$*. Let* $\mathbf{X}_r$ *be the best rank* r *approximation in the Frobenius norm. Calculate* $\mathbf{Q}$ *and* $\mathcal{P}_{\mathbf{Q}}$ *via* `RandomizedSVD` *using* $q = 0$ *and set* $\widetilde{\mathbf{X}} = \mathcal{P}_{\mathbf{Q}}\mathbf{X}$ *(which is rank* ℓ*). Then*

$$\mathbf{E}\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 \leq (1 + \epsilon)\,\|\mathbf{X} - \mathbf{X}_r\|_F^2 \text{ where } \epsilon = \frac{r}{\rho - 1}.$$

The theorem follows from the proof of Thm. 10.5 in [HMT11] (note that Thm. 10.5 is stated in terms of $\mathbf{E}\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F$ which is not the same as $\sqrt{\mathbf{E}\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2}$). The expectation is with respect to the Gaussian r.v. in `RandomizedSVD`. For the sake of our analysis, we cannot immediately truncate $\widetilde{\mathbf{X}}$ to rank r since then the error bound in [HMT11] is not tight enough. Thus, since $\widetilde{\mathbf{X}}$ is rank ℓ, in practice we even observe that $\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 < \|\mathbf{X} - \mathbf{X}_r\|_F^2$, especially for small r, as shown in Figure 4.13 later in the text. The figure also shows that using $q > 0$ power iterations is extremely helpful, though this is not taken into account in our analysis since there are no useful theoretical bounds (in the Frobenius norm). Note that variants for eigenvalues also exist; we refer to the equivalent of `RandomizedSVD` as `RandomizedEIG`, which has the property that $\mathbf{U} = \mathbf{V}$ and $\mathbf{\Sigma}$ need not be positive (cf., [HMT11, GM13])

Additional convex constraints

Consider the variant of (4.3):

$$\begin{aligned} \underset{\mathbf{X} \in \mathbb{R}^{m \times n}}{\text{minimize}} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \operatorname{rank}(\mathbf{X}) \leq r,\ \mathbf{X} \in \mathcal{C}, \end{aligned} \tag{4.25}$$

for a convex set $\mathcal{C}$. Our main interests are $\mathcal{C}_+ = \{\mathbf{X} : \mathbf{X} \succeq 0\}$ and the matrix simplex $\mathcal{C}_\Delta = \{\mathbf{X} : \mathbf{X} \succeq 0,\ \mathrm{trace}(\mathbf{X}) = 1\}$. In both cases the constraints are unitarily invariant and the projection onto these sets can be done by taking the eigenvalue decomposition and projecting the eigenvalues. Furthermore, for these specific $\mathcal{C}$, $\mathcal{P}_{\{\mathbf{X}:\mathrm{rank}(\mathbf{X})\leq r\}\cap\mathcal{C}} = \mathcal{P}_\mathcal{C} \circ \mathcal{P}_r$ (this is not obvious; see [KBCK13]).[9]

In general, any convex set $\mathcal{C}$ satisfying the above property is compatible with our algorithm, as long as $\mathbf{X}^\star \in \mathcal{C}$. We overload notation to use $\mathcal{P}_\mathcal{C}$ to denote both the projection of $\mathbf{X}$ onto the set as well as the projection of its eigenvalues onto the analogous set.

4.4.1 The RSVP algorithm

Our approach is based on the projected gradient descent algorithm:

$$\mathbf{X}_{i+1} = \mathcal{P}_r^\epsilon(\mathbf{X}_{i+1} - \mu_i \nabla f(\mathbf{X}_i)), \tag{4.26}$$

where $\mathbf{X}_i$ is the i-th iterate, $\nabla f(\cdot)$ is the gradient of the loss function, μ_i is a step-size, and $\mathcal{P}_r^\epsilon(\cdot)$ is the approximate projector onto rank r matrices given by `RandomizedSVD`. If we include a convex constraint $\mathcal{C}$, then the iteration is

$$\mathbf{X}_{i+1} = \mathcal{P}_\mathcal{C}(\mathcal{P}_r^\epsilon(\mathbf{X}_{i+1} - \mu_i \nabla f(\mathbf{X}_i))). \tag{4.27}$$

In practice, Nesterov acceleration improves performance:

$$\mathbf{Y}_{i+1} = (1+\beta_i)\mathbf{X}_i - \beta_i \mathbf{X}_{i-1} \tag{4.28}$$

$$\mathbf{X}_{i+1} = \mathcal{P}(\mathbf{Y}_i - \mu_i \nabla f(\mathbf{Y}_i)), \tag{4.29}$$

where β_i is chosen $\beta_i = (\alpha_{i-1}-1)/\alpha_i$ and $\alpha_0 = 1$, $2\alpha_{i+1} = 1+\sqrt{4\alpha_i^2+1}$ [Nes83] (see [KC14]). Theorem 12 holds for a stepsize μ_i based on the RIP constant, which is unknown. In practice, the algorithm consistently converges as long as $\mu_i \lesssim \frac{2}{\|\boldsymbol{\mathcal{A}}\|^2}$.

Algorithm 14 shows implementation details that are important for keeping low-memory requirements. The implementation of maps like `A` and `At` depends on the structure of $\boldsymbol{\mathcal{A}}$.

4.4.2 Convergence guarantees for RSVP

We assume the observations are generated by $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is a noise term, not to be confused with the approximation error ϵ. In the following theorem, we will assume that $\|\boldsymbol{\mathcal{A}}\|^2 \leq pn/m$, which is true for the quantum tomography example [Liu11]; if $\boldsymbol{\mathcal{A}}$ is a normalized Gaussian, then this assumption holds in expectation.

[9] This formula is literally true for $\mathcal{C}_+$ and $\{\mathbf{X} : \mathbf{X} \succeq 0,\ \mathrm{trace}(\mathbf{X}) \leq 1\}$. For $\mathcal{C} = \{\mathbf{X} : \mathbf{X} \succeq 0,\ \mathrm{trace}(\mathbf{X}) = 1\}$ constraints, $\mathcal{P}_\mathcal{C}$ can increase the rank, so formally we must work on a restricted subspace and then embed back in the larger space, but this poses no theoretical issues.

Algorithm 14 Efficient implementation of SVP, $\mathcal{K} = \{\mathbb{R}, \mathbb{C}\}$

Require: step-size $\mu > 0$, measurements $\boldsymbol{y}$, initial points $\mathbf{U}_0 \in \mathcal{K}^{m\times r},\ \mathbf{V}_0 \in \mathcal{K}^{n\times r},\ \mathbf{D}_0 \in \mathcal{K}^r$
Require: (optional) unitarily invariant convex set $\mathcal{C}$
Require: Function $\mathtt{A} : (\mathbf{U}, \mathbf{D}, \mathbf{V}) \mapsto \boldsymbol{\mathcal{A}}(\mathbf{U}\operatorname{diag}(\mathbf{D})\mathbf{V}^{\mathrm{H}})$
Require: Function $\mathtt{At} : (\mathbf{z}, \mathbf{W}) \mapsto \boldsymbol{\mathcal{A}}^*(\mathbf{z})\mathbf{W}$
Require: Function $\mathtt{At}^{\mathrm{H}} : (\mathbf{z}, \mathbf{W}) \mapsto (\boldsymbol{\mathcal{A}}^*(\mathbf{z}))^{\mathrm{H}}\mathbf{W}$
1: $\mathbf{V}_{-1} \leftarrow 0$, $\mathbf{U}_{-1} \leftarrow 0$, $\mathbf{D}_{-1} \leftarrow 0$
2: **for** $i = 0, 1, \ldots$ **do**
3: Compute β_i // See text
4: $\mathbf{U}_y \leftarrow [\mathbf{U}_i, \mathbf{U}_{i-1}]$, $\mathbf{V}_y \leftarrow [\mathbf{V}_i, \mathbf{V}_{i-1}]$
5: $\mathbf{D}_y \leftarrow [(1+\beta_i)\mathbf{D}_i, -\beta_i\mathbf{D}_{i-1}]$
6: $\mathbf{z} \leftarrow \boldsymbol{\mathcal{A}}(\mathbf{U}_y, \mathbf{D}_y, \mathbf{V}_y) - \boldsymbol{y}$ // Compute the residual
7: Define the functions
$\mathtt{h} : \mathbf{W} \mapsto \mathbf{U}_y \operatorname{diag}(\mathbf{D}_y)\mathbf{V}_y^{\mathrm{H}}\mathbf{W} - \mu\mathtt{At}(\mathbf{z}, \mathbf{W})$
$\mathtt{h}^{\mathrm{H}} : \mathbf{W} \mapsto \mathbf{V}_y \operatorname{diag}(\mathbf{D}_y)\mathbf{U}_y^{\mathrm{H}}\mathbf{W} - \mu\mathtt{At}^{\mathrm{H}}(\mathbf{z}, \mathbf{W})$
8: $(\mathbf{U}_{i+1}, \mathbf{D}_{i+1}, \mathbf{V}_{i+1}) \leftarrow \mathtt{RandomizedSVD}(\mathtt{h}, \mathtt{h}^{\mathrm{H}}, r)$ or $(\mathbf{U}_{i+1}, \mathbf{D}_{i+1}, \mathbf{U}_{i+1}) \leftarrow \mathtt{RandomizedEIG}(\mathtt{h}, \mathtt{h}^{\mathrm{H}}, r)$
9: $\mathbf{D}_{i+1} \leftarrow \mathcal{P}_{\mathcal{C}}(\mathbf{D}_{i+1})$ // Optional
10: **end for**
11: **return** $X \leftarrow \mathbf{U}_i\mathbf{D}_i\mathbf{V}_i^{\mathrm{H}}$ // If desired

Theorem 12. *(Iteration invariant) Pick an accuracy $\epsilon = \frac{r}{\rho - 1}$, where ρ is defined as in Theorem 11. Define $\ell = r + \rho$ and let c be an integer such that $\ell = (c-1)r$. Let $\mu_i = \frac{1}{2(1+\delta_{cr})}$ in* (4.26) *and assume $\|\boldsymbol{\mathcal{A}}\|^2 \leq pn/m$ and $f(\mathbf{X}_i) > C^2\|\boldsymbol{\varepsilon}\|^2$, where $C \geq 4$ is a constant. Then the descent scheme* (4.26) *or* (4.27) *has the following iteration invariant*

$$\mathbf{E} f(\mathbf{X}_{i+1}) \leq \theta f(\mathbf{X}_i) + \tau\|\boldsymbol{\varepsilon}\|^2, \tag{4.30}$$

in expectation, where

$$\theta \leq 12 \cdot \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(\frac{\epsilon}{1+\delta_{cr}} \cdot \frac{pn}{m} + (1+\epsilon)\frac{3\delta_{cr}}{1-\delta_{2r}} \right),$$

and

$$\tau \leq \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(12 \cdot (1+\epsilon)\left(1 + \frac{2\delta_{cr}}{1-\delta_{2r}}\right) + 8 \right).$$

The expectation is taken with respect to Gaussian random designs in `RandomizedSVD`*. If $\theta \leq \theta_\infty < 1$ for all iterations, then $\lim_{i\to\infty} \mathbf{E} f(\mathbf{X}_i) \leq \max\{C^2, \frac{\tau}{1-\theta_\infty}\}\|\boldsymbol{\varepsilon}\|^2$.*

Each call to `RandomizedSVD` draws a new Gaussian random variable, so the expected value does not depend on previous iterations. By Corollary 3.4 in [NT09a], $\delta_{cr} \leq c \cdot \delta_{2r}$, which allows us to put θ and τ in terms of δ_{2r} if desired, at a slight expense in sharpness.

The expected value of the function converges linearly at rate θ to within a constant of the noise level, and in particular, it converges to zero when there is no noise since C and τ are finite. Note that convergence of the iterates follows from convergence of the function f:

Corollary 5. *If $f(\mathbf{X}_i) \leq \gamma$, then $\|\mathbf{X}_i - \mathbf{X}^\star\|_F^2 \leq \frac{(\sqrt{\gamma} + \|\boldsymbol{\varepsilon}\|_2)^2}{1-\delta_{2r}}$.*

Proof. By the R-RIP and the triangle inequality,

$$\begin{aligned}\sqrt{1+\delta_{2r}(\boldsymbol{\mathcal{A}})}\|\mathbf{X}_i-\mathbf{X}^\star\|_F &\le \|\boldsymbol{\mathcal{A}}(\mathbf{X}_i)-\boldsymbol{\mathcal{A}}(\mathbf{X}^\star)\|_2\\ &= \|(\boldsymbol{\mathcal{A}}(\mathbf{X}_i)-\boldsymbol{y})-(\boldsymbol{\mathcal{A}}(\mathbf{X}^\star)-\boldsymbol{y})\|_2\\ &\le \|(\boldsymbol{\mathcal{A}}(\mathbf{X}_i)-\boldsymbol{y})\|_2+\|\boldsymbol{\varepsilon}\|_2\\ &\le \sqrt{\gamma}+\|\boldsymbol{\varepsilon}\|_2\end{aligned}$$

□

Corollary 6 (Exact computation). *If $\epsilon = 0$ and there is no additional convex constraint $\mathcal{C}$, then $\theta = \frac{2\delta_{2r}}{1-\delta_{2r}}(1+\frac{2}{C})$ and $\tau = 1+\frac{2\delta_{2r}}{1-\delta_{2r}}$, hence $\theta < 1$ if $\delta_{2r} < \frac{1}{3+4/C}$.*

Corollary 6 shows that without the approximate SVD, the R-RIP constants are quite reasonable. For example, with exact computation and no noise, any value of $\delta_{2r} < 1/3$ implies that $\lim_{i\to\infty}\mathbf{X}_i = \mathbf{X}^\star$. With noise, choosing $C = 4$ gives $\delta_{2r} = 1/5$ and $\theta = 3/4$, $\tau = 3/2$ and thus $\lim_{i\to\infty} f(\mathbf{X}_i) \le \max\{16,6\}\|\boldsymbol{\varepsilon}\|^2$.

Note that the theorem gives pessimistic values for ϵ. We want the bound on θ to be less than 1 in order to have a contraction, so we need

$$\underbrace{12\cdot\frac{1+\delta_{2r}}{1-\delta_{cr}}\cdot\frac{\epsilon}{1+\delta_{cr}}\cdot\frac{pn}{m}}_{\text{I}}+\underbrace{12(1+\epsilon)\cdot\frac{1+\delta_{2r}}{1-\delta_{cr}}\cdot\frac{3\delta_{cr}}{1-\delta_{2r}}}_{\text{II}}<1$$

For a rough analysis, we will give approximate conditions so that each of the I and II terms is less than 0.5. It is clear that the terms blow up if $\delta_{cr}\to 1$, so we will assume $\delta_{cr} \ll 1$ (and hence $\delta_{2r} \ll 1$). Then setting $1+\delta_{2r}\approx 1$ in the numerator of I, we require that

$$\frac{12}{1-\delta_{cr}^2}\cdot\frac{\epsilon\, pn}{m} < \frac{1}{2} \tag{4.31}$$

which means that we need $\epsilon \lesssim \frac{m}{24pn}$. For quantum tomography, $p = n$ and $m = \mathcal{O}(rn)$, so we require $\epsilon \lesssim \mathcal{O}(r/n)$. From Theorem 11, our bound on ϵ is $r/(\rho-1)$, so we require $\rho \simeq n$, which defeats the purpose of the randomized algorithm (in this case, one would just do a dense SVD). Numerical examples in the next section will show that ρ can be nearly a small constant, so the theory is not sharp.

For the II term, again approximate $1+\delta_{2r}\approx 1$ and then multiply the denominators and ignore the $\delta_{cr}\delta_{2r}$ term to get

$$72\delta_{cr}(1+\epsilon) \lesssim 1-\delta_{2r}-\delta_{cr}. \tag{4.32}$$

Since certainly $\epsilon \le 0.5$ and $\delta_{2r}+\delta_{cr} \le 0.5$, a sufficient condition is $\delta_{cr} < 1/216$, which is reasonable (cf. [JNS13]).

4.5 Solving the Robust PCA problem with Matrix ALPS

Robust Principal Component Analysis (RPCA) [CLMW11] deals with the challenge of recovering a low rank and a sparse matrix component from a *complete* data matrix. Here, we consider its generalization according to the following problem definition.

PROBLEM 5.2. *Given a linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ and a set of observations $\boldsymbol{y} \in \mathbb{R}^m$ (usually $m \ll p \cdot n$):*

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star + \boldsymbol{\varepsilon},$$

where $\mathbf{X}^\star := \mathbf{L}^\star + \mathbf{M}^\star \in \mathbb{R}^{p\times n}$ is the superposition of a rank-r $\mathbf{L}^\star$ and a k-sparse $\mathbf{M}^\star$ component that we desire to recover, identify a matrix $\widehat{\mathbf{L}} \in \mathbb{R}^{p\times n}$ of rank (at most) r and a matrix $\widehat{\mathbf{M}} \in \mathbb{R}^{p\times n}$ with sparsity level $\big\|\widehat{\mathbf{M}}\big\|_0 \leq k$ such that:

$$\big\{\widehat{\mathbf{L}},\, \widehat{\mathbf{M}}\big\} = \underset{\mathbf{L},\, \mathbf{M}:\, \mathrm{rank}(\mathbf{L})\leq r,\, \|\mathbf{M}\|_0\leq k}{\arg\min} \big\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{L}+\mathbf{M})\big\|_2. \tag{4.33}$$

Here, $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ represents the potential noise term. For different linear operator $\boldsymbol{\mathcal{A}}$ and signal $\mathbf{X}^\star$ configurations, the above problem arises in various research fields. In the case of RPCA, we acquire a finite set of observations $\mathbf{Y} \in \mathbb{R}^{p\times n}$ according to $\mathbf{Y} = \mathbf{L}^\star + \mathbf{M}^\star$ with $\mathbf{L}^\star \in \mathbb{R}^{p\times n}$ and $\mathbf{M}^\star \in \mathbb{R}^{p\times n}$, defined above. The "robust" characterization of the RPCA problem refers to $\mathbf{M}^\star$ having *gross* non-zero entries with *arbitrary* energy. Under mild assumptions concerning the incoherence between $\mathbf{L}^\star$ and $\mathbf{M}^\star$ [CLMW11], we can efficiently reconstruct both the low-rank and sparse components using convex and non-convex optimization approaches [CLMW11, ZT11].

While solving the RPCA problem itself is a difficult task, here we assume: (i) $\boldsymbol{\mathcal{A}}$ is an arbitrary linear operator satisfying both sparse- and rank-RIP (this assumption includes the identity linear map of RPCA as a special case) and, (ii) the total number of observations in $\boldsymbol{y}$ is much less compared to the total number of variables we want to recover, i.e., $m \ll p \cdot n$. Before we present our algorithm and its analysis, we note the following. The reconstruction of both $\mathbf{L}^\star$ and $\mathbf{M}^\star$ from $\boldsymbol{y}$ makes sense under mild conditions on $\mathbf{L}^\star$ and $\mathbf{M}^\star$. Borrowing from [CLMW11], we assume that the low rank component $\mathbf{L}^\star$ is not sparse and uniformly bounded with respect to its singuar vectors and the sparse compoment $\mathbf{M}^\star$ is not low rank with support set uniformly random over the entries of $\mathbf{M}^\star$.

An important ingredient for our matrix analysis is the following lemma—the proof can be found in [WSB11].

Lemma 24. *Let $\mathcal{F}$ be a support set with $|\mathcal{F}| \leq k$ and assume $\mathbf{L} \in \mathbb{R}^{p\times n}$ is a rank-r matrix, satisfying the conditions above. Then, given a general linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ satisfying both sparse- and rank-RIP, we have:*

$$\big\|(\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathbf{L})_{\mathcal{F}}\big\|_F \lesssim \delta_{k+r}(\boldsymbol{\mathcal{A}})\big\|\mathbf{L}\big\|_F, \;\; \textit{for } \min\{p,n\} \gg k \gg r.$$

where $\delta_{k+r}(\boldsymbol{\mathcal{A}})$ denotes the RIP constant of $\boldsymbol{\mathcal{A}}$ over (disjoint) sparse index and low-rank subspace sets where the combined cardinality is less than $k+r$.

Unfortunately, we cannot guarantee that the putative low rank and sparse solutions, i.e., $\mathbf{L}_i$ and $\mathbf{M}_i$, respectively, are uniformly bounded or have random support set patterns, respectively, at each iteration for arbitrary problem configurations. Although the potential optimization problem is non-convex, recent works on non-convex optimization [ABRS10, CIM11] establish mild conditions on the objective function and the regularization terms, that are satisfied in our setting, under which a stationary point to a non-convex problem can be obtained using memory-less or memory-based projected gradient descent methods.

Algorithm 15 MATRIX ALPS Instance

Input: $\boldsymbol{y}$, $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{A}}^*$, Tolerance η, MaxIterations, $\tau_i,\ \forall i$
Initialize: $\{\mathbf{Q}_0, \mathbf{M}_0, \mathbf{L}_0\} \leftarrow 0$, $\{\mathcal{L}_0, \mathcal{M}_0\} \leftarrow \{\emptyset\}$, $i \leftarrow 0$
repeat
  Low rank matrix estimation:
  $\mathcal{D}_i^{\mathcal{L}} \leftarrow \text{ortho}\left(\mathcal{P}_r(\nabla f(\mathbf{Q}_i))\right)$
  $\mathcal{S}_i^{\mathcal{L}} \leftarrow \mathcal{D}_i^{\mathcal{L}} \cup \mathcal{L}_i$
  $\mathbf{V}_i^{\mathcal{L}} \leftarrow \mathbf{Q}_i^{\mathcal{L}} - \frac{\mu_i^{\mathcal{L}}}{2}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\nabla f(\mathbf{Q}_i)$
  $\mathbf{L}_{i+1} \leftarrow \mathcal{P}_r(\mathbf{V}_i^{\mathcal{L}})$ with $\mathcal{L}_{i+1} \leftarrow \text{ortho}\,(\mathbf{L}_{i+1})$
  $\mathbf{Q}_{i+1}^{\mathcal{L}} \leftarrow \mathbf{L}_{i+1} + \tau_i(\mathbf{L}_{i+1} - \mathbf{L}_i)$
  $\mathbf{Q}_{i+1} \leftarrow \mathbf{Q}_{i+1}^{\mathcal{L}} + \mathbf{Q}_i^{\mathcal{M}}$
  Sparse matrix estimation:
  $\mathcal{D}_i^{\mathcal{M}} \leftarrow \text{supp}\left(\mathcal{P}_{\Sigma_k}(\nabla f(\mathbf{Q}_{i+1}))\right)$
  $\mathcal{S}_i^{\mathcal{M}} \leftarrow \mathcal{D}_i^{\mathcal{M}} \cup \mathcal{M}_i$
  $(\mathbf{V}_i^{\mathcal{M}})_{\mathcal{S}_i^{\mathcal{M}}} \leftarrow (\mathbf{Q}_i^{\mathcal{M}})_{\mathcal{S}_i^{\mathcal{M}}} - \frac{\mu_i^{\mathcal{M}}}{2}(\nabla f(\mathbf{Q}_{i+1}))_{\mathcal{S}_i^{\mathcal{M}}}$
  $\mathbf{M}_{i+1} \leftarrow \mathcal{P}_{\Sigma_k}(\mathbf{V}_i^{\mathcal{M}})$ with $\mathcal{M}_{i+1} \leftarrow \text{supp}\,(\mathbf{M}_{i+1})$
  $\mathbf{Q}_{i+1}^{\mathcal{M}} \leftarrow \mathbf{M}_{i+1} + \tau_i(\mathbf{M}_{i+1} - \mathbf{M}_i)$
  $\mathbf{Q}_{i+1} \leftarrow \mathbf{Q}_{i+1}^{\mathcal{L}} + \mathbf{Q}_{i+1}^{\mathcal{M}}$
  $i \leftarrow i + 1$
until $\|\mathbf{Y}_i - \mathbf{Y}_{i-1}\|_2 \leq \eta\|\mathbf{Y}_i\|_2$ or MaxIterations.

4.5.1 The MATRIX ALPS Framework for RPCA

We combine ALPS and MATRIX ALPS ideas for the RPCA case, based on acceleration techniques from convex analysis [Nes83, KC11]. At each iteration, we leverage both low rank and sparse matrix estimates from previous iterations to form a gradient surrogate with low-computational cost. Then, we update the current estimates using memory to gain momentum in convergence as proposed in Nesterov's optimal gradient methods; see Algorithm 15. A key ingredient is the selection of the momentum term τ—constant and adaptive momentum selection strategies can be found in [KC11].

To further improve the convergence speed, we replace the least-squares optimization steps with first-order gradient descent updates—the step size $\mu_i^{\mathcal{L}},\ \mu_i^{\mathcal{M}}$ selections follow from [KC11].

The best projection of an arbitrary matrix onto the set of low rank matrices requires sophisticated matrix decompositions such as Singular Value Decomposition (SVD). Using the Lanczos approach, we require $O(rpn)$ arithmetic operations to compute a rank-r matrix approximation for a given constant accuracy—a prohibitive time-complexity that does not scale well for many practical applications. Alternatives to SVD can be found in [HMT11, ZT11]. Furthermore, [KC14] includes ϵ-approximate low rank matrix projections in the recovery process and study their effects on the convergence.

The following theorem characterizes Algorithm 15 for the noiseless case using a constant momentum step size selection strategy.

Theorem 13. *Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ be a linear operator satisfying rank-RIP and sparse-RIP with constants $\delta_{4r}(\boldsymbol{\mathcal{A}}) \leq 0.09$ and $\delta_{4k}(\boldsymbol{\mathcal{A}}) \leq 0.095$, respectively. Furthermore, assume constant momentum step size selection with $\tau_i = 1/4$, $\forall i$. We consider the noiseless case where the set of observations satisfy $\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star$ for $\mathbf{X}^\star := \mathbf{L}^\star + \mathbf{M}^\star$ as defined in* PROBLEM 5.2. *Then, Algorithm 15 satisfies the following second-order linear system:*

$$\mathbf{x}(i+1) \leq (1+\tau)\boldsymbol{\Delta}\mathbf{x}(i) + \tau\boldsymbol{\Delta}\mathbf{x}(i-1), \tag{4.34}$$

where $\mathbf{x}(i) := \begin{bmatrix} \|\mathbf{L}_i - \mathbf{L}^\star\|_F \\ \|\mathbf{M}_i - \mathbf{M}^\star\|_F \end{bmatrix}$ *and* $\boldsymbol{\Delta} := \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}$ *depends on RIP constants $\delta_{4r}(\boldsymbol{\mathcal{A}})$ and $\delta_{4k}(\boldsymbol{\mathcal{A}})$. Furthermore, the above inequality can be transformed into the following first-order linear system:*

$$\mathbf{w}(i+1) \leq \underbrace{\begin{bmatrix} (1+\tau)\boldsymbol{\Delta} & \tau\boldsymbol{\Delta} \\ \mathbb{I} & \mathbf{0} \end{bmatrix}^i}_{\widehat{\boldsymbol{\Delta}}} \mathbf{w}(0), \tag{4.35}$$

for $\mathbf{w}(i) := [\mathbf{x}(i+1) \;\; \mathbf{x}(i)]^T$. We observe that $\lim_{i\to\infty} \mathbf{w}(i) = \mathbf{0}$ since $|\lambda_j(\widehat{\boldsymbol{\Delta}})| \leq 1$, $\forall j$.

4.6 Experiments

4.6.1 List of algorithms

For the ARM problem in **PROBLEM 4.1**, we compare the following algorithms: (i) the Singular Value Projection (SVP) algorithm [MJD10], a non-convex first-order projected gradient descent algorithm with *constant* step size selection (we study the case where $\mu = 1$), (ii) the inexact ALM algorithm [LCM10] based on augmented Langrance multiplier method, (iii) the OptSpace algorithm [KMO10], a gradient descent algorithm on the Grassmann manifold, (iv) the Grassmannian Rank-One Update Subspace Estimation (GROUSE) and the Grassmannian Robust Adaptive Subspace Tracking methods (GRASTA) [BNR10, HBL11], two stochastic gradient descent algorithms that operate on the Grassmannian—moreover, to allay the impact of outliers in the subspace selection step, GRASTA incorporates the augmented Lagrangian of ℓ_1-norm loss function into the Grassmannian optimization framework, (v) the Riemannian Trust Region Matrix Completion algorithm (RTRMC) [BA11], a matrix completion method using first- and second-order Riemannian trust-region approaches, (vi) the Low rank Matrix Fitting algorithm (LMatFit) [WYZ12], a nonlinear successive over-relaxation algorithm and (vii) the algorithms MATRIX ALPS I, ADMiRA [LB10], MATRIX ALPS II and Randomized MATRIX ALPS II with QR Factorization (referred shortly as MATRIX ALPS II with QR) presented in this paper.

For the problem of RPCA in PROBLEM 5.2, we compare MATRIX ALPS II with GoDec [ZT11], a state-of-the-art projected gradient descent algorithm.

4.6.2 Implementation details

To properly compare the algorithms in the above list, we preset a set of parameters that are common. We denote the ratio between the number of observed samples and the number of variables in $\mathbf{X}^\star$ as SR $:= m/(p\cdot n)$ (sampling ratio). Furthemore, we reserve FR to represent the degree of freedom in a rank-r

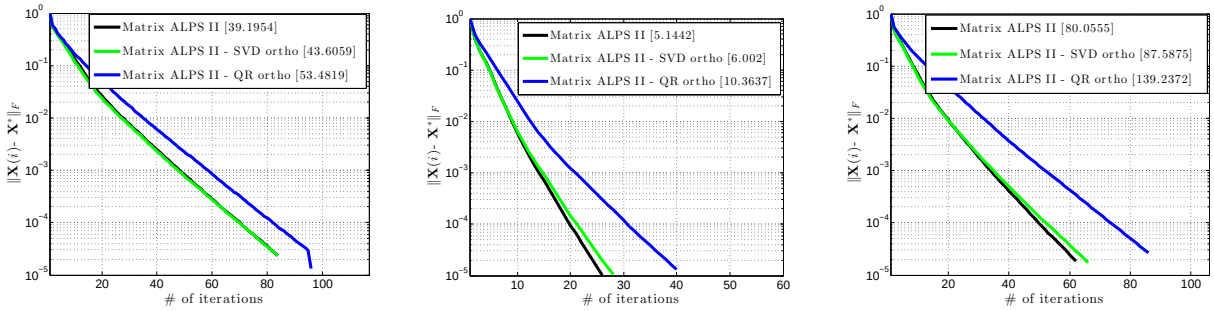


Figure 4.5: *Median error per iteration for* MATRIX ALPS II *variants over 10 Monte-Carlo repetitions. In brackets, we present the mean time consumed for convergene in seconds. (a)* $n = 1024, p = 256, m = 0.25n^2$, *and rank* $r = 20$. *(b)* $n = 2048, p = 512, m = 0.25n^2$, *and rank* $r = 60$. *(c)* $n = 1000, p = 500, m = 0.25n^2$, *and rank* $r = 50$.

matrix to the number of observations—this corresponds to the following definition FR $:= (r(m+n-r))/m$. In most of the experiments, we fix the number of observable data $m = 0.3pn$ and vary the dimensions and the rank r of the matrix $\mathbf{X}^\star$. This way, we create a wide range of different problem configurations with variable FR.

Most of the algorithms in comparison as well as the proposed schemes are implemented in MATLAB. We note that the LMaFit software package contains parts implemented in C that reduce the per iteration computational time. This provides insights for further time savings in our schemes; we leave a fully optimized implementation of our algorithms as future work. In this paper, we mostly test cases where $m \ll n$. Such settings can be easily found in real-world problems such as recommender systems (e.g. Netflix, Amazon, etc.) where the number of products, movies, etc. is much greater than the number of active users.

In all algorithms, we fix the maximum number of iterations to 500, unless otherwise stated. To solve a least squares problem over a restricted low-rank subspace, we use conjugate gradients with maximum number of iterations given by $\mathrm{cg\_maxiter} := 500$ and tolerance parameter $\mathrm{cg\_tol} := 10^{-10}$. We use the same stopping criteria for the majority of algorithms under consideration:

$$\frac{\left\|\boldsymbol{X}(i) - \boldsymbol{X}(i-1)\right\|_F}{\left\|\boldsymbol{X}(i)\right\|_F} \leq \mathrm{tol}, \tag{4.36}$$

where $\boldsymbol{X}(i),\ \boldsymbol{X}(i-1)$ denote the current and the previous estimate of $\mathbf{X}^\star$ and $\mathrm{tol} := 5 \cdot 10^{-5}$. If this is not the case, we tweak the algorithms to minimize the total execution time and achieve similar reconstruction performance as the rest of the algorithms. For SVD calculations, we use the lansvd implementation in PROPACK package [Lar]—moreover, all the algorithms in comparison use the same linear operators $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^*$ for gradient and SVD calculations and conjugate-gradient least-squares minimizations. For fairness, we modified all the algorithms so that they *exploit the true rank*. Small deviations from the true rank result in relatively small degradation in terms of the reconstruction performance. In case the rank of $\mathbf{X}^\star$ is unknown, one has to predict the dimension of the principal singular space. The authors in [MJD10], based on ideas in [KMO10], propose to compute singular values incrementally until a significant gap between singular values is found. Similar strategies can be found in [LCM10] for the convex case.

In MATRIX ALPS II and MATRIX ALPS II with QR, we perform $\mathcal{Q}_i \leftarrow \mathrm{ortho}(\mathcal{X}_i \cup \mathcal{X}_{i+1})$ to construct a set of orthonormal rank-1 matrices that span the subspace, spanned by $\mathcal{X}_i \cup \mathcal{X}_{i+1}$. While such operation can be implemented using factorization procedures (such as SVD or QR decompositions), in practice this

degrades the time complexity of the algorithm substantially as the rank r and the problem dimensionality increase. In our implementations, we simply *union* the set of orthonormal rank-1 matrices, without further orthogonalization. Thus, we employ *inexact* projections for computational efficiency which results in faster convergence. Figure 4.5 shows the time overhead due to the additional orthogonalization process. We compare three algorithms: MATRIX ALPS II (no orthogonalization step), MATRIX ALPS II using SVD for orthogonalization and, MATRIX ALPS II using QR for orthogonalization. In Figures 4.5(a)-(b), we use subsampled and permuted noiselets for linear map $\boldsymbol{\mathcal{A}}$ and in Figure 5(c), we test the MC problem. In all the experimental cases considered in this work, we observed identical performace in terms of reconstruction accuracy for the three variants, as can be also seen in Figure 4.5. To this end, for the rest of the paper, we use MATRIX ALPS II where $\mathcal{Q}_i \leftarrow \mathcal{X}_i \cup \mathcal{X}_{i+1}$.

4.6.3 Synthetic data

General affine rank minimization using noiselets: In this experiment, the set of observations $\boldsymbol{y} \in \mathbb{R}^m$ satisfy:

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star + \boldsymbol{\varepsilon} \tag{4.37}$$

Here, we use permuted and subsampled noiselets for the linear operator $\boldsymbol{\mathcal{A}}$ [WSB11]. The signal $\mathbf{X}^\star$ is generated as the multiplication of two low-rank matrices, $\mathbf{L} \in \mathbb{R}^{p\times r}$ and $\mathbf{R} \in \mathbb{R}^{n\times r}$, such that $\mathbf{X}^\star = \mathbf{L}\mathbf{R}^T$ and $\left\|\mathbf{X}^\star\right\|_F = 1$. Both $\mathbf{L}$ and $\mathbf{R}$ have random independent and identically distributed (iid) Gaussian entries with zero mean and unit variance. In the noisy case, the additive noise term $\boldsymbol{\varepsilon} \in \mathbb{R}^m$ contains entries drawn from a zero mean Gaussian distribution with $\left\|\boldsymbol{\varepsilon}\right\|_2 \in \{10^{-3}, 10^{-4}\}$.

We compare the following algorithms: SVP, ADMiRA, MATRIX ALPS I, MATRIX ALPS II and MATRIX ALPS II with QR for various problem configurations, as depicted in Table 4.1 (there is no available code with arbitrary sensing operators for the rest algorithms). In Table 4.1, we show the median values of reconstruction error, number of iterations and execution time over 50 Monte Carlo iterations. For all cases, we assume SR $= 0.3$ and we set the maximum number of iterations to 500. Bold font denotes the fastest execution time. Furthermore, Figure 4.6 illustrates the effectiveness of the algorithms for some representative problem configurations.

In Table 4.1, MATRIX ALPS II and MATRIX ALPS II with QR obtain accurate low-rank solutions much faster than the rest of the algorithms in comparison. In high dimensional settings, MATRIX ALPS II with QR scales better as the problem dimensions increase, leading to faster convergence. Moreover, its execution time is at least a few orders of magnitude smaller compared to SVP, ADMiRA and MATRIX ALPS I implementations.

Robust matrix completion: We design matrix completion problems in the following way. The signal of interest $\mathbf{X}^\star \in \mathbb{R}^{p\times n}$ is synthesized as a rank-r matrix, factorized as $\mathbf{X}^\star := \mathbf{L}\mathbf{R}^T$ with $\left\|\mathbf{X}^\star\right\|_F = 1$ where $\mathbf{L} \in \mathbb{R}^{p\times r}$ and $\mathbf{R} \in \mathbb{R}^{n\times r}$ as defined above. In sequence, we subsample $\mathbf{X}^\star$ by observing $m = 0.3pn$ entries, drawn uniformly at random. We denote the set of ordered pairs that represent the coordinates of the observable entries as $\Omega = \{(i,j) : [\mathbf{X}^\star]_{ij} \text{ is known}\} \subseteq \{1,\dots,p\} \times \{1,\dots,n\}$ and let $\boldsymbol{\mathcal{A}}_\Omega$ denote the linear operator (mask) that samples a matrix according to Ω. Then, the set of observations satisfies:

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}_\Omega\mathbf{X}^\star + \boldsymbol{\varepsilon}, \tag{4.38}$$

i.e., the known entries of $\mathbf{X}^\star$ are structured as a vector $\boldsymbol{y} \in \mathbb{R}^m$, disturbed by a dense noise vector $\boldsymbol{\varepsilon} \in \mathbb{R}^m$

Table 4.1: General ARM using Noiselets.

Configuration				FR	SVP			ADMiRA			MATRIX ALPS I		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
256	512	5	0	0.097	38	$2.2 \cdot 10^{-4}$	0.78	27	$4.4 \cdot 10^{-5}$	2.26	13.5	$1 \cdot 10^{-5}$	0.7
256	512	5	10^{-3}	0.097	38	$6 \cdot 10^{-4}$	0.91	700	$2 \cdot 10^{-3}$	65.94	16	$7 \cdot 10^{-4}$	0.92
256	512	5	10^{-4}	0.097	38	$2.1 \cdot 10^{-4}$	0.94	700	$4.1 \cdot 10^{-4}$	69.03	11.5	$7.9 \cdot 10^{-5}$	0.72
256	512	10	0	0.193	50	$3.4 \cdot 10^{-4}$	1.44	38	$5 \cdot 10^{-5}$	4.42	13	$3.9 \cdot 10^{-5}$	0.92
256	512	10	10^{-3}	0.193	50	$9 \cdot 10^{-4}$	1.39	700	$1.7 \cdot 10^{-3}$	56.94	29	$1.2 \cdot 10^{-3}$	1.78
256	512	10	10^{-4}	0.193	50	$3.5 \cdot 10^{-4}$	1.38	700	$9.3 \cdot 10^{-5}$	64.69	14	$1.4 \cdot 10^{-4}$	0.93
256	512	20	0	0.38	86	$7 \cdot 10^{-4}$	3.32	700	$4.1 \cdot 10^{-5}$	81.93	45	$2 \cdot 10^{-4}$	4.09
256	512	20	10^{-3}	0.38	86	$1.5 \cdot 10^{-3}$	3.45	700	$4.2 \cdot 10^{-2}$	77.35	69	$2.3 \cdot 10^{-3}$	5.05
256	512	20	10^{-4}	0.38	86	$7 \cdot 10^{-4}$	3.26	700	$4 \cdot 10^{-2}$	79.47	46	$4 \cdot 10^{-4}$	4.1
512	1024	30	0	0.287	66	$4.9 \cdot 10^{-4}$	8.79	295	$5.4 \cdot 10^{-5}$	143.53	24	$1 \cdot 10^{-4}$	8.01
512	1024	40	0	0.38	86	$7 \cdot 10^{-4}$	10.09	700	$4.3 \cdot 10^{-2}$	251.27	45	$2 \cdot 10^{-4}$	11.08
1024	2048	50	0	0.24	57	$4.3 \cdot 10^{-4}$	42.88	103	$5.2 \cdot 10^{-5}$	312.62	18	$5.7 \cdot 10^{-5}$	35.86

					MATRIX ALPS II			MATRIX ALPS II with QR		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time
256	512	5	0	0.097	8	$7.1 \cdot 10^{-6}$	0.42	10	$9.1 \cdot 10^{-6}$	**0.39**
256	512	5	10^{-3}	0.097	9	$7 \cdot 10^{-4}$	**0.56**	20	$7 \cdot 10^{-4}$	0.93
256	512	5	10^{-4}	0.097	8	$7 \cdot 10^{-5}$	0.5	10	$7.8 \cdot 10^{-5}$	**0.46**
256	512	10	0	0.193	10	$2.3 \cdot 10^{-5}$	0.68	13	$2.4 \cdot 10^{-5}$	**0.64**
256	512	10	10^{-3}	0.193	19	$1 \cdot 10^{-3}$	**1.29**	27	$1 \cdot 10^{-3}$	1.35
256	512	10	10^{-4}	0.193	10	$1.1 \cdot 10^{-4}$	0.68	13	$1.1 \cdot 10^{-4}$	**0.62**
256	512	20	0	0.38	21	$1 \cdot 10^{-4}$	1.92	24	$1 \cdot 10^{-4}$	**1.26**
256	512	20	10^{-3}	0.38	36	$1.5 \cdot 10^{-3}$	2.67	39	$1.5 \cdot 10^{-3}$	**1.69**
256	512	20	10^{-4}	0.38	21	$2 \cdot 10^{-4}$	1.87	24	$2 \cdot 10^{-4}$	**1.22**
512	1024	30	0	0.287	14	$4.5 \cdot 10^{-5}$	4.7	18	$3.3 \cdot 10^{-5}$	**4.15**
512	1024	40	0	0.38	21	$1 \cdot 10^{-4}$	6.01	24	$1 \cdot 10^{-4}$	**4.53**
1024	2048	50	0	0.24	12	$2.5 \cdot 10^{-5}$	22.76	15	$3.3 \cdot 10^{-5}$	**17.94**

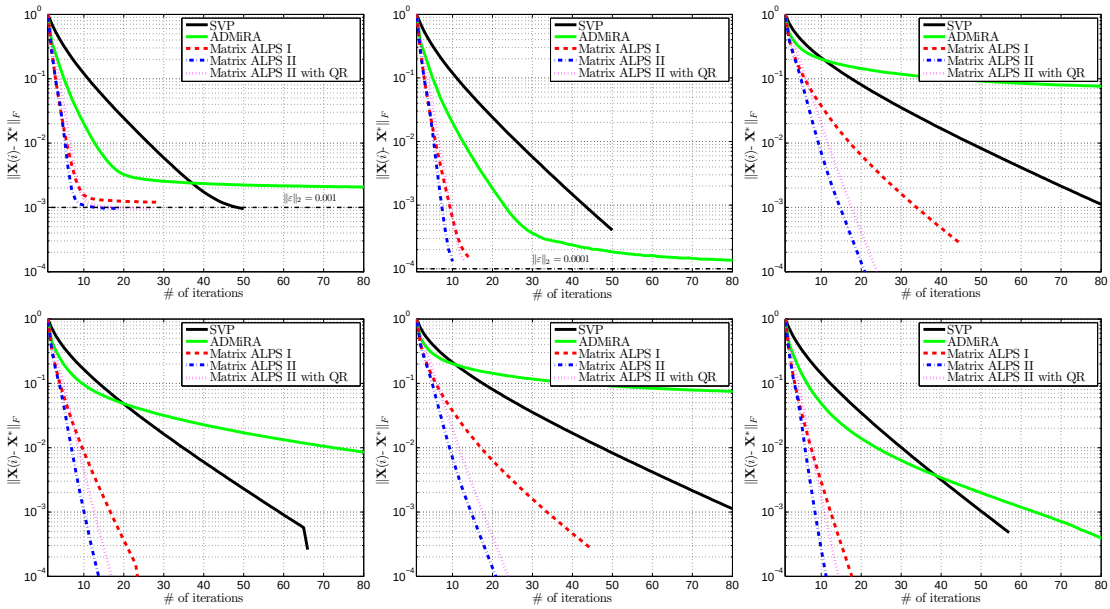


Figure 4.6: *Low rank signal reconstruction using noiselet linear operator. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $m = 0.3pn$. *(a)* $p = 256$, $n = 512$, $r = 10$ *and* $\|\boldsymbol{\epsilon}\|_2 = 10^{-3}$. *(b)* $p = 256$, $n = 512$, $r = 10$ *and* $\|\boldsymbol{\epsilon}\|_2 = 10^{-4}$. *(c)* $p = 256$, $n = 512$, $r = 20$ *and* $\|\boldsymbol{\epsilon}\|_2 = 0$. *(d)* $p = 512$, $n = 1024$, $r = 30$ *and* $\|\boldsymbol{\epsilon}\|_2 = 0$. *(e)* $p = 512$, $n = 1024$, $r = 40$ *and* $\|\boldsymbol{\epsilon}\|_2 = 0$. *(f)* $p = 1024$, $n = 2048$, $r = 50$ *and* $\|\boldsymbol{\epsilon}\|_2 = 0$.

with fixed-energy, which is populated by iid zero-mean Gaussians.

To demonstrate the reconstruction accuracy and the convergence speeds, we generate various problem

configurations (both noisy and noiseless settings), according to (4.38). The energy of the additive noise takes values $\|\boldsymbol{\varepsilon}\|_2 \in \{10^{-3}, 10^{-4}\}$. All the algorithms are tested for the same signal-matrix-noise realizations. A summary of the results can be found in Tables 4.2, 4.3 and, 4.4 where we present the median values of reconstruction error, number of iterations and execution time over 50 Monte Carlo iterations. For all cases, we assume SR $= 0.3$ and set the maximum number of iterations to 700. Bold font denotes the fastest execution time. Some convergence error curves for specific cases are illustrated in Figures 4.7 and 4.8.

In Table 4.2, LMaFit [WYZ12] implementation has the fastest convergence for small scale problem configuration where $p = 300$ and $n = 600$. We note that part of LMaFit implementation uses C code for acceleration. GROUSE [BNR10] is a competitive low-rank recovery method with attractive execution times for the *extreme low rank* problem settings due to stochastic gradient descent techniques. Nevertheless, its execution time performance degrades significantly as we increase the rank of $\mathbf{X}^\star$. Moreover, we observe how randomized low rank projections accelerate the convergence speed where MATRIX ALPS II with QR converges faster than MATRIX ALPS II. In Tables 4.3 and 4.4, we increase the problem dimensions. Here, MATRIX ALPS II with QR has faster convergence for most of the cases and scales well as the problem size increases. We note that we do not exploit stochastic gradient descent techniques in the recovery process to accelerate convergence which is left for future work.

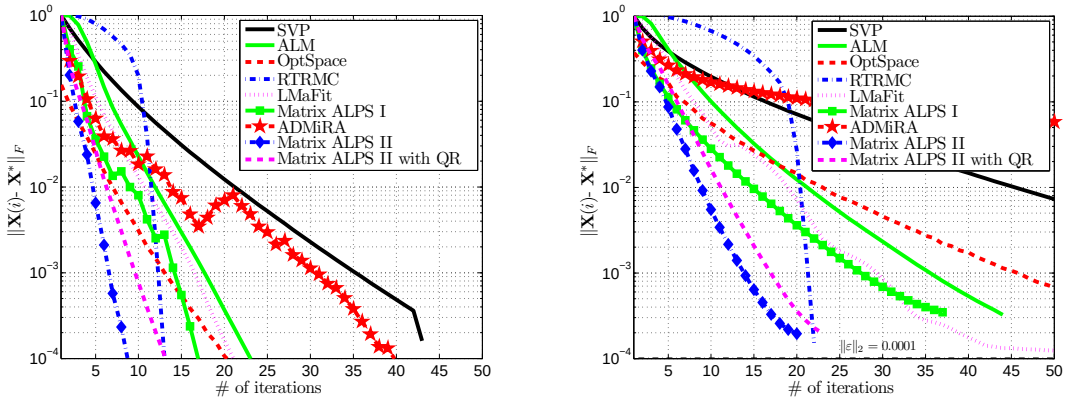


Figure 4.7: *Low rank matrix recovery for the matrix completion problem. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $m = 0.3pn$. *(a)* $p = 300$, $n = 600$, $r = 5$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(b)* $p = 300$, $n = 600$, $r = 20$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-4}$.

4.6.4 Image compression

We use real images to highlight the reconstruction performance of the proposed schemes. In particular, we perform grayscale image denoising from an incomplete set of observed pixels—similar experiments can be found in [WYZ12]. Based on the matrix completion setting, we observe a limited number of pixels from the original image and perform a low rank approximation based only on the set of measurements. While the true underlying image might not be low-rank, we apply our solvers to obtain low-rank approximations.

Figures 4.9 and 4.10 depict the reconstruction results. In the first test case, we use a 512×512 grayscale image as shown in the top left corner of Figure 4.9. For this case, we observe only the 35% of the total number of pixels, randomly selected—a realization is depicted in the top right plot in Figure 4.9. In sequel, we fix the desired rank to $r = 40$. The best rank-40 approximation using SVD is shown in the top

Table 4.2: Matrix Completion problem for $p = 300$ and $n = 600$. "$-$" depicts no information or not applicable due to time overhead.

Configuration				FR	SVP			GROUSE			TFOCS		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	43	$2.9 \cdot 10^{-4}$	0.59	–	$1.52 \cdot 10^{-4}$	0.08	–	$8.69 \cdot 10^{-5}$	3.36
300	600	5	10^{-3}	0.083	42	$6 \cdot 10^{-4}$	0.65	–	$2 \cdot 10^{-4}$	0.082	–	$5 \cdot 10^{-4}$	3.85
300	600	5	10^{-4}	0.083	43	$3 \cdot 10^{-4}$	0.64	–	$2 \cdot 10^{-4}$	0.079	–	$1 \cdot 10^{-4}$	3.5
300	600	10	0	0.165	54	$4 \cdot 10^{-4}$	0.9	–	$4.5 \cdot 10^{-6}$	0.22	–	$2 \cdot 10^{-4}$	6.43
300	600	10	10^{-3}	0.165	54	$9 \cdot 10^{-4}$	0.89	–	$2 \cdot 10^{-4}$	0.16	–	$8 \cdot 10^{-4}$	7.83
300	600	10	10^{-4}	0.165	54	$4 \cdot 10^{-4}$	0.91	–	$2 \cdot 10^{-4}$	0.16	–	$1 \cdot 10^{-4}$	6.75
300	600	20	0	0.326	85	$8 \cdot 10^{-4}$	2.04	–	$1 \cdot 10^{-4}$	0.81	–	$2 \cdot 10^{-4}$	30.04
300	600	40	0	0.637	241	$3.4 \cdot 10^{-3}$	11.1	–	$3.1 \cdot 10^{-3}$	13.94	–	–	–
					Inexact ALM			OptSpace			GRASTA		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	24	$6.7 \cdot 10^{-5}$	0.47	31	$2.8 \cdot 10^{-6}$	2.41	–	$2.2 \cdot 10^{-4}$	2.07
300	600	5	10^{-3}	0.083	24	$6 \cdot 10^{-4}$	0.49	297	$5 \cdot 10^{-4}$	22.82	–	$1 \cdot 10^{-4}$	2.07
300	600	5	10^{-4}	0.083	24	$1 \cdot 10^{-4}$	0.49	267	$1 \cdot 10^{-4}$	21.56	–	$8 \cdot 10^{-5}$	2.1
300	600	10	0	0.165	26	$1 \cdot 10^{-4}$	0.6	37	$2.3 \cdot 10^{-6}$	8.42	–	$8.6 \cdot 10^{-6}$	4.5
300	600	10	10^{-3}	0.165	26	$8 \cdot 10^{-4}$	0.59	304	$8 \cdot 10^{-4}$	66.02	–	$5.5 \cdot 10^{-3}$	3.43
300	600	10	10^{-4}	0.165	26	$1 \cdot 10^{-4}$	0.61	304	$1 \cdot 10^{-4}$	65.56	–	$5.3 \cdot 10^{-3}$	3.44
300	600	20	0	0.326	44	$3 \cdot 10^{-4}$	1.37	–	–	–	–	$5 \cdot 10^{-4}$	10.51
300	600	40	0	0.637	134	$1.6 \cdot 10^{-3}$	7.08	–	–	–	–	$5.2 \cdot 10^{-3}$	251.34
					RTRMC			LMaFit			MATRIX ALPS I		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	13	$1.2 \cdot 10^{-4}$	0.59	20	$2.2 \cdot 10^{-4}$	**0.054**	22	$1.8 \cdot 10^{-5}$	0.76
300	600	5	10^{-3}	0.083	13	$1 \cdot 10^{-4}$	0.59	19	$5 \cdot 10^{-4}$	**0.049**	37	$7 \cdot 10^{-4}$	1.34
300	600	5	10^{-4}	0.083	13	$2 \cdot 10^{-4}$	0.59	21	$1 \cdot 10^{-4}$	**0.052**	18	$1 \cdot 10^{-4}$	0.61
300	600	10	0	0.165	16	$1.1 \cdot 10^{-3}$	1.03	23	$1 \cdot 10^{-4}$	**0.064**	16	$1 \cdot 10^{-4}$	0.65
300	600	10	10^{-3}	0.165	17	$1 \cdot 10^{-4}$	1.09	26	$8 \cdot 10^{-4}$	**0.077**	30	$1.1 \cdot 10^{-3}$	1.16
300	600	10	10^{-4}	0.165	17	$2 \cdot 10^{-4}$	1.09	32	$1 \cdot 10^{-4}$	**0.097**	16	$1 \cdot 10^{-4}$	0.63
300	600	20	0	0.326	22	$4 \cdot 10^{-4}$	2.99	37	$2 \cdot 10^{-4}$	**0.12**	37	$2 \cdot 10^{-4}$	2.05
300	600	40	0	0.637	35	$3 \cdot 10^{-5}$	11.83	233	$4.9 \cdot 10^{-4}$	**2.52**	500	$6.5 \cdot 10^{-2}$	45.67
					ADMiRA			MATRIX ALPS II			MATRIX ALPS II with QR		
p	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
300	600	5	0	0.083	59	$5.2 \cdot 10^{-5}$	2.86	10	$1.7 \cdot 10^{-5}$	0.34	14	$3.2 \cdot 10^{-5}$	0.45
300	600	5	10^{-3}	0.083	700	$4 \cdot 10^{-3}$	30.96	12	$6 \cdot 10^{-4}$	0.44	24	$6 \cdot 10^{-4}$	0.81
300	600	5	10^{-4}	0.083	700	$4.5 \cdot 10^{-3}$	31.45	10	$1 \cdot 10^{-4}$	0.36	14	$1 \cdot 10^{-4}$	0.47
300	600	10	0	0.165	47	$1 \cdot 10^{-3}$	2.56	12	$3 \cdot 10^{-5}$	0.48	16	$3.4 \cdot 10^{-5}$	0.49
300	600	10	10^{-3}	0.165	700	$1.5 \cdot 10^{-3}$	28.49	19	$9 \cdot 10^{-4}$	0.74	29	$9 \cdot 10^{-4}$	0.95
300	600	10	10^{-4}	0.165	700	$1 \cdot 10^{-4}$	31.99	12	$1 \cdot 10^{-4}$	0.49	16	$1 \cdot 10^{-4}$	0.54
300	600	20	0	0.326	700	$1.2 \cdot 10^{-3}$	41.86	20	$1 \cdot 10^{-4}$	1.16	23	$1 \cdot 10^{-4}$	0.79
300	600	20	0	0.326	–	–	–	72	$2 \cdot 10^{-4}$	7.21	68	$2 \cdot 10^{-4}$	2.6

middle of Figure 4.9 where the full set of pixels is observed. Given a fixed common tolerance and the same stopping criteria, Figure 4.9 shows the recovery performance achieved by a range of algorithms. We repeat the same experiment for the second image in Figure 4.10. Here, the size of the image is 256×256, the desired rank is set to $r = 30$ and we observe the 33% of the image pixels. In constrast to the image denoising procedure above, we measure the reconstruction error of the computed solutions with respect to the *best rank*-30 *approximation* of the true image. In both cases, we note that MATRIX ALPS II has a better phase transition performance as compared to the rest of the algorithms.

4.6.5 Quantum tomography

We apply Algorithm 14 to the quantum tomography problem, which is a particular instance of (4.3). For details, we refer to [GLF+10, FGLE12]. The salient features are that the variable $\mathbf{X} \in \mathbb{C}^{n \times n}$ is constrained to be Hermitian positive-definite, and that, unlike many low-rank recovery problems, the linear operator $\boldsymbol{\mathcal{A}}$ satisfies the R-RIP: [Liu11] establishes that Pauli measurements (which comprise $\boldsymbol{\mathcal{A}}$) have R-RIP with overwhelming probability when $m = \mathcal{O}(rn \log^6 n)$. In the ideal case, $\mathbf{X}^\star$ is exactly rank 1, but it may have larger rank due to some (non-Gaussian) noise processes, in addition to AWGN $\boldsymbol{\epsilon}$. Furthermore,

Table 4.3: Matrix Completion problem for $p = 700$ and $n = 1000$. "$-$" depicts no information or not applicable due to time overhead.

Configuration				FR	SVP			Inexact ALM			GROUSE		
m	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
700	1000	5	0	0.04	34	$1.9 \cdot 10^{-4}$	1.77	23	$6.5 \cdot 10^{-5}$	1.69	–	$3.5 \cdot 10^{-5}$	**0.23**
700	1000	5	$10-3$	0.04	34	$4.2 \cdot 10^{-4}$	1.92	23	$3.7 \cdot 10^{-4}$	1.87	–	$3.1 \cdot 10^{-4}$	**0.24**
700	1000	30	0	0.239	61	$4.6 \cdot 10^{-4}$	6.39	29	$1.2 \cdot 10^{-4}$	3.91	–	$3.2 \cdot 10^{-5}$	3.15
700	1000	30	10^{-3}	0.239	61	$1.1 \cdot 10^{-3}$	6.33	29	$1 \cdot 10^{-3}$	3.87	–	$8 \cdot 10^{-4}$	3.14
700	1000	50	0	0.393	95	$8.5 \cdot 10^{-4}$	14.47	49	$3.2 \cdot 10^{-4}$	9.02	–	$1.3 \cdot 10^{-5}$	10.31
700	1000	50	10^{-3}	0.393	95	$1.6 \cdot 10^{-3}$	15.15	49	$1.4 \cdot 10^{-3}$	9.11	–	$8 \cdot 10^{-4}$	10.34
700	1000	110	0	0.833	683	$1.2 \cdot 10^{-2}$	253.1	374	$5.8 \cdot 10^{-3}$	152.61	–	$1.2 \cdot 10^{-1}$	110.93
700	1000	110	10^{-3}	0.833	682	$1.3 \cdot 10^{-2}$	256.21	374	$6.8 \cdot 10^{-3}$	154.34	–	$1.05 \cdot 10^{-1}$	111.05
					LMaFit			MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
700	1000	5	0	0.04	24	$7.2 \cdot 10^{-6}$	0.67	8	$1.5 \cdot 10^{-5}$	1.15	15	$8.3 \cdot 10^{-5}$	1.05
700	1000	5	10^{-3}	0.04	17	$3.7 \cdot 10^{-4}$	0.5	10	$4.5 \cdot 10^{-4}$	1.38	15	$3.8 \cdot 10^{-4}$	1.1
700	1000	30	0	0.239	34	$9.2 \cdot 10^{-6}$	**1.95**	14	$4.5 \cdot 10^{-5}$	3.69	35	$1.1 \cdot 10^{-4}$	2.6
700	1000	30	10^{-3}	0.239	30	$1 \cdot 10^{-3}$	**1.71**	25	$1.1 \cdot 10^{-3}$	6.1	35	$1 \cdot 10^{-3}$	2.61
700	1000	50	0	0.393	53	$2.7 \cdot 10^{-5}$	4.59	25	$8.6 \cdot 10^{-5}$	8.87	57	$1.6 \cdot 10^{-5}$	**4.47**
700	1000	50	10^{-3}	0.393	52	$1.4 \cdot 10^{-3}$	4.53	40	$1.6 \cdot 10^{-3}$	14.38	57	$1.4 \cdot 10^{-3}$	**4.49**
700	1000	110	0	0.833	584	$9 \cdot 10^{-4}$	101.95	280	$8 \cdot 10^{-4}$	214.93	553	$7 \cdot 10^{-4}$	**51.72**
700	1000	110	10^{-3}	0.833	584	$3.7 \cdot 10^{-3}$	102.15	336	$4.7 \cdot 10^{-3}$	261.98	551	$3.7 \cdot 10^{-3}$	**51.62**

Table 4.4: Matrix Completion problem for $p = 500$ and $n = 2000$. "$-$" depicts no information or not applicable due to time overhead.

Configuration				FR	SVP			Inexact ALM			GROUSE		
m	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
500	2000	30	0	0.083	64	$5.3 \cdot 10^{-4}$	10.18	32	$1.9 \cdot 10^{-4}$	6.47	–	$1.6 \cdot 10^{-4}$	**2.46**
500	2000	30	10^{-3}	0.083	64	$1.1 \cdot 10^{-3}$	6.69	32	$1 \cdot 10^{-3}$	4.51	–	$6 \cdot 10^{-4}$	**1.94**
500	2000	30	10^{-4}	0.083	64	$5.4 \cdot 10^{-4}$	10.14	32	$2.2 \cdot 10^{-4}$	6.51	–	$1.6 \cdot 10^{-4}$	**2.46**
500	2000	50	0	0.408	103	$1.1 \cdot 10^{-4}$	15.74	54	$5 \cdot 10^{-4}$	10.8	–	$8 \cdot 10^{-5}$	7.32
500	2000	50	10^{-3}	0.408	103	$1.8 \cdot 10^{-3}$	24.97	54	$1.55 \cdot 10^{-3}$	16.14	–	$9 \cdot 10^{-4}$	8.6
500	2000	50	10^{-4}	0.408	102	$1.1 \cdot 10^{-3}$	24.85	54	$5 \cdot 10^{-4}$	16.17	–	$7 \cdot 10^{-5}$	8.59
500	2000	80	0	0.645	239	$3.5 \cdot 10^{-3}$	92.91	134	$1.7 \cdot 10^{-3}$	59.33	–	$1 \cdot 10^{-4}$	79.64
500	2000	80	10^{-3}	0.645	239	$4.2 \cdot 10^{-3}$	94.86	134	$2.8 \cdot 10^{-3}$	60.68	–	$1 \cdot 10^{-4}$	79.98
500	2000	80	10^{-4}	0.645	239	$3.6 \cdot 10^{-3}$	93.95	134	$1.8 \cdot 10^{-3}$	60.76	–	$1 \cdot 10^{-4}$	79.48
500	2000	100	0	0.8	523	$1.1 \cdot 10^{-2}$	259.13	307	$6 \cdot 10^{-3}$	173.14	–	$4.5 \cdot 10^{-2}$	143.41
500	2000	100	10^{-3}	0.8	525	$1.2 \cdot 10^{-2}$	262.19	308	$7 \cdot 10^{-3}$	176.04	–	$5.2 \cdot 10^{-2}$	142.85
500	2000	100	10^{-4}	0.8	523	$1.1 \cdot 10^{-2}$	262.11	307	$6 \cdot 10^{-3}$	170.47	–	$5.1 \cdot 10^{-2}$	144.78
					LMaFit			MATRIX ALPS II			MATRIX ALPS II with QR		
m	n	r	$\|\boldsymbol{\epsilon}\|_2$		iter.	err.	time	iter.	err.	time	iter.	err.	time
500	2000	30	0	0.083	37	$1.3 \cdot 10^{-5}$	3.05	13	$3.1 \cdot 10^{-5}$	4.84	37	$1.2 \cdot 10^{-5}$	4.04
500	2000	30	10^{-3}	0.083	37	$1 \cdot 10^{-3}$	2.52	22	$1.1 \cdot 10^{-3}$	5.35	37	$1 \cdot 10^{-3}$	3.32
500	2000	30	10^{-4}	0.083	35	$1 \cdot 10^{-4}$	2.86	13	$1.3 \cdot 10^{-4}$	4.85	37	$1.6 \cdot 10^{-4}$	4.05
500	2000	50	0	0.408	60	$6 \cdot 10^{-5}$	6.06	22	$1 \cdot 10^{-4}$	7.6	60	$2 \cdot 10^{-4}$	**5.67**
500	2000	50	10^{-3}	0.408	60	$1.4 \cdot 10^{-3}$	7.26	36	$1.6 \cdot 10^{-3}$	19.64	59	$1.6 \cdot 10^{-3}$	**6.91**
500	2000	50	10^{-4}	0.408	60	$2 \cdot 10^{-4}$	7.29	22	$2 \cdot 10^{-4}$	11.87	59	$2 \cdot 10^{-4}$	**6.75**
500	2000	80	0	0.645	183	$3 \cdot 10^{-4}$	33.65	61	$2 \cdot 10^{-4}$	49.53	151	$3 \cdot 10^{-4}$	**18.66**
500	2000	80	10^{-3}	0.645	183	$2.3 \cdot 10^{-3}$	33.48	92	$2.4 \cdot 10^{-3}$	75.51	151	$2.3 \cdot 10^{-3}$	**18.87**
500	2000	80	10^{-4}	0.645	183	$3 \cdot 10^{-4}$	33.47	61	$4 \cdot 10^{-4}$	49.52	151	$3 \cdot 10^{-4}$	**18.92**
500	2000	100	0	0.8	519	$1.5 \cdot 10^{-3}$	115.11	148	$4 \cdot 10^{-4}$	153.74	429	$7 \cdot 10^{-4}$	**55.1**
500	2000	100	10^{-3}	0.8	529	$3.6 \cdot 10^{-3}$	117.7	228	$3.7 \cdot 10^{-3}$	239.92	427	$3.4 \cdot 10^{-3}$	**55.7**
500	2000	100	10^{-3}	0.8	520	$1.6 \cdot 10^{-3}$	116.66	148	$6 \cdot 10^{-4}$	154.46	428	$8 \cdot 10^{-4}$	**55.07**

it is known that the true solution $\mathbf{X}^\star$ has trace 1, which is also possible to exploit in our algorithmic framework.

Since $\mathbf{X}$ is Hermitian, the $\mathbf{U}$ and $\mathbf{V}$ terms in the algorithm are identical. Several computations can be simplified and there is a version of Algorithm 12 which exploits the positive-definiteness to incorporate a Nyström approximation (and also forces the approximation to be positive-definite); see [HMT11, GM13]. Here, we focus on showing how the functions `A` and `At` can be computed (due to the complex symmetry,

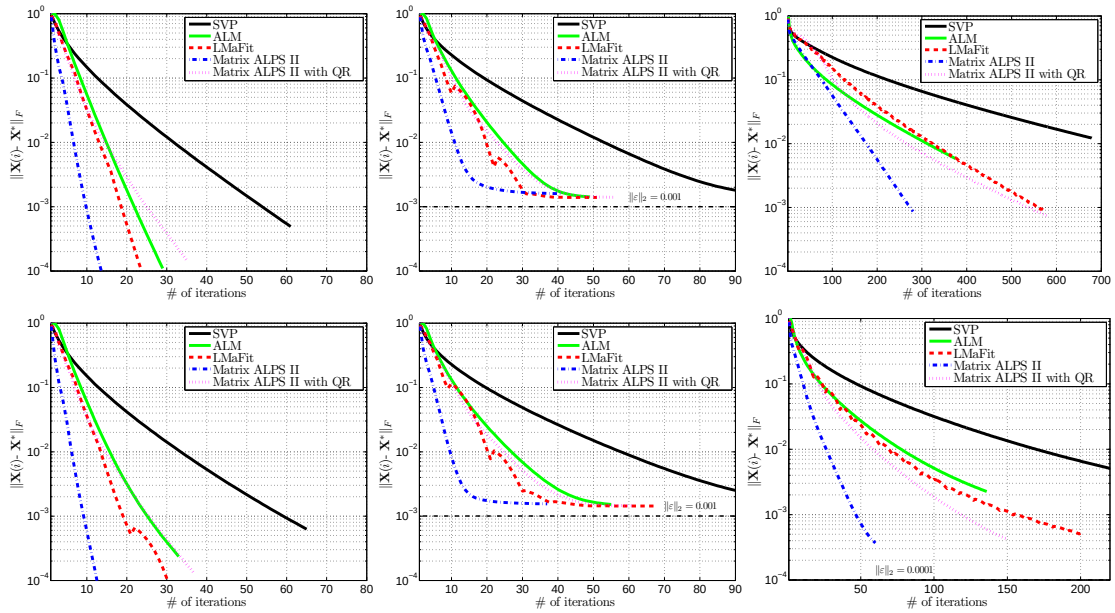


Figure 4.8: *Low rank matrix recovery for the matrix completion problem. The error curves are the median values across 50 Monte-Carlo realizations over each iteration. For all cases, we assume* $m = 0.3pn$. *(a)* $p = 700$, $n = 1000$, $r = 30$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(b)* $p = 700$, $n = 1000$, $r = 50$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-3}$. *(c)* $p = 700$, $n = 1000$, $r = 110$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(d)* $p = 500$, $n = 2000$, $r = 10$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 0$. *(e)* $p = 500$, $n = 2000$, $r = 50$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-3}$. *(f)* $p = 500$, $n = 2000$, $r = 80$ *and* $\|\boldsymbol{\varepsilon}\|_2 = 10^{-4}$.

$\texttt{At}^{\mathrm{H}} = \texttt{At}$).

In quantum tomography, the linear operator has the form $(\boldsymbol{\mathcal{A}}(\mathbf{X}))_j = \langle \mathbf{E}_j, \mathbf{X} \rangle$ where $\mathbf{E}_j = \mathbf{E}_j^{\mathrm{H}}$ is the Kronecker product of 2×2 Pauli matrices. There are four possible Pauli matrices $\sigma_{x,y,z}$ if we define σ_I to be the 2×2 identity matrix. For a q_b-qubit system, $\mathbf{E}_j = \sigma_{j1} \otimes \sigma_{j2} \otimes \ldots \otimes \sigma_{jq_b}$. For roughly 12 qubits and fewer, it is simple to calculate $\boldsymbol{\mathcal{A}}(\mathbf{X})$ by explicitly forming $\mathbf{E}_j$ and then creating a sparse matrix $\mathbf{A}$ with the j^{th} row of $\mathbf{A}$ equal to $\mathrm{vec}(\mathbf{E}_j)$ so that $\boldsymbol{\mathcal{A}}(\mathbf{X}) = \mathbf{A}\,\mathrm{vec}(\mathbf{X})$. For larger systems, storing this sparse matrix is impractical since there are $m \geq n$ rows and each row has exactly n non-zero entries, so there are over n^2 entries in $\mathbf{A}$.

To keep memory low, we exploit the Kronecker-product nature of $\mathbf{E}_j$ and store it with only q_b numbers. When $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{H}}$, we compute $\langle \mathbf{E}_j, \mathbf{X} \rangle = \mathrm{trace}(\mathbf{E}_j \mathbf{x}\mathbf{x}^{\mathrm{H}}) = \mathrm{trace}(\mathbf{x}^{\mathrm{H}} \mathbf{E}_j \mathbf{x})$, and $\mathbf{E}_j \mathbf{x}$ can be computed in $\mathcal{O}(q_b n)$ time. This gives us `A`. The output of `A` is real even when $\mathbf{X}$ is complex.

To compute $\texttt{At}(\mathbf{z}, \mathbf{W})$ when the dimensions are small, we just explicitly form the matrix $\mathbf{M} = \boldsymbol{\mathcal{A}}(\mathbf{z})$ and then multiply $\mathbf{MW}$. To form $\mathbf{M}$, we use the same sparse matrix $\mathbf{A}$ as above and reshape the n^2 vector $\mathbf{A}^* \mathbf{z}$ into a $n \times n$ matrix. For larger dimensions, when it is impractical to store $\mathbf{A}$, we implicitly represent $\mathbf{M} = \sum_{j=1}^m \mathbf{z}_j \mathbf{E}_j$ and thus $\mathbf{MW} = \sum_{j=1}^m \mathbf{z}_j \mathbf{E}_j \mathbf{W}$. In general, the output is complex. However, if it is known *a priori* that $\mathbf{X}$ is real-valued, this can be exploited by taking the real part of $\mathbf{M}$. This leads to a considerable time savings ($2\times$ to $4\times$), and all experiments shown below make this assumption.

In our numerical implementation, we code both `A` and `At` in C and parallelize the code since this is the most computationally expensive calculation. Our parallelization implementation uses both `pthreads` on local cores as well as message passing among different computers. There are two approaches to

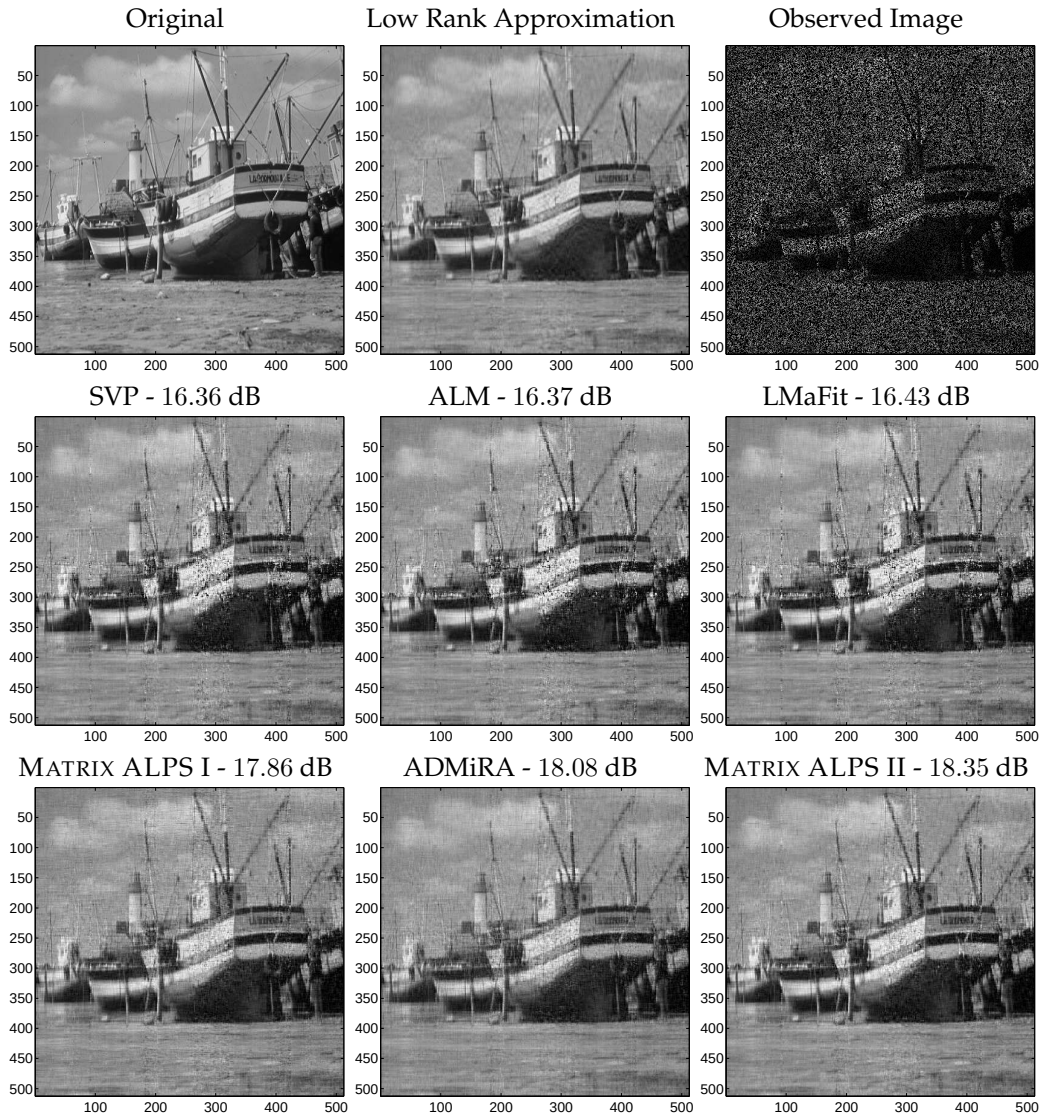


Figure 4.9: Reconstruction performance in image denoising settings. The image size is 512×512 and the desired rank is preset to $r = 40$. We observe 35% of the pixels of the true image. Furthermore, we depict the median reconstruction error with respect to the true image in dB over 10 Monte Carlo realizations.

parallelization: divide the indices $j = 1, \ldots, m$ among different cores, or, when $\mathbf{X}$ or $\mathbf{W}$ has several columns, send different columns to the different cores. Both approaches are efficient in terms of message passing since $\boldsymbol{\mathcal{A}}$ is parameterized and static. The latter approach only works when $\mathbf{X}$ or $\mathbf{W}$ has a significant number of columns, and so it does not apply to Lanczos methods that perform only matrix-vector multiplies.

Recording error metrics can be costly if not done correctly. Let $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{H}}$ and $\mathbf{Y} = \mathbf{y}\mathbf{y}^{\mathrm{H}}$ be rank-r factorizations. For the Frobenius norm error $\|\mathbf{X} - \mathbf{Y}\|_F$ which requires n^2 operations naively, we expand the term and use the cyclic invariance of trace to get $\|\mathbf{X} - \mathbf{Y}\|_F^2 = \mathrm{trace}(\mathbf{x}^{\mathrm{H}}\mathbf{x}\mathbf{x}^{\mathrm{H}}\mathbf{x}) + \mathrm{trace}(\mathbf{y}^{\mathrm{H}}\mathbf{y}\mathbf{y}^{\mathrm{H}}\mathbf{y}) - 2\mathrm{trace}(\mathbf{x}^{\mathrm{H}}\mathbf{y}\mathbf{y}^{\mathrm{H}}\mathbf{x})$, which requires only $\mathcal{O}(nr^2)$ flops. In quantum information, another common metric is the trace distance [NC10] $\|\mathbf{X} - \mathbf{Y}\|_*$, where $\|\cdot\|_*$ is the nuclear norm. This calculation requires $\mathcal{O}(n^3)$ flops if calculated directly but can also be calculated cheaply via `factoredSVD` on $\mathbf{U} = \mathbf{V} = [\mathbf{x}, \mathbf{y}]$ and

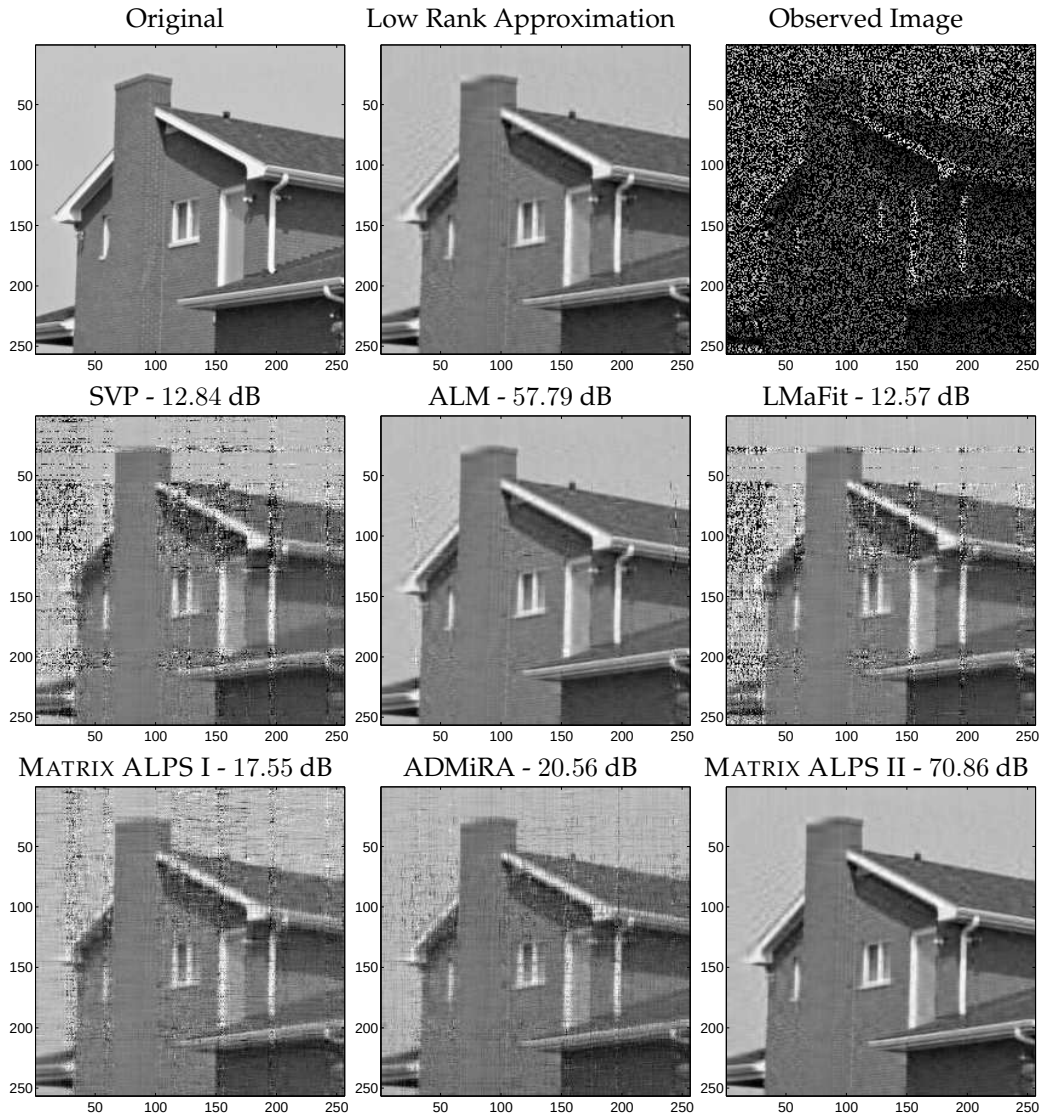


Figure 4.10: Reconstruction performance in image denoising settings. The image size is 256×256 and the desired rank is preset to $r = 30$. We observe 33% of the pixels of the best rank-30 approximation of the image. Furthermore, we depict the median reconstruction with respect to the best rank-30 approximation in dB over 10 Monte Carlo realizations

$\mathbf{D} = [\mathbb{I}, \mathbf{0}; \mathbf{0}, -\mathbb{I}]$. The third common metric is the fidelity [NC10] given by $\|\mathbf{X}^{1/2}\mathbf{Y}^{1/2}\|_*$. If either $\mathbf{X}$ or $\mathbf{Y}$ is rank-1, this can be calculated cheaply as well.

Results: Figure 4.11 (left) plots convergence and accuracy results for a quantum tomography problem with 8 qubits and $m = 4rn$ with $r = 1$. The SVP algorithm works well on noisy problems but we focus here on a noiseless (and truly low-rank) problem in order to examine the effects of approximate SVD/eigenvalue computations. The figure shows that the power method with $q \geq 1$ is extremely effective even though it lacks theoretical guarantees; without the power method, take $\rho \simeq 20$ and we see convergence, albeit slower. When m is smaller and the R-RIP is not satisfied, taking ρ or q too small can lead to non-convergence.

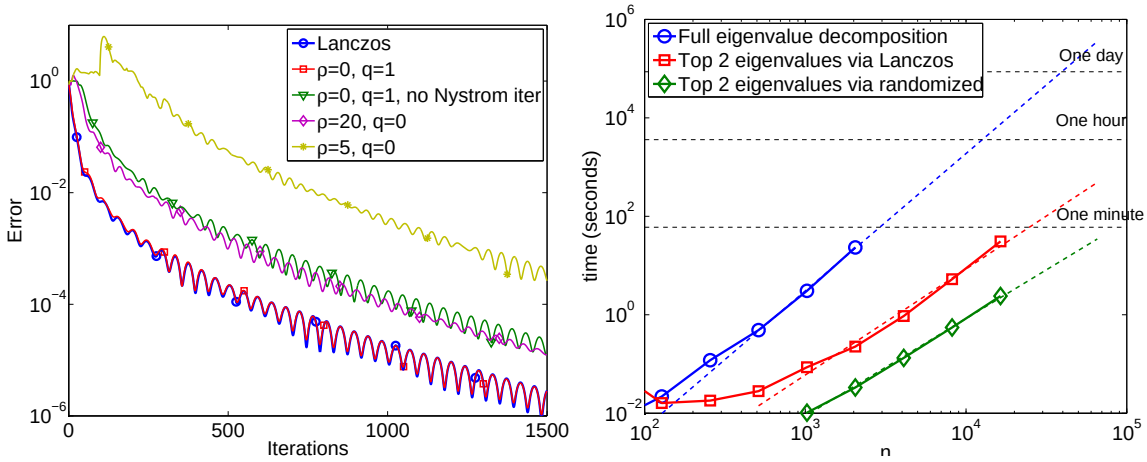


Figure 4.11: (Left) Convergence rate as a function of parameters to `RandomizedSVD`/`RandomizedEIG`. (Right) Comparison of just eigenvalue computation times via three methods.

Figure 4.11 (right) is a direct comparison of `RandomizedEIG` (with $\rho = 5$ and $q = 3$) and the Lanczos method for multiplies of the type encountered in the algorithm. The `RandomizedEIG` has the same asymptotic complexity but much better constants.

Figure 4.12 shows that because the eigenvalue decomposition is a significant portion of the computational cost, using `RandomizedEIG` instead of Lanczos makes a difference. The difference is not pronounced in the small-scale full-memory implementation because the variable $\mathbf{X}$ is explicitly formed and matrix multiplies are relatively cheap compared to other operations in the code. For larger dimensions with the low-memory code, $\mathbf{X}$ is never explicitly formed and multiplying with the gradient is quite costly. The randomized method requires fewer multiplies, explaining its benefit. For 12 qubits, the Lanczos method averages 98.4 seconds/iteration, whereas the randomized method averages just 59.2 seconds. The right subfigure shows that the low-memory implementation (which has memory requirement $\mathcal{O}(rn)$) still has only $\mathcal{O}(n^2)$ time complexity per iteration.

Figure 4.14: Accuracy comparison of several algorithms, as a function of number of samples m. Each point is the median of the results of 20 simulations.

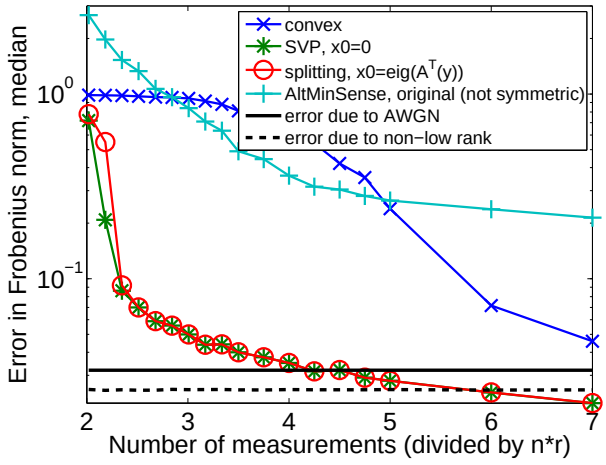


Figure 4.13 tests Theorem 11 by plotting the value of

$$\widetilde{\epsilon} = \|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 / \|\mathbf{X} - \mathbf{X}_r\|_F^2 - 1$$

(which is bounded by ϵ) for matrices $\mathbf{X}$ that are generated by the iterates of the algorithm. The algorithm is set for $r = 1$ (so $\mathbf{X}$ is the sum of a rank 2 term, which includes the Nesterov term, and the full rank gradient), but the plots consider a range of r and a range of oversampling parameters ρ. The plots use $q = 0, 1$ (top row, left to right) and $q = 2$ (bottom row, left) power iterations. Because $\widetilde{\mathbf{X}}$ has rank $\ell = r + \rho$, it is possible for $\widetilde{\epsilon} < 0$, as we observe in the plots when r is small and ρ is large. For two power iterations, the error is excellent. In all cases, the observed error $\widetilde{\epsilon}$ is much better than the bound ϵ

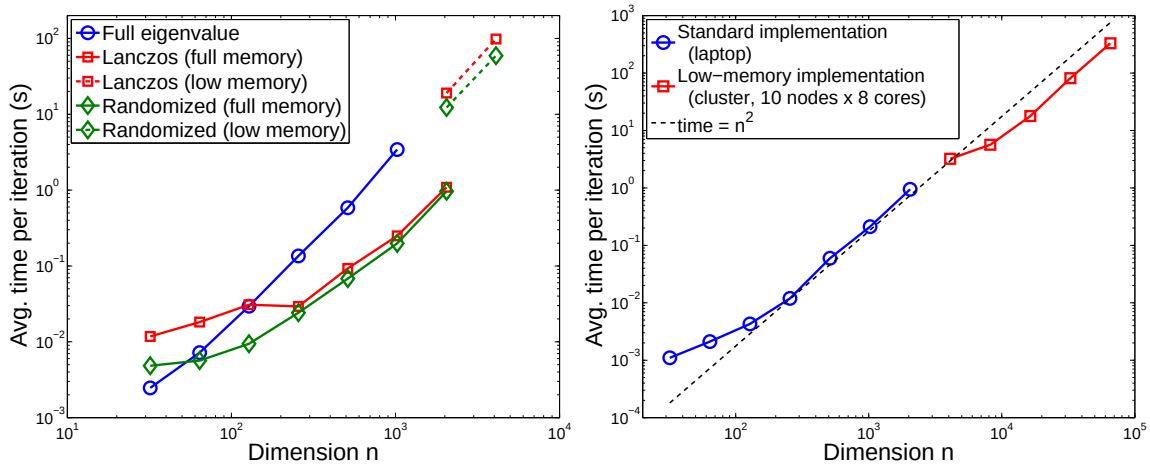


Figure 4.12: Mean time of 10 iterations: this includes the matrix multiplications as well as eigenvalue computations. (Left) shows times for a complete iteration of our method on a single computer using sparse matrix multiplies ("full memory") and, above 11 qubits, the custom low-memory implementation as well (not multi-threaded) on the same computer. (Right) shows times for just the `RandomizedSVD`/`RandomizedEIG`.

(shown bottom row, right) from Theorem 11, suggesting that it may be possible to have a more refined analysis.

Finally, to test scaling to very large data, we compute a 16 qubit state ($n = 65536$), using a known quantum state as input, with realistic quantum mechanical perturbations (global depolarizing noise of level $\gamma = 0.01$; see [FGLE12]) as well as AWGN to give a SNR of 30 dB, and $m = 5n = 327680$ measurements. The first iteration uses Lanczos and all subsequent iterations use `RandomizedEIG` using $\rho = 5$ and $q = 3$ power iterations. On a cluster with 10 computers, the mean time per iteration is 401 seconds. The table in Fig. 4.15 (left) shows the error metrics of the recovered matrix, and Fig. 4.15 (right) plots the convergence rate of the Frobenius-norm error and trace distance.

Figure 4.14 reports the median error on 20 test problems across a range of m. Here, $\mathbf{X}^\star$ is only approximately low rank and $\mathbf{Y}$ is contaminated with noise. We compare the convex approach [FGLE12], the "AltMinSense" approach [JNS13], and a standard splitting approach. AltMinSense and the convex approach have poor accuracy; the accuracy of AltMinSense can be improved by incorporating symmetry, but this changes the algorithm fundamentally and the theoretical guarantees are lost. The splitting approach, if initialized correctly, is accurate, but lacks guarantees. Furthermore, it is slower in practice due to slower convergence, though for some simple problems (i.e., no convex constraints $\mathcal{C}$) it is possible to accelerate using L-BFGS [Lau12].

4.6.6 Video background subtraction via RPCA

We consider the problem of background subtraction in video sequences: static brackground scenes are considered low-rank while moving foreground objects are sparse data. Using the complete set of measurements, this problem falls under the RPCA framework. We apply the GoDec algorithm [ZT11] and the MATRIX ALPS scheme on a 144 x 176 x 200 video sequence. Both solvers use the same low-rank projection operators based on randomized QR factorization ideas [HMT11, ZT11]. Representative results

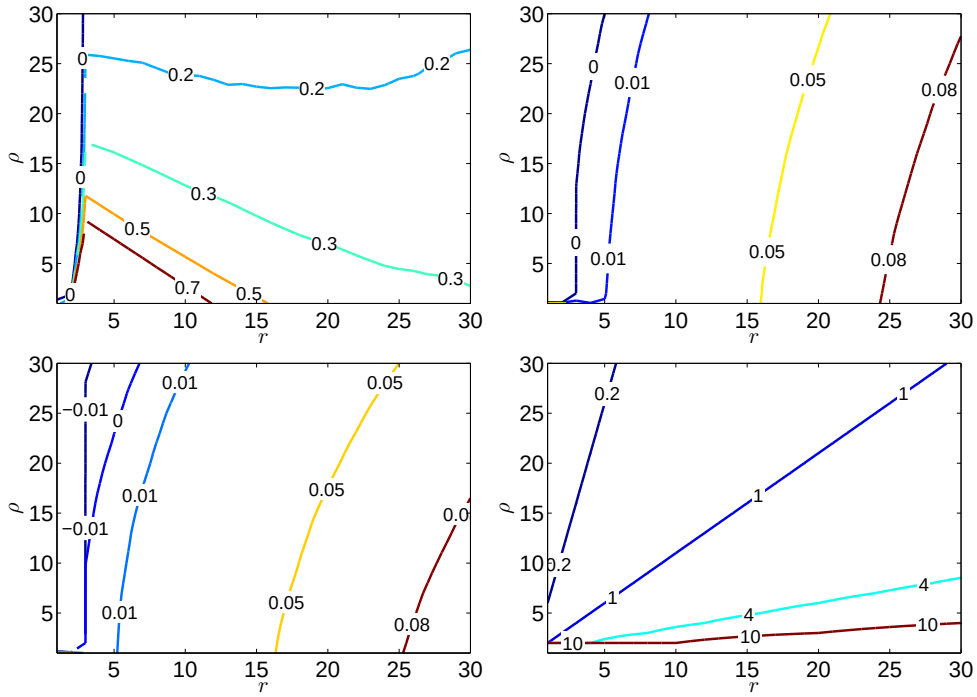


Figure 4.13: Top row: $\widetilde{\epsilon}$ for (left) $q = 0$ and (right) $q = 1$ power iterations. Bottom row: $\widetilde{\epsilon}$ for $q = 2$ power iterations (left), and (right) shows the bound ϵ.

are depicted in Figure 4.16.

4.7 Discussion

In this chapter, we present new strategies and review existing ones for hard thresholding methods to recover low-rank matrices from dimensionality reducing, linear projections. Our discussion starts and revolves around four basic building blocks that exploit the problem structure to reduce computational complexity without sacrificing stability.

In theory, constant μ_i selection schemes are accompanied with strong RIP constant conditions but empirical evidence reveal signal reconstruction vulnerabilities. While convergence derivations of adaptive schemes are characterized by weaker bounds, the performance gained by this choice in terms of convergence rate, is quite significant. Memory-based methods lead to convergence speed with (almost) no extra cost on the complexity of hard thresholding methods—we provide theoretical evidence for convergence for simple cases but more theoretical justification is needed to generalize this part as future work. Lastly, further estimate refinement over low rank subspaces using gradient update steps or pseudo-inversion optimization techniques provides signal reconstruction efficacy, but more computational power is needed per iteration.

We connect ϵ-approximation low-rank revealing schemes with first-order gradient descent algorithms to solve general affine rank minimization problems; to the best of our knowledge, this is the first attempt to theoretically characterize the performance of iterative greedy algorithms with ϵ-approximation schemes. In all cases, experimental results illustrate the effectiveness of the proposed schemes on different problem

$\|\mathbf{X}-\mathbf{X}^\star\|_F$	Trace distance $\|\mathbf{X}-\mathbf{X}^\star\|_*$	Fidelity $F(\mathbf{X},\mathbf{X}^\star)$
0.0256	0.0363	0.9998

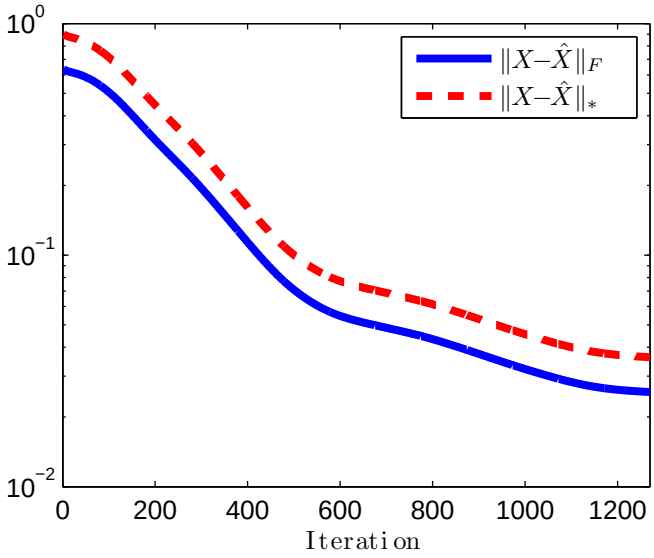


Figure 4.15: The table (left) shows error metrics for the noisy rank-1 16-qubit recovery. The figure (right) shows the convergence rate for the same simulation.

configurations.

Randomization is a powerful tool to accelerate and scale optimization algorithms, and it can be rigorously included in algorithms that are robust to small errors. Within the low-rank recovery context, we leverage randomized approximations to remove memory bottlenecks by merging the two-key steps of most recovery algorithms in affine rank minimization problems: gradient calculation and low-rank projection. Unfortunately, the current black-box approximation guarantees, such as Theorem 11, are too pessimistic to be directly used in theoretical characterizations of our approach. For future work, motivated by the overwhelming empirical evidence of the good performance of our approach, we plan to directly analyze the impact of randomization in characterizing the algorithmic performance.

Finally, we study the general problem of sparse plus low rank matrix recovery from incomplete and noisy data. In essence, the problem under consideration includes various low-dimensional models as special cases such as sparse signal reconstruction, affine rank minimization and robust PCA. Based on this algorithm, we derive improved conditions on the restricted isometry constants that guarantee the success of reconstruction. Furthermore, we show that the memory-based scheme provides great computational advantage over both the convex and the non-convex approaches.

The discussion in this chapter leads to the following open problem/extension: Let us consider a content-data structure in the form of a matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, where each row represent a specific content item and each column represent a single server. Then $C_{ij} = 1$ if server j contains item i. We expect each server to contain a small fraction of the overall content, so that each column of $\mathbf{C}$ will have few non-zero entries.

We can further group the contents into categories, such as sport movies, documentaries, news, Bollywood movies and so on. We expect that if a server contains an item from category k, then it will be more likely to contain other items of the same category.

According to the discussion above, one might be interested in the following problem:

Figure 4.16: Background subtraction in video sequence. Median execution times over 10 Monte-Carlo iterations. GoDec: 34.8 sec—MATRIX ALPS: 15.8 sec.

Open question 6. *Let $\mathbf{C} \in \mathbb{R}^{p \times n}$ be a given (possible binary) matrix that partially indices items to servers. Furthermore, let $\mathbf{C}^{(i)},\ i = 1, \ldots, l$ be a set of l submatrices (of known but not necessarily equal size) as a non-overlapping fragmentation of $\mathbf{C}$ such that $\cup_i \mathbf{C}^{(i)} \to \mathbf{C}$; here, $\cup$ denotes the union/remapping of the set of submatrices into full matrix. Given Ω as the set of index pairs corresponding to the observable (non-zero) entries in $\mathbf{C}$, we define the linear operator $\mathcal{A} : \mathbb{R}^{p \times n} \to \mathbb{R}^m$ as a linear mask over the observable entries. To this end, we are interested in the following optimization problem:*

$$\underset{\mathbf{C}^{(i),\ i=1,\ldots,p}}{\text{minimize}} \quad \sum_i \text{rank}(\mathbf{C}^{(i)}) \quad \text{subject to} \quad \mathbf{y} = \mathcal{A}(\cup_i \mathbf{C}^{(i)}) \tag{4.39}$$

where $\mathbf{y} = \mathcal{A}(\mathbf{C})$ denotes the set of observable entries.

Appendix

In this section, we present some lemmas that are useful in our subsequent developments—these lemmas are consequences of the R-RIP of $\boldsymbol{\mathcal{A}}$.

Lemma 25. *[LB10] Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ be a linear operator that satisfies the R-RIP with constant δ_r and let $\boldsymbol{\mathcal{A}}^* : \mathbb{R}^m \to \mathbb{R}^{p\times n}$ be the adjoint operation. Then, $\forall \boldsymbol{v} \in \mathbb{R}^m$, the following holds true:*

$$\left\|\mathcal{P}_{\mathcal{S}}(\boldsymbol{\mathcal{A}}^*\boldsymbol{v})\right\|_F \leq \sqrt{1+\delta_r}\left\|\boldsymbol{v}\right\|_2, \tag{4.40}$$

where $\mathcal{S}$ is a set of orthonormal, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that $rank(\mathcal{P}_{\mathcal{S}}\mathbf{X}) \leq r,\ \forall \mathbf{X} \in \mathbb{R}^{p\times n}$.

Lemma 26. *[LB10] Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ be a linear operator that satisfies the R-RIP with constant δ_r. Then, $\forall \mathbf{X} \in \mathbb{R}^{p\times n}$, the following holds true:*

$$(1-\delta_r)\left\|\mathcal{P}_{\mathcal{S}}\mathbf{X}\right\|_F \leq \left\|\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}}\mathbf{X}\right\|_F \leq (1+\delta_r)\left\|\mathcal{P}_{\mathcal{S}}\mathbf{X}\right\|_F, \tag{4.41}$$

where $\mathcal{S}$ is a set of orthonormal, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that $rank(\mathcal{P}_{\mathcal{S}}\mathbf{X}) \leq r,\ \forall \mathbf{X} \in \mathbb{R}^{p\times n}$.

Lemma 27. *[GM11] Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ be a linear operator that satisfies the R-RIP with constant δ_r and $\mathcal{S}$ be a set of orthonormal, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that $rank(\mathcal{P}_{\mathcal{S}}\mathbf{X}) \leq r,\ \forall \mathbf{X} \in \mathbb{R}^{p\times n}$. Then, for $\mu > 0$, $\boldsymbol{\mathcal{A}}$ satisfies:*

$$\lambda(\mu\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}}) \in [\mu(1-\delta_r), \mu(1+\delta_r)]. \tag{4.42}$$

where $\lambda(\boldsymbol{\mathcal{B}})$ represents the range of eigenvalues of the linear operator $\boldsymbol{\mathcal{B}} : \mathbb{R}^m \to \mathbb{R}^{p\times n}$. Moreover, $\forall \mathbf{X} \in \mathbb{R}^{p\times n}$, it follows that:

$$\left\|(\mathbf{I} - \mu\mathcal{P}_{\mathcal{S}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}})\mathcal{P}_{\mathcal{S}}\mathbf{X}\right\|_F \leq \max\left\{\mu(1+\delta_r)-1, 1-\mu(1-\delta_r)\right\}\left\|\mathcal{P}_{\mathcal{S}}\mathbf{X}\right\|_F. \tag{4.43}$$

Lemma 28. *[GM11] Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ be a linear operator that satisfies the R-RIP with constant δ_r and $\mathcal{S}_1, \mathcal{S}_2$ be two sets of orthonormal, rank-1 matrices in $\mathbb{R}^{p\times n}$ such that*

$$rank(\mathcal{P}_{\mathcal{S}_1\cup\mathcal{S}_2}\mathbf{X}) \leq r,\ \forall \mathbf{X} \in \mathbb{R}^{p\times n}. \tag{4.44}$$

Then, the following inequality holds: $\left\|\mathcal{P}_{\mathcal{S}_1}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_1^\perp}\mathbf{X}\right\|_F \leq \delta_r\left\|\mathcal{P}_{\mathcal{S}_1^\perp}\mathbf{X}\right\|_F, \forall \mathbf{X} \in span(\mathcal{S}_2)$.

A well-known lemma used in the convergence rate proofs of this class of greedy hard thresholding algorithms is defined next.

Lemma 29. *[Ber95] Let $\mathcal{J} \subseteq \mathbb{R}^{p\times n}$ be a closed convex set and $f : \mathcal{J} \to \mathbb{R}$ be a smooth objective function defined over $\mathcal{J}$. Let $\mathbf{X}^* \in \mathcal{J}$ be a local minimum of the objective function f over the set $\mathcal{J}$. Then*

$$\langle \nabla f(\mathbf{X}^*), \mathbf{X} - \mathbf{X}^* \rangle \geq 0, \quad \forall \mathbf{X} \in \mathcal{J}. \tag{4.45}$$

Remark 8. *Let $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \in \mathbb{R}^{p\times n}$ and $\mathbf{Y} = \widetilde{\mathbf{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\mathbf{V}}^T \in \mathbb{R}^{p\times n}$. Assume two sets: i) $\mathcal{S}_1 = \{\boldsymbol{u}_i\boldsymbol{u}_i^T : i \in \mathcal{I}_1\}$ where $\boldsymbol{u}_i$ is the i-th singular vector of $\mathbf{X}$ and $\mathcal{I}_1 \subseteq \{1, \ldots, rank(\mathbf{X})\}$ and, ii) $\mathcal{S}_2 = \{\boldsymbol{u}_i\boldsymbol{u}_i^T, \tilde{\boldsymbol{u}}_j\tilde{\boldsymbol{u}}_j{}^T : i \in$*

$\mathcal{I}_2,\ j \in \mathcal{I}_3\}$ where $\tilde{\boldsymbol{u}}_i$ is the i-th singular vector of $\mathbf{Y}$, $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq \{1,\dots,rank(\mathbf{X})\}$ and, $\mathcal{I}_3 \subseteq \{1,\dots,rank(\mathbf{Y})\}$. We observe that the subspaces defined by $\boldsymbol{u}_i\boldsymbol{u}_i^T$ and $\tilde{\boldsymbol{u}}_j\tilde{\boldsymbol{u}}_j{}^T$ are not necessarily orthogonal. To this end, let $\widehat{\mathcal{S}}_2 = ortho(\mathcal{S}_2)$; this operation can be easily computed via SVD. Then, the following commutativity property holds true for any matrix $\mathbf{W} \in \mathbb{R}^{p\times n}$:

$$\mathcal{P}_{\mathcal{S}_1}\mathcal{P}_{\widehat{\mathcal{S}}_2}\mathbf{W} = \mathcal{P}_{\widehat{\mathcal{S}}_2}\mathcal{P}_{\mathcal{S}_1}\mathbf{W}. \tag{4.46}$$

Proof of Lemma 21

Given $\mathcal{X}^* \leftarrow \mathcal{P}_r(\mathbf{X}^\star)$ using SVD factorization, we define the following quantities: $\mathcal{S}_i \leftarrow \mathcal{X}_i \cup \mathcal{D}_i,\ \mathcal{S}_i^* \leftarrow$ ortho $(\mathcal{X}_i \cup \mathcal{X}^*)$. Then, given the structure of the sets $\mathcal{S}_i$ and $\mathcal{S}_i^*$

$$\mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{(\mathcal{S}_i^*)^\perp} = \mathcal{P}_{\mathcal{D}_i}\mathcal{P}_{(\mathcal{X}^*\cup\mathcal{X}_i)^\perp} \ \text{ and } \ \mathcal{P}_{\mathcal{S}_i^*}\mathcal{P}_{\mathcal{S}_i^\perp} = \mathcal{P}_{\mathcal{X}^*}\mathcal{P}_{(\mathcal{D}_i\cup\mathcal{X}_i)^\perp}. \tag{4.47}$$

Since the subspace defined in $\mathcal{D}_i$ is the best rank-r subspace, orthogonal to the subspace spanned by $\mathcal{X}_i$:

$$\left\|\mathcal{P}_{\mathcal{D}_i}\mathcal{P}_{\mathcal{X}_i^\perp}\nabla f(\mathbf{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{X}^*}\mathcal{P}_{\mathcal{X}_i^\perp}\nabla f(\mathbf{X}(i))\right\|_F^2 \Rightarrow \left\|\mathcal{P}_{\mathcal{S}_i}\nabla f(\mathbf{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{S}_i^*}\nabla f(\mathbf{X}(i))\right\|_F^2$$

Removing the common subspaces in $\mathcal{S}_i$ and $\mathcal{S}_i^*$ by the commutativity property of the projection operation and using the shortcut $\mathcal{P}_{\mathcal{A}\setminus\mathcal{B}} \equiv \mathcal{P}_{\mathcal{A}}\mathcal{P}_{\mathcal{B}^\perp}$ for sets $\mathcal{A},\ \mathcal{B}$, we get:

$$\begin{aligned}\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\nabla f(\mathbf{X}(i))\right\|_F^2 \geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\nabla f(\mathbf{X}(i))\right\|_F^2 \Rightarrow \\ \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}(i)) + \mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F\end{aligned} \tag{4.48}$$

Next, we assume that $\mathcal{P}_{(\mathcal{A}\setminus\mathcal{B})^\perp}$ denotes the orthogonal projection onto the subspace spanned by $\mathcal{P}_{\mathcal{A}}\mathcal{P}_{\mathcal{B}^\perp}$. Then, on the left hand side of (4.48), we have:

$$\begin{aligned}&\left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \overset{(i)}{\leq} \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \\ &\overset{(ii)}{=} \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\mathbf{X}^\star - \mathbf{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \\ &\overset{(iii)}{=} \left\|(\mathbf{I} - \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*})(\mathbf{X}^\star - \mathbf{X}(i)) + \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \\ &\leq \left\|(\mathbf{I} - \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*})(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F \\ &\overset{(iv)}{\leq} \delta_{3r}\left\|\mathbf{X}^\star - \mathbf{X}(i)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F \\ &\overset{(v)}{\leq} \delta_{3r}\left\|\mathbf{X}^\star - \mathbf{X}(i)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F + \delta_{3r}\left\|\mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F \\ &\overset{(vi)}{\leq} 2\delta_{3r}\left\|\mathbf{X}^\star - \mathbf{X}(i)\right\|_F + \left\|\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F\end{aligned} \tag{4.49}$$

where (i) due to triangle inequality over Frobenius metric norm, (ii) since $\mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\mathbf{X}(i) - \mathbf{X}^\star) = \mathbf{0}$, (iii) by using the fact that $\mathbf{X}(i) - \mathbf{X}^\star := \mathcal{P}_{\mathcal{S}_i\setminus\mathcal{S}_i^*}(\mathbf{X}(i) - \mathbf{X}^\star) + \mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}(i) - \mathbf{X}^\star)$, (iv) due to Lemma 27, (v) due to Lemma 28 and (vi) since $\left\|\mathcal{P}_{(\mathcal{S}_i\setminus\mathcal{S}_i^*)^\perp}(\mathbf{X}^\star - \mathbf{X}(i))\right\|_F \leq \left\|\mathbf{X}(i) - \mathbf{X}^\star\right\|_F$.

For the right hand side of (4.48), we calculate:

$$\begin{aligned}\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star-\mathbf{X}(i))+\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F &\geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\mathbf{X}^\star-\mathbf{X}(i))\right\|_F-\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{(\mathcal{S}_i^*\setminus\mathcal{S}_i)^\perp}(\mathbf{X}^\star-\mathbf{X}(i))\right\|_F\\ &\quad-\left\|(\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}-\mathbf{I})(\mathbf{X}^\star-\mathbf{X}(i))\right\|_F-\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F\\ &\geq \left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}(\mathbf{X}^\star-\mathbf{X}(i))\right\|_F-2\delta_{2r}\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F-\left\|\mathcal{P}_{\mathcal{S}_i^*\setminus\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\right\|_F\end{aligned} \tag{4.50}$$

by using Lemmas 27 and 28. Combining (4.49) and (4.50) in (4.48), we get:

$$\left\|\mathcal{P}_{\mathcal{X}^*\setminus\mathcal{S}_i}\mathbf{X}^\star\right\|_F\leq(2\delta_{2r}+2\delta_{3r})\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F+\sqrt{2(1+\delta_{2r})}\left\|\boldsymbol{\varepsilon}\right\|_2.$$

Proof of Theorem 6

Let $\mathcal{X}^*\leftarrow\mathcal{P}_r(\mathbf{X}^\star)$ be a set of orthonormal, rank-1 matrices that span the range of $\mathbf{X}^\star$. In Algorithm 1, $\mathbf{W}(i)\leftarrow\mathcal{P}_r(\mathbf{V}(i))$. Thus:

$$\begin{aligned}\left\|\mathbf{W}(i)-\mathbf{V}(i)\right\|_F^2\leq\left\|\mathbf{X}^\star-\mathbf{V}(i)\right\|_F^2\Rightarrow\\ \left\|\mathbf{W}(i)-\mathbf{X}^\star+\mathbf{X}^\star-\mathbf{V}(i)\right\|_F^2\leq\left\|\mathbf{X}^\star-\mathbf{V}(i)\right\|_F^2\Rightarrow\left\|\mathbf{W}(i)-\mathbf{X}^\star\right\|_F^2\leq2\langle\mathbf{W}(i)-\mathbf{X}^\star,\mathbf{V}(i)-\mathbf{X}^\star\rangle\end{aligned} \tag{4.51}$$

From Algorithm 1, $i)$ $\mathbf{V}(i)\in\text{span}(\mathcal{S}_i)$, $ii)$ $\mathbf{X}(i)\in\text{span}(\mathcal{S}_i)$ and $iii)$ $\mathbf{W}(i)\in\text{span}(\mathcal{S}_i)$. We define $\mathcal{E}\leftarrow\text{ortho}(\mathcal{S}_i\cup\mathcal{X}^*)$ where $\text{rank}(\text{span}(\mathcal{E}))\leq3r$ and let $\mathcal{P}_\mathcal{E}$ be the orthogonal projection onto the subspace defined by $\mathcal{E}$.

Since $\mathbf{W}(i)-\mathbf{X}^\star\in\text{span}(\mathcal{E})$ and $\mathbf{V}(i)-\mathbf{X}^\star\in\text{span}(\mathcal{E})$, the following hold true:

$$\mathbf{W}(i)-\mathbf{X}^\star=\mathcal{P}_\mathcal{E}(\mathbf{W}(i)-\mathbf{X}^\star)\ \text{ and }\ \mathbf{V}(i)\qquad\qquad-\mathbf{X}^\star=\mathcal{P}_\mathcal{E}(\mathbf{V}(i)-\mathbf{X}^\star).$$

Then, (4.51) can be written as:

$$\begin{aligned}\left\|\mathbf{W}(i)-\mathbf{X}^\star\right\|_F^2&\leq2\langle\mathcal{P}_\mathcal{E}(\mathbf{W}(i)-\mathbf{X}^\star),\mathcal{P}_\mathcal{E}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle\\ &=\underbrace{2\langle\mathcal{P}_\mathcal{E}(\mathbf{W}(i)-\mathbf{X}^\star),\mathcal{P}_\mathcal{E}(\mathbf{X}(i)-\mathbf{X}^\star-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}(i)-\mathbf{X}^\star))\rangle}_{\doteq A}+\underbrace{2\mu_i\langle\mathcal{P}_\mathcal{E}(\mathbf{W}(i)-\mathbf{X}^\star),\mathcal{P}_\mathcal{E}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle}_{\doteq B}\end{aligned} \tag{4.52}$$

In B, we observe:

$$\begin{aligned}B:=2\mu_i\langle\mathcal{P}_\mathcal{E}(\mathbf{W}(i)-\mathbf{X}^\star),\mathcal{P}_\mathcal{E}\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle&\overset{(i)}{=}2\mu_i\langle\mathbf{W}(i)-\mathbf{X}^\star,\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\rangle\\ &\overset{(ii)}{\leq}2\mu_i\left\|\mathbf{W}(i)-\mathbf{X}^\star\right\|_F\left\|\mathcal{P}_{\mathcal{S}_i}(\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon})\right\|_F\\ &\overset{(iii)}{\leq}2\mu_i\sqrt{1+\delta_{2r}}\left\|\mathbf{W}(i)-\mathbf{X}^\star\right\|_F\left\|\boldsymbol{\varepsilon}\right\|_2\end{aligned} \tag{4.53}$$

where (i) holds since $\mathcal{P}_{\mathcal{S}_i}\mathcal{P}_\mathcal{E}=\mathcal{P}_\mathcal{E}\mathcal{P}_{\mathcal{S}_i}=\mathcal{P}_{\mathcal{S}_i}$ for $\text{span}(\mathcal{S}_i)\in\text{span}(\mathcal{E})$, (ii) is due to Cauchy-Schwarz inequality and, (iii) is easily derived using Lemma 25.

In A, we perform the following motions:

$$\begin{aligned}
A &:= 2\langle \mathbf{W}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star) - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\rangle \\
&\overset{(i)}{=} 2\langle \mathbf{W}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star) - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \big[\mathcal{P}_{\mathcal{S}_i} + \mathcal{P}_{\mathcal{S}_i^\perp}\big] \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\rangle \\
&= 2\langle \mathbf{W}(i) - \mathbf{X}^\star, (\mathbf{I} - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i}) \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\rangle - 2\mu_i \langle \mathbf{W}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\rangle \\
&\overset{(ii)}{\leq} 2\big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F \big\|(\mathbf{I} - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i}) \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F + 2\mu_i \big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F \big\|\mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F
\end{aligned} \tag{4.54}$$

where (i) is due to $\mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star) := \mathcal{P}_{\mathcal{S}_i} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star) + \mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)$ and (ii) follows from Cauchy-Schwarz inequality. Since $\frac{1}{1+\delta_{2r}} \leq \mu_i \leq \frac{1}{1-\delta_{2r}}$, Lemma 27 implies:

$$\lambda(\mathbf{I} - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i}) \in \left[1 - \frac{1-\delta_{2r}}{1+\delta_{2r}}, \frac{1+\delta_{2r}}{1-\delta_{2r}} - 1\right] \leq \frac{2\delta_{2r}}{1-\delta_{2r}}.$$

and thus:

$$\big\|(\mathbf{I} - \mu_i \mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i}) \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F \leq \frac{2\delta_{2r}}{1-\delta_{2r}} \big\|\mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F.$$

Furthermore, according to Lemma 28:

$$\big\|\mathcal{P}_{\mathcal{S}_i} \boldsymbol{\mathcal{A}}^* \boldsymbol{\mathcal{A}} \mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F \leq \delta_{3r} \big\|\mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F$$

since $\text{rank}(\mathcal{P}_{\mathcal{K}}\mathbf{X}) \leq 3r,\ \forall \mathbf{X} \in \mathbb{R}^{p \times n}$ for $\mathcal{K} \leftarrow \text{ortho}(\mathcal{E} \cup \mathcal{S}_i)$. Since $\mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star) = \mathcal{P}_{\mathcal{X}^* \setminus (\mathcal{D}_i \cup \mathcal{X}_i)} \mathbf{X}^\star$ where

$$\mathcal{D}_i \leftarrow \mathcal{P}_k \left(\mathcal{P}_{\mathcal{X}_i^\perp} \nabla f(\mathbf{X}(i))\right),$$

then:

$$\big\|\mathcal{P}_{\mathcal{S}_i^\perp} \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i) - \mathbf{X}^\star)\big\|_F = \big\|\mathcal{P}_{\mathcal{X}^* \setminus (\mathcal{D}_i \cup \mathcal{X}_i)} \mathbf{X}^\star\big\|_F \leq (2\delta_{2r} + 2\delta_{3r}) \big\|\mathbf{X}(i) - \mathbf{X}^\star\big\|_F + \sqrt{2(1+\delta_{2r})} \big\|\boldsymbol{\varepsilon}\big\|_2,$$

using Lemma 21. Combining the above in (4.54), we compute:

$$A \leq \Big(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r} + 2\delta_{3r}) \frac{2\delta_{3r}}{1-\delta_{2r}}\Big) \big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F \cdot \big\|\mathbf{X}(i) - \mathbf{X}^\star\big\|_F + \frac{2\delta_{3r}}{1-\delta_{2r}} \big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F \sqrt{2(1+\delta_{2r})} \big\|\boldsymbol{\varepsilon}\big\|_2 \tag{4.55}$$

Combining (4.53) and (4.55) in (4.52), we get:

$$\big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F \leq \Big(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r} + 2\delta_{3r}) \frac{2\delta_{3r}}{1-\delta_{2r}}\Big) \big\|\mathbf{X}(i) - \mathbf{X}^\star\big\|_F + \Big(\frac{2\sqrt{1+\delta_{2r}}}{1-\delta_{2r}} + \frac{2\delta_{3r}}{1-\delta_{2r}} \sqrt{2(1+\delta_{2r})}\Big) \big\|\boldsymbol{\varepsilon}\big\|_2 \tag{4.56}$$

Focusing on steps 5 and 6 of Algorithm 1, we perform similar motions to obtain:

$$\big\|\mathbf{X}(i+1) - \mathbf{X}^\star\big\|_F \leq \Big(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\Big) \big\|\mathbf{W}(i) - \mathbf{X}^\star\big\|_F + \frac{\sqrt{1+\delta_r}}{1-\delta_r} \big\|\boldsymbol{\varepsilon}\big\|_2 \tag{4.57}$$

Combining the recursions in (4.56) and (4.57), we finally compute:

$$\left\|\mathbf{X}(i+1)-\mathbf{X}^\star\right\|_F \leq \rho\left\|\mathbf{X}(i)-\mathbf{X}^\star\right\|_F + \gamma\left\|\boldsymbol{\varepsilon}\right\|_2,$$

for $\rho := \left(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\right)\left(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r}+2\delta_{3r})\frac{2\delta_{3r}}{1-\delta_{2r}}\right)$ and

$$\gamma := \left(\left(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\right)\left(\frac{2\sqrt{1+\delta_{2r}}}{1-\delta_{2r}} + \frac{2\delta_{3r}}{1-\delta_{2r}}\sqrt{2(1+\delta_{2r})}\right) + \frac{\sqrt{1+\delta_r}}{1-\delta_r}\right)$$

For the convergence parameter ρ, further compute:

$$\left(\frac{1+2\delta_{2r}}{1-\delta_{2r}}\right)\left(\frac{4\delta_{2r}}{1-\delta_{2r}} + (2\delta_{2r}+2\delta_{3r})\frac{2\delta_{3r}}{1-\delta_{2r}}\right) \leq \frac{1+2\delta_{3r}}{(1-\delta_{3r})^2}\left(4\delta_{3r}+8\delta_{3r}^2\right) =: \hat{\rho}. \tag{4.58}$$

for $\delta_r \leq \delta_{2r} \leq \delta_{3r}$. Calculating the roots of this expression, we easily observe that $\rho < \hat{\rho} < 1$ for $\delta_{3r} < 0.1235$.

Proof of Theorem 7

Before we present the proof of Theorem 7, we list a series of lemmas that correspond to the motions Algorithm 2 performs.

Lemma 30. *[Error norm reduction via least-squares optimization] Let $\mathcal{S}_i$ be a set of orthonormal, rank-1 matrices that span a rank-2r subspace in $\mathbb{R}^{p\times n}$. Then, the least squares solution $\mathbf{V}(i)$ given by:*

$$\mathbf{V}(i) \leftarrow \underset{\mathbf{V}:\mathbf{V}\in span(\mathcal{S}_i)}{\arg\min} \left\|\boldsymbol{y}-\boldsymbol{\mathcal{A}}\mathbf{V}\right\|_2^2 \qquad \textit{satisfies:} \tag{4.59}$$

$$\left\|\mathbf{V}(i)-\mathbf{X}^\star\right\|_F \leq \frac{1}{\sqrt{1-\delta_{3r}^2(\boldsymbol{\mathcal{A}})}}\left\|\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i)-\mathbf{X}^\star)\right\|_F + \frac{\sqrt{1+\delta_{2r}}}{1-\delta_{3r}}\left\|\boldsymbol{\varepsilon}\right\|_2. \tag{4.60}$$

Proof. We observe that $\left\|\mathbf{V}(i)-\mathbf{X}^\star\right\|_F^2$ is decomposed as follows:

$$\left\|\mathbf{V}(i)-\mathbf{X}^\star\right\|_F^2 = \left\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\right\|_F^2 + \left\|\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i)-\mathbf{X}^\star)\right\|_F^2. \tag{4.61}$$

In (4.59), $\mathbf{V}(i)$ is the minimizer over the low-rank subspace spanned by $\mathcal{S}_i$ with $\text{rank}(\text{span}(\mathcal{S}_i)) \leq 2r$. Using the optimality condition (Lemma 29) over the convex set $\Theta = \{\mathbf{X} : \text{span}(\mathbf{X}) \in \mathcal{S}_i\}$, we have:

$$\langle \nabla f(\mathbf{V}(i)), \mathcal{P}_{\mathcal{S}_i}(\mathbf{X}^\star - \mathbf{V}(i))\rangle \geq 0 \Rightarrow \langle \boldsymbol{\mathcal{A}}\mathbf{V}(i)-\boldsymbol{y}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle \leq 0. \tag{4.62}$$

for $\mathcal{P}_{\mathcal{S}_i}\mathbf{X}^\star \in \text{span}(\mathcal{S}_i)$. Given condition (4.62), the first term on the right hand side of (4.61) becomes:

$$\begin{aligned}\left\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\right\|_F^2 &= \langle \mathbf{V}(i)-\mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle \\ &\overset{(4.62)}{\leq} \langle \mathbf{V}(i)-\mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle - \langle \boldsymbol{\mathcal{A}}\mathbf{V}(i)-\boldsymbol{y}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle \\ &\leq |\langle \mathbf{V}(i)-\mathbf{X}^\star, (\mathbf{I}-\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle| + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle \end{aligned} \tag{4.63}$$

Focusing on the term $|\langle \mathbf{V}(i) - \mathbf{X}^\star, (\mathbf{I} - \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle|$, we derive the following:

$$\begin{aligned}
|\langle \mathbf{V}(i) - \mathbf{X}^\star, (\mathbf{I} - \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle| &= |\langle \mathbf{V}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle - \langle \mathbf{V}(i) - \mathbf{X}^\star, \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle| \\
&\overset{(i)}{=} |\langle \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}(\mathbf{V}(i) - \mathbf{X}^\star), \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle \\
&\quad - \langle \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}(\mathbf{V}(i) - \mathbf{X}^\star), \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle| \\
&\overset{(ii)}{=} |\langle \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}(\mathbf{V}(i) - \mathbf{X}^\star), \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle \\
&\quad - \langle \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}(\mathbf{V}(i) - \mathbf{X}^\star), \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle| \\
&= |\langle \mathbf{V}(i) - \mathbf{X}^\star, (\mathbf{I} - \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle|
\end{aligned}$$

where (i) follows from the facts that $\mathbf{V}(i) - \mathbf{X}^\star \in \text{span}(\text{ortho}(\mathcal{S}_i \cup \mathcal{X}^*))$ and thus $\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}(\mathbf{V}(i) - \mathbf{X}^\star) = \mathbf{V}(i) - \mathbf{X}^\star$ and (ii) is due to $\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\mathcal{P}_{\mathcal{S}_i} = \mathcal{P}_{\mathcal{S}_i}$ since $\text{span}(\mathcal{S}_i) \subseteq \text{span}(\text{ortho}(\mathcal{S}_i \cup \mathcal{X}^*))$. Then, (4.63) becomes:

$$\begin{aligned}
\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F^2 &\leq |\langle \mathbf{V}(i) - \mathbf{X}^\star, (\mathbf{I} - \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle| + \langle \boldsymbol{\varepsilon}, \boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\rangle \\
&\overset{(i)}{\leq} \|\mathbf{V}(i) - \mathbf{X}^\star\|_F \|(\mathbf{I} - \mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i \cup \mathcal{X}^*})\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F + \|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\varepsilon}\|_F \|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \\
&\overset{(ii)}{\leq} \delta_{3r}\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \|\mathbf{V}(i) - \mathbf{X}^\star\|_F + \sqrt{1+\delta_{2r}}\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \|\boldsymbol{\varepsilon}\|_2, \qquad (4.64)
\end{aligned}$$

where (i) comes from Cauchy-Swartz inequality and (ii) is due to Lemmas 25 and 27. Simplifying the above quadratic expression, we obtain:

$$\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \leq \delta_{3r}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F + \sqrt{1+\delta_{2r}}\|\boldsymbol{\varepsilon}\|_2. \qquad (4.65)$$

As a consequence, (4.61) can be upper bounded by:

$$\|\mathbf{V}(i) - \mathbf{X}^\star\|_F^2 \leq \left(\delta_{3r}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F + \sqrt{1+\delta_{2r}}\|\boldsymbol{\varepsilon}\|_2\right)^2 + \|\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F^2. \qquad (4.66)$$

We form the quadratic polynomial for this inequality assuming as unknown variable the quantity $\|\mathbf{V}(i) - \mathbf{X}^\star\|_F$. Bounding by the largest root of the resulting polynomial, we get:

$$\|\mathbf{V}(i) - \mathbf{X}^\star\|_F \leq \frac{1}{\sqrt{1 - \delta_{3r}^2(\boldsymbol{\mathcal{A}})}}\|\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F + \frac{\sqrt{1+\delta_{2r}}}{1-\delta_{3r}}\|\boldsymbol{\varepsilon}\|_2. \qquad (4.67)$$

□

The following Lemma characterizes how subspace *pruning* affects the recovered energy:

Lemma 31. *[Best rank-r subspace selection] Let $\mathbf{V}(i) \in \mathbb{R}^{p \times n}$ be a rank-2r proxy matrix in the subspace spanned by $\mathcal{S}_i$ and let $\mathbf{X}(i+1) \leftarrow \mathcal{P}_r(\mathbf{V}(i))$ denote the best rank-r approximation to $\mathbf{V}(i)$, according to (4.7). Then:*

$$\|\mathbf{X}(i+1) - \mathbf{V}(i)\|_F \leq \|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \leq \|\mathbf{V}(i) - \mathbf{X}^\star\|_F. \qquad (4.68)$$

Proof. Since $\mathbf{X}(i+1)$ denotes the best rank-r approximation to $\mathbf{V}(i)$, the following inequality holds for any rank-r matrix $\mathbf{X} \in \mathbb{R}^{p \times n}$ in the subspace spanned by $\mathcal{S}_i$, i.e. $\forall \mathbf{X} \in \text{span}(\mathcal{S}_i)$:

$$\|\mathbf{X}(i+1) - \mathbf{V}(i)\|_F \leq \|\mathbf{X} - \mathbf{V}(i)\|_F. \qquad (4.69)$$

Since $\mathcal{P}_{\mathcal{S}_i}\mathbf{V}(i) = \mathbf{V}(i)$, the left inequality in (4.68) is satisfied for $\mathbf{X} := \mathcal{P}_{\mathcal{S}_i}\mathbf{X}^\star$ in (4.69). □

Lemma 32. *Let $\mathbf{V}(i)$ be the least squares solution in Step 2 of the ADMiRA algorithm and let $\mathbf{X}(i+1)$ be a proxy, rank-r matrix to $\mathbf{V}(i)$ according to: $\mathbf{X}(i+1) \leftarrow \mathcal{P}_k(\mathbf{V}(i))$. Then, $\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_F$ can be expressed in terms of the distance from $\mathbf{V}(i)$ to $\mathbf{X}^\star$ as follows:*

$$\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_F \leq \sqrt{1+3\delta_{3r}^2}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F + \sqrt{1+3\delta_{3r}^2}\sqrt{\frac{3(1+\delta_{2r})}{1+3\delta_{3r}^2}}\|\boldsymbol{\varepsilon}\|_2. \tag{4.70}$$

Proof. We observe the following

$$\begin{aligned}\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_F^2 &= \|\mathbf{X}(i+1) - \mathbf{V}(i) + \mathbf{V}(i) - \mathbf{X}^\star\|_F^2 \\ &= \|\mathbf{V}(i) - \mathbf{X}^\star\|_F^2 + \|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F^2 \\ &\quad - 2\langle \mathbf{V}(i) - \mathbf{X}^\star, \mathbf{V}(i) - \mathbf{X}(i+1)\rangle. \end{aligned} \tag{4.71}$$

Focusing on the right hand side of expression (4.71), $\langle \mathbf{V}(i) - \mathbf{X}^\star, \mathbf{V}(i) - \mathbf{X}(i+1)\rangle = \langle \mathbf{V}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}(i+1))\rangle$ can be similarly analysed as in Lemma 10 where we obtain the following expression:

$$|\langle \mathbf{V}(i) - \mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}(i+1))\rangle| \leq \delta_{3r}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F\|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F + \sqrt{1+\delta_{2r}}\|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F\|\boldsymbol{\varepsilon}\|_2. \tag{4.72}$$

Now, expression (4.71) can be further transformed as:

$$\begin{aligned}\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_F^2 &\overset{(i)}{\leq} \|\mathbf{V}(i) - \mathbf{X}^\star\|_F^2 + \|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F^2 \\ &\quad + 2(\delta_{3r}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F\|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F \\ &\quad + \sqrt{1+\delta_{2r}}\|\mathbf{V}(i) - \mathbf{X}(i+1)\|_F\|\boldsymbol{\varepsilon}\|_2) \end{aligned} \tag{4.73}$$

where (i) is due to (4.72). Using Lemma 31, we further have:

$$\begin{aligned}\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_F^2 &\leq \|\mathbf{V}(i) - \mathbf{X}^\star\|_F^2 + \|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F^2 \\ &\quad + 2\Big(\delta_{3r}\|\mathbf{V}(i) - \mathbf{X}^\star\|_F\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F \\ &\quad + \sqrt{1+\delta_{2r}}\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{V}(i) - \mathbf{X}^\star)\|_F\|\boldsymbol{\varepsilon}\|_2\Big) \end{aligned} \tag{4.74}$$

Furthermore, replacing $\|\mathcal{P}_{\mathcal{S}_i}(\mathbf{X}^\star - \mathbf{V}(i))\|_F$ with its upper bound defined in (4.65), we get:

$$\|\mathbf{X}(i+1) - \mathbf{X}^\star\|_2^2 \overset{(i)}{\leq} \left(1+3\delta_{3r}^2\right)\left(\|\mathbf{V}(i) - \mathbf{X}^\star\|_2 + \sqrt{\frac{3(1+\delta_{2r})}{1+3\delta_{3r}^2}}\|\boldsymbol{\varepsilon}\|\right)^2 \tag{4.75}$$

where (i) is obtained by completing the squares and eliminating negative terms. □

Applying basic algebra tools in (4.70) and (4.60), we get:

$$\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F \leq \sqrt{\frac{1+3\delta_{3r}^2}{1-\delta_{3r}^2}}\|\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i)-\mathbf{X}^\star)\|_F + \Big(\frac{\sqrt{1+3\delta_{3r}^2}}{1-\delta_{3r}}+\sqrt{3}\Big)\sqrt{1+\delta_{2r}}\|\boldsymbol{\varepsilon}\|_2.$$

Since $\mathbf{V}(i)\in\text{span}(\mathcal{S}_i)$, we observe $\mathcal{P}_{\mathcal{S}_i^\perp}(\mathbf{V}(i)-\mathbf{X}^\star) = -\mathcal{P}_{\mathcal{S}_i^\perp}\mathbf{X}^\star = -\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\mathbf{X}^\star$. Then, using Lemma 21, we obtain:

$$\begin{aligned}\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F \leq (2\delta_{2r}+2\delta_{3r})\sqrt{\frac{1+3\delta_{3r}^2}{1-\delta_{3r}^2}}\|\mathbf{X}^\star-\mathbf{X}(i)\|_F + \Bigg[\sqrt{\frac{1+3\delta_{3r}^2}{1-\delta_{3r}^2}}\sqrt{2(1+\delta_{3r})} \\ + \Big(\frac{\sqrt{1+3\delta_{3r}^2}}{1-\delta_{3r}}+\sqrt{3}\Big)\sqrt{1+\delta_{2r}}\Bigg]\|\boldsymbol{\varepsilon}\|_2 \end{aligned} \tag{4.76}$$

Given $\delta_{2r}\leq\delta_{3r}$, ρ is upper bounded by $\rho < 4\delta_{3r}\sqrt{\frac{1+3\delta_{3r}}{1-\delta_{3r}^2}}$. Then, $4\delta_{3r}\sqrt{\frac{1+3\delta_{3r}}{1-\delta_{3r}^2}} < 1 \Leftrightarrow \delta_{3r} < 0.2267$.

Proof of Theorem 8

Let $\mathcal{X}^* \leftarrow \mathcal{P}_r(\mathbf{X}^\star)$ be a set of orthonormal, rank-1 matrices that span the range of $\mathbf{X}^\star$. In Algorithm 3, $\mathbf{X}(i+1)$ is the best rank-r approximation of $\mathbf{V}(i)$. Thus:

$$\|\mathbf{X}(i+1)-\mathbf{V}(i)\|_F^2 \leq \|\mathbf{X}^\star-\mathbf{V}(i)\|_F^2 \Rightarrow \|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F^2 \leq 2\langle\mathbf{X}(i+1)-\mathbf{X}^\star, \mathbf{V}(i)-\mathbf{X}^\star\rangle \tag{4.77}$$

From Algorithm 3, $i)$ $\mathbf{V}(i)\in\text{span}(\mathcal{S}_i)$, $ii)$ $\boldsymbol{Q}_i\in\text{span}(\mathcal{S}_i)$ and $iii)$ $\mathbf{W}(i)\in\text{span}(\mathcal{S}_i)$. We define $\mathcal{E}\leftarrow$ ortho$(\mathcal{S}_i\cup\mathcal{X}^*)$ where we observe rank(span$(\mathcal{E})) \leq 4r$ and let $\mathcal{P}_\mathcal{E}$ be the orthogonal projection onto the subspace defined by $\mathcal{E}$.

Since $\mathbf{X}(i+1)-\mathbf{X}^\star\in\text{span}(\mathcal{E})$ and $\mathbf{V}(i)-\mathbf{X}^\star\in\text{span}(\mathcal{E})$, the following hold true:

$$\mathbf{X}(i+1)-\mathbf{X}^\star = \mathcal{P}_\mathcal{E}(\mathbf{X}(i+1)-\mathbf{X}^\star),$$

and,

$$\mathbf{V}(i)-\mathbf{X}^\star = \mathcal{P}_\mathcal{E}(\mathbf{V}(i)-\mathbf{X}^\star).$$

$$\begin{aligned} g(i+1) &\leq \left[b_1\Big(\frac{\alpha(1+\tau_i)+\sqrt{\Delta}}{2}\Big)^{i+1} + b_2\Big(\frac{\alpha(1+\tau_i)-\sqrt{\Delta}}{2}\Big)^{i+1}\right]\|\mathbf{X}(0)-\mathbf{X}^\star\|_F \\ &\leq \left[(b_1+b_2)\Big(\frac{\alpha(1+\tau_i)+\sqrt{\Delta}}{2}\Big)^{i+1}\right]\|\mathbf{X}(0)-\mathbf{X}^\star\|_F \end{aligned} \tag{4.78}$$

Then, (4.101) can be written as:

$$\begin{aligned}
\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F^2 &\leq 2\langle \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i+1)-\mathbf{X}^\star), \mathcal{P}_{\mathcal{E}}(\mathbf{V}(i)-\mathbf{X}^\star)\rangle \\
&= 2\langle \mathcal{P}_{\mathcal{E}}(\mathbf{X}(i+1)-\mathbf{X}^\star), \mathcal{P}_{\mathcal{E}}\left(\boldsymbol{Q}_i+\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{X}^\star-\boldsymbol{Q}_i)-\mathbf{X}^\star\right)\rangle \\
&\overset{(i)}{=} 2\langle \mathbf{X}(i+1)-\mathbf{X}^\star, \mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}[\mathcal{P}_{\mathcal{S}_i}+\mathcal{P}_{\mathcal{S}_i^\perp}]\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\rangle \\
&= 2\langle \mathbf{X}(i+1)-\mathbf{X}^\star, (\mathbf{I}-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\rangle - 2\mu_i\langle \mathbf{X}(i+1)-\mathbf{X}^\star, \mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\rangle \\
&\overset{(ii)}{\leq} 2\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F\|(\mathbf{I}-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F \\
&+ 2\mu_i\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F
\end{aligned} \tag{4.79}$$

where (i) is due to $\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star) := \mathcal{P}_{\mathcal{S}_i}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)+\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)$ and (ii) follows from Cauchy-Schwarz inequality. Since $\frac{1}{1+\delta_{3r}} \leq \mu_i \leq \frac{1}{1-\delta_{3r}}$, Lemma 27 implies:

$$\lambda(\mathbf{I}-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i}) \in \left[1-\frac{1-\delta_{3r}}{1+\delta_{3r}}, \frac{1+\delta_{3r}}{1-\delta_{3r}}-1\right] \leq \frac{2\delta_{3r}}{1-\delta_{3r}}.$$

and thus:

$$\|(\mathbf{I}-\mu_i\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i})\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F \leq \frac{2\delta_{3r}}{1-\delta_{3r}}\|\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F.$$

Furthermore, according to Lemma 28:

$$\|\mathcal{P}_{\mathcal{S}_i}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F \leq \delta_{4r}\|\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F$$

since $\text{rank}(\mathcal{P}_{\mathcal{K}}\boldsymbol{Q}) \leq 4r,\ \forall \boldsymbol{Q} \in \mathbb{R}^{p\times n}$ where $\mathcal{K} \leftarrow \text{ortho}(\mathcal{E}\cup\mathcal{S}_i)$. Since $\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star) = \mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\mathbf{X}^\star$ where

$$\mathcal{D}_i \leftarrow \mathcal{P}_k\left(\mathcal{P}_{\mathcal{Q}_i^\perp}\nabla f(\boldsymbol{Q}_i)\right),$$

then:

$$\|\mathcal{P}_{\mathcal{S}_i^\perp}\mathcal{P}_{\mathcal{E}}(\boldsymbol{Q}_i-\mathbf{X}^\star)\|_F = \|\mathcal{P}_{\mathcal{X}^*\setminus(\mathcal{D}_i\cup\mathcal{X}_i)}\mathbf{X}^\star\|_F \leq (2\delta_{3r}+2\delta_{4r})\|\boldsymbol{Q}_i-\mathbf{X}^\star\|_F, \tag{4.80}$$

using Lemma 21. Using the above in (4.79), we compute:

$$\|\mathbf{X}(i+1)-\mathbf{X}^\star\|_F \leq \left(\frac{4\delta_{3r}}{1-\delta_{3r}}+(2\delta_{3r}+2\delta_{4r})\frac{2\delta_{3r}}{1-\delta_{3r}}\right)\|\boldsymbol{Q}_i-\mathbf{X}^\star\|_F \tag{4.81}$$

Furthermore:

$$\begin{aligned}
\|\boldsymbol{Q}_i-\mathbf{X}^\star\|_F &= \|\mathbf{X}(i)+\tau_i(\mathbf{X}(i)-\mathbf{X}(i-1))\|_F \\
&= \|(1+\tau_i)(\mathbf{X}(i)-\mathbf{X}^\star)+\tau_i(\mathbf{X}^\star-\mathbf{X}(i-1))\|_F \\
&\leq (1+\tau_i)\|\mathbf{X}(i)-\mathbf{X}^\star\|_F+\tau_i\|\mathbf{X}(i-1)-\mathbf{X}^\star\|_F
\end{aligned} \tag{4.82}$$

Combining (4.81) and (4.108), we get:

$$\begin{aligned}\big\|\mathbf{X}(i+1)-\mathbf{X}^\star\big\|_F \leq (1+\tau_i)\Big(\frac{4\delta_{3r}}{1-\delta_{3r}}+(2\delta_{3r}+2\delta_{4r})\frac{2\delta_{3r}}{1-\delta_{3r}}\Big)\big\|\mathbf{X}(i)-\mathbf{X}^\star\big\|_F \\ +\tau_i\Big(\frac{4\delta_{3r}}{1-\delta_{3r}}+(2\delta_{3r}+2\delta_{4r})\frac{2\delta_{3r}}{1-\delta_{3r}}\Big)\big\|\mathbf{X}(i-1)-\mathbf{X}^\star\big\|_F \end{aligned} \tag{4.83}$$

Let $\alpha := \frac{4\delta_{3r}}{1-\delta_{3r}}+(2\delta_{3r}+2\delta_{4r})\frac{2\delta_{3r}}{1-\delta_{3r}}$ and $g(i) := \big\|\mathbf{X}(i+1)-\mathbf{X}^\star\big\|_F$. Then, (4.83) defines the following homogeneous recurrence:

$$g(i+1)-\alpha(1+\tau_i)g(i)+\alpha\tau_i g(i-1)\leq 0 \tag{4.84}$$

Using the *method of characteristic roots* to solve the above recurrence, we assume that the homogeneous linear recursion has solution of the form $g(i)=r^i$ for $r\in\mathbb{R}$. Thus, replacing $g(i)=r^i$ in (4.84) and factoring out $r^{(i-2)}$, we form the following characteristic polynomial:

$$r^2-\alpha(1+\tau_i)r-\alpha\tau_i\leq 0 \tag{4.85}$$

Focusing on the worst case where (4.85) is satisfied with equality, we compute the roots $r_{1,2}$ of the quadratic characteristic polynomial as:

$$r_{1,2}=\frac{\alpha(1+\tau_i)\pm\sqrt{\Delta}}{2},\ \text{where}\ \Delta:=\alpha^2(1+\tau_i)^2+4\alpha\tau_i.$$

Then, as a general solution, we combine the above roots with unknown coefficients b_1,b_2 to obtain (4.78). Using the initial condition $g(0):=\big\|\mathbf{X}(0)-\mathbf{X}^\star\big\|_F \overset{\mathbf{X}(0)=\mathbf{0}}{=} \big\|\mathbf{X}^\star\big\|_F=1$, we get $b_1+b_2=1$. Thus, we conclude to the following recurrence:

$$\big\|\mathbf{X}(i+1)-\mathbf{X}^\star\big\|_F\leq\Big(\frac{\alpha(1+\tau_i)+\sqrt{\Delta}}{2}\Big)^{i+1}.$$

Proof of Theorem 12

There are three aspects to the proof. Even without approximate SVD calculations, the problem is non-convex, so we must leverage the R-RIP to prove that iterates converge. Mixed in with this calculation is the approximate nature of our rank ℓ point $\widetilde{\mathbf{X}}_{i+1}$, where we will apply the bounds from Theorem 11. Finally, we relate $\widetilde{\mathbf{X}}_{i+1}$ to its rank r version $\mathbf{X}_{i+1}$.

An important definition for our subsequent developments is the following:

Definition 14 (ϵ-approximate low-rank projection)**.** *Let $\mathbf{X}$ be an arbitrary matrix. For any $\epsilon>0$, $\mathcal{P}^{\epsilon}_{r',\ell'}(\mathbf{X})$ provides a rank-ℓ' matrix approximation to $\mathbf{X}$ such that*

$$\mathbf{E}\big\|\mathcal{P}^{\epsilon}_{r',\ell'}(\mathbf{X})-\mathbf{X}\big\|_F^2\leq(1+\epsilon)\big\|\mathcal{P}_{r'}(\mathbf{X})-\mathbf{X}\big\|_F^2, \tag{4.86}$$

where $\mathcal{P}_{r'}(\mathbf{X})\in\arg\min_{\mathbf{Y}:r(\mathbf{Y})\leq r'}\|\mathbf{X}-\mathbf{Y}\|_F$.

Let $\mathbf{X}_i$ be the putative rank r solution at the i-th iteration, $\mathbf{X}^\star$ be the rank r matrix we are looking for and $\widetilde{\mathbf{X}}_{i+1}$ be the rank l matrix, obtained using approximate SVD calculations. Define $L:=2(1+\delta_{r+\ell})$ and

$M := 2(1-\delta_{2r})$. Then, we have:

$$\begin{aligned} f(\widetilde{\mathbf{X}}_{i+1}) &= f(\mathbf{X}_i) + \langle \nabla f(\mathbf{X}_i),\ \widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i\rangle + \|\boldsymbol{\mathcal{A}}(\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i)\|_F^2 \\ &\leq f(\mathbf{X}_i) + \langle \nabla f(\mathbf{X}_i),\ \widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i\rangle + \frac{L}{2}\|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i\|_F^2 \\ &= f(\mathbf{X}_i) - \frac{1}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 + \frac{L}{2}\left(\|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i\|_F^2 + 2\langle \frac{1}{L}\nabla f(\mathbf{X}_i),\ \widetilde{\mathbf{X}}_{i+1} - \mathbf{X}_i\rangle + \frac{1}{L^2}\|\nabla f(\mathbf{X}_i)\|_F^2\right) \\ &= f(\mathbf{X}_i) - \frac{1}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 + \frac{L}{2}\|\widetilde{\mathbf{X}}_{i+1} - \left(\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i)\right)\|_F^2. \end{aligned} \tag{4.87}$$

By construction $\widetilde{\mathbf{X}}_{i+1} \in \mathcal{P}^{\epsilon}_{r,\ell}\left(\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i)\right)$ (since the step-size is $\mu = 1/L$), so, for $\overline{\mathbf{X}}_{i+1} \in \mathcal{P}_r\left(\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i)\right)$,

$$\begin{aligned} \mathbf{E}\|\widetilde{\mathbf{X}}_{i+1} - (\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i))\|_F^2 &\leq (1+\epsilon)\|\overline{\mathbf{X}}_{i+1} - (\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i))\|_F^2 \\ &\leq (1+\epsilon)\|\mathbf{X}^\star - (\mathbf{X}_i - \frac{1}{L}\nabla f(\mathbf{X}_i))\|_F^2 \end{aligned} \tag{4.88}$$

by the definition of $\mathcal{P}_r(\cdot)$ (since $r(\mathbf{X}^\star) = r$). Combining (4.88) with (5.48), we obtain:

$$\begin{aligned} \mathbf{E}f(\widetilde{\mathbf{X}}_{i+1}) &\leq f(\mathbf{X}_i) - \frac{1}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 + \frac{L}{2}(1+\epsilon)\|\mathbf{X}^\star - \mathbf{X}_i + \frac{1}{L}\nabla f(\mathbf{X}_i)\|_F^2 \\ &= f(\mathbf{X}_i) - \frac{1}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 + (1+\epsilon)\left(\frac{1}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 + \langle \nabla f(\mathbf{X}_i),\ \mathbf{X}^\star - \mathbf{X}_i\rangle + \frac{L}{2}\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2\right) \\ &\leq (1+\epsilon)\left[f(\mathbf{X}_i) + \langle \nabla f(\mathbf{X}_i),\ \mathbf{X}^\star - \mathbf{X}_i\rangle + \frac{L}{2}\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2\right] + \frac{\epsilon}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 \end{aligned} \tag{4.89}$$

where we use the fact that $f(\mathbf{X}_i) \geq 0$ in the last inequality. Due to the restricted strong convexity of f that follows from the restricted isometry property, we have:

$$\begin{aligned} f(\mathbf{X}^\star) &\geq f(\mathbf{X}_i) + \langle \nabla f(\mathbf{X}_i), \mathbf{X}^\star - \mathbf{X}_i\rangle + \frac{M}{2}\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2 \\ f(\mathbf{X}^\star) - \frac{M}{2}\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2 &\geq f(\mathbf{X}_i) + \langle \nabla f(\mathbf{X}_i), \mathbf{X}^\star - \mathbf{X}_i\rangle \end{aligned}$$

which, combined with (4.89), leads to:

$$\mathbf{E}f(\widetilde{\mathbf{X}}_{i+1}) \leq (1+\epsilon)\left[f(\mathbf{X}^\star) + \frac{L-M}{2}\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2\right] + \frac{\epsilon}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 \tag{4.90}$$

Due to the R-RIP,

$$\|\mathbf{X}^\star - \mathbf{X}_i\|_F^2 \leq \frac{\|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_i)\|_2^2}{1-\delta_{2r}} \tag{4.91}$$

Now define a constant C and assume $f(\mathbf{X}_i) = \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{X}_i\|_2^2 > C^2\|\boldsymbol{\varepsilon}\|_2^2$ (if the assumption fails, it means $\mathbf{X}_i$ is already close to $\mathbf{X}^\star$). In particular, in the noiseless case $\|\boldsymbol{\varepsilon}\| = 0$, we may pick C arbitrarily large and

set all $1/C$ terms to zero.

$$\begin{aligned}\|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_i)\|_F^2 &= \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i) - \boldsymbol{\varepsilon}\|_2^2 \\ &= \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i)\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 - 2\langle \boldsymbol{\varepsilon}, \boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i)\rangle \\ &\leq f(\mathbf{X}_i) + \|\boldsymbol{\varepsilon}\|_2^2 + 2\|\boldsymbol{\varepsilon}\|_2 \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i)\|_2 \\ &\leq f(\mathbf{X}_i) + \|\boldsymbol{\varepsilon}\|_2^2 + \frac{2}{C} f(\mathbf{X}_i) \end{aligned} \tag{4.92}$$

Substituting (4.92) and (5.52) into (5.50), expanding the values of L and M, and noting that $f(\mathbf{X}^\star) = \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}^\star)\|_2^2 = \|\boldsymbol{\varepsilon}\|_2^2$, gives

$$\begin{aligned}\mathbf{E} f(\widetilde{\mathbf{X}}_{i+1}) &\leq (1+\epsilon)\left[\|\boldsymbol{\varepsilon}\|_2^2 + \frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\left(f(\mathbf{X}_i) + \|\boldsymbol{\varepsilon}\|_2^2 + \frac{2}{C} f(\mathbf{X}_i)\right)\right] + \frac{\epsilon}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 \\ &\leq (1+\epsilon)\left[\frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\left(1+\frac{2}{C}\right) f(\mathbf{X}_i) + \left(1 + \frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\right)\|\boldsymbol{\varepsilon}\|_2^2\right] + \frac{\epsilon}{2L}\|\nabla f(\mathbf{X}_i)\|_F^2 \end{aligned} \tag{4.93}$$

We bound $\|\nabla f(\mathbf{X}_i)\|$ using our assumption on the magnitude of $\|\boldsymbol{\mathcal{A}}\|$:

$$\|\nabla f(\mathbf{X}_i)\|_F^2 = 4\|\boldsymbol{\mathcal{A}}^*\left(\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i)\right)\|_F^2 \leq 4\|\boldsymbol{\mathcal{A}}^*\|^2 \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_i)\|_2^2 = 4\|\boldsymbol{\mathcal{A}}\|^2 f(\mathbf{X}_i) \leq 4\frac{mn}{m} f(\mathbf{X}_i) \tag{4.94}$$

For quantum tomography, we even have $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{A}}^* = \frac{mn}{m}\mathcal{I}$, so the inequality holds with equality (and $m = n$).

Combining (4.93) with (4.94) and by the definition of L, we obtain:

$$\begin{aligned}\mathbf{E} f(\widetilde{\mathbf{X}}_{i+1}) &\leq (1+\epsilon)\left[\frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\left(1+\frac{2}{C}\right) f(\mathbf{X}_i) + \left(1 + \frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\right)\|\boldsymbol{\varepsilon}\|_2^2\right] + \frac{\epsilon}{1+\delta_{r+\ell}} \cdot \frac{mn}{m} f(\mathbf{X}_i) \\ &= \underbrace{\left(\frac{\epsilon}{1+\delta_{r+\ell}} \cdot \frac{mn}{m} + (1+\epsilon)\frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\left(1+\frac{2}{C}\right)\right)}_{\theta'} f(\mathbf{X}_i) + \underbrace{(1+\epsilon)\left(1 + \frac{\delta_{r+\ell}+\delta_{2r}}{1-\delta_{2r}}\right)}_{\tau'}\|\boldsymbol{\varepsilon}\|_2^2 \end{aligned} \tag{4.95}$$

Note that if an exact SVD computation is used, then not only is $\epsilon = 0$ but also $\widetilde{\mathbf{X}}_{i+1}$ is rank r, so we are done and can use $\theta = \theta'$ and $\tau = \tau'$. To finish the proof, we now relate $\mathbf{E} f(\mathbf{X}_{i+1})$ to $\mathbf{E} f(\widetilde{\mathbf{X}}_{i+1})$. In the algorithm, $\mathbf{X}_{i+1}$ is the output of `RandomizedSVD`, and $\widetilde{\mathbf{X}}_{i+1}$ is the intermediate value $U\Sigma V^{\mathrm{H}}$ on line 10 of Algo. 12. Given $\widetilde{\mathbf{X}}_{i+1}$ with $r(\widetilde{\mathbf{X}}_{i+1}) = \ell > r$, $\mathbf{X}_{i+1}$ is defined as the best rank-r approximation to $\widetilde{\mathbf{X}}_{i+1}$.[10] Thus, the following inequality holds true:

$$\begin{aligned}\|\mathbf{X}_{i+1} - \mathbf{X}^\star\|_F &= \|\mathbf{X}_{i+1} - \widetilde{\mathbf{X}}_{i+1} + \widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F \\ &\leq \|\mathbf{X}_{i+1} - \widetilde{\mathbf{X}}_{i+1}\|_F + \|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F \\ &\leq 2\|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F \end{aligned} \tag{4.96}$$

since $\|\mathbf{X}_{i+1} - \widetilde{\mathbf{X}}_{i+1}\|_F \leq \|\mathbf{X}^\star - \widetilde{\mathbf{X}}_{i+1}\|_F$. In particular, since the above is valid for any value of the random variable $\widetilde{\mathbf{X}}_{i+1}$, $\mathbf{E}\ \|\mathbf{X}_{i+1} - \mathbf{X}^\star\|_F^2 \leq \mathbf{E}\ 4\|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F^2$. This bound is pessimistic and in practice the

[10] If we include a convex constraint $\mathcal{C}$ then instead of defining $\mathbf{X}_{i+1} = \mathcal{P}_r(\widetilde{\mathbf{X}}_{i+1})$ we have $\mathbf{X}_{i+1} = \mathcal{P}_C(\mathcal{P}_r(\widetilde{\mathbf{X}}_{i+1}))$. In this case,

$$\|\mathcal{P}_C(\mathcal{P}_r(\widetilde{\mathbf{X}}_{i+1})) - \mathbf{X}^\star\|_F = \|\mathcal{P}_C(\mathcal{P}_r(\widetilde{\mathbf{X}}_{i+1}) - \mathbf{X}^\star)\|_F \leq \|\mathcal{P}_r(\widetilde{\mathbf{X}}_{i+1}) - \mathbf{X}^\star\|_F.$$

The first equality follows from $\mathbf{X}^\star \in \mathcal{C}$ and the second is true since the projection onto a non-empty closed convex set is non-expansive. Hence the result in (4.96) still applies when we include the $\mathcal{C}$ constraints.

constant is close to 1 rather than 4.

We will again assume that $f(\widetilde{\mathbf{X}}_{i+1}), f(\mathbf{X}_{i+1}) \geq C^2\|\boldsymbol{\varepsilon}\|_2^2$, and $C > 2$, since otherwise the current point is a good-enough solution. We have:

$$\begin{aligned} f(\mathbf{X}_{i+1}) = \|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_{i+1})\|_2^2 &= \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1}) + \boldsymbol{\varepsilon}\|_2^2 \\ &= \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 + 2\langle \boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1}), \boldsymbol{\varepsilon}\rangle \\ &= \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 + 2\langle \boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_{i+1}) - \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle \\ &= \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 + 2\langle \boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_{i+1}), \boldsymbol{\varepsilon}\rangle + 2\langle -\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle \\ &\leq \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 + \|\boldsymbol{\varepsilon}\|_2^2 + 2\|\boldsymbol{y} - \boldsymbol{\mathcal{A}}(\mathbf{X}_{i+1})\|_2\|\boldsymbol{\varepsilon}\|_2 - 2\|\boldsymbol{\varepsilon}\|_2^2 \\ &\leq \|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 - \|\boldsymbol{\varepsilon}\|_2^2 + \frac{2}{C} f(\mathbf{X}_{i+1}) \end{aligned}$$

which, if $1 - 2/C \geq 0$, implies

$$f(\mathbf{X}_{i+1}) \leq \frac{1}{1-2/C}\|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 - \frac{1}{1-2/C}\|\boldsymbol{\varepsilon}\|_2^2 \tag{4.97}$$

By the R-RIP assumption, we have:

$$\|\boldsymbol{\mathcal{A}}(\mathbf{X}^\star - \mathbf{X}_{i+1})\|_2^2 \leq (1+\delta_{2r})\|\mathbf{X}^\star - \mathbf{X}_{i+1}\|_F^2. \tag{4.98}$$

Using (4.96) and (4.98) in (4.97), we obtain:

$$f(\mathbf{X}_{i+1}) \leq \frac{4(1+\delta_{2r})}{1-2/C}\|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F^2 - \frac{1}{1-2/C}\|\boldsymbol{\varepsilon}\|_2^2 \tag{4.99}$$

Using the R-RIP property again, the following sequence of inequalities holds:

$$\begin{aligned} \|\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star\|_F^2 &\leq \frac{\|\boldsymbol{\mathcal{A}}(\widetilde{\mathbf{X}}_{i+1} - \mathbf{X}^\star)\|_F^2}{1-\delta_{r+\ell}} \\ &\leq \frac{1+2/C}{1-\delta_{r+\ell}} f(\widetilde{\mathbf{X}}_{i+1}) + \frac{1}{1-\delta_{r+\ell}}\|\boldsymbol{\varepsilon}\|_2^2 \end{aligned} \tag{4.100}$$

where the second inequality is obtained following same motions as (4.92). Combining (4.99)-(4.100) with (4.95), we obtain:

$$\mathbf{E} f(\mathbf{X}_{i+1}) \leq \underbrace{\frac{4(1+\delta_{2r})}{1-2/C} \cdot \frac{1+2/C}{1-\delta_{r+\ell}} \cdot \theta'}_{\theta} \cdot f(\mathbf{X}_i) + \underbrace{\left(\frac{4(1+\delta_{2r})}{1-2/C} \cdot \frac{1+2/C}{1-\delta_{r+\ell}} \cdot \tau' + \frac{4(1+\delta_{2r})}{1-2/C} \cdot \frac{1}{1-\delta_{r+\ell}} - \frac{1}{1-2/C} \right)}_{\tau} \|\boldsymbol{\varepsilon}\|_2^2$$

Now we simplify the result to make it more interpretable. Define $\rho = \ell - r$. Let c be the smallest integer such that $\ell \geq (c-1)r$ (and for simplicity, assume $\ell = (c-1)r$) so that $\delta_{r+\ell} = \delta_{cr}$ and $\delta_{r+\ell} + \delta_{2r} \leq 2\delta_{cr}$. By Theorem 11, $\epsilon \leq \frac{r}{\rho - 1} = \frac{r}{(c-2)r-1}$. For concreteness, take $C \geq 4$ so that $1 + 2/C \leq 3/2$ and $(1-2/C)^{-1} \leq 2$. Then

$$\theta \leq 12 \cdot \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(\frac{\epsilon}{1+\delta_{cr}} \cdot \frac{mn}{m} + (1+\epsilon)\frac{3\delta_{cr}}{1-\delta_{2r}} \right)$$

and

$$\begin{aligned}\tau &\leq \left(12 \cdot \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot (1+\epsilon)\left(1+\frac{\delta_{2r}+\delta_{cr}}{1-\delta_{2r}}\right) + \frac{8(1+\delta_{2r})}{1-\delta_{cr}}\right) \\ &\leq \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(12 \cdot (1+\epsilon)\left(1+\frac{2\delta_{cr}}{1-\delta_{2r}}\right) + 8\right)\end{aligned}$$

Proof of Theorem 13

Here, we prove the convergence of Algorithm 2, both for the low rank and the sparse matrix estimate part, and then combine the corresponding theoretical results. Let $\mathcal{L}^* \leftarrow \text{ortho}(\mathbf{L}^\star)$ be a set of orthonormal, rank-1 matrices that span the range of $\mathbf{L}^\star$ and $\mathcal{M}^*$ be the set of indices of the non-zero elemetns in $\mathbf{M}^*$. For the low rank matrix estimate, we observe the following:

$$\begin{aligned}\left\|\mathbf{L}_{i+1} - \mathbf{V}_i^{\mathcal{L}}\right\|_F^2 \leq \left\|\mathbf{L}^\star - \mathbf{V}_i^{\mathcal{L}}\right\|_F^2 \Rightarrow \\ \left\|\mathbf{L}_{i+1} - \mathbf{L}^\star + \mathbf{L}^\star - \mathbf{V}_i^{\mathcal{L}}\right\|_F^2 \leq \left\|\mathbf{L}^\star - \mathbf{V}_i^{\mathcal{L}}\right\|_F^2 \Rightarrow \\ \left\|\mathbf{L}_{i+1} - \mathbf{L}^\star\right\|_F^2 + \left\|\mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star\right\|_F^2 + 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathbf{L}^\star - \mathbf{V}_i^{\mathcal{L}}\rangle \leq \left\|\mathbf{L}^\star - \mathbf{V}_i^{\mathcal{L}}\right\|_F^2 \Rightarrow \\ \left\|\mathbf{L}_{i+1} - \mathbf{L}^\star\right\|_F^2 \leq 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star\rangle \end{aligned} \tag{4.101}$$

From Algorithm 2, it is obvious that (i) $\mathbf{V}_i^{\mathcal{L}} \in \text{span}(\mathcal{S}_i^{\mathcal{L}})$, (ii) $\mathbf{Q}_i^{\mathcal{L}} \in \text{span}(\mathcal{S}_i^{\mathcal{L}})$ and (iii) $\mathbf{L}_{i+1} \in \text{span}(\mathcal{S}_i^{\mathcal{L}})$. We define $\mathcal{E} := \mathcal{S}_i^{\mathcal{L}} \cup \mathcal{L}^*$ where $\text{rank}(\text{span}(\mathcal{E})) \leq 4r$ and let $\mathcal{P}_{\mathcal{E}}$ be the orthogonal projection onto the subspace defined by $\mathcal{E}$. We highlight that $\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}} = \mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}$.

Since $\mathbf{L}_{i+1} - \mathbf{L}^\star \in \text{span}(\mathcal{E})$ and $\mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star \in \text{span}(\mathcal{E})$, the following hold true:

$$\mathbf{L}_{i+1} - \mathbf{L}^\star = \mathcal{P}_{\mathcal{E}}(\mathbf{L}_{i+1} - \mathbf{L}^\star) \;\text{ and }\; \mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star = \mathcal{P}_{\mathcal{E}}(\mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star).$$

Then, (4.101) can be written as:

$$\begin{aligned}&\left\|\mathbf{L}_{i+1} - \mathbf{L}^\star\right\|_F^2 \\ &\leq 2\langle \mathcal{P}_{\mathcal{E}}(\mathbf{L}_{i+1} - \mathbf{L}^\star), \mathcal{P}_{\mathcal{E}}(\mathbf{V}_i^{\mathcal{L}} - \mathbf{L}^\star)\rangle \\ &= 2\langle \mathcal{P}_{\mathcal{E}}(\mathbf{L}_{i+1} - \mathbf{L}^\star), \mathcal{P}_{\mathcal{E}}\left(\mathbf{Q}_i^{\mathcal{L}} + \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*(\boldsymbol{y} - \boldsymbol{\mathcal{A}}\mathbf{Q}_i) - \mathbf{L}^\star\right)\rangle\end{aligned}$$

$$\begin{aligned}&= 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) + \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\left(\boldsymbol{\mathcal{A}}^*(\boldsymbol{\mathcal{A}}(\mathbf{L}^\star + \mathbf{M}^\star) - \boldsymbol{\mathcal{A}}\mathbf{Q}_i)\right)\rangle \qquad (4.102) \\ &= 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) + \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\left(\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{L}^\star + \mathbf{M}^\star) - \boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{L}} + \mathbf{Q}_i^{\mathcal{M}})\right)\rangle \\ &= 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star)\rangle \\ &= 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\rangle - 2\mu_i^{\mathcal{L}}\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star)\rangle \\ &= 2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\left[\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}} + \mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\right]\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\rangle \\ &- 2\mu_i^{\mathcal{L}}\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star)\rangle \qquad (4.103)\end{aligned}$$

due to $\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) := \mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) + \mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)$. The first term in (4.103) satisfies:

$$
\begin{aligned}
&2\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star) - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}[\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}} + \mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}]\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\rangle \\
&\leq 2\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\big\|(I - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}})\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F + 2\mu_i^{\mathcal{L}}\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\big\|\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F \\
&\leq \frac{4\delta_{3r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F + \frac{2\delta_{4r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}(2\delta_{3r}(\boldsymbol{\mathcal{A}}) + 2\delta_{4r}(\boldsymbol{\mathcal{A}}))\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F \quad (4.104)
\end{aligned}
$$

where (4.104) holds, since $\frac{1}{1+\delta_{3r}(\boldsymbol{\mathcal{A}})} \leq \mu_i^{\mathcal{L}} \leq \frac{1}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}$, using Lemma 3 in [**?**]:

$$
\lambda(\mathbf{I} - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}) \in \left[1 - \frac{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}{1+\delta_{3r}(\boldsymbol{\mathcal{A}})}, \frac{1+\delta_{3r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})} - 1\right] \leq \frac{2\delta_{3r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}. \quad (4.105)
$$

and thus:

$$
\big\|(\mathbf{I} - \mu_i^{\mathcal{L}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}})\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F \leq \frac{2\delta_{3r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}\big\|\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F.
$$

Furthermore, according to Lemma 4 in [KC11]:

$$
\big\|\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F \leq \delta_{4r}(\boldsymbol{\mathcal{A}})\big\|\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F
$$

since $\text{rank}(\mathcal{P}_{\mathcal{E}\cup\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{Q}) \leq 4r,\ \forall \boldsymbol{Q} \in \mathcal{R}^{p\times n}$. Moreover:

$$
\big\|\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}^{\perp}\mathcal{P}_{\mathcal{E}}(\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star)\big\|_F \leq (2\delta_{3r}(\boldsymbol{\mathcal{A}}) + 2\delta_{4r}(\boldsymbol{\mathcal{A}}))\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F,
$$

using ideas from Lemma 21.

The second term in (4.103) satsifies:

$$
\begin{aligned}
2\mu_i^{\mathcal{L}}\langle \mathbf{L}_{i+1} - \mathbf{L}^\star, \mathcal{P}_{\mathcal{E}}\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star)\rangle &\leq \frac{2}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\big\|\mathcal{P}_{\mathcal{S}_i^{\mathcal{L}}}\boldsymbol{\mathcal{A}}^*\boldsymbol{\mathcal{A}}(\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star)\big\|_F \\
&\leq \frac{2}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F\delta_{3r+3k}(\boldsymbol{\mathcal{A}})\big\|\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star\big\|_F
\end{aligned}
$$

using Lemma 3.2 in [WSB11]. Replacing the above results in (4.103), we compute:

$$
\big\|\mathbf{L}_{i+1} - \mathbf{L}^\star\big\|_F \leq \alpha\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F + \beta\big\|\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star\big\|_F, \quad (4.106)
$$

where $\alpha := \left(\frac{4\delta_{3r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})} + \frac{2\delta_{4r}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}(2\delta_{3r}(\boldsymbol{\mathcal{A}}) + 2\delta_{4r}(\boldsymbol{\mathcal{A}}))\right)$ and $\beta := \frac{2\delta_{3r+3k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3r}(\boldsymbol{\mathcal{A}})}$. Following similar steps for the sparse matrix estimate part, we end up with the following inequality bound for $\mathbf{M}_{i+1}$:

$$
\big\|\mathbf{M}_{i+1} - \mathbf{M}^\star\big\|_F \leq \gamma\big\|\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star\big\|_F + \zeta\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F, \quad (4.107)
$$

where $\gamma := \frac{2(\delta_{4k}(\boldsymbol{\mathcal{A}}) + \delta_{3k}(\boldsymbol{\mathcal{A}}))}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}$ and $\zeta := \frac{2\delta_{3r+4k}(\boldsymbol{\mathcal{A}})}{1-\delta_{3k}(\boldsymbol{\mathcal{A}})}$.

Furthermore:

$$
\begin{aligned}
\big\|\mathbf{Q}_i^{\mathcal{L}} - \mathbf{L}^\star\big\|_F &= \big\|\mathbf{L}_i + \tau_i(\mathbf{L}_i - \mathbf{L}_{i-1}) - \mathbf{L}^\star\big\|_F \\
&= \big\|(1+\tau_i)(\mathbf{L}_i - \mathbf{L}^\star) + \tau_i(\mathbf{L}^\star - \mathbf{L}_{i-1})\big\|_F \\
&\leq (1+\tau_i)\big\|\mathbf{L}_i - \mathbf{L}^\star\big\|_F + \tau_i\big\|\mathbf{L}_{i-1} - \mathbf{L}^\star\big\|_F \quad (4.108)
\end{aligned}
$$

and

$$\begin{aligned}\left\|\mathbf{Q}_i^{\mathcal{M}} - \mathbf{M}^\star\right\|_F &= \left\|\mathbf{M}_i + \tau_i(\mathbf{M}_i - \mathbf{M}_{i-1}) - \mathbf{M}^\star\right\|_F \\ &= \left\|(1+\tau_i)(\mathbf{M}_i - \mathbf{M}^\star) + \tau_i(\mathbf{M}^\star - \mathbf{M}_{i-1})\right\|_F \\ &\leq (1+\tau_i)\left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F + \tau_i\left\|\mathbf{M}_{i-1} - \mathbf{M}^\star\right\|_F \end{aligned} \tag{4.109}$$

Combining (4.108), (4.109) into (4.106) and (4.107), we get:

$$\begin{aligned}\left\|\mathbf{L}_{i+1} - \mathbf{L}^\star\right\|_F \leq \alpha(1+\tau_i)\left\|\mathbf{L}_i - \mathbf{L}^\star\right\|_F + \alpha\tau_i\left\|\mathbf{L}_{i-1} - \mathbf{L}^\star\right\|_F \\ + \beta(1+\tau_i)\left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F + \beta\tau_i\left\|\mathbf{M}_{i-1} - \mathbf{M}^\star\right\|_F \end{aligned} \tag{4.110}$$

and

$$\begin{aligned}\left\|\mathbf{M}_{i+1} - \mathbf{M}^\star\right\|_F \leq \gamma(1+\tau_i)\left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F + \gamma\tau_i\left\|\mathbf{M}_{i-1} - \mathbf{M}^\star\right\|_F \\ + \zeta(1+\tau_i)\left\|\mathbf{L}_i - \mathbf{L}^\star\right\|_F + \zeta\tau_i\left\|\mathbf{L}_{i-1} - \mathbf{L}^\star\right\|_F \end{aligned} \tag{4.111}$$

The inequalities (4.110) and (4.111) define the following coupled set of inequalities:

$$\begin{bmatrix} \left\|\mathbf{L}_{i+1} - \mathbf{L}^\star\right\|_F \\ \left\|\mathbf{M}_{i+1} - \mathbf{M}^\star\right\|_F \end{bmatrix} \leq (1+\tau_i)\boldsymbol{\Delta}\begin{bmatrix} \left\|\mathbf{L}_i - \mathbf{L}^\star\right\|_F \\ \left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F \end{bmatrix} + \tau_i\boldsymbol{\Delta}\begin{bmatrix} \left\|\mathbf{L}_{i-1} - \mathbf{L}^\star\right\|_F \\ \left\|\mathbf{M}_{i-1} - \mathbf{M}^\star\right\|_F \end{bmatrix} \tag{4.112}$$

where $\boldsymbol{\Delta} := \begin{bmatrix}\alpha & \beta \\ \zeta & \gamma\end{bmatrix}$. Furthermore, we define $\mathbf{x}(i) := \begin{bmatrix} \left\|\mathbf{L}_i - \mathbf{L}^\star\right\|_F \\ \left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F \end{bmatrix}$ to obtain inequality (4.34). We can convert this second-order linear system into a two-dimensional first-order system where the variables of the linear system are multi-dimensional. To achieve this, we define a new state variable $\mathbf{y}(i)$ where:

$$\mathbf{y}(i) := \mathbf{x}(i+1).$$

and thus, $\mathbf{y}(i+1) := \mathbf{x}(i+2)$. Using the new variable above, we define the following two-dimensional first-order system:

$$\begin{cases} \mathbf{y}(i+1) - (1+\tau_i)\boldsymbol{\Delta}\mathbf{y}(i) - \tau_i\boldsymbol{\Delta}\mathbf{x}(i) \leq 0, \\ \mathbf{x}(i+1) \leq \mathbf{y}(i). \end{cases}$$

which, moreover, defines the following linear system that characterizes the evolution of two state variables, $\{\mathbf{y}(i), \mathbf{x}(i)\}$:

$$\begin{bmatrix}\mathbf{y}(i+1) \\ \mathbf{x}(i+1)\end{bmatrix} \leq \begin{bmatrix}(1+\tau_i)\boldsymbol{\Delta} & \tau_i\boldsymbol{\Delta} \\ \mathbb{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{y}(i) \\ \mathbf{x}(i)\end{bmatrix} \Rightarrow \begin{bmatrix}\mathbf{x}(i+2) \\ \mathbf{x}(i+1)\end{bmatrix} \leq \begin{bmatrix}(1+\tau_i)\boldsymbol{\Delta} & \tau_i\boldsymbol{\Delta} \\ \mathbb{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{x}(i+1) \\ \mathbf{x}(i)\end{bmatrix},$$

with well-defined initial conditions $\mathbf{x}(0) := \begin{bmatrix} \left\|\mathbf{L}^\star\right\|_F \\ \left\|\mathbf{M}^\star\right\|_F \end{bmatrix}$ and $\mathbf{y}(0) := \mathbf{x}(1) = (1+\tau_i)\boldsymbol{\Delta}\mathbf{x}(0)$. For $\mathbf{w}(i) := \begin{bmatrix}\mathbf{x}(i+1) \\ \mathbf{x}(i)\end{bmatrix}$, we obtain the linear system:

$$\mathbf{w}(i+1) \leq \underbrace{\begin{bmatrix}(1+\tau_i)\boldsymbol{\Delta} & \tau_i\boldsymbol{\Delta} \\ \mathbb{I} & \mathbf{0}\end{bmatrix}}_{\widehat{\boldsymbol{\Delta}}}\mathbf{w}(i).$$

Unfolding the recursion, we get the inequality (4.35):

$$\mathbf{w}(i+1) \leq \widehat{\mathbf{\Delta}}^i \mathbf{w}(0).$$

Assuming $\boldsymbol{\mathcal{A}} : \mathbb{R}^{p\times n} \to \mathbb{R}^m$ is a linear operator satisfying rank-RIP and sparse-RIP with constants $\delta_{4r}(\boldsymbol{\mathcal{A}}) \leq 0.09$ and $\delta_{4k}(\boldsymbol{\mathcal{A}}) \leq 0.095$, respectively, and satisfies jointly the low rank- and sparse-RIP with constant $\delta_{3r+3k}(\boldsymbol{\mathcal{A}}) \leq 0.095$, we observe that the eigenvalues of $\widehat{\mathbf{\Delta}}$ are distinct and real and satisfy $|\lambda_j(\widehat{\mathbf{\Delta}})| < 1,\ \forall j$. Furthermore, $|\mathbb{I} - \widehat{\mathbf{\Delta}}| \neq 0$. To complete the proof, we use the following Theorem from **[?]** — the proof is omitted:

Theorem 14 (Necessary and Sufficient Conditions for Global Stability: Distinct Real Eigenvalues)**.** *Consider the system* $\mathbf{w}(i+1) = \widehat{\mathbf{\Delta}}\mathbf{w}(i) + \mathbf{B}$ *where* $\mathbf{w}(0)$ *is given. We assume that* $|\mathbb{I} - \widehat{\mathbf{\Delta}}| \neq 0$ *and* $\widehat{\mathbf{\Delta}}$ *has distinct real eigenvalues. Then:*

- *The steady-state equilibrium* $\widetilde{\mathbf{w}} = [\mathbb{I} - \widehat{\mathbf{\Delta}}]^{-1}\mathbf{B}$ *is globally stable if and only if* $|\lambda_j(\widehat{\mathbf{\Delta}})| < 1,\ \forall j$.
- $\lim_{i\to\infty} \mathbf{w}(i) = \widetilde{\mathbf{w}}$ *if and only if* $|\lambda_j(\widehat{\mathbf{\Delta}})| < 1,\ \forall j$.

In our simple case, we consider $\mathbf{B} := 0$. Thus, the steady-state equilibrium in (4.35) satisfies $\widetilde{\mathbf{w}} = \mathbf{0}$. Then, we conclude $\lim_{i\to\infty} \mathbf{w}(i) = \mathbf{0}$ and, thus:

$$\left\|\mathbf{L}_i - \mathbf{L}^\star\right\|_F \to 0 \quad \text{and} \quad \left\|\mathbf{M}_i - \mathbf{M}^\star\right\|_F \to 0,$$

as $i \to \infty$.

5 Convex approaches in low-dimensional modeling

Introduction

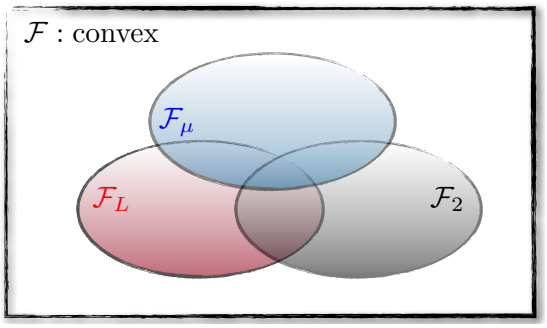
It is obvious so far that mathematical optimization is used in many applications: from portfolio optimization and kernel density estimation to quantum state tomography and video background subtraction, from image processing in biology to compressed sensing and neuronal spike detection. In all cases and given the resources available, we are interested in finding the best solution $\mathbf{x}^\star$ that best fits / interprets the problem at hand.

As already mentioned, for this purpose, one should fully understand the nature of the problem in order to formulate it with maximum fidelity. As concrete examples, in Chapter 3 we provide both convex and non-convex model descriptions of sparsity models that appear in practice such as rooted-connected sparsity model or pairwise overlapping group model. We show that, depending on the model followed at this stage, there are different tools to be used in order to simulate and finally predict the real underlying process.

Hitherto, most of our discussions so far focused on the case of *greedy, non-convex* methods where one operates directly on the *discrete model*. However, while the discrete model might be often closer to what we expect from the physical process, it is absolutely necessary to highlight the consequences of such selection. E.g., while in the case of compressed sensing and affine rank minimization problems one can use *greedy* algorithms for fast and accurate solutions (see Chapters 2 and 4), there are problem cases where *non-convexity* cannot guarantee convergence to a "good" solution (i.e., in the best case, we cannot hope for more than a locally optimally point). Moreover, deviations from the strict discrete model in the non-convex case usually lead to severe degradations in signal reconstruction.

From a computational point of view, it is almost common sense to assume that non-convex problem formulations are more difficult to solve in their entirety. Most of the non-convex models presented in this thesis are discrete and their usage in practice might lead to some NP-hard problem formulations; e.g., consider the ℓ_0 "norm" minimization formulation in the case of compressed sensing.

Based on the above, researchers very often in practice lean to choose "less-good" models—i.e., models that do not fully comply with the problem and might lead to model discrepancies; see Chapter 3—than "good" models that are difficult to handle in practice. E.g., convex relaxations are usually less susceptible to model mismatches and result into better recovery performance in compressive image recovery; e.g., see Figures 3.7-3.8. This fact is also mirrored by the *computational practice* over the past decades: both in



Class	Property $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f),\ \mathbf{v} \in \mathbb{R}^n, 0 \leq \mu \leq L < +\infty$
$\mathcal{F}_L$	$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L \|\mathbf{x} - \mathbf{y}\|$
$\mathcal{F}_\mu$	$\frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2 + f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y})$
$\mathcal{F}_2$	$\|\varphi'''(t)\| \leq 2\varphi''(t)^{3/2}$: $\varphi(t) = f(\mathbf{x} + t\mathbf{v}),\ t \in \mathbb{R}$

Figure 5.1: Common structural assumptions on the smooth function f.

academia and industry, the most widespread optimization models followed in practice are that of *convex non-linear optimization*.[1]

However, *convexity by itself does not imply tractability*. Our purpose in this chapter is to highlight that, in order to apply convex optimization techniques with provable guarantees, one needs to be aware of the underlying nature of the problem at hand and exploit the structure, as well as the theory that applies. To show this, we focus on problems of the following composite form:[2]

$$F^* := \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}) \mid F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x})\}, \tag{5.1}$$

where f and g are both closed and convex, and n is the problem dimension. In the canonical setting of the composite minimization problem (5.1), the functions f and g are assumed to be smooth and non-smooth, respectively [Nes13]. The literature on the formulation, analysis, and applications of *composite convex minimization* is ever expanding due to its broad applications in machine learning, signal processing, and statistics. For instance, such composite objectives naturally arise in maximum a posteriori model estimation, where we regularize a model likelihood function as measured by a data-driven smooth term f with a non-smooth model prior g, which carries some notion of model complexity (e.g., sparsity, low-rankness, etc.).

A vignette of algorithmic approaches

In theory, many convex problem instances of the form (5.1) have a well-understood structure, and hence high accuracy solutions can be efficiently obtained with polynomial time methods, such as interior point methods (IPM) after transforming them into conic quadratic programming or semidefinite programming formulations [BTN01, GBY06, NN94]. *In practice*, however, the curse-of-dimensionality renders these methods impractical for large-scale problems. Moreover, the presence of a non-smooth term g prevents direct applications of scalable smooth optimization techniques, such as sequential linear or quadratic programming.

Fortunately, we can provably trade-off accuracy with computation for large-scale applications by further exploiting the individual structures of f and g. Existing methods invariably rely on two structural assumptions that particularly stand out among many others. First, we often assume that f has Lipschitz

[1] This includes the ancestor of convex optimization, linear optimization.

[2] According to conventional wisdom, regularized convex optimization formulations is preferred over constrained ones since *unconstrained* optimization is generally easier to solve than constrained one. Thus, in this chapter we focus on the former formulation.

continuous gradient (i.e., $f \in \mathcal{F}_L$: cf., Fig. 5.1). Second, we assume that the proximity operator of g is somewhat easy to compute; see also Sections 6.3-6.4. On the basis of these structures, we can design algorithms featuring a full spectrum of (nearly) dimension-independent, global convergence rates with well-understood analytical complexity (see Table 5.1).

Table 5.1: Taxonomy of convex optimization methods when $f \in \mathcal{F}_L$ to reach $F(\mathbf{x}^k) - F^* \leq \epsilon$.

Order	Method example	Main oracle	Analytical complexity
1-st	[Accelerated] gradient	∇f	$[\mathcal{O}(\epsilon^{-1/2})]$ $\mathcal{O}(\epsilon^{-1})$
1^+-th	Proximal quasi-Newton	$\mathbf{H}_k, \nabla f$	$\mathcal{O}(\log \epsilon^{-1})$ or faster
2-nd	Proximal Newton	$\nabla^2 f, \nabla f$	$\mathcal{O}(\log \log \epsilon^{-1})$

[BF12, LSS12, Nes04, NW06].

Unfortunately, existing large-scale algorithms have become inseparable with the Lipschitz gradient assumption on f and are still being applied to solve (5.1) in applications where this assumption does not hold. For instance, when proximity operation is not easy to compute, it is still possible to establish convergence—albeit slower—with smoothing, splitting or primal-dual decomposition techniques [CP11, EB92, Nes05a, Nes05b, TDSD13]. However, when $f \notin \mathcal{F}_L$, the composite problems of the form (5.1) are not within the full theoretical grasp. In particular, there is no known global convergence rate. One kludge to handle $f \notin \mathcal{F}_L$ is to use sequential quadratic approximation of f to reduce the subproblems to the Lipschitz gradient case. For local convergence of these methods, we need *strong regularity* assumptions on f (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$) near the optimal solution. Attempts at global convergence require a *globalization strategy* such as line search procedures, as we describe next. However, neither the strong regularity nor the line search assumptions can be certified *a priori*.

Self-concordance in composite convex minimization

To this end, we address the following question in this paper: "Is it possible to efficiently solve large-scale instances of (5.1) for non-global Lipschitz continuous gradient f with rigorous global convergence guarantees?" The answer is positive (at least for a broad class of functions): We can still cover a full spectrum of global convergence rates with well-characterizable computation and accuracy trade-offs (akin to Table 5.1 for $f \in \mathcal{F}_L$) for *self-concordant* f (in particular, self-concordant barriers) [NT09b, NN94]:

Definition 15 (Self-concordant (barrier) functions)**.** *A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant (i.e., $f \in \mathcal{F}_M$) with parameter M, if $|\varphi'''(t)| \leq M\varphi''(t)^{3/2}$, where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$. When $M = 2$, the function f is said to be a standard self-concordant, i.e., $f \in \mathcal{F}_2$.[a] A standard self-concordant function $f \in \mathcal{F}_2$ is a ν-self-concordant barrier of a given convex set Ω with parameter $\nu > 0$, i.e., $f \in \mathcal{F}_\nu$, when φ also satisfies $|\varphi'(t)| \leq \sqrt{\nu}\varphi''(t)^{1/2}$ and $f(\mathbf{x}) \to +\infty$ as $\mathbf{x} \to \partial\Omega$, the boundary of Ω.*

[a] We use this constant for convenience in the derivations since if $f \in \mathcal{F}_M$, then $(M^2/4)f \in \mathcal{F}_2$.

While there are other definitions of self-concordant functions and self-concordant barriers [BV04, NT09b, NN94, Nes04], we use Definition 15 in the sequel, unless otherwise stated.

Why is the assumption $f \in \mathcal{F}_2$ interesting for composite minimization?

The assumption $f \in \mathcal{F}_2$ in (5.1) is quite natural for two reasons. First, several important applications directly feature a self-concordant f, which does not have global Lipschitz continuous gradient. Second, self-concordant composite problems can enable approximate solutions of general constrained convex problems where the constraint set is endowed with a ν-self-concordant barrier function.[3] We highlight three examples below, based on compositions with the $\log$-functions.

Log-determinant: The matrix variable function $f(\mathbf{\Theta}) := -\log\det\mathbf{\Theta}$ is self-concordant with $\mathrm{dom}(f) := \{\mathbf{\Theta} \in \mathbb{S}^p \mid \mathbf{\Theta} \succ 0\}$. As a stylized application, consider learning a Gaussian Markov random field (GMRF) of p nodes/variables from a dataset $\mathcal{D} := \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_m\}$, where $\boldsymbol{\phi}_j \in \mathcal{D}$ is a p-dimensional random vector with Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. Let $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ be the inverse covariance (or the precision) matrix for the model. To satisfy the conditional dependencies with respect to the GMRF, $\mathbf{\Theta}$ must have zero in $(\mathbf{\Theta})_{ij}$ corresponding to the absence of an edge between node i and node j; cf., [Dem72].

We can learn GMRF's with theoretical guarantees from as few as $\mathcal{O}(d^2 \log p)$ data samples, where d is the graph node degree, via ℓ_1-norm regularization formulation (see [RWRY11]):

$$\mathbf{\Theta}^* := \arg\min_{\mathbf{\Theta} \succ 0} \Big\{ \underbrace{-\log\det(\mathbf{\Theta}) + \mathrm{tr}(\widehat{\mathbf{\Sigma}}\mathbf{\Theta})}_{=:f(\mathbf{\Theta})} + \underbrace{\rho\|\mathbf{vec}(\mathbf{\Theta})\|_1}_{=:g(\mathbf{\Theta})} \Big\}, \tag{5.2}$$

where $\rho > 0$ parameter balances a Gaussian model likelihood and the sparsity of the solution, $\widehat{\mathbf{\Sigma}}$ is the empirical covariance estimate, and $\mathbf{vec}$ is the vectorization operator. The formulation also applies for learning models beyond GMRF's, such as the Ising model, since $f(\mathbf{\Theta})$ acts also as a Bregman distance [BEGd08].

Numerical solution methods for solving problem (5.2) have been extensively studied, e.g. in [BEGd08, HSDR11, LSS12, Lu10, OONR12, RRG+12, SR09, SMG10, Yua12]. However, none so far exploits self-concordance and feature global convergence guarantees.

Log-barrier for linear inequalities: The function $f(\mathbf{x}) := -\log(\mathbf{a}^T\mathbf{x} - b)$ is a self-concordant barrier with $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T\mathbf{x} > b\}$. As a stylized application, consider the low-light imaging problem in signal processing [HMW12, FBD10], where the imaging data is collected by counting photons hitting a detector over the time. In this setting, we wish to accurately reconstruct an image in low-light, which leads to noisy measurements due to low photon count levels. We can express our observation model using the Poisson distribution as:

$$\mathbb{P}(\mathbf{y}|\mathcal{A}(\mathbf{x})) = \prod_{i=1}^{m} \frac{(\mathbf{a}_i^T\mathbf{x})^{y_i}}{y_i!} e^{-\mathbf{a}_i^T\mathbf{x}},$$

where $\mathbf{x}$ is the true image, $\mathcal{A}$ is a linear operator that projects the scene onto the set of observations, $\mathbf{a}_i$ is the i-th row of $\mathcal{A}$, and $\mathbf{y} \in \mathbb{Z}_+^m$ is a vector of observed photon counts.

[3] Let us consider a constrained convex minimization $\mathbf{x}_C^* := \arg\min_{\mathbf{x}\in C} g(\mathbf{x})$, where the feasible convex set C is endowed with a ν-self-concordant barrier $\Psi_C(\mathbf{x})$. If we let $f(\mathbf{x}) := \frac{\epsilon}{\nu}\Psi_C(\mathbf{x})$, then the solution $\mathbf{x}^*$ of the composite minimization problem (5.1) well-approximates $\mathbf{x}_C^*$, as $g(\mathbf{x}^*) \leq g(\mathbf{x}_C^*) + (\nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*))^T(\mathbf{x}^* - \mathbf{x}_C^*) + \epsilon$. The middle term can be controlled by accuracy at which we solve the composite minimization problem [Nes11, Nes13].

Via the log-likelihood formulation, we stumble upon a composite minimization problem:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ \underbrace{\sum_{i=1}^m \mathbf{a}_i^T\mathbf{x} - \sum_{i=1}^m y_i \log(\mathbf{a}_i^T\mathbf{x})}_{=:f(\mathbf{x})} + g(\mathbf{x}) \Big\}, \tag{5.3}$$

where $f(\mathbf{x})$ is self-concordant (but not standard). In the above formulation, the typical image priors $g(\mathbf{x})$ include the ℓ_1-norm for sparsity in a known basis, total variation semi-norm of the image, and the positivity of the image pixels. While the formulation (5.3) seems specific to imaging, it is also common in sparse regression with unknown noise variance [SBdG12], heteroschedastic LASSO [DHMS13], and barrier approximations of, e.g., the Dantzig selector [CT07] as well.

The current state of the art solver is called SPIRAL-TAP [HMW12], which biases the logarithmic term (i.e., $\log(\mathbf{a}_i^T\mathbf{x} + \varepsilon) \to \log(\mathbf{a}_i^T\mathbf{x})$, where $\varepsilon \ll 1$) and then applies non-monotone composite gradient descent algorithms for $\mathcal{F}_L$ with a Barzilai-Borwein step-size as well as other line-search strategies.

Logarithm of concave quadratic functions: The function $f(\mathbf{x}) := -\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2\right)$ is self-concordant with $\mathrm{dom}(f) := \left\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 < \sigma^2\right\}$. As a stylized application, we consider the basis pursuit denoising (BPDN) formulation [VDBF08] as:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ g(\mathbf{x}) \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \le \sigma^2 \right\}. \tag{5.4}$$

The BPDN criteria is commonly used in magnetic resonance imaging (MRI) where $\mathbf{A}$ is a subsampled Fourier operator, $\mathbf{y}$ is the MRI scan data, and σ^2 is a known machine noise level (i.e., obtained during a pre-scan). In (5.4), g is an image prior, e.g., similar to the Poisson imaging problem. Approximate solutions to (5.4) can be obtained via a barrier formulation:

$$\mathbf{x}_t^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ \underbrace{-t\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2\right)}_{=:f(\mathbf{x})} + g(\mathbf{x}) \Big\}, \tag{5.5}$$

where $t > 0$ is a penalty parameter which controls the quality of the approximation. The BPDN formulation is quite generic and has several other applications in statistical regression, geophysics, and signal processing.

Several different approaches solve the BPDN problem (5.4), some of which require projections onto the constraint set, including Douglas-Rachford splitting [ABDF11], proximal methods [ABDF10, CW05a], and the SPGL$_1$ method [VDBF08].

Chapter roadmap

Interior point methods are always an option while solving the self-concordant composite problems (5.1) numerically by means of disciplined convex programming [GBY06, LÖ4]. More concretely, in the IPM setting, we set up an equivalent problem to (5.1) that typically avoids the non-smooth term $g(x)$ in the objective by lifting the problem dimensions with slack variables and introducing additional constraints. The new constraints may then be embedded into the objective through a barrier function. We then solve

a sequence of smooth problems (e.g., with Newton methods) and "path-follow"[4] to obtain an accurate solution [NT09b, Nes04]. In this loop, many of the underlying structures within the original problem, such as sparsity, can be lost due to pre-conditioning or Newton direction scaling (e.g., Nesterov-Todd scaling, [NT97]). The efficiency and the memory bottlenecks of the overall scheme then heavily depends on the workhorse algorithm that solves the smooth problems.

In this chapter, we introduce an algorithmic framework that directly handles the composite minimization problem (5.1) without increasing the original problem dimensions. Instead of solving a sequence of smooth problems, we solve a sequence of non-smooth proximal problems with a variable metric (i.e., our workhorse). Fortunately, these proximal problems feature the composite form (5.1) with a Lipschitz gradient (and oft-times strongly convex) smooth term. Hence, we leverage the tremendous amount of research on large-scale algorithms (cf., Table 5.1) done over the last decades. Surprisingly, we can even retain the original problem structures that lead to computational ease in many cases (e.g., see Section 5.4.2).

In particular:

1. In Section 5.3. we propose a new *variable metric* framework for minimizing the sum $f + g$ of a self-concordant function f and a convex, possibly nonsmooth function g. Our approach relies on the solution of a convex subproblem obtained by linearizing and regularizing the first term f. To achieve monotonic descent, we develop a new set of *analytic* step-size selection and correction procedures based on the structure of the problem. We establish both the global and the local convergence of different variable metric strategies.

   As an extension, we pay particular attention to diagonal variable metrics as many of the proximal subproblems can be solved exactly (i.e., in closed form). We derive conditions on when these variants achieve locally linear convergence.

2. We apply our algorithms to the aforementioned large-scale real-world and synthetic problems (Section 5.4) to highlight the strengths and the weaknesses of our scheme. For instance, in the graph learning problem (5.2), our framework can avoid matrix inversions as well as Cholesky decompositions in learning graphs. In Poisson intensity reconstruction (5.3), up to around $80\times$ acceleration is possible over the state-of-the-art solver (Section 5.4).

This chapter is based on the joint work with Volkan Cevher, Quoc Tran-Dinh and Rabeeh Mahabadi Karimi [TDKC13c, TDKC13b, KC13, TDKC13a, KMTDC14].

5.1 Preliminaries

Notation: We reserve lower-case and bold lower-case letters for scalar and vector representation, respectively. Upper-case bold letters denote matrices. We denote $\mathbb{S}^p_{++}$ for the set of symmetric positive definite matrices of size $p \times p$. For a proper, lower semicontinuous convex function f from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$, we denote its domain by $\mathrm{dom}(f)$, i.e., $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}$ (see, e.g., [Roc70]).

Weighted norm and local norm: Given a matrix $\mathbf{H} \in \mathbb{S}^n_{++}$, we define the weighted norm $\|\mathbf{x}\|_{\mathbf{H}} := \sqrt{\mathbf{x}^T\mathbf{H}\mathbf{x}}$, $\forall \mathbf{x} \in \mathbb{R}^n$; its dual norm is defined as $\|\mathbf{x}\|^*_{\mathbf{H}} := \max_{\|\mathbf{y}\|_{\mathbf{H}} \leq 1} \mathbf{y}^T\mathbf{x} = \sqrt{\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}}$. Let $f \in \mathcal{F}_2$ and

[4] It is also referred to as a homotopy method.

$\mathbf{x} \in \mathrm{dom}(f)$ so that $\nabla^2 f(\mathbf{x})$ is positive definite. For a given vector $\mathbf{v} \in \mathbb{R}^n$, the local norm around $\mathbf{x} \in \mathrm{dom}(f)$ with respect to f is defined as $\|\mathbf{v}\|_{\mathbf{x}} := \left(\mathbf{v}^T \nabla^2 f(\mathbf{x}) \mathbf{v}\right)^{1/2}$, while the corresponding dual norm is given by $\|\mathbf{v}\|_{\mathbf{x}}^* = \left(\mathbf{v}^T \nabla^2 f(\mathbf{x})^{-1} \mathbf{v}\right)^{1/2}$.

Subdifferential and subgradient: Given a proper, lower semicontinuous convex function, we define the subdifferential of g at $\mathbf{x} \in \mathrm{dom}(g)$ as

$$\partial g(\mathbf{x}) := \left\{ \mathbf{v} \in \mathbb{R}^n \mid g(\mathbf{y}) - g(\mathbf{x}) \geq \mathbf{v}^T (\mathbf{y} - \mathbf{x}),\ \forall \mathbf{y} \in \mathrm{dom}(g) \right\}.$$

If $\partial g(\mathbf{x}) \neq \emptyset$ then each element in $\partial g(\mathbf{x})$ is called a subgradient of g at $\mathbf{x}$. In particular, if g is differentiable, we use $\nabla g(\mathbf{x})$ to denote its derivative at $\mathbf{x} \in \mathrm{dom}(g)$, and $\partial g(\mathbf{x}) \equiv \{\nabla f(\mathbf{x})\}$.

Proximity operator: A basic tool to handle the nonsmoothness of a convex function g is its proximity operator, whose definition is given in Chapter 6. For notational convenience in our derivations, we alter this definition in the sequel as follows: Let g be a proper lower semicontinuous and convex in $\mathbb{R}^n$ and $\mathbf{H} \in \mathbb{S}_{++}^n$. We define

$$P_{\mathbf{H}}^g(\mathbf{u}) := (\mathbf{H} + \partial g)^{-1}(\mathbf{u}) = \arg\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ g(\mathbf{x}) + \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} - \mathbf{u}^T \mathbf{x} \right\}, \quad \forall \mathbf{u} \in \mathbb{R}^n, \tag{5.6}$$

as the proximity operator for the nonsmooth g, which has the following properties.

Lemma 33. *The operator $P_{\mathbf{H}}^g$ in (5.6) is single-valued and satisfies the following property:*

$$(P_{\mathbf{H}}^g(\mathbf{u}) - P_{\mathbf{H}}^g(\mathbf{v}))^T (\mathbf{u} - \mathbf{v}) \geq \|P_{\mathbf{H}}^g(\mathbf{u}) - P_{\mathbf{H}}^g(\mathbf{v})\|_{\mathbf{H}}^2, \tag{5.7}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. Consequently, $P_{\mathbf{H}}^g$ is a non-expansive mapping, i.e.,

$$\|P_{\mathbf{H}}^g(\mathbf{u}) - P_{\mathbf{H}}^g(\mathbf{v})\|_{\mathbf{H}} \leq \|\mathbf{u} - \mathbf{v}\|_{\mathbf{H}}^*. \tag{5.8}$$

Proof. The single-valuedness of $P_{\mathbf{H}}^g$ is obvious due to the strong convexity of the objective function in (5.6). Let $\boldsymbol{\xi}_{\mathbf{u}} := P_{\mathbf{H}}^g(\mathbf{u})$ and $\boldsymbol{\xi}_{\mathbf{v}} := P_{\mathbf{H}}^g(\mathbf{v})$. By the definition of $P_{\mathbf{H}}^g$, we have $\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{u}})$ and $\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{v}})$. Since g is convex, we have $(\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} - (\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{v}}))^T (\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) \geq 0$. This inequality leads to $(\mathbf{u}-\mathbf{v})^T(\boldsymbol{\xi}_{\mathbf{u}}-\boldsymbol{\xi}_{\mathbf{v}}) \geq (\boldsymbol{\xi}_{\mathbf{u}}-\boldsymbol{\xi}_{\mathbf{v}})^T \mathbf{H} (\boldsymbol{\xi}_{\mathbf{u}}-\boldsymbol{\xi}_{\mathbf{v}}) = \|\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}\|_{\mathbf{H}}^2$ which is indeed (5.7). Via the generalized Cauchy-Schwarz inequality, (5.7) leads to (5.8). □

Key self-concordant bounds: Based on [Nes04, Theorems 4.1.7 and 4.1.8], for a given standard self-concordant function f, we recall the following inequalities

$$\omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + f(\mathbf{x}) \leq f(\mathbf{y}), \tag{5.9}$$

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) + \omega_*(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}), \tag{5.10}$$

where $\omega : \mathbb{R} \to \mathbb{R}_+$ is defined as $\omega(t) := t - \ln(1 + t)$ and $\omega_* : [0, 1] \to \mathbb{R}_+$ is defined as $\omega_*(t) := -t - \ln(1 - t)$. These functions are both nonnegative, strictly convex and increasing. Hence, (5.9) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$, and (5.10) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < 1$. In contrast to the

"global" inequalities for the function classes $\mathcal{F}_L$ and $\mathcal{F}_\mu$ (cf., Fig. 5.1), the self-concordant inequalities are based on "local" quantities. Moreover, these bounds are no longer quadratic which prevents naive applications of the methods from $\mathcal{F}_{\mu,L}$.

5.2 Related work

One of the main approaches in this setting is based on operator splitting. By presenting the optimality condition of problem (5.1) as an inclusion of two monotone operators, one can apply splitting techniques, such as forward-backward or Douglas-Rachford methods, to solve the resulting monotone inclusion [BAC11, FP03, GO09]. In our context, several variants of this approach have been studied. For example, projected gradient or proximal-gradient methods and fast proximal-gradient methods have been considered, see, e.g., [BT09a, MF81, Nes13]. In all these methods, the main assumption required to prove the convergence is the global Lipschitz continuity of the gradient of the smooth function f. Unfortunately, when $f \notin \mathcal{F}_L$, these theoretical results on the global convergence and the global convergence rates are no longer applicable.

Other mainstream approaches for (5.1) include augmented Lagrangian and alternating techniques: cf., [BPC$^+$11, GM12]. These methods have empirically proven to be quite powerful in specific applications. The main disadvantage of these methods is the manual tuning of the penalty parameter in the augmented Lagrangian function, which is not yet well-understood for general problems. Consequently, the analysis of global convergence as well as the convergence rate is an issue since the performance of the algorithms strongly depends on the choice of this penalty parameter in practice. Moreover, as indicated in a recent work [GOS12], alternating direction methods of multipliers as well as alternating linearization methods can be viewed as splitting methods in the convex optimization context. Hence, it is unclear if this line of work is likely to lead to any rigorous guarantees.

An emerging direction for solving composite minimization problems (5.1) is based on the proximal-Newton method. The origins of this method can be traced back to the work of [Bon94], which relies on the concept of *strong regularity* introduced by [Rob80] for generalized equations. In the convex case, this method has been studied by several authors such as [BF12, LSS12, SRB11]. So far, methods along this line are applied to solve a generic problem of the form (5.1). The convergence analysis of these methods is encouraged by standard Newton methods and requires the strong regularity of the Hessian of f near the optimal solution (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$). Moreover, the global convergence can only be proved by applying a certain globalization strategy such as line-search or trust-region. Unfortunately, none of these assumptions can be verified before the algorithm execution for the intended applications.

5.3 The Self-Concordant Optimization (SCOPT) framework

In this section, we propose a *variable metric* optimization framework that rigorously trades off computation and accuracy of solutions without transforming (5.1) into a higher dimension smooth convex optimization problem. We assume theoretically that the proximal subproblems can be solved exactly. Moreover, our theory can be extended to include the *inexact case*, where we solve parts of the optimization up to a sufficiently high accuracy (typically, it is at least higher than (e.g., 0.1ε) the desired accuracy ε of (5.1) at the few last iterations), see, e.g., [TDNSD13]. In our theoretical characterizations, we only rely on the following assumption:

Assumption A.1. *The function f is convex and standard self-concordant (see Definition 15). The function g from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ is proper, lower semicontinuous, convex and possibly nonsmooth with a tractable proximity operator.*

Unique solvability of (5.1) and its optimality condition: First, we show that problem (5.1) is uniquely solvable. The proof of this lemma can be done similarly as [Nes04, Theorem 4.1.11] and is provided in the appendix.

Lemma 34. *Suppose that the functions f and g of problem (5.1) satisfy Assumption 1. If $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|^*_{\mathbf{x}^*} < 1$, for some $\mathbf{x} \in \mathrm{dom}(F)$ and $\mathbf{v} \in \partial g(\mathbf{x})$, the solution $\mathbf{x}^*$ of (5.1) exists and is unique.*

Thus, we show that a local condition $\lambda(\mathbf{x}) < 1$ for some $\mathbf{x}$ provides us with some global information on f.

Since this problem is convex, the following optimality condition is necessary and sufficient:

$$\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*). \tag{5.11}$$

The solution $\mathbf{x}^*$ is called *strongly regular* if $\nabla^2 f(\mathbf{x}^*) \succ 0$. In this case, $\infty > \sigma^*_{\max} \geq \sigma^*_{\min} > 0$, where $\sigma^*_{\min}$ and $\sigma^*_{\max}$ are the smallest and the largest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$.

Fixed-point characterization: Let $\mathbf{H} \in \mathbb{S}^n_{++}$. We define $S_{\mathbf{H}}(\mathbf{x}) := \mathbf{H}\mathbf{x} - \nabla f(\mathbf{x})$. Then, from (5.11), we have

$$S_{\mathbf{H}}(\mathbf{x}^*) \equiv \mathbf{H}\mathbf{x}^* - \nabla f(\mathbf{x}^*) \in \mathbf{H}\mathbf{x}^* + \partial g(\mathbf{x}^*).$$

By using the definition of $P^g_{\mathbf{H}}(\cdot)$ in (5.6), one can easily derive the fixed-point expression

$$\mathbf{x}^* = P^g_{\mathbf{H}}\left(S_{\mathbf{H}}(\mathbf{x}^*)\right), \tag{5.12}$$

that is, $\mathbf{x}^*$ is the fixed-point of the mapping $R^g_{\mathbf{H}}(\cdot)$, where $R^g_{\mathbf{H}}(\cdot) := P^g_{\mathbf{H}}(S_{\mathbf{H}}(\cdot))$. The formula in (5.12) suggests that we can generate an iterative sequence based on the fixed-point principle, i.e., $\mathbf{x}^{k+1} := R^g_{\mathbf{H}}(\mathbf{x}^k)$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ for $k \geq 0$. Theoretically, under certain assumptions, one can ensure that the mapping $R^g_{\mathbf{H}}$ is contractive and the sequence generated by this scheme is convergent.

We note that if $g \equiv 0$ and $\mathbf{H} \in \mathbb{S}^n_{++}$, then $P^g_{\mathbf{H}}$ defined by (5.6) reduces to $P^g_{\mathbf{H}}(\cdot) = \mathbf{H}^{-1}(\cdot)$. Consequently, the fixed-point formula (5.12) becomes $\mathbf{x}^* = \mathbf{x}^* - \mathbf{H}^{-1}\nabla f(\mathbf{x}^*)$, which is equivalent to $\nabla f(\mathbf{x}^*) = 0$.

Our variable metric framework: Given a point $\mathbf{x}^k \in \mathrm{dom}(F)$ and a symmetric positive definite matrix $\mathbf{H}_k$, we consider the function

$$Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) := f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T\mathbf{H}_k(\mathbf{x} - \mathbf{x}^k), \tag{5.13}$$

for $\mathbf{x} \in \mathrm{dom}(F)$. The function $Q(\cdot; \mathbf{x}^k, \mathbf{H}_k)$ is—seemingly—a quadratic approximation of f around $\mathbf{x}^k$. Now, we study the following scheme to generate a sequence $\{\mathbf{x}^k\}_{k\geq 0}$:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \tag{5.14}$$

where $\alpha_k \in (0, 1]$ is a step size and $\mathbf{d}^k$ is a search direction.

Let $\mathbf{s}^k$ be a solution of the following problem:

$$\mathbf{s}^k \in \mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) := \underset{\mathbf{x}\in\mathrm{dom}(F)}{\arg\min} \left\{Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) + g(\mathbf{x})\right\} = P^g_{\mathbf{H}_k}\left(\mathbf{H}_k\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right). \tag{5.15}$$

Since we do not assume that $\mathbf{H}_k$ to be positive definite, the solution $\mathbf{s}^k$ may not exist. We require the following assumption:

Assumption A.2. *The subproblem* (5.15) *has at least one solution* $\mathbf{s}^k$*, i.e.,* $\mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) \neq \emptyset$*.*

In particular, if $\mathbf{H}_k \in \mathbb{S}^n_{++}$, then the solution $\mathbf{s}^k$ of (5.15) exists and is unique, i.e., $\mathcal{S}(\mathbf{x}^k, \mathbf{H}_k) = \left\{\mathbf{s}^k\right\} \neq \emptyset$. Up to now, we have not required the uniqueness of $\mathbf{s}^k$. This assumption will be specified later in the next sections. Throughout this paper, we assume that both Assumptions $\mathbf{A}.1$ and $\mathbf{A}.2$ are satisfied without referring to them specifically.

Now, given $\mathbf{s}^k$, the direction $\mathbf{d}^k$ is computed as

$$\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k. \tag{5.16}$$

If we define $\mathbf{G}_k := \mathbf{H}_k\mathbf{d}^k$, then $\mathbf{G}_k$ is called the *gradient mapping* of (5.1) [Nes04], which behaves similarly as gradient vectors in non-composite minimization. Since problem (5.15) is solvable due to Assumption $\mathbf{A}.2$, we can write its optimality condition as

$$\mathbf{0} \in \nabla f(\mathbf{x}^k) + \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) + \partial g(\mathbf{s}^k). \tag{5.17}$$

It is easy to see that if $\mathbf{d}^k = 0$, i.e., $\mathbf{s}^k \equiv \mathbf{x}^k$, then (5.17) reduces to $0 \in \nabla f(\mathbf{x}^k) + \partial g(\mathbf{x}^k)$, which is exactly (5.11). Hence, $\mathbf{x}^k$ is a solution of (5.1).

In the variable metric framework, depending on the choice of $\mathbf{H}_k$, the iteration scheme (5.14) leads to different methods for solving (5.1). For instance,

1. If $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method (5.14) is a *proximal-Newton* method.
2. If $\mathbf{H}_k$ is a symmetric positive definite matrix approximation of $\nabla^2 f(\mathbf{x}^k)$, then the method (5.14) is a *proximal-quasi Newton* method.
3. If $\mathbf{H}_k := L_k\mathbb{I}$, where L_k is, say, an approximation for the local Lipschitz constant of f and $\mathbb{I}$ is the identity matrix, then the method (5.14) is a *proximal-gradient* method.

Many of these above methods have been studied for (5.1) when $f \in \mathcal{F}_L$: cf., [BF12, BT09a, CPR13, LSS12]. Note however that, since the self-concordant part f of F is not (necessarily) globally Lipschitz continuously differentiable, these approaches are generally not applicable in theory.

Given the search direction $\mathbf{d}^k$ defined by (5.16), we define the following proximal-Newton decrement[5] λ_k and the weighted norm β_k:

$$\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k} = \left((\mathbf{d}^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k\right)^{1/2} \text{ and } \beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k}. \tag{5.18}$$

[5]This notion is borrowed from standard the Newton decrement defined in [Nes04, Chapter 4].

In the sequel, we study only the proximal-Newton method; a complete description of all variable metric strategies is given in [TDKC13a].

Remark 9. *If $g \equiv 0$ and $\nabla^2 f(\mathbf{x}^k) \in \mathbb{S}^n_{++}$, then $\mathbf{d}^k = -\nabla^2 f(\mathbf{x}^k)^{-1}\nabla f(\mathbf{x}^k)$ is the standard Newton direction. In this case, λ_k defined by* (5.18) *reduces to $\lambda_k \equiv \|\nabla f(\mathbf{x}^k)\|^*_{\mathbf{x}^k}$, the Newton decrement defined in [Nes04, Chapter 4]. Moreover, we have $\lambda_k \equiv \lambda(\mathbf{x}^k)$, as defined in Lemma 34.*

5.3.1 A proximal-Newton method

If we choose $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method described in (5.14) is called the *proximal Newton* algorithm. For notational ease, we redefine $\mathbf{s}^k_n := \mathbf{s}^k$ and $\mathbf{d}^k_n := \mathbf{d}^k$, where the subscript n is used to distinguish proximal Newton related quantities from the other variable metric strategies. Moreover, we use the shorthand notation $P^g_{\bar{\mathbf{x}}} := P^g_{\nabla^2 f(\bar{\mathbf{x}})}$, whenever $\bar{\mathbf{x}} \in \mathrm{dom}(f)$. Using (5.15) and (5.16), $\mathbf{s}^k_n$ and $\mathbf{d}^k_n$ are given by

$$\mathbf{s}^k_n := P^g_{\mathbf{x}^k}\left(\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right), \quad \mathbf{d}^k_n := \mathbf{s}^k_n - \mathbf{x}^k. \tag{5.19}$$

Then, the proximal-Newton method generates a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ according to

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k_n, \tag{5.20}$$

where $\alpha_k \in (0, 1]$ is a step size. If $\alpha_k < 1$, then the iteration (5.20) is called the *damped proximal-Newton* iteration. If $\alpha_k = 1$, then it is called the *full-step proximal-Newton* iteration.

Global convergence: We first show that with an appropriate choice of the step-size $\alpha_k \in (0, 1]$, the iterative sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by the damped-step proximal Newton scheme (5.20) is a decreasing sequence; i.e., $F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\sigma)$ whenever $\lambda_k \geq \sigma$, where $\sigma > 0$ is fixed. The following theorem provides an explicit formula for the step size α_k whose proof can be found in the appendix.

Theorem 15. *If $\alpha_k := \frac{1}{1+\lambda_k} \in (0, 1]$, then the scheme in* (5.20) *generates $\mathbf{x}^{k+1}$ satisfies:*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k). \tag{5.21}$$

Moreover, the step α_k is optimal. The number of iterations to reach the point $\mathbf{x}^k$ such that $\lambda_k < \sigma$ for some $\sigma \in (0, 1)$ is $k_{\max} := \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$.

Local quadratic convergence rate: We now establish the local quadratic convergence of the scheme (5.20). A complete proof of this theorem can be found in the appendix.

Theorem 16. *Assume that* $\mathbf{x}^*$ *is the unique solution of* (5.1) *and is strongly regular. Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be a sequence generated by the proximal Newton scheme* (5.20) *with* $\alpha_k \in (0, 1]$. *Then:*

a) *If* $\alpha_k\lambda_k < 1 - \frac{1}{\sqrt{2}}$, *then it holds that*

$$\lambda_{k+1} \leq \left(\frac{1 - \alpha_k + (2\alpha_k^2 - \alpha_k)\lambda_k}{1 - 4\alpha_k\lambda_k + 2\alpha_k^2\lambda_k^2}\right)\lambda_k. \tag{5.22}$$

b) *If the sequence* $\{\mathbf{x}^k\}_{k\geq 0}$ *is generated by the damped proximal-Newton scheme* (5.20), *starting from* $\mathbf{x}^0$ *such that* $\lambda_0 \leq \bar{\sigma} := \sqrt{5} - 2 \approx 0.236068$ *and* $\alpha_k := (1 + \lambda_k)^{-1}$, *then* $\{\lambda_k\}_k$ *locally converges to* 0^+ *at a quadratic rate.*

c) *Alternatively, if the sequence* $\{\mathbf{x}^k\}_{k\geq 0}$ *is generated by the full-step proximal-Newton scheme* (5.20) *starting from* $\mathbf{x}^0$ *such that* $\lambda_0 \leq \bar{\sigma} := 0.25(5 - \sqrt{17}) \approx 0.219224$ *and* $\alpha_k = 1$, *then* $\{\lambda_k\}_k$ *locally converges to* 0^+ *at a quadratic rate.*

Consequently, the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ *also locally converges to* $\mathbf{x}^*$ *at a quadratic rate in both cases* b) *and* c), *i.e.,* $\{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}\}_{k\geq 0}$ *locally converges to* 0^+ *at a quadratic rate.*

A two-phase algorithm for solving (5.1): Now, by the virtue of the above analysis, we can propose a two-phase proximal-Newton algorithm for solving (5.1). Initially, we perform the dub-step proximal-Newton iterations until we reach the quadratic convergence region (Phase 1). Then, we perform full-step proximal-Newton iterations, until we reach the desired accuracy (Phase 2). The pseudocode of the algorithm is presented in Algorithm 16.

Algorithm 16 (*Proximal-Newton algorithm*)

Inputs: $\mathbf{x}^0 \in \mathrm{dom}(F)$, tolerance $\varepsilon > 0$.
Initialization: Select a constant $\sigma \in (0, \frac{(5-\sqrt{17})}{4}]$, e.g., $\sigma := 0.2$.

for $k = 0$ **to** $K_{\max}$ **do**
1. Compute the proximal-Newton search direction $\mathbf{d}_n^k$ as in (5.19).
2. Compute $\lambda_k := \|\mathbf{d}_n^k\|_{\mathbf{x}^k}$.
3. **if** $\lambda_k > \sigma$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}_n^k$, where $\alpha_k := (1 + \lambda_k)^{-1}$.
4. **elseif** $\lambda_k > \varepsilon$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}_n^k$.
5. **else** terminate.

end for

The radius σ of the quadratic convergence region in Algorithm 16 can be fixed at any value in $(0, \bar{\sigma}]$, e.g., at its upper bound $\bar{\sigma}$. An upper bound $K_{\max}$ of the iterations can also be specified, if necessary. The computational bottleneck in Algorithm 16 is typically incurred Step 1 in Phase 1 and Phase 2, where we need to solve the subproblem (5.15) to obtain a search direction $\mathbf{d}_n^k$. When problem (5.15) is strongly convex, i.e., $\nabla^2 f(\mathbf{x}^k) \in \mathbb{S}_{++}^n$, one can apply first order methods to efficiently solve this problem with a linear convergence rate (see, e.g., [BT09a, Nes04, Nes13]) and make use of a *warm-start* strategy by employing the information of the previous iterations.

Iteration-complexity analysis. The choice of σ in Algorithm 16 can trade-off the number of iterations between the damped-step and full-step iterations. If we fix $\sigma = 0.2$, then the complexity of the full-step Newton phase becomes $\mathcal{O}\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)$. The following theorem summarizes the complexity of the proposed algorithm.

Theorem 17. *The maximum number of iterations required in Algorithm 1 does not exceed* $K_{\max} := \left\lfloor \frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{0.017} \right\rfloor + \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 2$ *provided that* $\sigma = 0.2$ *to obtain* $\lambda_k \leq \varepsilon$. *Consequently,* $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq 2\varepsilon$, *where* $\mathbf{x}^*$ *is the unique solution of* (5.1).

Proof. Let $\sigma = 0.2$. From the estimate (5.22) of Theorem 16 and $\alpha_{k-1} = 1$ we have $\lambda_k \leq (1 - 4\lambda_{k-1} + 2\lambda_{k-1}^2)^{-1}\lambda_{k-1}^2$ for $k \geq 1$. Since $\lambda_0 \leq \sigma$, by induction, we can easily show that $\lambda_k \leq (1-4\sigma+2\sigma^2)^{-1}\lambda_{k-1}^2 \leq c\lambda_{k-1}^2$, where $c := 3.57$. This implies $\lambda_k \leq c^{2^k-1}\lambda_0^{2^k} \leq c^{2^k-1}\sigma^{2^k}$. The stopping criterion $\lambda_k \leq \varepsilon$ in Algorithm 16 is ensured if $(c\sigma)^{2^k} \leq c\varepsilon$. Since $c\sigma \approx 0.71 < 1$, the last condition leads to $k \geq (\ln 2)^{-1}\ln\left(\frac{-\ln(c\sigma)}{-\ln(c\varepsilon)}\right)$. By using $c = 3.57$, $\sigma = 0.2$ and the fact that $\ln(2)^{-1} < 1.5$, we can show that the last requirement is fulfilled if $k \geq \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 1$. Now, combining the last conclusion and Theorem 15 with noting that $\omega(\sigma) > 0.017$ we obtain $K_{\max}$ as in Theorem 17.

Finally, we prove $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq 2\varepsilon$. Indeed, we have $\mathbf{r}_k := \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}}{1-\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}} + \|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^*} = \frac{\lambda_k}{1-\mathbf{r}_k} + \mathbf{r}_{k+1}$, whenever $\mathbf{r}_k < 1$. Next, using (5.77) with $\alpha_k = 1$, we have $\mathbf{r}_{k+1} \leq \frac{(3-\mathbf{r}_k)\mathbf{r}_k^2}{1-4\mathbf{r}_k+2\mathbf{r}_k^2}$. Combining these inequalities, we obtain $\frac{(1-\mathbf{r}_k)(1-7\mathbf{r}_k+3\mathbf{r}_k^2)\mathbf{r}_k}{1-4\mathbf{r}_k+2\mathbf{r}_k^2} \leq \lambda_k \leq \varepsilon$. Since the function $s(\mathbf{r}) := \frac{(1-\mathbf{r})(1-7\mathbf{r}+3\mathbf{r}^2)\mathbf{r}}{1-4\mathbf{r}+2\mathbf{r}^2}$ attains a maximum at $\mathbf{r}^* \approx 0.08763$ and it is increasing on $[0, \mathbf{r}^*]$. Moreover, $\frac{(1-\mathbf{r}_k)(1-7\mathbf{r}_k+3\mathbf{r}_k^2)}{1-4\mathbf{r}_k+2\mathbf{r}_k^2} \geq 0.5$ for $\mathbf{r}_k \in [0, \mathbf{r}^*]$, which leas to $0.5\mathbf{r}_k \leq \frac{(1-\mathbf{r}_k)(1-7\mathbf{r}_k+3\mathbf{r}_k^2)\mathbf{r}_k}{1-4\mathbf{r}_k+2\mathbf{r}_k^2} \leq \varepsilon$. Hence, $\mathbf{r}_k \leq 2\varepsilon$ provided that $\mathbf{r}_k \leq \mathbf{r}_0 \leq \mathbf{r}^* \approx 0.08763$. □

Remark 10. *When* $g \equiv 0$, *we can modify the proof of estimate* (5.22) *to obtain a tighter bound* $\lambda_{k+1} \leq \frac{\lambda_k^2}{(1-\lambda_k)^2}$. *This estimate is exactly [Nes04,], which implies that the radius of the quadratic convergence region is* $\bar{\sigma} := (3-\sqrt{5})/2$.

A modification of the proximal-Newton method: In Algorithm 16, if we remove Step 4 and replace analytic step-size selection calculation in Step 3 with a backtracking line-search, then we reach the proximal Newton method of [LSS12]. Hence, this approach *in practice* might lead to reduced overall computation since our step-size α_k is selected optimally with respect to the worst case problem structures as opposed to the particular instance of the problem. Since the backtracking approach always starts with the full-step, we also do not need to know whether we are within the quadratic convergence region. Moreover, the cost of evaluating the objective at the full-step in certain applications may not be significantly worse than the cost of calculating α_k or may be dominated by the cost of calculating the Newton direction.

In stark contrast to backtracking, our new theory behooves us to propose a new forward line-search procedure as illustrated by Figure 5.2. The idea is quite simple: we start with the "optimal" step-size α_k and increase it towards full-step with a stopping condition based on the objective evaluations. Interestingly, when we analytically calculate the step, we also have access to the side information on whether or not we are within the quadratic convergence region, and hence, we can automatically switch to Step 4 in Algorithm 16. Alternatively, calculation of the analytic step-size can enhance backtracking

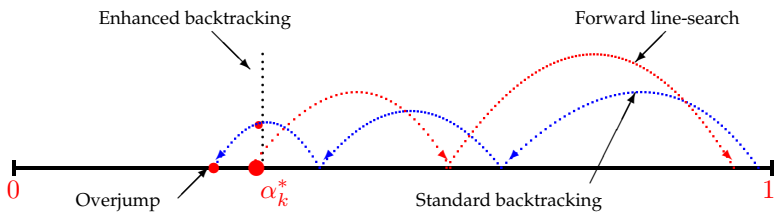


Figure 5.2: Illustration of step-size selection procedures

since the knowledge of α_k reduces the backtracking range from $(0, 1]$ to $(\alpha_k, 1]$ with the side-information as to when to automatically take the full-step without function evaluation.

5.4 Experiments

In this section, we illustrate our optimization framework via numerical experiments on the variants discussed in this chapter. All the tests are performed in MATLAB 2011b running on a PC Intel Xeon X5690 at 3.47GHz per core with 94Gb RAM.[6]

5.4.1 Empirical performance comparison

By using the graph selection problem, we first show that the modifications on the proximal-Newton method provides advantages in practical convergence as compared to state-of-the-art strategies and provides a safeguard for line-search procedures in optimization routines. We then highlight the impact of different subsolvers for (5.23) in the practical convergence of the algorithms.

Comparison of different step-size selection procedures

We apply four different step-size selection procedures in our proximal-Newton framework to solve problem (5.2). Specifically, we test the algorithm based on the following configuration:

(i) We implement Algorithm 17 in MATLAB using FISTA [BT09a] to solve the dual subproblem with the following stopping criterion: $\|\boldsymbol{\Theta}_{i+1} - \boldsymbol{\Theta}_i\|_F \leq 10^{-8} \times \max\{\|\boldsymbol{\Theta}_{i+1}\|_F, 1\}$.

(ii) We consider four different globalization procedures, whose details can be found in Section 5.3.1: $a)$ `NoLS` which uses the analytic step size $\alpha_k^* = (1 + \lambda_k)^{-1}$, $b)$ `BtkLS` which is an instance of the proximal-Newton framework of [LSS12] and uses the standard backtracking line-search based on the Amirjo's rule, $c)$ `E-BtkLS` which is based on the standard backtracking line-search enhanced by the lower bound α_k^* and, $d)$ `FwLS` as the forward line-search by starting from α_k^* and increasing the step size until either infeasibility or the objective value does not improve.

(iii) We test our implementation on four problem cases: The first problem is a synthetic examples of size $p = 10$, where the data is generated as in [KC13]. We run this test for 10 times and report computational primitives in average. Three remaining problems are based on real data from

[6] We also provide MATLAB implementations of the examples in this section as a software package (SCOPT) at http://lions.epfl.ch/software.

http://ima.umn.edu/~maxxa007/send_SICS/, where the regularization parameters are chosen as the standard values (cf., [TDKC13c, LSS12, HSDR11]).

The numerical results are summarized in Table 5.2. Here, #iter denotes the (average) number of iterations, #chol represents the (average) number of Cholesky decompositions and #Mm is the (average) number of matrix-matrix multiplications.

Table 5.2: METADATA FOR THE LINE SEARCH STRATEGY COMPARISON

LS SCHEME	Synthetic ($\rho = 0.01$)			Arabidopsis ($\rho = 0.5$)			Leukemia ($\rho = 0.1$)			Hereditary ($\rho = 0.1$)		
	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm
NoLS	25.4	-	3400	18	-	1810	44	-	9842	72	-	20960
BtkLS	25.5	37.0	2436	11	25	718	15	50	1282	19	63	2006
E-BtkLS	25.5	36.2	2436	11	24	718	15	49	1282	15	51	1282
FwLS	18.1	**26.2**	**1632**	10	**17**	**612**	12	**34**	**844**	14	**44**	**1126**

We can see that our new step-size selection procedure FwLS shows superior empirical performance as compared to the rest: The standard approach NoLS usually starts with pessimistic step-sizes which are designed for worst-case problem structures. Therefore, we find it advantageous to continue with a forward line-search procedure. Whenever it reaches the quadratic convergence, no Cholesky decompositions are required. This makes a difference, compared to standard backtracking line-search BtkLS where we need to evaluate the objective value at every iteration. While there is no free lunch, the cost of computing λ_k is $\mathcal{O}(p^2)$ in FwLS, which turns out to be quite cheap in this application. The E-BtkLS combines both backtrack line-search and our analytic step-size $\alpha_k^* := (1 + \lambda_k)^{-1}$, which outperforms BtkLS as the regularization parameter becomes smaller. Finally, we note that the NoLS variant needs more iterations but it does not require any Cholesky decompositions, which might be advantageous in homogeneous computational platforms.

5.4.2 Graphical model selection

We customize our optimization framework to solve the graph selection problem (5.2). For notational convenience, we maintain a matrix variable $\boldsymbol{\Theta}$ instead of vectorizing it. We observe that $f(\boldsymbol{\Theta}) := -\log(\det(\boldsymbol{\Theta})) + \operatorname{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})$ is a standard self-concordant function, while $g(\boldsymbol{\Theta}) := \rho \left\| \mathbf{vec}(\boldsymbol{\Theta}) \right\|_1$ is convex and nonsmooth. The gradient and the Hessian of f can be computed explicitly as $\nabla f(\boldsymbol{\Theta}) := \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}^{-1}$ and $\nabla^2 f(\boldsymbol{\Theta}) := \boldsymbol{\Theta}^{-1} \otimes \boldsymbol{\Theta}^{-1}$, respectively. Next, we formulate our proposed framework to construct two algorithmic variants for (5.2).

Dual proximal-Newton algorithm

We consider a second order algorithm via a dual solution approach for (5.15). This approach is first introduced in our earlier work [TDKC13c], which did not consider the new modifications we propose in Section 5.3.1.

We begin by deriving the following dual formulation of the convex subproblem (5.15). Let $\mathbf{p}_k := \nabla f(\mathbf{x}^k)$,

the convex subproblem (5.15) can then be written equivalently as

$$\min_{\mathbf{x}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{x}^T\mathbf{H}_k\mathbf{x}+(\mathbf{p}_k-\mathbf{H}_k\mathbf{x}^k)^T\mathbf{x}+g(\mathbf{x})\right\}. \tag{5.23}$$

By using the min-max principle, we can write (5.23) as

$$\max_{\mathbf{u}\in\mathbb{R}^n}\min_{\mathbf{x}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{x}^T\mathbf{H}_k\mathbf{x}+(\mathbf{p}_k-\mathbf{H}_k\mathbf{x}^k)^T\mathbf{x}+\mathbf{u}^T\mathbf{x}-g^*(\mathbf{u})\right\}, \tag{5.24}$$

where g^* is the Fenchel conjugate function of g, i.e. $g^*(\mathbf{u}) := \sup_{\mathbf{x}}\left\{\mathbf{u}^T\mathbf{x}-g(\mathbf{x})\right\}$. Solving the inner minimization in (5.24) we obtain

$$\min_{\mathbf{u}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u}+\tilde{\mathbf{p}}_k^T\mathbf{u}+g^*(\mathbf{u})\right\}, \tag{5.25}$$

where $\tilde{\mathbf{p}}_k := \mathbf{H}_k^{-1}\mathbf{p}_k-\mathbf{x}^k$. Note that the objective function $\varphi(\mathbf{u}) := g^*(\mathbf{u})+\frac{1}{2}\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u}+\tilde{\mathbf{p}}_k^T\mathbf{u}$ of (5.25) is strongly convex, one can apply the fast projected gradient methods with a linear convergence rate for solving this problem, see [Nes13, BT09a].

In order to recover the solution of the primal subproblem (5.15), we note that the solution of the parametric minimization problem in (5.24) is given by $\mathbf{x}^*(\mathbf{u}) := \mathbf{x}^k-\mathbf{H}_k^{-1}(\mathbf{p}_k+\mathbf{u})$. Let $\mathbf{u}^*_{\mathbf{x}^k}$ be the optimal solution of (5.25). We can recover the primal proximal-Newton search direction $\mathbf{d}^k$ of the subproblem (5.15) as

$$\mathbf{d}_n^k = -\nabla^2 f(\mathbf{x}^k)^{-1}\left(\nabla f(\mathbf{x}^k)+\mathbf{u}^*_{\mathbf{x}^k}\right). \tag{5.26}$$

To compute the quantity λ_k defined by (5.18) in Algorithm 16, we use (5.26) such that

$$\lambda_k = \|\mathbf{d}_n^k\|_{\mathbf{x}^k} = \left\|\nabla f(\mathbf{x}^k)+\mathbf{u}^*_{\mathbf{x}^k}\right\|^*_{\mathbf{x}^k}. \tag{5.27}$$

Note that computing λ_k by (5.27) requires the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$.

Surprisingly, this dual approach allows us to avoid matrix inversion as well as Cholesky decomposition in computing the gradient $\nabla f(\mathbf{\Theta}_i)$ and the Hessian $\nabla^2 f(\mathbf{\Theta}_i)$ of f in graph selection. An alternative is of course to solve (5.15) in its primal form. Though, in such case, we need to compute $\mathbf{\Theta}_i^{-1}$ at each iteration i (say, via Cholesky decompositions).

The dual subproblem (5.25) becomes as:

$$\mathbf{U}^* = \underset{\|\mathbf{vec}(\mathbf{U})\|_\infty\leq 1}{\arg\min}\left\{\frac{1}{2}\mathrm{tr}((\mathbf{\Theta}_i\mathbf{U})^2)+\mathrm{tr}(\widetilde{\mathbf{Q}}\mathbf{U})\right\}, \tag{5.28}$$

for the graph selection, where $\widetilde{\mathbf{Q}} := \rho^{-1}[\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i-2\mathbf{\Theta}_i]$. Given the dual solution $\mathbf{U}^*$ of (5.28), the primal proximal-Newton search direction (i.e. the solution of (5.15)) is computed as

$$\mathbf{\Delta}_i := -\left((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}-\mathbb{I})\mathbf{\Theta}_i+\rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\right). \tag{5.29}$$

The quantity λ_i defined in (5.27) can be computed as follows, where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}}+\rho\mathbf{U}^*)$:

$$\lambda_i := \left(p-2\cdot\mathrm{tr}\left(\mathbf{W}_i\right)+\mathrm{tr}\left(\mathbf{W}_i^2\right)\right)^{1/2}. \tag{5.30}$$

Algorithm 17 summarizes the description above. Overall, this proximal-Newton (PN) algorithm *does*

Algorithm 17 (*Dual PN for graph selection* (DPNGS))

Input: Matrix $\widehat{\mathbf{\Sigma}} \succ 0$ and a given tolerance $\varepsilon > 0$. Set $\sigma := 0.25(5 - \sqrt{17})$.
Initialization: Find a starting point $\mathbf{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Set $\widetilde{\mathbf{Q}} := \rho^{-1}\left(\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i - 2\mathbf{\Theta}_i\right)$.
2. Compute $\mathbf{U}^*$ in (5.28).
3. Compute λ_i by (5.30), where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}} + \rho\mathbf{U}^*)$.
4. If $\lambda_i \leq \varepsilon$ terminate.
5. Compute $\mathbf{\Delta}_i := -\left((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}} - \mathbb{I})\mathbf{\Theta}_i + \rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\right)$.
6. If $\lambda_i > \sigma$, then set $\alpha_i := (1 + \lambda_i)^{-1}$. Otherwise, set $\alpha_i = 1$.
7. Update $\mathbf{\Theta}_{i+1} := \mathbf{\Theta}_i + \alpha_i\mathbf{\Delta}_i$.

end for

not require any matrix inversions or Cholesky decompositions. It only needs matrix-vector and matrix-matrix calculations, which might be attractive for different computational platforms (such as GPUs or simple parallel implementations) or appropriate matrix multiplication approximations can lead to accelerations [KVZ14]. Note however that as we work through the dual problem, the primal solution can be dense even if majority of the entries are rather small (e.g., smaller than 10^{-6}).[7]

We now explain the underlying costs of each step in Algorithm 17, which is useful when we consider different strategies for the selection of the step size α_k. The computation of $\widetilde{\mathbf{Q}}$ and $\mathbf{\Delta}_i$ require basic matrix multiplications. For the computation of λ_i, we require two trace operations: trace($\mathbf{W}_i$) in $\mathcal{O}(p)$ time-complexity and trace($\mathbf{W}_i^2$) in $\mathcal{O}(p^2)$ complexity. We note here that, while $\mathbf{W}_i$ is a *dense* matrix, the trace operation in the latter case requires only the computation of the diagonal elements of $\mathbf{W}_i^2$. Given $\mathbf{\Theta}_i$, α_i and $\mathbf{\Delta}_i$, the calculation of $\mathbf{\Theta}_{i+1}$ has $\mathcal{O}(p^2)$ complexity. In contrast, evaluation of the objective can be achieved through Cholesky decompositions, which has $\mathcal{O}(p^3)$ time complexity.

To compute (5.28), we can use the fast proximal-gradient method (FPGM) [Nes13, BT09a] with step size $1/L$ where L is the Lipschitz constant of the gradient of the objective function in (5.28). It is easy to observe that $L := \gamma_{\max}^2(\mathbf{\Theta}_i)$ where $\gamma_{\max}(\mathbf{\Theta}_i)$ is the largest eigenvalue of $\mathbf{\Theta}_i$. For sparse $\mathbf{\Theta}_i$, we can approximately compute $\gamma_{\max}(\mathbf{\Theta}_i)$ is $O(p^2)$ by using *iterative power methods* (typically, 10 iterations suffice). The projection onto $\|\mathbf{vec}(\mathbf{U})\|_\infty \leq 1$ clips the elements by unity in $O(p^2)$ time. Since FPGM requires a constant number of iterations $k_{\max}$ (independent of p) to achieve an ε_{in} solution accuracy, the time-complexity for the solution in (5.28) is $O(k_{\max}M)$, where M is the cost of matrix multiplication. We have also implemented block coordinate descent and active set methods which scale $O(p^2)$ in practice when the solution is quite sparse.

Overall, the major operation with general proximal maps in the algorithm is typically the matrix-matrix multiplications of the form $\mathbf{\Theta}_i\mathbf{U}\mathbf{\Theta_i}$, where $\mathbf{\Theta}_i$ and $\mathbf{U}$ are symmetric positive definite. This operation can naturally be computed (e.g., in a GPU) in a parallel or distributed manner. For more details of such computations we refer the reader to [BT89]. It is important to note that without Cholesky decompositions used in objective evaluations, the basic DPNGS approach theoretically scales with the cost of matrix-matrix multiplications.

[7] In our MATLAB code, we made no attempts to sparsify of the primal solution. The overall efficiency can be improved via thresholding tricks, both in terms of time-complexity (e.g., less number of iterations) and matrix estimation quality.

Proximal-gradient algorithm

For our experiments, we only describe the proximal gradient method of our approach to be used in the experiments for the case of the graph selection problem. Since $g(\mathbf{\Theta}) := \rho \|\mathbf{vec}(\mathbf{\Theta})\|_1$ and $\nabla f(\mathbf{\Theta}_i) = \mathbf{vec}(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})$, the subproblem (5.15) becomes

$$\mathbf{\Delta}_{i+1} := \mathcal{T}_{\tau_i \rho}\left(\mathbf{\Theta}_i - \tau_i(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})\right) - \mathbf{\Theta}_i, \tag{5.31}$$

where $\mathcal{T}_\tau : \mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$ is the component-wise matrix thresholding operator which is defined as $\mathcal{T}_\tau(\mathbf{\Theta}) := \max\{0, |\mathbf{\Theta}| - \tau\}$. We also note that the computation of $\mathbf{\Delta}_{i+1}$ requires a matrix inversion $\mathbf{\Theta}_i^{-1}$. Since $\mathbf{\Theta}_i$ is positive definite, one can apply Cholesky decompositions to compute $\mathbf{\Theta}_i^{-1}$ in $O(p^3)$ operations. To compute the quantity λ_i, we have $\lambda_i := \|\mathbf{\Delta}_i\|_{\mathbf{\Theta}_i} = \left\|\mathbf{\Theta}_i^{-1}\mathbf{\Delta}_i\right\|_2$. We also choose $L_i := 0.5\left\|\nabla^2 f(\mathbf{\Theta}_i)\right\|_2 = 0.5\left\|\mathbf{\Theta}_i^{-1}\right\|_2^2$. The above are summarized in Algorithm 18.

Algorithm 18 (*Proximal-gradient method for graph selection* (`ProxGrad1`))

Initialization: Choose a starting point $\mathbf{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Compute $\mathbf{\Theta}_i^{-1}$ via Cholesky decomposition.
2. Set $\tau_i := L_i^{-1}$.
3. Compute the search direction $\mathbf{\Delta}_i$ as (5.31).
4. Compute $\beta_i := L_i \|\mathbf{vec}(\mathbf{\Delta}_i)\|_2$ and $\lambda_i := \left\|\mathbf{\Theta}_i^{-1}\mathbf{\Delta}_i\right\|_2$.
5. Determine the step size $\alpha_i := \frac{\beta_i}{\lambda_i(\lambda_i+\beta_i)}$.
6. Update $\mathbf{\Theta}_{i+1} := \mathbf{\Theta}_i + \alpha_i\mathbf{\Delta}_i$.

end for

The per iteration complexity is dominated by matrix-matrix multiplications and Cholesky decompositions for matrix inversion calculations. In particular, Step 1 requires a Cholesky decomposition with $O(p^3)$ time-complexity. Step 2 requires to compute ℓ_2-norm of a symmetric positive matrix, which can be done by a power-method in $O(p^2)$ time-complexity. The complexity of Steps 3, 4 and 6 requires $O(p^2)$ operations. Step 2 may require additional bisection steps whenever $\lambda_k < 1$.

Impact of different solvers for the subproblems

As mentioned in the introduction, an important step in our second order algorithmic framework is the solution of the subproblem (5.15). If the variable matrix $\mathbf{H}_k$ is not diagonal, computing $\mathbf{s}_{\mathbf{H}_k}^k$ corresponds to solving a convex subproblem. For a given regularization term g, we can exploit different existing approaches to tackle this problem. We illustrate that the overall framework is quite robust against the solution accuracy of the individual subsolver.

In this test, we consider the broadly used ℓ_1-norm function as the regularizer. Hence, (5.15) collapses to an unconstrained LASSO problem; cf. [WNF09]. We implement the proximal-Newton algorithm to solve the graph learning problem (5.2) where $g(\mathbf{x}) := \rho\|\mathbf{x}\|_1$. To show the impact of the subsolver in (5.2), we implement the following methods, which are all available in our software package SCOPT:

(i) `pFISTA` and `dFISTA`: in these cases, we use the FISTA algorithm [BT09a] for solving the primal (5.23) and the dual subproblem (5.25). Morever, to speedup the computations, we further run these methods on the GPU [NVIDIA Quadro 4000].

Table 5.3: METADATA FOR THE SUBSOLVER EFFICIENCY COMPARISON— $\rho = 0.5$

SUB-SOLVERS	Estrogen ($p = 692$)			Arabidopsis ($p = 834$)			Leukemia ($p = 1255$)			Hereditary($p = 1869$)		
	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]
pFISTA	9	29	13.10	10	35	24.76	**9**	31	286.57	17	80	1608.66
pFISTA[gpu]	9	29	10.70	10	35	16.81	**9**	31	231.97	17	80	1265.97
dFISTA	8	**16**	4.66	10	**17**	10.92	14	**22**	50.19	14	**27**	147.86
dFISTA[gpu]	8	**16**	**4.16**	10	**17**	**7.89**	14	**22**	**43.53**	14	**27**	**120.16**
FastAS	**7**	24	28.69	**8**	27	96.93	**9**	31	532.11	**11**	40	1682.28
BCDC	8	25	90.35	9	28	227.27	**9**	31	549.80	12	47	3452.82
MatQUIC	11	29	21.61	10	35	50.67	10	35	119.06	14	44	891.29
ProxGrad1	175	175	8.82	226	226	17.78	230	230	44.06	660	660	350.52

Table 5.4: METADATA FOR THE SUBSOLVER EFFICIENCY COMPARISON — $\rho = 0.1$

SUB-SOLVERS	Estrogen ($p = 692$)			Arabidopsis ($p = 834$)			Leukemia ($p = 1255$)			Hereditary($p = 1869$)		
	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]
pFISTA	34	101	357.25	57	148	1056.90	143	242	7490.27	-	-	-
pFISTA[gpu]	34	101	300.90	57	148	730.07	143	242	6083.06	-	-	-
dFISTA	14	**32**	12.51	**12**	**35**	15.53	**12**	**34**	38.73	**14**	**44**	150.03
dFISTA[gpu]	14	**32**	**11.18**	**12**	**35**	**11.18**	**12**	**34**	**33.45**	**14**	**44**	121.37
FastAS	-	-	-	-	-	-	-	-	-	-	-	-
BCDC	**13**	48	1839.17	15	50	4806.62	-	-	-	-	-	-
MatQUIC	30	88	573.87	36	95	1255.13	36	95	4260.97	-	-	-
ProxGrad1	4345	4345	224.95	6640	6640	532.77	9225	9225	1797.49	-	-	-

(ii) FastAS: this method corresponds to the exact implementation of the fast active-set method proposed in [KP10] for solving the primal-dual (5.23).

(iii) BCDC: here, we consider the block-coordinate descent method implemented in [HSDR11] for solving the primal subproblem (5.23).

We also compare the above variants of the *proximal-Newton approach* with (i) the proximal-gradient method (Algorithm 18) denoted by ProxGrad1 and (ii) a precise MATLAB implementation of QUIC (MatQUIC), as described in [HSDR11]. For the proximal-Newton and MatQUIC approaches, we terminate the execution if the maximum number of iterations exceeds 200 or the total execution time exceeds the 5 hours. The maximum number of iterations in ProxGrad1 is set to 10^4.

The results are reported in Tables 5.3-5.4. Overall, we observe that dFISTA shows superior performance across the board in terms of computational time and the total number of Cholesky decompositions required. Here, #nnz represents the number of nonzero entries in the final solution. The notation "$-$" indicates that the algorithms exceed either the maximum number of iterations or the time limit (5 hours).

If the parameter ρ is relatively large (i.e., the solution is expected to be quite sparse), FastAS, BCDC and MatQUIC perform well and converge in a reasonable time. This is expected since all three approaches vastly rely on the sparsity of the solution: the sparser the solution is, the faster their computations are performed, as restricted on the active set of variables. However, when ρ is small, the performance of these methods significantly degrade due to the increased number of active (non-zero) entries.

Aside from the above, ProxGrad1 performs well in terms of computational time, as compared to the rest of the methods. Unfortunately, the number of Cholesky decompositions in this method can become

as many as the number of iterations, which indicates a computational bottleneck in high-dimensional problem cases. Moreover, when ρ is small, this method also slows down and requires more iterations to converge.

On the other hand, we also note that `pFISTA` is rather sensitive to the accuracy of the subsolver within the quadratic convergence region. In fact, while `pFISTA` reaches medium scale accuracies in a manner similar to `dFISTA`, it spends most of its iterations trying to achieve the higher accuracy values. However, this could also be an artifact of our MATLAB implementation.

5.4.3 Sparse covariance estimation

Let $\{\mathbf{x}_j\}_{j=1}^m$ be a collection of n-variate random vectors, i.e., $\mathbf{x}_j \in \mathbb{R}^n$, drawn from a joint probability distribution with positive definite covariance matrix $\mathbf{\Sigma}$. In this context, assume there may exist unknown marginal independences among the variables to discover; we note that $(\mathbf{\Sigma})_{kl} = 0$ when the k-th and l-th variables are independent. Thus, $\mathbf{\Sigma}$ ranges from being diagonal, where every component is independent to every other component, to being a fully dense matrix, where all the components are *unconditionally* pairwise dependent. In this work, we consider problems where $\mathbf{\Sigma}$ is unknown and *sparse*, i.e., only a small number of entries are nonzero. Our goal is to recover the nonzero pattern of $\mathbf{\Sigma}$, as well as compute a decent approximation, from a (possibly) limited sample corpus.

Covariance estimation is an important problem, found in diverse research areas. In classic portfolio optimization [Mar52], the covariance matrix over the asset returns is unknown and thus it is approximated from historical data. Unfortunately though, "...financial data is typically non-stationary. This limits the amount of data that can be used to meaningfully estimate (...) the covariance of the asset return vector" [Pol12]. In this context, even the estimation of the most significant dependencies among assets might lead to meaningful decisions for portfolio optimization. Moreover, shrinkage operations over the covariance estimates, such as sparsity-inducing regularization, mitigate the instability (due to the small sample size) of classic sample covariance estimators [BL08]. In bioinformatics, we are interested in inferring the dependency network among genes [SS05]: groups might be completely independent from other groups. In its simplest form, this problem boils down to the covariance estimation problem from insufficiently small amount of gene expression data, where sparsity regularization has shown to help in practice [KSB09]. Other applications of the sparse covariance estimation include fMRI imaging [VGPT10], data mining [AKMZ02], etc. Overall, sparse covariance matrices come with nice properties such as natural graphical interpretation, whereas are easy to be transfered and stored.

Optimization Criteria for Sparse Covariance Estimation

Given $\{\mathbf{x}_j\}_{j=1}^m$, the empirical covariance[8] matrix $\widehat{\mathbf{\Sigma}} = \frac{1}{m}\sum_{j=1}^m (\mathbf{x}_j - \widehat{\boldsymbol{\mu}})(\mathbf{x}_j - \widehat{\boldsymbol{\mu}})^T$, where $\widehat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{j=1}^m \mathbf{x}_j$, turns out to be (near) optimal *only* in fixed, low-dimensional settings where sufficient data is provided (chapter 3.2, [And58]). Unfortunately, such traditional estimation techniques are prone to errors when the dimensionality of the problem increases and $m \ll n$ [Joh01].

To mitigate such phenomena and based on observations in [Dem72], recent works utilize general thresholding techniques to compute a succinct solution that fits the model adequately. According to these

[8] In what follows, one can safely work with *correlation* matrices instead of covariance ones: In such case, the true correlation matrix has the same sparsity pattern as the true covariance matrix and its diagonal entries have unit values. The analysis employed in this paper apply for both cases.

approaches, we approximate $\mathbf{\Sigma}$ with $\mathbf{\Theta}^\star$ where:

$$\mathbf{\Theta}^\star \in \arg\min_{\mathbf{\Theta}} \left\{ \frac{1}{2}\|\mathbf{\Theta} - \widehat{\mathbf{\Sigma}}\|_F^2 + \lambda \|\mathbf{\Theta}\|_{\#} \right\}; \tag{5.32}$$

here, $\lambda > 0$ is a regularization parameter that controls the sparsity of the solution and $\|\cdot\|_{\#}$ denotes a sparsity-inducing regularization norm; a non-exhaustive list of such thresholding functions includes entrywise hard thresholding [BL08] and soft-thresholding [RLZ09].[9] One can easily identify that (5.32) is the proximity operator proposed by Moreau [Mor62]; a generalization of the Euclidean projection where additional structure is incorporated as a regularization to "bias" the estimate.

Unfortunately, the solution of (5.32) is not guaranteed to be a positive definite matrix (in the non-asymptotic sense), according to the following theorem; the proof is provided in [GR13, GR12]:

Theorem 18 ([GR13, GR12]). *Let $\mathbf{A} \in \mathbb{S}_{++}^n$ and $\mathcal{H}_k(\beta) = \beta \cdot \mathbb{1}_{|\beta|>k}$ and $\mathcal{S}_\lambda(\beta) = sign(\beta)\cdot\max\{|\beta|-\lambda, 0\}$ denote the elementwise hard- and soft-thresholding operations for $\beta \in \mathbb{R}$, with parameters k and λ, respectively. Even if $\mathbf{A}$ is a sparse matrix, there is no guarantee in general that $\mathcal{H}_k(\mathbf{A}) \succ 0$ and $\mathcal{S}_\lambda(\mathbf{A}) \succ 0$, for any k, λ; i.e., there are no* universal *values $k_0 > 0$ and $\lambda_0 > 0$ such that $\mathcal{H}_{k_0}(\mathbf{A}) \succ 0$ and $\mathcal{S}_{\lambda_0}(\mathbf{A}) \succ 0$ for any $\mathbf{A} \in \mathbb{S}_{++}^n$, except for some trivial cases.*

Therefore, even if we asssume $\widehat{\mathbf{\Sigma}} \succ 0$, the positive-definite cone is not invariant with respect to general elementwise thresholding operations such as hard- and soft-thresholding[10]; this leads to the surprising result that sparsity-inducing regularization cannot preserve positive definiteness at the same time.

To this end, consider the following estimator for $\mathbf{\Sigma}$ where we force the positive definiteness of the solution with the constraint $\mathbf{\Theta} \succ 0$:

$$\mathbf{\Theta}^\star = \arg\min_{\mathbf{\Theta}\succ 0} \left\{ \frac{1}{2}\|\mathbf{\Theta} - \widehat{\mathbf{\Sigma}}\|_F^2 + \lambda \|\mathbf{\Theta}\|_{\#} \right\}. \tag{5.33}$$

For easily computable proximity operators (e.g., both hard- and soft-thresholding operations apply elementwise and incur very little computational cost), the main computational bottleneck of solving (5.33) is due to the constraint $\mathbf{\Theta} \succ 0$: the putative solution must be projected onto the positive definite cone $\mathbb{S}_{++}^n$ by "forcing" its eigenvalues to be positive.

To overcome this difficulty, we consider a variant of (5.33) with $\# = 1$, as described next

SPARSE COVARIANCE ESTIMATION: *Given a set of n-dimensional samples $\{\mathbf{x}_j\}_{j=1}^m$, drawn from a joint probability density function with* unknown *sparse covariance $\mathbf{\Sigma} \in \mathbb{S}_{++}^n$, we approximate $\mathbf{\Sigma}$ as the solution to the following optimization problem:*

$$\mathbf{\Theta}^\star = \arg\min_{\mathbf{\Theta}} \underbrace{\{\frac{1}{2\rho}\|\mathbf{\Theta} - \widehat{\mathbf{\Sigma}}\|_F^2 - \log\det(\mathbf{\Theta})}_{=f(\mathbf{\Theta})} + \underbrace{\frac{\lambda}{\rho}\|\mathbf{\Theta}\|_1}_{=g(\mathbf{\Theta})}\}. \tag{5.34}$$

[9] There are several works that consider $\|\cdot\|_{\#}$, where the norm operates only on the off-diagonal elements. Our work naturally extends to these cases.

[10] The authors in [GR13, GR12] actually show that the positive definite cone is invariant to such thresholding operations if $\widehat{\mathbf{\Sigma}}$ is a sparse matrix with a specific tree-based nonzero pattern.

In this case, we incorporate an additional regularization term to promote positive definite solutions with no constraints. The $\log\det(\boldsymbol{\Theta})$ term in (5.34) operates as a log-barrier function for the positive-definite cone set. Though, while we avoid the computation of the projection onto $\mathbb{S}^n_{++}$, we note that this criterion introduces an additional regularization parameter ρ in the objective function, which needs to be carefully selected.

Prior work

In [XMZ12], the authors propose a Alternating Direction Method of Multipliers (ADMM) variant to solve (5.33) where $\# = 1$. Such approaches are quite powerful in practice for specific applications. Their main drawback is the manual tuning of the penalty parameter in the augmented Lagrangian function: The authors found that a constant step size selection $\mu = 2$ (eq. (6) in [XMZ12]) leads to a fast and stable implementation of their proposed scheme. Unfortunately, the analysis of the global convergence as well as the convergence rate of such schemes is an issue since they strongly depend on the choice of μ.

From a different perspective and inspired by [DVR08], [BT11] derives the following optimization criterion:

$$\boldsymbol{\Theta}^\star = \underset{\boldsymbol{\Theta}\succ 0}{\arg\min} \left\{ \log\det(\boldsymbol{\Theta}) + \text{trace}(\boldsymbol{\Theta}^{-1}\widehat{\boldsymbol{\Sigma}}) + \lambda\|\boldsymbol{\Theta}\|_1 \right\}, \tag{5.35}$$

under the assumption that *variables* $\mathbf{x}$ *satisfy sub-Gaussian tail conditions*. Since (5.35) is highly nonconvex, as the sum of convex and concave functions, the authors follow a majorization-minimization strategy, where the $\log\det(\cdot)$ part is linearized to be the majorizer function.[11]

To the best of our knowledge, only the work presented in [Rot12] considers (5.34) where an iterative block-coordinatewise graphical LASSO approach is utilized with the following guarantees:

Theorem 19 ([Rot12]). *Assume* $\boldsymbol{\Sigma} \in \mathbb{S}^n_{++}$ *with sparsity* $\|\boldsymbol{\Sigma}\|_0 \leq k$, *such that* $0 < \epsilon_1 \leq \lambda_{min}(\boldsymbol{\Sigma}) \leq \lambda_{max}(\boldsymbol{\Sigma}) \leq \epsilon_2 < \infty$, *for* ϵ_1, ϵ_2 *constants. Moreover, assume that the samples* $\{\mathbf{x}_j\}_{j=1}^m$ *satisfy* $\mathbb{E}\left[|\mathbf{x}_j|^{2\alpha}\right] \leq c < \infty$ *for all* j *and for* $\alpha \geq 2$. *Then, if* $\lambda = \mathcal{O}\left(\sqrt{\frac{n^{4/\alpha}}{N}}\right)$, $\rho = \mathcal{O}\left(\frac{\sqrt{\frac{k\cdot n^{4/\alpha}}{N}}}{\|\mathbf{R}^{-1}\|_F}\right)$ *and* $\left((k+1)n^{4/\alpha}\right) \in o(N)$, *then:*

$$\|\boldsymbol{\Theta}^\star - \boldsymbol{\Sigma}\|_F = \mathcal{O}\left(\sqrt{\frac{(k+1)n^{4/\alpha}}{N}}\right), \quad \textit{where } \mathbf{R} \textit{ denotes the true correlation matrix.} \tag{5.36}$$

We underline that the above theorem holds for all probability distributions and better bounds can be obtained under Gaussian assumptions. Moreover, no convergence guarantee is provided for this scheme.

A summary of the discussion above is given in Table 5.5.

How to solve subproblem for sparse covariance estimation: An important ingredient for our scheme is the calculation of the descent direction. For simplicity we use FISTA, a fast ℓ_1-norm regularized gradient method for solving the proximal subproblem, and describe how to efficiently implement such solver for our case.[12]

[11] In practice, a Alternating Direction Method of Multipliers (ADMM) algorithm is proposed which has slow convergence in general for arbitrary step size selections.

[12] We note that one can solve (5.39) using a dual approach. In this case, the subproblem becomes

Table 5.5: Summary of related work.

	[XMZ12]	[BT11]	[Rot12]	[Wan12]	This work
Complexity per iteration	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$
# of tuning parameters	2	1	2	1	2
Convergence guarantee	✓	–	✓	–	✓
Convergence rate	Linear	Linear	–[†]	–[†]	Quadratic
Covariate distribution	Any	Gaussian	Any	Gaussian	Any

[†]To the best of our knowledge, block coordinate descent algorithms have known convergence *only* for the case of Lipschitz continuous gradient objective functions [BT13].

Given the current estimate of $\mathbf{x}_i$ in the i-th iteration, the gradient and the Hessian of function $f(\cdot)$ around $\mathbf{x}_i$ can be computed respectively as:

$$\nabla f(\mathbf{x}_i) = \frac{1}{\rho}\left(\mathbf{x}_i - \mathbf{vec}(\widehat{\boldsymbol{\Sigma}})\right) - \mathbf{vec}\left(\mathbf{mat}(\mathbf{x}_i)^{-1}\right) \in \mathbb{R}^{p\times 1}, \tag{5.37}$$

and

$$\nabla^2 f(\mathbf{x}_i) = \frac{\partial \nabla f(\mathbf{x}_i)}{\partial \mathbf{x}_i} = \frac{\mathbf{I}_{p\times p}}{\rho} + (\mathbf{mat}(\mathbf{x}_i)^{-1} \otimes \mathbf{mat}(\mathbf{x}_i)^{-1}) \in \mathbb{R}^{p\times p}. \tag{5.38}$$

Using (5.37), (5.38), let $\mathbf{h} := \nabla f(\mathbf{x}_i) - \nabla^2 f(\mathbf{x}_i)\mathbf{x}_i$. One can easily observe that the proximal subproblem is equivalent to:

$$\boldsymbol{\delta}_i = \arg\min_{\boldsymbol{\delta}} \left\{ \underbrace{\frac{1}{2}\boldsymbol{\delta}^T \nabla^2 f(\mathbf{x}_i)\boldsymbol{\delta} + \mathbf{h}^T\boldsymbol{\delta}}_{\varphi(\boldsymbol{\delta})} + g(\boldsymbol{\delta}) \right\}, \tag{5.39}$$

where $\varphi(\cdot)$ is a smooth convex function[13] with Lipschitz constant:

$$L := \left\| \frac{\mathbf{I}}{\rho} + (\mathbf{mat}(\mathbf{x}_i)^{-1} \otimes \mathbf{mat}(\mathbf{x}_i)^{-1}) \right\|_{2\to 2} = \frac{1}{\rho} + \frac{1}{\lambda^2_{\min}(\mathbf{mat}(\mathbf{x}_i))}. \tag{5.40}$$

Combining the above quantities in a ISTA-like gradient descent procedure [DDDM04], we have:

$$\boldsymbol{\delta}^{k+1} = \mathcal{S}_{\frac{\lambda}{L\rho}}\left(\boldsymbol{\delta}^k - \frac{1}{L}\nabla\varphi(\boldsymbol{\delta}^k)\right), \tag{5.41}$$

where we use superscript k to denote the k-th iteration of the ISTA procedure (as opposed to the subscript i for the i-th iteration). Here, $\nabla\varphi(\boldsymbol{\delta}^k) = \nabla^2 f(\mathbf{x}_i)\boldsymbol{\delta}^k + \mathbf{h}$ and $\mathcal{S}_{\frac{\lambda}{L\rho}}(\mathbf{x}) := \mathrm{sign}(\mathbf{x})\max\{|\mathbf{x}| - \frac{\lambda}{L\rho}, 0\}$. Furthermore, to achieve an $\mathcal{O}(1/k^2)$ convergence rate, one can use acceleration techniques that lead to FISTA algorithm [BT09b], based on Nesterov's seminal work [Nes83].

Implementation details: We observe that L and $\mathbf{h}$ can be precomputed once. Given $\mathbf{x}_i$, we compute $\lambda_{\min}(\mathbf{mat}(\mathbf{x}_i))$ in $\mathcal{O}(n^3)$ time complexity, while $\mathbf{h}$ can be computed with $\mathcal{O}(n^3)$ time cost using the Kro-

$\min_{\|\mathbf{u}\|_\infty \le 1}\left\{\frac{1}{2}\mathbf{u}^T\left(\nabla^2 f(\mathbf{x}_i)\right)^{-1}\mathbf{u} + \tilde{\mathbf{q}}^T\mathbf{u}\right\}$, where $\tilde{\mathbf{q}} = \frac{\rho}{\lambda}\left(\left(\nabla^2 f(\mathbf{x}_i)\right)^{-1}\mathbf{q} - \mathbf{x}_i\right)$. However, the inversion of the Hessian creates a computational bottleneck due its $\mathcal{O}(p^3)$ (i.e., $\mathcal{O}(n^6)$) time cost.

[13]If $\nabla^2 f(\mathbf{x}_i) \succ \mu\mathbf{I}$ where μ is known, then $\varphi(\cdot)$ is *strongly* convex and more acceleration in the convergence rate sense can be achieved.

necker product property $\mathbf{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\mathbf{vec}(\mathbf{X})$. Similarly, $\nabla\varphi(\boldsymbol{\delta}^k)$ can be iteratively computed in $\mathcal{O}(n^3)$ time cost. Overall, the above procedure has $\mathcal{O}(K \cdot n^3)$ computational cost, where K is the total number of iterations.

In this section, we conduct extensive experiments to compare the numerical performance of several methods on both real and synthetic datasets. All approaches under comparison are optimally implementing in MATLAB code with no C-coded parts. Our experiments are executed using a MATLAB based environment on a MacBook Air, equipped with a 1.8 GHz Intel Core i7 processor and 4GByte 1333 MHz DDR3 main memory. Overall, the proposed scheme achieves the desiderata with smaller computational cost and better covariance recovery performance, as opposed to the rest of the schemes under comparison.

Time efficiency of SCOPT in sparse covariance estimation

To the best of our knowledge, only the work of A. Rothman [Rot12] considers the same objective function (5.34) for the problem of sparse covariance estimation. According to [Rot12], the proposed algorithm follows similar motions with the graphical LASSO method [FHT08] and the graphical elastic net algorithm [CPS11]: Every covariate is computed via a column- and row-wise cyclical coordinate descent method where each column and row of the estimate is estimated using a ℓ_1-norm LASSO type of optimization.

We generate the following three synthetic examples [Rot12]:

(i) In the first scenario, $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_1$ where:

$$\boldsymbol{\Sigma}_1(i,j) = \begin{cases} 1 & \text{if } i = j, \\ 0.4 & \text{if } |i-j| = 1, \end{cases}$$

i.e., $\boldsymbol{\Sigma}_1$ is a tridiagonal covariance matrix. This model might occur in random processes, where the correlations are localized in time, i.e., the current variable depends heavily only on the recent and future variable, but weakly on the rest.

(ii) In the second scenario, $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_2$ is a block-sparse covariance matrix with overlapping blocks of dependencies and unit diagonal entries. In particular, we assume that there are b blocks of dependent variables $\mathcal{B}_q,\ q \in [b]$, where each block $\mathcal{B}_q$ of variables has variable size. Then:

$$\boldsymbol{\Sigma}_2(i,j) = \begin{cases} 1 & \text{if } i = j, \\ 0.4 & \text{if } i,j \in \mathcal{B}_q \text{ for some } q \in [b], \\ 0.8 & \text{if } i,j \text{ lie in adjacent blocks}, \\ 0 & \text{elsewhere.} \end{cases}$$

We tested various number of blocks $b = 5, 10, 50, 100$ for the cases $n = 500, 1000, 2000, 5000$. This model generalizes scenario (i) where larger sets of (consecutive in time) variables are dependent; furthermore, variables that belonging to two blocks simultaneously show higher correlation.

(iii) Finally, in the third scenario, $\boldsymbol{\Sigma} \equiv \boldsymbol{\Sigma}_3$ is a random positive definite covariance matrix with $\|\boldsymbol{\Sigma}_3\|_0 = k$. In our experiments, we test sparsity levels k such that $\frac{k}{n^2} = \{0.05, 0.1, 0.2\}$.

In all cases and without loss of generality, we assume that the variables are drawn from a joint Gaussian probability distribution. Given $\boldsymbol{\Sigma}$, we generate $\{\mathbf{x}_j\}_{j=1}^m$ random n-variate vectors according to $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $n \in \{500, 1000, 2000, 5000\}$ and $m = \frac{n}{2}$. We highlight that the sample covariance matrix $\widehat{\boldsymbol{\Sigma}} =$

$\frac{1}{m}\sum_{j=1}^{m}\mathbf{x}_j\mathbf{x}_j^T$ is ill-conditioned in all cases with $\text{rank}(\widehat{\mathbf{\Sigma}}) \leq \frac{n}{2}$. We observe that the number of unknowns is $\binom{n}{2} = \frac{n(n-1)}{2}$; in our testbed, this corresponds to estimation of 4950 up to $12,497,500$ variables, depending on the value of n.

Implementation-wise, we set the total number of iterations to $I^{\max} = 500$ and the error tolerance to $\gamma = 10^{-10}$. The approximation tolerance of the subsolver (5.39) ϵ equals to 10^{-8}. To compute L in (5.40), we use a power method scheme with $\texttt{Pw} = 20$ iterations. All algorithms under comparison are initialized with $\mathbf{x}_0 = \textbf{vec}\left(\text{diag}(\widehat{\mathbf{\Sigma}})\right)$. As an execution wall time, we set $\texttt{T} = 3600$ seconds (1 hour). In all cases, we set $\rho = 0.1$.

Table 5.6: Summary of comparison results for time efficiency.

Model				$F(\mathbf{\Theta}^\star)$ $(\times 10^2)$			Time (secs)		
	n		λ	[Rot12]	SCOPT	SCOPT FLS	[Rot12]	SCOPT	SCOPT FLS
$\mathbf{\Sigma}_1$	500		0.3	6.942	**6.884**	**6.884**	25.752	21.722	**2.592**
	1000		0.2	26.644	**26.266**	**26.266**	229.031	184.221	**23.697**
	2000		0.1	233.729	**232.726**	**232.726**	247.064	202.134	**82.932**
	5000		0.1	–	–	**582.585**	$>$ T	$>$ T	**1330.641**
$\mathbf{\Sigma}_2$	500	$b = 5$	0.5	260.232	**245.591**	**245.591**	62.965	342.067	96.777
	1000	$b = 10$	0.5	553.821	**502.550**	**502.550**	477.548	3272.567	**174.414**
	2000	$b = 50$	0.2	–	–	**1358.171**	$>$ T	$>$ T	**471.932**
	5000	$b = 100$	0.1	–	–	**7529.334**	$>$ T	$>$ T	**3394.512**
$\mathbf{\Sigma}_3$	100	$\frac{k}{n^2} = 0.05$	1	32.013	**31.919**	**31.919**	8.288	9.996	**3.584**
		$\frac{k}{n^2} = 0.1$	0.5	36.190	**34.689**	**34.689**	10.470	12.761	**5.012**
		$\frac{k}{n^2} = 0.2$	0.5	62.143	**53.081**	**53.081**	18.446	14.720	**6.257**
	1000	$\frac{k}{n^2} = 0.05$	1	–	–	**2711.931**	$>$ T	$>$ T	**759.724**
		$\frac{k}{n^2} = 0.1$	1	–	–	**4734.251**	$>$ T	$>$ T	**875.344**
		$\frac{k}{n^2} = 0.2$	1	–	–	**5553.508**	$>$ T	$>$ T	**1059.709**
	2000	$\frac{k}{n^2} = 0.05$	1	–	–	**3244.956**	$>$ T	$>$ T	**1121.377**
		$\frac{k}{n^2} = 0.1$	1	–	–	**3847.061**	$>$ T	$>$ T	**2157.029**
		$\frac{k}{n^2} = 0.2$	1	–	–	–	$>$ T	$>$ T	$>$ T

Table 5.6 contains the summary of results. Overall, we observe that the proposed framework shows superior performance across diverse configuration settings, both in terms of time complexity and objective function minimization efficiency: both SCOPT and SCOPT FLS (Forward Line Search as described in the previous subsections) find solutions with lower objective function value, as compared to [Rot12], within the same time frame. The regular SCOPT algorithm performs relatively well in terms of computational time as compared to the rest of the methods. However, its convergence rate heavily depends on the *conservative* τ_i selection.

Reconstruction efficiency of SCOPT in sparse covariance estimation

In this subsection, we measure the $\mathbf{\Sigma}$ reconstruction efficacy of solving (5.34), as compared to other optimization formulations for sparse covariance estimation. To this end, we compare the $\mathbf{\Theta}^\star$ estimates as computed by: (i) the Alternating Direction Method of Multipliers (ADMM) implementation [XMZ12] of (5.33) for $\# = 1$, (ii) the coordinate descent algorithm for solving (5.35) as presented in [Wan12] and, (iii)

our presented algorithm for the problem described in (5.34). It is obvious that the direct comparison of the achieved objective function values has no clear interpretation. Thus, we use as comparison metric the normalized distance $\frac{\|\boldsymbol{\Theta}^\star - \boldsymbol{\Sigma}\|_F}{\|\boldsymbol{\Sigma}\|_F}$ for each estimate as well as the captured sparsity pattern in $\boldsymbol{\Sigma}$.

Table 5.7 aggregates the experimental results. Without loss of generality, we fix $\lambda = 0.5, \rho = 0.1$ for the case $n = 100$ and, $\lambda = 1.5, \rho = 0.1$ for the cases $n = 2000,\ 5000$. SCOPT framework is at least as competitive with the state-of-the-art implementations for sparse covariance estimation. It is evident that the proposed SCOPT variant, based on self-concordant analysis, is at least one order of magnitude faster than the rest of algorithms under comparison. In terms of reconstruction efficacy, using our proposed scheme, we can achieve marginally better $\boldsymbol{\Sigma}$ reconstruction performance, as compared to [XMZ12]. However, SCOPT FLS recovers $\gtrsim 60\%$ of the true sparsity pattern; at least 82% better sparsity recovery than [XMZ12] and [Wan12].[14]

Table 5.7: Summary of comparison results for reconstruction of efficiency.

Model			$\|\boldsymbol{\Theta}^\star - \boldsymbol{\Sigma}\|_F / \|\boldsymbol{\Sigma}\|_F$			Time			Support Recovery (%)		
	n	N	[Wan12]	[XMZ12]	SCOPT FLS	[Wan12]	[XMZ12]	SCOPT FLS	[Wan12]	[XMZ12]	SCOPT FLS
$\boldsymbol{\Sigma}_3$	100	$n/2$	1.180	0.912	**0.908**	0.456	**0.252**	2.604	9.49	38.76	**66.87**
		n	0.9201	0.554	**0.542**	0.494	**0.108**	0.155	9.47	34.01	**71.01**
		$10n$	0.396	0.192	**0.190**	0.451	0.108	**0.054**	9.50	42.29	**75.87**
	2000	$n/2$	–	**0.428**	**0.428**	$> \mathtt{T}$	350.145	**203.515**	–	32.80	**69.42**
		n	–	**0.352**	**0.352**	$> \mathtt{T}$	385.340	167.688	–	45.23	**71.89**
		$10n$	–	0.211	**0.209**	$> \mathtt{T}$	401.970	**122.535**	–	52.32	**74.77**
	5000	$n/2$	–	–	**0.424**	$> \mathtt{T}$	$> \mathtt{T}$	**2496.112**	–	–	**59.78**
		n	–	–	**0.350**	$> \mathtt{T}$	$> \mathtt{T}$	**1792.086**	–	–	62.65
		$10n$	–	–	**0.258**	$> \mathtt{T}$	$> \mathtt{T}$	**1558.192**	–	–	65.41

Application to classic portfolio optimization

Introduced by Harry Markowitz [Mar52], mean-variance optimization (MVO) lies at the heart of classic portfolio optimization theory as a means of asset allocation recommendations with minimum risk. Shortly, assume we possess historical stock market data $\{\mathbf{r}^{(i)}\}_{i=1}^{\mathtt{T}_{\text{total}}}$ of n stocks, where $\mathbf{r}^{(i)} \in \mathbb{R}^n, \forall i$, represent the actual return of the i-th asset over a time period $\mathtt{T}_{\text{total}}$; both monthly- and daily-based data apply. Our goal is to propose a portfolio $\mathbf{w} \in \mathbb{R}^n$ such that its application in future stock market sessions would result into a desired stock return μ with the minimum possible risk.

In mathematical terms, the above describe the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{r} = \mu \\ & \sum_i w_i = C, \ \ w_i \geq 0, \ \forall i. \end{aligned} \tag{5.42}$$

Here, $\boldsymbol{\Sigma} \in \mathbb{S}_+^n$ is the *true* covariance matrix over the asset returns, $\mathbf{r} \in \mathbb{R}^n$ denotes the *true* asset returns, $\mathbf{w}$ represents a weighted probability distribution over the set of assets such that $\sum_i w_i = C$ and C is the total capital to be invested. Without loss of generality, one can assume a normalized capital such that $\sum_i w_i = 1$. In such case, $\mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w}$ is both the risk of the investment as well as a metric of *variance* of the portfolio selection.

[14] The solutions returned by SCOPT FLS are not fully dense but capture most of the actual nonzero pattern of $\boldsymbol{\Sigma}$.

In practice, both $\mathbf{r}$ and $\mathbf{\Sigma}$ are unknown and MVO requires an estimation for both. One can easily observe that the total number of variables to estimate is $n + \binom{n}{2}$. While for small-sized portfolios, i.e., $n = 10$, the sample estimates $\widehat{\mathbf{r}} = \frac{1}{\mathtt{T}} \sum_{i \in \mathtt{T}} \mathbf{r}^{(i)}$ and $\widehat{\mathbf{\Sigma}} = \frac{1}{\mathtt{T}-1} \sum_{i \in \mathtt{T}} \left(\mathbf{r}^{(i)} - \widehat{\mathbf{r}}\right) \left(\mathbf{r}^{(i)} - \widehat{\mathbf{r}}\right)^T$, $\mathtt{T} \subset \mathtt{T}_{\text{total}}$, are reliable approximations, they quickly become problematic in the large scale: the amount of data required increases quadratically to be commensurate with the degree of dimensionality. Due to such difficulties, even a simple *equal weighted portfolio* $\mathbf{w}$ such that $w_i = 1/n, \ \forall i$, is often preferred in practice [DGU09].

Nevertheless, data analysts and practitioners regularly assume that many elements of the covariance matrix are zero, a property which is appealing due to its interpretability and ease of estimation. Moreover, there are cases in practice where most of the variables are correlated to only a few others.

In the discussion below, (i) we highlight a small part of the independences observed among stock variables, based on a real stock market dataset, further stressing the belief that forcing sparsity in covariance estimates might be a favorable strategy, (ii) we compare the out-of-sample performance using different covariance estimates, based on synthetic data.

Dataset and methodology: All simulations for this application are based on daily financial data, crawled from the Yahoo Finance website[15] over the period between $01.09.2009$ and $31.08.2013$. The complete description of the dataset is given in Table 5.8. Stocks are retrieved from stock markets in the America (e.g., Dow Jones, NYSE, etc.), Europe (e.g., London Stock Exchange, Paris Stock parket, etc.), Asia (e.g., Nikkei, etc) and Africa (e.g., South Africa's stock exchange).

Table 5.8: Stock dataset description

Stock market period	Number of stocks s	Trading days d
$01.09.2009 - 31.08.2013$	2833	1038

For our experimental setup, we follow the next strategy[BDDM$^+$09]: using historical daily observations over a sliding time window of 3- or 6-month period, we compute a sparse covariance $\mathbf{\Theta}^\star$ via the proposed method, using only stock records within this period.

Results: Here, we conduct experiments to showcase: (i) possible correlations/anti-correlations and independence between stocks, induced by the estimated sparse covariance matrix, (ii) the out-of-sample performance of MVO when a sparse covariance matrix is used between the proposed portfolios and well-established strategies.

Dependencies in stocks: Based on the estimated sparse covariances, both positive and negative correlations, as well as full independece cases are reported. The top plot of Figure 5.3 shows a case of nearly independence: based on covariance estimates using the SCOPT algorithm, Coca-Cola stock behavior is uncorrelated to that of Blinx stock, an Internet Media platform service. The same holds for many pairs of stocks in the data set at hand: background knowledge on the model governing the data indicate that assets belonging to different stock sectors are more likely to be independent. On the other hand, the middle plot of Figure 5.3 shows positive correlation between the Galaxy Entertainment Group, an investment holding company in the Hong Kong Stock Exchange, and SAP enterprise software corporation.[16] Finally, we show

[15] http://finance.yahoo.com

[16] We mention that in 2006, Galaxy Entertainment Group "...chose the SAP ERP Human Capital Management and SAP ERP Financials solutions to enable its business to grow and launch new resorts and casinos with minimal impact to the business operation..." [sap].

two of the stock variables with the most negative correlation estimated during the period in September 2009 and September 2013: IP Group, a british intellectual property business company, and Petroneft resources company, a gas and oil extraction company – in this case, further underlying information might be unknown to us for understanding their negative correlation.

Figures 5.4 and 5.5 show some representative correlation estimates we observed during the period 01.09.2009 and 31.08.2013. Using the SCOPT algorithm with regularization parameters $\lambda = 0.1$ and $\rho = 1$, we solve (5.34) with tolerance $\gamma = 10^{-12}$ and $\epsilon = 10^{-10}$. Here, the sample covariance $\widehat{\mathbf{\Sigma}}$ uses all the provided data within the time period of interest. By sorting the non-diagonal elements of $\mathbf{\Theta}^\star$ and keeping the most important correlations, we obtain the infographics provided in Figures 5.4 and 5.5.

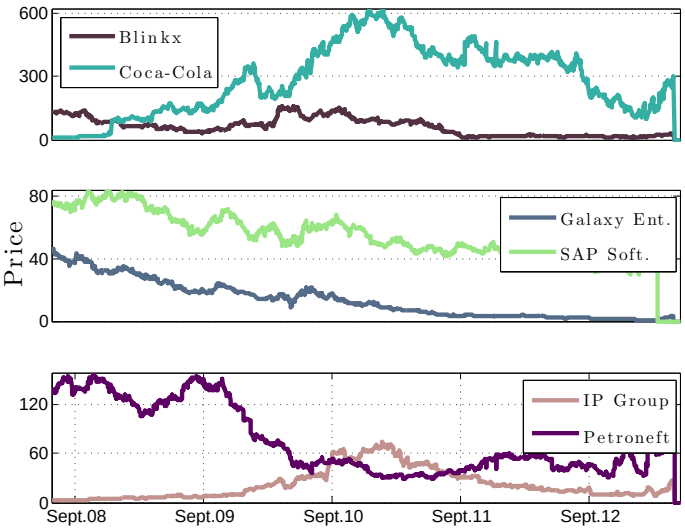


Figure 5.3: Representative stock behavior.

Out-of-sample performance with synthetic data: From the discussion above, it is apparent that both strong and weak correlations among stock assets are evident in practice. As pointed out in [HR11], the behavior of non-diagonal entries in correlation matrix estimates is such that it is not easily distinguishable whether small values indicate weak dependence between variables or estimation fluctuations, especially in the large dimension setting with small sample corpus. Under these settings, [HR11] propose that small values should be considered as zeros while only large values can be considered as good covariate estimates. Thus, assuming a sparse covariance matrix $\mathbf{\Sigma}$ in the true underlying model is sustainable.

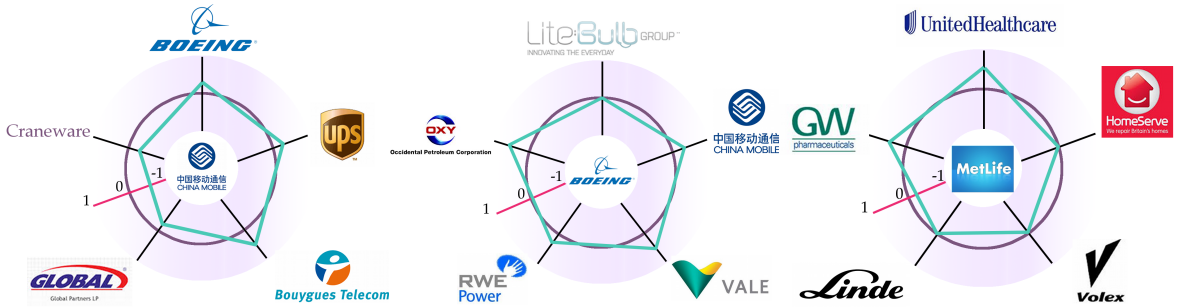
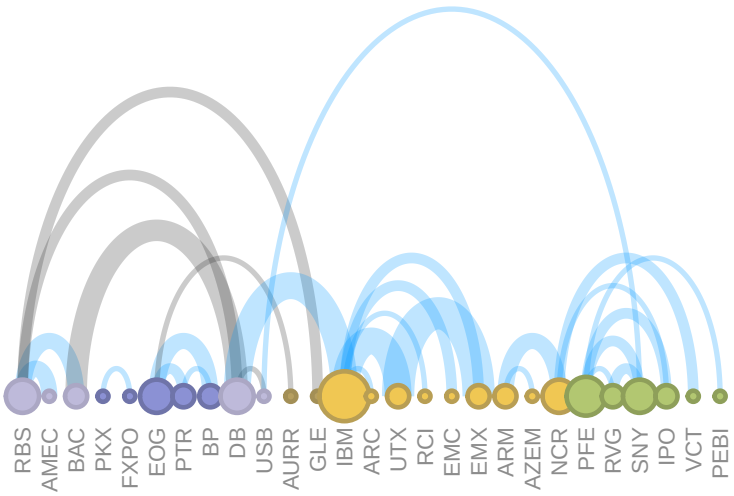


Figure 5.4: Top five (in magnitude) correlations for three stock assets: Chine mobile (left panel), Boeing (center panel) and, MetLife insurance (right panel).

To measure the performance of using a sparse covariance estimate in MVO, we assume the following synthetic case: Let $\mathbf{\Sigma} \in \mathbb{S}^n_{++}$ be synthetically generated as a Gaussian covariance matrix to represent the correlations among assets. Furthermore, assume that only k entries of $\mathbf{\Sigma}$ are "significant": we construct their absolute values to be at least two orders of magnitude larger than the rest of the entries; this assumption is partially supported by the analysis above on real datasets. In our experiments below we



Stock Abbr.	Company name
RBS	Royal Bank of Scotland
AMEC	AMEC Group
BAC	Bank of America
PKX	Posco
FXPO	Ferrexpo
EOG	EOG Resources
PTR	PetroChina
BP	BP
DB	Deutsche Bank
USB	U.S. Bank Corp.
AURR	Aurora Russia
GLE	Glencore
IBM	IBM
ARC	Arc Document
UTX	United Technologies
RCI	Rogers Communications
EMC	EMC Corporation
EMX	EMX Industries
ARM	ARM Holdings
AZEM	Azem Chemicals
NCR	NCR Electronics
PFE	Pfizer Inc.
RVG	RetroScreen Virology
SNY	Sanofi healthcare
IPO	Intellectual Property
VCT	Victrex Chemicals
PEBI	Port Erin BioFarma

Figure 5.5: We focus on three sectors: (i) bank industry (light purple), (ii) petroleum industry (dark purple), (iii) Computer science and microelectronics industry (light yellow), (iv) Pharmaceuticals/Chemistry industry (green). Any miscellaneous companies are denoted with dart yellow. Positive correlations are denoted with blue arcs; negative correlations with black arcs. The width of the arcs denotes the strength of the correlation - here, the maximum correlation (in magnitude) is 0.3934.

set $n = 500,\ 1000$ and consider a sampling time window of $N = 90,\ 180$ days (i.e., an approximately 3- and 6-month sampling period).

Given the above, both $\widehat{\mathbf{\Sigma}}$ and $\mathbf{\Theta}^\star$ are calcuated – we use our algorithm for the latter. Using these two quantities, we then solve (5.42) for $\mathbf{\Sigma} \leftarrow \widehat{\mathbf{\Sigma}}$ and $\mathbf{\Sigma} \leftarrow \mathbf{\Theta}^\star$ for various expected returns μ and record the computed minimum risk portfolios $\mathbf{w}_{\text{sample}}$ and $\mathbf{w}_{\text{SCOPT}}$, respectively. Finally, given $\mathbf{w}_{\text{sample}}$ and $\mathbf{w}_{\text{SCOPT}}$, as well as the equal-weight portfolio $\mathbf{w}_{\text{equal}} := \frac{1}{n} \cdot \mathbb{1}_{n\times 1}$, we report the risk/variances achieved by the constructed portfolios using the ground truth covariance $\mathbf{\Sigma}$.

Overall, we report lower variances $\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}$ for minimum variance portfolios when $\mathbf{\Theta}^\star$ is used, compared with the risk achieved by the equally-weighted portfolio or the sample covariance estimation. We provide some representative evaluations in Table 5.9. By using our approach, we achieve the minimum risk over all the configurations considered; of course, such approach comes with some complexity to compute $\mathbf{\Theta}^\star$ as compared to the rest of the approaches. The empirical covariance strategy with $\mathbf{w}_{\text{sample}}$ has the worst performance in terms of minimum risk achieved for most of our testings; we point out that, in this case, $\widehat{\mathbf{\Sigma}}$ is a rank-defficient positive semidefinite matrix.

Table 5.9: Summary of comparison results for reconstruction of efficiency – all strategies considered achieve the requested return μ.

Model			Risk $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$		
	λ	$\frac{k}{n^2}$ (%)	$\mathbf{w}_{\text{sample}}$	$\mathbf{w}_{\text{equal}}$	$\mathbf{w}_{\text{SCOPT}}$
$\boldsymbol{\Sigma}_3$ ($n = 500$, $N = 90$)	1.1	0.5	0.0347	0.0094	**0.0066**
	1.3	1	0.0393	0.0125	**0.0096**
	1.8	5	0.0801	0.0216	**0.0166**
	2.0	7	0.1336	0.0256	**0.0200**
	2.1	10	0.1118	0.0315	**0.0272**
	2.3	15	0.1328	0.0379	**0.0314**
	2.3	20	0.1920	0.0451	**0.0442**
	2.5	30	0.2280	**0.0559**	0.0695

Model			Risk $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$		
	λ	$\frac{k}{n^2}$ (%)	$\mathbf{w}_{\text{sample}}$	$\mathbf{w}_{\text{equal}}$	$\mathbf{w}_{\text{SCOPT}}$
$\boldsymbol{\Sigma}_3$ ($n = 500$, $N = 180$)	0.8	0.5	0.0183	0.0105	**0.0074**
	1.0	1	0.0174	0.0110	**0.0081**
	1.5	5	0.0405	0.0227	**0.0165**
	1.7	7	0.0481	0.0270	**0.0193**
	1.8	10	0.0521	0.0315	**0.0229**
	2.0	15	0.0616	0.0376	**0.0272**
	2.0	20	0.0713	0.0444	**0.0313**
	2.2	30	0.0976	0.0531	**0.0403**

Model			Risk $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$		
	λ	$\frac{k}{n^2}$ (%)	$\mathbf{w}_{\text{sample}}$	$\mathbf{w}_{\text{equal}}$	$\mathbf{w}_{\text{SCOPT}}$
$\boldsymbol{\Sigma}_3$ ($n = 1000$, $N = 90$)	1.4	0.5	0.0760	0.0065	**0.0053**
	1.7	1	0.0810	0.0078	**0.0059**
	2.3	5	0.0902	0.0158	**0.0129**
	2.7	7	0.1968	0.0188	**0.0159**
	3.0	10	0.2232	0.0223	**0.0196**
	3.8	15	0.2463	0.0267	**0.0231**
	4.5	20	0.2408	0.0307	**0.0257**
	4.5	30	0.4925	0.0375	**0.0365**

Model			Risk $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$		
	λ	$\frac{k}{n^2}$ (%)	$\mathbf{w}_{\text{sample}}$	$\mathbf{w}_{\text{equal}}$	$\mathbf{w}_{\text{SCOPT}}$
$\boldsymbol{\Sigma}_3$ ($n = 1000$, $N = 180$)	1.4	0.5	0.0223	0.0066	**0.0050**
	1.7	1	0.0233	0.0076	**0.0072**
	2.3	5	0.0513	0.0157	**0.0115**
	2.7	7	0.0529	0.0183	**0.0139**
	3.0	10	0.0706	0.0217	**0.0177**
	3.8	15	0.0876	0.0264	**0.0202**
	4.5	20	0.0872	0.0307	**0.0227**
	4.5	30	0.1075	0.0373	**0.0291**

5.5 Discussion

In this chapter, we propose a variable metric method for minimizing convex functions that are compositions of proximity functions with self-concordant smooth functions. Our framework does not rely on the usual Lipschitz gradient assumption on the smooth part for its convergence theory. A highlight of this work is the new set of analytic step-size selection and correction procedures, which are best matched to the underlying problem structures. Our empirical results illustrate that the new theory leads to significant improvements in the practical performance of the algorithmic instances when tested on a variety of different applications.

In this work, we present a convergence proof for composite minimization problems under the assumption of *exact algorithmic calculations* at each step of the methods. An interesting problem to pursue is the extension of this analysis to include *inexact calculations* and study how these errors propagate into the convergence and convergence rate guarantees [KMTDC14]. We hope this paper triggers future efforts along this direction.

We highlight three key practical contributions to numerical optimization. First, in the proximal-Newton method, our analytical step-size procedures allow us to do away with any globalization strategy (e.g., line-search). This has a significant practical impact when the evaluation of the functions is expensive. We show how to combine the analytical step-size selection with the standard backtracking or forward line-search procedures to enhance the global convergence of our method. Our analytical quadratic convergence characterization helps us adaptively switch from *damped* step-size to a *full* step-size. Second, in the proximal-gradient method setting, we establish a step-size selection and correction mechanism. The step-size selection procedure can be considered as a predictor, where existing step-size rules that leverage local information can be used. The step-size corrector then adapts the local information of the function to achieve the best theoretical decrease in the objective function. While our procedure does not require any function evaluations, we can further enhance convergence whenever we are allowed function evaluations. Finally, our framework, as we demonstrate in [TDKC13b], accommodates a path-following

strategy, which enable us to approximately solve constrained non-smooth convex minimization problems with rigorous guarantees.

As a possible application of the proposed framework we propose the following: Consider the sparse PCA problem where the data matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{maximize}} & \mathbf{x}^T \mathbf{A} \mathbf{x} \\ \text{subject to} & \|\mathbf{x}\|_0 \leq k, \;\; \|\mathbf{x}\|_2 = 1 \end{array} \tag{5.43}$$

We can transform this problem by "lifting" it to the matrix case and introducing a regularizer parameter:

$$\begin{array}{ll} \underset{\mathbf{Z}}{\text{maximize}} & \text{trace}(\mathbf{A}\mathbf{Z}) - \lambda \sqrt{\|\mathbf{Z}\|_0} \\ \text{subject to} & \mathbf{Z} \succeq 0, \; \text{trace}(\mathbf{Z}) = 1, \; \text{rank}(\mathbf{Z}) = 1 \end{array} \tag{5.44}$$

where $\mathbf{Z} := \mathbf{x}\mathbf{x}^T$. Usually $\sqrt{\|\mathbf{Z}\|_0}$ is substituted by its convex surrogate (for fixed scale) $\|\cdot\|_1$ since $\|\mathbf{Z}\|_1 \leq \sqrt{\|\mathbf{Z}\|_0}\|\mathbf{Z}\|_F = \sqrt{\|\mathbf{Z}\|_0}$. This leads to:

$$\begin{array}{ll} \underset{\mathbf{Z}}{\text{maximize}} & \text{trace}(\mathbf{A}\mathbf{Z}) - \lambda \|\mathbf{Z}\|_1 \\ \text{subject to} & \mathbf{Z} \succeq 0, \; \text{trace}(\mathbf{Z}) = 1, \; \text{rank}(\mathbf{Z}) = 1 \end{array}$$

Open question 7. *Instead of relaxing the constraints, use the problem formulation presented above to propose a projected algorithm as follows:*

$$\underset{\mathbf{Z}}{\textit{maximize}} \quad \textit{trace}(\mathbf{A}\mathbf{Z}) - \lambda \|\mathbf{Z}\|_1 \quad \textit{subject to} \quad \mathbf{Z} \in \mathcal{C}$$

where $C = \{\mathbf{X} : \mathbf{Z} \succeq 0, \; \textit{trace}(\mathbf{Z}) = 1, \; \textit{rank}(\mathbf{Z}) = 1\}$. *We know from Chapter 2 that the projection* $\mathcal{P}_{\mathcal{C}}(\mathbf{B})$*:*

$$\underset{\mathbf{Z}}{\textit{minimize}} \quad \|\mathbf{B} - \mathbf{Z}\|_F \quad \textit{subject to} \quad \mathbf{Z} \in \mathcal{C}$$

can be computed exactly. Hows does this approach work in practice?

Open question 8. *Since the* $\mathbf{Z} \succeq 0$ *constraint usually requires a eigenvalue decomposition, we use this constraint to create a self-concordant barrier function and regularize the objective. This way, we can define a path-following barrier scheme [TDKC13b] using proximal operations (without transforming the non-smooth part* $\|\cdot\|_1$ *into linear constraints):*

$$\begin{array}{ll} \underset{\mathbf{Z}}{\textit{maximize}} & t\,(\textit{trace}(\mathbf{A}\mathbf{Z}) - \lambda \|\mathbf{Z}\|_1) + \log\det(\mathbf{Z}) \\ \textit{subject to} & \textit{trace}(\mathbf{Z}) = 1, \;\; \textit{rank}(\mathbf{Z}) = 1 \end{array} \tag{5.45}$$

Appendix

We provide the detailed proofs of the theoretical results in the main text here.

Proof of Lemma 34

Since g is convex, we have

$$g(\mathbf{y}) \geq g(\mathbf{x}) + \mathbf{v}^T(\mathbf{y}-\mathbf{x}), \ \ \forall \mathbf{v} \in \partial g(\mathbf{x}).$$

By adding this inequality to (5.9) and notting that $F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}), \ \forall \mathbf{x}$, we obtain

$$\begin{aligned} F(\mathbf{y}) &\geq F(\mathbf{x}) + \left(\nabla f(\mathbf{x}) + \mathbf{v}\right)^T (\mathbf{y}-\mathbf{x}) + \omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}) \\ &\geq F(\mathbf{x}) - \lambda(\mathbf{x}) \|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} + \omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}). \end{aligned} \tag{5.46}$$

Here, the last inequality is due to the generalized Cauchy-Schwartz inequality and $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|_{\mathbf{x}}^*$. Let $\mathcal{L}_F(F(\mathbf{x})) := \{\mathbf{y} \in \mathrm{dom}(F) \mid F(\mathbf{y}) \leq F(\mathbf{x})\}$ be a sublevel set of F. Then, for any $\mathbf{y} \in \mathcal{L}_F(F(\mathbf{x}))$, we have $F(\mathbf{y}) \leq F(\mathbf{x})$ which leads to

$$\lambda(\mathbf{x}) \|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} \geq \omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}),$$

due to (5.46). Since ω is convex and strictly increasing, the equation $\lambda(\mathbf{x})t - \omega(t) = 0$ has unique solution $t^* > 0$, if $\lambda(\mathbf{x}) < 1$. Therefore, for any $0 \leq t \leq t^*$, we have $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} \leq t^*$. This implies that $\mathcal{L}_F(F(\mathbf{x}))$ is bounded. Hence, $\mathbf{x}^*$ exists due to the well-known Weierstrass theorem. The uniqueness of $\mathbf{x}^*$ follows from the monotonicity of $\omega(\cdot)$. □

Proofs of global convergence: Theorem 15

In this subsection, we provide the proof of Theorem 15. We first provide a key result quantifying the improvement of the objective as a function of the step-size α_k.

Maximum decrease of the objective function: Let $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}^k}$, $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$ and $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k$, where $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k+\beta_k^2)} \in (0,1]$. We will prove below that the following holds at each iteration of the algorithms

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega\left(\frac{\beta_k^2}{\lambda_k}\right). \tag{5.47}$$

Moreover, the choice of α_k is *optimal* (in the worse-case sense).

Proof. Indeed, since g is convex and $\alpha_k \in (0,1]$, we have $g(\mathbf{x}^{k+1}) = g\left((1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k\right) \leq (1-\alpha_k) g(\mathbf{x}^k) + \alpha_k g(\mathbf{s}^k)$, which leads to

$$g(\mathbf{x}^{k+1}) - g(\mathbf{x}^k) \leq \alpha_k (g(\mathbf{s}^k) - g(\mathbf{x}^k)). \tag{5.48}$$

Combining (5.48) with the self-concordant property (5.10) of f, we obtain

$$\begin{aligned} F(\mathbf{x}^{k+1}) &\leq F(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}^{k+1}-\mathbf{x}^k) + \omega_*\left(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}\right) + \alpha_k\left(g(\mathbf{s}^k) - g(\mathbf{x}^k)\right) \\ &\overset{(5.16)}{\leq} F(\mathbf{x}^k) + \alpha_k \nabla f(\mathbf{x}^k)^T \mathbf{d}^k + \omega_*\left(\alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}\right) + \alpha_k\left(g(\mathbf{s}^k) - g(\mathbf{x}^k)\right). \end{aligned} \tag{5.49}$$

Since $\mathbf{s}^k$ is the unique solution of (5.15), by using the optimality condition (5.17), we get

$$\begin{aligned} -\nabla f(\mathbf{x}^k) - \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) \in \partial g(\mathbf{s}^k) \Rightarrow \\ -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|^2_{\mathbf{H}_k} \in (\mathbf{s}^k - \mathbf{x}^k)^T \partial g(\mathbf{s}^k). \end{aligned} \tag{5.50}$$

Combining (5.50) with $g(\mathbf{x}^k) - g(\mathbf{s}^k) \geq \mathbf{v}^T(\mathbf{x}^k - \mathbf{s}^k),\ \mathbf{v} \in \partial(\mathbf{s}^k)$, due to the convexity of $g(\cdot)$, we have

$$g(\mathbf{s}^k) - g(\mathbf{x}^k) \leq -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|^2_{\mathbf{H}_k}. \tag{5.51}$$

Using (5.51) in (5.49) together with the definitions of β_k and λ_k, we obtain

$$F(\mathbf{x}^{k+1}) \overset{(5.16)}{\leq} F(\mathbf{x}^k) - \alpha_k\beta_k^2 + \omega_*(\alpha_k\lambda_k). \tag{5.52}$$

Let us consider the function $\varphi(\alpha) := \alpha\beta_k^2 - \omega_*(\alpha\lambda_k)$. By the definition of $\omega_*(\cdot)$, we can easily show that $\varphi(\alpha)$ attains the maximum

$$\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k + \beta_k^2)},$$

provided that $\alpha_k \in (0, 1]$. Moreover, $\varphi(\alpha_k) = \omega(\beta_k^2/\lambda_k)$, which proves (5.47). Since α_k maximizes φ over $[0, 1]$, this value is optimal. □

Since $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, we observe $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k} \equiv \|\mathbf{d}^k\|_{\mathbf{x}_k} =: \lambda_k$, where $\mathbf{d}^k \equiv \mathbf{d}_n^k$. In this case, the step size α_k in (5.47) becomes $\alpha_k = \frac{\lambda_k}{1+\lambda_k}$ which is in $(0, 1)$. Moreover, (5.47) reduces to

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k),$$

which is indeed (5.21).

Finally, we assume that, for a given $\sigma \in (0, 1)$, we have $\lambda_k \geq \sigma$ for $0 \leq k \leq k_{\max} - 1$. Since ω strictly increases, it follows from (5.21) by induction that

$$F(\mathbf{x}^*) \leq F(\mathbf{x}^k) \leq F(\mathbf{x}^0) - \sum_{j=0}^{k-1} \omega(\lambda_j) \leq F(\mathbf{x}^0) - k\omega(\sigma).$$

This estimate shows that the number of iterations to reach $\lambda_k < \sigma$ is at least $k_{\max} = \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$. □

Proofs of local convergence: Theorem 16

Optimality conditions as fixed-point formulations: Let f be a given standard self-concordant function, g be a given proper, lower semicontinuous and convex function, and $\mathbf{H}_k$ be a given symmetric positive definite matrix. Besides the two key inequalities (5.9) and (5.10), we also need the following inequality [NN94, Nes04, Theorem 4.1.6] in the proofs below:

$$(1 - \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}})^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq (1 - \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}})^{-2} \nabla^2 f(\mathbf{x}), \tag{5.53}$$

for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < 1$.

For a fixed $\bar{\mathbf{x}} \in \mathrm{dom}(F)$, where $F := f + g$, we redefined the following operators:

$$P_{\bar{\mathbf{x}}}^{g}(\mathbf{z}) := \left(\nabla^2 f(\bar{\mathbf{x}}) + \partial g\right)^{-1}(\mathbf{z}), \;\; S_{\bar{\mathbf{x}}}(\mathbf{z}) := \nabla^2 f(\bar{\mathbf{x}})\mathbf{z} - \nabla f(\mathbf{z}), \tag{5.54}$$

and

$$\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{z}) := \left(\nabla^2 f(\bar{\mathbf{x}}) - \mathbf{H}_k\right)(\mathbf{z} - \mathbf{x}^k). \tag{5.55}$$

Here, $P_{\bar{\mathbf{x}}}^{g}$ and $S_{\bar{\mathbf{x}}}$ can be considered as a generalized proximal operator of g and the gradient step of f, respectively. While $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ measures the error between $\nabla^2 f(\bar{\mathbf{x}})$ and $\mathbf{H}_k$ along the direction $z - \mathbf{x}^k$.

Next, given $\mathbf{s}^k$ is the unique solution of (5.15), we characterize the optimality condition of the original problem (5.1) and the subproblem (5.15) based on the $P_{\bar{\mathbf{x}}}^{g}$, $S_{\bar{\mathbf{x}}}$ and $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ operators. From (5.17), we have

$$S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k) \in \nabla^2 f(\bar{\mathbf{x}})\mathbf{s}^k + \partial g(\mathbf{s}^k).$$

By the definition of $P_{\bar{\mathbf{x}}}^{g}$ in (5.54), the above expression leads to

$$\mathbf{s}^k = P_{\bar{\mathbf{x}}}^{g}\left(S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k)\right). \tag{5.56}$$

By replacing $\bar{\mathbf{x}}$ with $\mathbf{x}^*$, i.e., the unique solution of (5.1), into (5.56) we obtain

$$\mathbf{s}^k = P_{\mathbf{x}^*}^{g}\left(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)\right). \tag{5.57}$$

Moreover, if we replace $\mathbf{H}_k$ by $\nabla^2 f(\mathbf{x}^*)$ in the above fixed-point expression, we finally have

$$\mathbf{x}^* = P_{\mathbf{x}^*}^{g}\left(S_{\mathbf{x}^*}(\mathbf{x}^*)\right). \tag{5.58}$$

Formulas (5.56) to (5.58) represent the fixed-point formulation of the optimality conditions.

Key estimates: Let $\mathbf{r}_k := \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ and λ_k be defined by (5.18). For any $\alpha_k \in (0, 1]$:

$$\|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2\lambda_k^2}{1 - \alpha_k\lambda_k} + \frac{2\alpha_k\lambda_k - \alpha_k^2\lambda_k^2}{(1 - \alpha_k\lambda_k)^2}\|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}, \tag{5.59}$$

$$\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\mathbf{r}_k^2}{1 - \mathbf{r}_k} + \|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*, \text{provided that } \alpha_k\lambda_k < 1 \text{ and } \mathbf{r}_k < 1. \tag{5.60}$$

Proof. First, by using the nonexpansiveness of $P_{\mathbf{x}^k}^{g}$ in Lemma (33), it follows from (5.56) that

$$\begin{aligned}
\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_{\mathbf{x}^k} &= \left\|P_{\mathbf{x}^k}^{g}(S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1})) - P_{\mathbf{x}^k}^{g}(S_{\mathbf{x}^k}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k))\right\|_{\mathbf{x}^k} \\
&\overset{(5.8)}{\leq} \left\|S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*)\right\|_{\mathbf{x}^*}^* \\
&\overset{(i)}{\leq} \left\|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k)\right\|_{\mathbf{x}^k}^* \\
&+ \left\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\right\|_{\mathbf{x}^k}^* \\
&\overset{(ii)}{=} \left\|\int_0^1 \left(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\right)(\mathbf{x}^{k+1} - \mathbf{x}^k)d\tau\right\|_{\mathbf{x}^k}^* \\
&+ \left\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\right\|_{\mathbf{x}^k}^*,
\end{aligned} \tag{5.61}$$

where (i) and (ii) are due to the triangle inequality and the mean-value theorem, respectively.

Second, we estimate the first term in (5.61). For this purpose, we define

$$\begin{aligned}\boldsymbol{\Sigma}_k &:= \int_0^1 \left(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\right) d\tau, \\ \mathbf{M}_k &:= \nabla^2 f(\mathbf{x}^k)^{-1/2} \boldsymbol{\Sigma}_k \nabla^2 f(\mathbf{x}^k)^{-1/2}.\end{aligned} \tag{5.62}$$

Based on the proof of [Nes04, Theorem 4.1.14], we can show that

$$\|\mathbf{M}_k\|_2 \leq \frac{\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}^k}}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}}.$$

Using this estimate, the definition (5.62) and noting that $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$, we obtain

$$\begin{aligned}\|\boldsymbol{\Sigma}_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\|^*_{\mathbf{x}^k} = \|\mathbf{M}_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\|_{\mathbf{x}^k} &\overset{(i)}{\leq} \|\mathbf{M}_k\|_2 \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} \\ &\leq \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}} = \frac{\alpha_k^2 \|\mathbf{d}^k\|^2_{\mathbf{x}^k}}{1 - \alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}},\end{aligned} \tag{5.63}$$

where (i) is due to the Cauchy-Schwartz inequality.

Third, we consider the second term in (5.61) for $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$. By the definition of $\mathbf{e}_{\bar{\mathbf{x}}}$, it is obvious that $\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k) = 0$. Hence, we have

$$\begin{aligned}\mathcal{T}_2 &:= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k)\right\|^*_{\mathbf{x}^k} \\ &= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})\right\|^*_{\mathbf{x}^k} \\ &= \left\|\left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right)\mathbf{d}^{k+1}\right\|^*_{\mathbf{x}^k}.\end{aligned} \tag{5.64}$$

We now define the following quantity, whose spectral norm we bound below

$$\mathbf{N}_k := \nabla^2 f(\mathbf{x}^k)^{-1/2} \left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right) \nabla^2 f(\mathbf{x}^k)^{-1/2}. \tag{5.65}$$

By applying (5.53) with $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{y} = \mathbf{x}^{k+1}$, we can bound the spectral norm of $\mathbf{N}_k$ as follows

$$\begin{aligned}\|\mathbf{N}_k\|_2 &\leq \max\left\{1 - \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^2, \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^{-2} - 1\right\} \\ &= \frac{2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k})^2}.\end{aligned} \tag{5.66}$$

Therefore, from (5.64) we can obtain the following estimate

$$\begin{aligned}(\mathcal{T}_2)^2 &= \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})^T \nabla^2 f(\mathbf{x}^k)^{-1} \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1}) \\ &= (\mathbf{d}^{k+1})^T \, \nabla^2 f(\mathbf{x}^k)^{1/2} \, \mathbf{N}_k^2 \, \nabla^2 f(\mathbf{x}^k)^{1/2} \, \mathbf{d}^{k+1} \\ &\leq \|\mathbf{N}_k\|_2^2 \, \|\mathbf{d}^{k+1}\|^2_{\mathbf{x}^k}.\end{aligned} \tag{5.67}$$

By substituting (5.66) into (5.67) and noting that $\alpha_k \mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$, we obtain

$$\mathcal{T}_2 \leq \frac{2\alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k} - \alpha_k^2 \left\|\mathbf{d}^k\right\|^2_{\mathbf{x}^k}}{(1 - \alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k})^2} \|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}. \tag{5.68}$$

Now, by substituting (5.63) and (5.68) into (5.61) and noting that $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, $\mathbf{s}^k \equiv \mathbf{s}_n^k$, $\mathbf{d}^k \equiv \mathbf{d}_n^k$ and $\lambda_k \equiv \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}$, we obtain

$$\left\|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\right\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2 \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}^2}{1 - \alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}} + \frac{2\alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k} - \alpha_k^2 \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}^2}{(1 - \alpha_k \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k})^2} \|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k}.$$

which is indeed (5.59).

Similarly to Proof of (5.61) and (5.63), we have

$$\begin{aligned}
\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} &\overset{(5.58)}{=} \left\| P_{\mathbf{x}^*}^g(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)) - P_{\mathbf{x}^*}^g(S_{\mathbf{x}^*}(\mathbf{x}^*)) \right\|_{\mathbf{x}^*} \\
&\overset{(5.8)}{\leq} \left\| S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*) \right\|_{\mathbf{x}^*}^* \\
&\leq \left\| \int_0^1 \left(\nabla^2 f(\mathbf{x}^* + \tau(\mathbf{x}^k - \mathbf{x}^*)) - \nabla^2 f(\mathbf{x}^*) \right) (\mathbf{x}^k - \mathbf{x}^*) d\tau \right\|_{\mathbf{x}^*}^* + \left\| \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) \right\|_{\mathbf{x}^*}^* \\
&\overset{(5.63)}{\leq} \frac{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}^2}{1 - \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}} + \left\| \left(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*) \right) \mathbf{d}^k \right\|_{\mathbf{x}^*}^*, \qquad (5.69)
\end{aligned}$$

which is indeed (5.60) since $\mathbf{r}_k = \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$. □

Proof of Theorem 16: Since $\mathbf{x}^k = \mathbf{s}_n^k - \mathbf{d}_n^k$ due to (5.20), we have $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}_n^k = \mathbf{s}_n^k - (1 - \alpha_k)\mathbf{d}_n^k$, which leads to

$$\mathbf{d}_n^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{x}^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1 - \alpha_k)\mathbf{d}_n^k.$$

By applying the triangle inequality to the above expression, we have

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} = \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1 - \alpha_k)\mathbf{d}_n^k\|_{\mathbf{x}^k} \leq \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} + (1 - \alpha_k)\|\mathbf{d}_n^k\|_{\mathbf{x}^k}. \qquad (5.70)$$

Substituting (5.59) into (5.70) we obtain

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2 \lambda_k^2}{1 - \alpha_k \lambda_k} + \frac{2\alpha_k \lambda_k - \alpha_k^2 \lambda_k^2}{(1 - \alpha_k \lambda_k)^2} \|\mathbf{d}^{k+1}\|_{\mathbf{x}^k} + (1 - \alpha_k)\lambda_k.$$

Rearranging this inequality we get

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \left(\frac{(1 - \alpha_k \lambda_k)\left(1 - \alpha_k + (2\alpha_k^2 - \alpha_k)\lambda_k\right)}{1 - 4\alpha_k \lambda_k + 2\alpha_k^2 \lambda_k^2} \right) \lambda_k, \qquad (5.71)$$

provided that $1 - 4\alpha_k \lambda_k + 2\alpha_k^2 \lambda_k^2 > 0$. Now, by applying (5.53) with $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{y} = \mathbf{x}^{k+1}$, one can show that

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^{k+1}} \leq \frac{\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k}}{1 - \alpha_k \|\mathbf{d}_n^k\|_{\mathbf{x}^k}}. \qquad (5.72)$$

We note that $1 - 4\alpha_k \lambda_k + 2\alpha_k^2 \lambda_k^2 > 0$ if $\alpha_k \lambda_k < 1 - 1/\sqrt{2}$. By combining (5.71) and (5.72) we obtain

$$\lambda_k \left\|\mathbf{d}_n^{k+1}\right\|_{\mathbf{x}^{k+1}} \leq \left(\frac{1 - \alpha_k + (2\alpha_k^2 - \alpha_k)\lambda_k}{1 - 4\alpha_k \lambda_k + 2\alpha_k^2 \lambda_k^2} \right) \lambda_k,$$

which is (5.22).

Next, we consider the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by dub-step proximal Newton method (5.20) with the step size $\alpha_k = (1+\lambda_k)^{-1}$. Then, (5.22) is transformed into

$$\lambda_{k+1} \leq \frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2}\lambda_k. \tag{5.73}$$

Assuming $\lambda_k \leq \bar{\sigma} := \sqrt{5}-2$, we can easily deduce that $\frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2} \leq 1$ and thus, $\lambda_{k+1} \leq \lambda_k$. By induction, if $\lambda_0 \leq \bar{\sigma}$ then, $\lambda_{k+1} \leq \lambda_k$ for all $k \geq 0$. Moreover, we have $\lambda_{k+1} \leq \frac{2}{1-2\bar{\sigma}-\bar{\sigma}^2}\lambda_k^2$, which shows that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate, which completes the proof of part b).

Now, since $\alpha_k = 1$, the estimate (5.22) reduces to $\lambda_{k+1} \leq \frac{\lambda_k^2}{1-4\lambda_k+2\lambda_k^2}$. By the same argument as in the proof of part b), we can show that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate.

Finally, we prove the last statement in Theorem 16. By substituting $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$ into (5.59), we obtain

$$\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \|(\nabla^2 f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*. \tag{5.74}$$

Let $\mathcal{T}_3 := \|(\nabla^2 f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|_{\mathbf{x}^*}^*$. Similarly to the proof of (5.68), we can show that

$$\mathcal{T}_3 \leq \left[\frac{2\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*} - \|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}^2}{(1-\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*})^2}\right]\|\mathbf{d}^k\|_{\mathbf{x}^*} \leq \alpha_k\frac{(2-\mathbf{r}_k)\mathbf{r}_k}{(1-\mathbf{r}_k)^2}(\mathbf{r}_{k+1}+\mathbf{r}_k). \tag{5.75}$$

Here the second inequality follows from the fact that $\|\mathbf{d}^k\|_{\mathbf{x}^*} = \alpha_k\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^*} \leq \alpha_k[\|\mathbf{x}^{k+1}-\mathbf{x}^*\|_{\mathbf{x}^*} + \|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}] = \alpha_k(\mathbf{r}_{k+1}+\mathbf{r}_k)$. We also have $\mathbf{r}_{k+1} = \|\mathbf{x}^{k+1}-\mathbf{x}^*\|_{\mathbf{x}^*} = \|(1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq (1-\alpha_k)\mathbf{r}_k + \alpha_k\|\mathbf{s}_k-\mathbf{x}^*\|_{\mathbf{x}^*}$. Using these inequalities, (5.75) and (5.74) we get

$$\mathbf{r}_{k+1} \leq (1-\alpha_k)\mathbf{r}_k + \alpha_k\frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \alpha_k^2\frac{(2-\mathbf{r}_k)\mathbf{r}_k}{(1-\mathbf{r}_k)^2}(\mathbf{r}_{k+1}+\mathbf{r}_k). \tag{5.76}$$

Rearranging this inequality to obtain

$$\mathbf{r}_{k+1} \leq \left(\frac{1-\alpha_k+(2\alpha_k^2+3\alpha_k-2)\mathbf{r}_k+(1-\alpha_k-\alpha_k^2)\mathbf{r}_k^2}{1-2(1+\alpha_k^2)\mathbf{r}_k+(1+\alpha_k^2)\mathbf{r}_k^2}\right)\mathbf{r}_k. \tag{5.77}$$

We consider two cases:

Case 1: $\alpha_k = 1$: We have $\mathbf{r}_{k+1} \leq \frac{3-\mathbf{r}_k}{1-4\mathbf{r}_k+2\mathbf{r}_k^2}\mathbf{r}_k^2$. Hence, if $\mathbf{r}_k < 1-1/\sqrt{2}$ then $1-4\mathbf{r}_k+2\mathbf{r}_k^2 > 0$. Moreover, $\mathbf{r}_{k+1} \leq \mathbf{r}_k$ if $3\mathbf{r}_k - \mathbf{r}_k^2 < 1-4\mathbf{r}_k+2\mathbf{r}_k^2$, which is satisfied if $\mathbf{r}_k < (7-\sqrt{37})/6 \approx 0.152873$. Now, if we assume that $\mathbf{r}_0 \leq \sigma \in (0,(7-\sqrt{37})/6)$, then, by induction, we have $\mathbf{r}_{k+1} \leq \frac{3-\sigma}{1-4\sigma+2\sigma^2}\mathbf{r}_k^2$. This shows that $\{\mathbf{r}_k\}_{k\geq 0}$ locally converges to 0^+ at a quadratic rate. Since $\mathbf{r}_k := \left\|\mathbf{x}^k-\mathbf{x}^*\right\|_{\mathbf{x}^*}$, we can conclude that $\mathbf{x}^k \to \mathbf{x}^*$ at a quadratic rate as $k \to \infty$.

Case 2: $\alpha_k = (1+\lambda_k)^{-1}$: Since $\lambda_k = \left\|\mathbf{x}^{k+1}-\mathbf{x}^k\right\|_{\mathbf{x}^k} \leq \frac{\left\|\mathbf{x}^{k+1}-\mathbf{x}^*\right\|_{\mathbf{x}^*}+\left\|\mathbf{x}^k-\mathbf{x}^*\right\|_{\mathbf{x}^*}}{1-\|\mathbf{x}^k-\mathbf{x}^*\|_{\mathbf{x}^*}} = \frac{\mathbf{r}_{k+1}+\mathbf{r}_k}{1-\mathbf{r}_k}$. We have $1-\alpha_k \leq \frac{\mathbf{r}_{k+1}+\mathbf{r}_k}{(1+\lambda_k)(1-\mathbf{r}_k)} \leq \frac{\mathbf{r}_{k+1}+\mathbf{r}_k}{1-\mathbf{r}_k}$. Substituting this into (5.76) and using the fact that $\alpha_k \leq 1$, we have

$$\mathbf{r}_{k+1} \leq \frac{(\mathbf{r}_{k+1}+\mathbf{r}_k)\mathbf{r}_k}{1-\mathbf{r}_k} + \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \frac{(2-\mathbf{r}_k)\mathbf{r}_k}{(1-\mathbf{r}_k)^2}(\mathbf{r}_{k+1}+\mathbf{r}_k).$$

Rearranging this inequality, we finally get

$$\mathbf{r}_{k+1} \leq \frac{4 - 3\mathbf{r}_k}{1 - 5\mathbf{r}_k + 3\mathbf{r}_k^2}\mathbf{r}_k^2. \tag{5.78}$$

Since $1 - 5\mathbf{r}_k + 3\mathbf{r}_k^2 > 0$ if $\mathbf{r}_k < (5 - \sqrt{13})/6$, we can see from (5.78) that $\mathbf{r}_k < (9 - \sqrt{57})/12 \approx 0.120847$ then $\mathbf{r}_{k+1} \leq \mathbf{r}_k$. By induction, if we choose $\mathbf{r}_0 \leq \bar{\sigma} \in (0, (9 - \sqrt{57})/12)$ then $\mathbf{r}_{k+1} \leq \frac{4-3\bar{\sigma}}{1-5\bar{\sigma}+3\bar{\sigma}^2}\mathbf{r}_k^2$, which shows that $\{\mathbf{r}_k\}_{k\geq 0}$ converges to 0^+ at a quadratic rate. Consequently, the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a quadratic rate. □

Conclusions

In this thesis, our key contention is that while the ambient dimension is large in many machine learning/signal processing problems, the relevant state information therein often resides in a much lower dimensional space. Such observation has led and can still guide exciting developments under different low-dimensional modeling frameworks, such as compressive sensing, matrix completion, portfolio optimization, graph model selection, image processing and, nonparametric Bayesian inference, while revealing new measurement systems, tools and methods for information extraction from low-dimensional or incomplete data.

It is our belief that real progress on high-dimensional statistics and optimization requires a coordinated effort based on combinatorial and geometric foundations that unify convex and combinatorial optimization frameworks. As a first step towards this direction, this thesis presents algorithmic solutions that benefit from both worlds: a salient feature of our approach is computational thinking to not only best leverage our current computational infrastructure but also to best exploit new developments in approximate linear algebra methods. Based on this premise, we propose scalable and accurate algorithmic frameworks (both convex and non-convex) that scale well to accommodate this data "deluge", promising substantial reductions in acquisition time, communication bandwidth, digital storage, and computational resources. Our confidence that substantial progress can be made is backed up by a great body of promising preliminary empirical results.

We believe that this research thrust demonstrates that the underlying mathematical framework extends far beyond a concrete application. We rigorously show how various seemingly different-and extremely fundamental-scientific and engineering applications, such as optimal financial portfolio design, density learning from data, etc., can be readily handled by our theoretical and algorithmic developments.

However, we identify that there are several aspects that have not been considered in this thesis. As a representative and—we think—very interesting research direction to follow is that of *combining theoretical computer science results with machine learning/signal processing problems*. Such techniques are still relatively new and not widespread, since only recently has the randomized linear algebra community achieved nearly-optimal error bounds, and there are no standard implementations yet. E.g., as shown, finding low-cost SVD or eigenvalue decomposition approximations is a challenging task. Although the randomized techniques require roughly the same arithmetic operations as the "exact" approaches, they usually reveal more degrees of freedom on the strategies that can be followed, which is essential to take advantage of modern computer architecture; e.g., can randomized techniques lead to simpler and parallel implementations in practice?

We hope that the research presented in this thesis triggers a lot of interesting questions to pursue, such as the ones shown at the end of each chapter of this manuscript.

More applications

To further highlight the potential importance of the material presented in this thesis, we mention some applications that we believe can find algorithmic solutions in the proposed schemes.

Applications in biomedical imaging and genetics: An important class of low-dimensional models is based on groups of variables that should either be selected or discarded together. These structures naturally arise in applications such as neuronal imaging [GK09b, JGM+11], gene expression inference [STM+05, OJV11] and bioinformatics [RBV08, ZSSL10]. For example, in cancer research, the groups might represent genetic pathways that constitute cellular processes. Identifying which processes lead to the development of a tumor can allow biologists to directly target certain groups of genes instead of others [STM+05]. Incorrect identification of the active/inactive groups can thus have a rather dramatic effect on the speed at which cancer therapies are developed.

In bioinformatics, we are interested in inferring the dependency network among genes: groups might be completely independent from other groups. In its simplest form, this problem boils down to the covariance estimation problem from insufficiently small amount of gene expression data, where low-dimensional modeling and reconstruction has shown to help in practice [KSB09]. However, we believe there is a lot of space for improvements upon the state-of-the-art approaches.

Applications in neuroscience: In order to understand the functioning of the human brain, it is necessary to identify and study the behavior of neuronal cell membranes under rapid change in the electric potential. However, to observe such phenomena, electrical activities on neurons need to be recorded using specialized microscopy equipment. Such low-light imaging problems have also been identified in other signal processing problems [HMW12], where the imaging data is collected by counting photons hitting a detector over time.

In this context, one wishes to accurately reconstruct the underlying phenomena under the presence of noise. As an illustrative example of how our algorithms are useful in this setting, neuronal cell membranes can be considered as the static background of the recorded phenomenon over time; such information is well approximated as a *low-rank* component. Furthermore, any time dependent electrical activity induced in-vitro can be considered as *sparse* activity. Identification and decomposition of such components can be performed using the proposed algorithms from a limited number of measurements. This way we can facilitate the interpretation of the signals in terms of the chosen structures, revealing information that could be used to better understand their properties.

Applications in quantum computing: Quantum information theory enables solutions of scientific and engineering problems, such as fast integer factorizations and database searches, that are not within the reach of our conventional technology. Realizing the full potential of the quantum computation systems is believed to be one of the most important problems of our current century.

While quantum information theory is too far in its infancy to build a quantum computer, quantum tomography measurements are being performed now [GLF+10]. The proposed research contributes in three related research themes, currently developed within this context: (i) provides scalable and approximate projection methods for quantum systems, which are important ingredients for large-scale optimization procedures, (ii) supports but also further contributes in the current theory for scalable and accurate quantum state tomography, (iii) describes low-dimensionality reducing mechanisms and algorithms, imperative for the development of efficient physical quantum tomography systems.

6 Appendix A: Mathematical prerequisites

This chapter contains the necessary background to support the developments presented in this thesis. Our intension is to provide a complete set of preliminary tools that makes reading this dissertation easy. *No attempt has been made though to connect the different pieces presented in this chapter.*

6.1 Norms, convexity and (sub)gradients

Norms: We define the ℓ_p^n-norm in n-dimensions as:

$$\|\mathbf{x}\|_p = \begin{cases} \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} & \text{if } p \in (0,\infty), \\ \max_i |x_i| & \text{if } p = \infty. \end{cases}$$

The ℓ_0 pseudo-norm is defined as: $\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|$.

The nuclear norm of a matrix $\mathbf{X} \in \mathbb{R}^{p\times n}$ with $\text{rank}(\mathbf{X}) = k$ is defined as:

$$\|\mathbf{X}\|_\star = \sum_{i=1}^k \sigma_i,$$

where σ_i represents the i-th singular value of $\mathbf{X}$.

The total-variation norm of a matrix $\mathbf{X}$ is given by:

$$\|\mathbf{X}\|_{\text{TV}} := \begin{cases} \sum_{i,j} |\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}| + |\mathbf{X}_{i+1,j} - \mathbf{X}_{i,j}| & \text{anisotropic case,} \\ \sum_{i,j} \sqrt{|\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}|^2 + |\mathbf{X}_{i+1,j} - \mathbf{X}_{i,j}|^2} & \text{isotropic case} \end{cases}$$

One can easily extend this norm to vectors by "vectorizing" properly $\mathbf{X}$.

Convexity basics: For completeness, we briefly define two important notions in optimization: *convex functions* and *convex sets*.

Definition 16. *Let $\mathcal{B} \subseteq \mathbb{R}^n$ be a subset of points in n-dimensions. Then, $\mathcal{B}$ is a convex set if and only if, $\forall \mathbf{x},\ \mathbf{y} \in \mathcal{B}$, every point on the line segment that connects $\mathbf{x}$ and $\mathbf{y}$ belongs also in $\mathcal{B}$.*

The same definition extends to matrices, etc. The following definition declares one of the many conditions that a function should satisfy such to be convex.

Definition 17. *Let $f : \mathcal{B} \to \mathbb{R}$ be a function, defined over the convex set $\mathcal{B} \subseteq \mathbb{R}^n$. Then, f is a convex function if and only if, $\forall \mathbf{x},\ \mathbf{y} \in \mathcal{B}$ and $\forall \gamma \in [0, 1]$, the following holds:*

$$f(\gamma \mathbf{x} + (1 - \gamma)\mathbf{y}) \leq \gamma f(\mathbf{x}) + (1 - \gamma) f(\mathbf{y}),$$

i.e., the line segment between any two points on the graph of f lies above the graph.

Subgradient and gradient: Given a proper, lower semicontinuous convex function f, we define its subdifferential at $\mathbf{x} \in \mathrm{dom}(f)$ as

$$\partial f(\mathbf{x}) := \left\{ \mathbf{v} \in \mathbb{R}^n \mid f(\mathbf{y}) - f(\mathbf{x}) \geq \mathbf{v}^T(\mathbf{y} - \mathbf{x}),\ \forall \mathbf{y} \in \mathrm{dom}(f) \right\}.$$

If $\partial f(\mathbf{x}) \neq \emptyset$ then each element in $\partial f(\mathbf{x})$ is called a subgradient of f at $\mathbf{x}$. In particular, if f is differentiable, we use $\nabla f(\mathbf{x})$ to denote its derivative at $\mathbf{x} \in \mathrm{dom}(f)$, and $\partial f(\mathbf{x}) \equiv \{\nabla f(\mathbf{x})\}$.

Let f be a twice differentiable, smooth function, i.e., the subdifferential $\partial f(\mathbf{x})$ is constituted only by the gradient $\nabla f(\mathbf{x})$. The Hessian matrix of f at $\mathbf{w} \in \mathrm{dom}(f)$ is computed as:

$$\nabla^2 f(\mathbf{w}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2}(\mathbf{w}) & \dfrac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{w}) & \cdots & \dfrac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{w}) \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{w}) & \dfrac{\partial^2 f}{\partial x_2^2}(\mathbf{w}) & \cdots & \dfrac{\partial^2 f}{\partial x_2 \partial x_n}(\mathbf{w}) \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{w}) & \dfrac{\partial^2 f}{\partial x_n \partial x_2}(\mathbf{w}) & \cdots & \dfrac{\partial^2 f}{\partial x_n^2}(\mathbf{w}) \end{bmatrix}.$$

6.2 Low-dimensional models

A key notion that defines low-dimensional models (LDMs) is the *sparse synthesis model*: In such model, using the appropriate collection of *atoms*, an object of interest (e.g., vector, matrix, tensor, etc.) follows an given LDM if it can be well-described as a *sparse* linear combination/superposition of atoms that "live" in the underlying LDM. Here, by "sparse linear combination/superposition" we refer to the latent degrees of freedom that the object actually has, a compared to its ambient dimension. To motivate our discussion, consider two cases: (i) the vector case with the sparsity LDM and, (ii) the matrix case with the low-rankness LDM.

In the first case, using the appropriate basis $\mathbf{\Psi} \in \mathbb{R}^{n \times n}$, an n-dimensional $\mathbf{x}$ can be well-described as a k-sparse $(k \ll n)$ linear combination of atoms $\{\boldsymbol{\psi}_i\}_{i=1}^n$ that correspond to columns of $\mathbf{\Psi}$. Typical examples of sparse-inducing bases are wavelet transform for piecewise smooth signals [Huo99], Fourier transform for smooth and periodic signals, curvelets for images with edges [CD00], etc. Similarly, in the low-rankness LDM, a $(p \times n)$-dimensional r-rank matrix $\mathbf{X}$ can be well-approximated as the r-sparse superposition of 1-rank orthogonal matrices matrices that live in the range space of $\mathbf{X}$.

The seminal work in [CDS98] is one of the first to present a unifying framework using *atoms* for the sparse synthesis model. Based on the above, [CRPW12, RRN12] propose the following mathematical

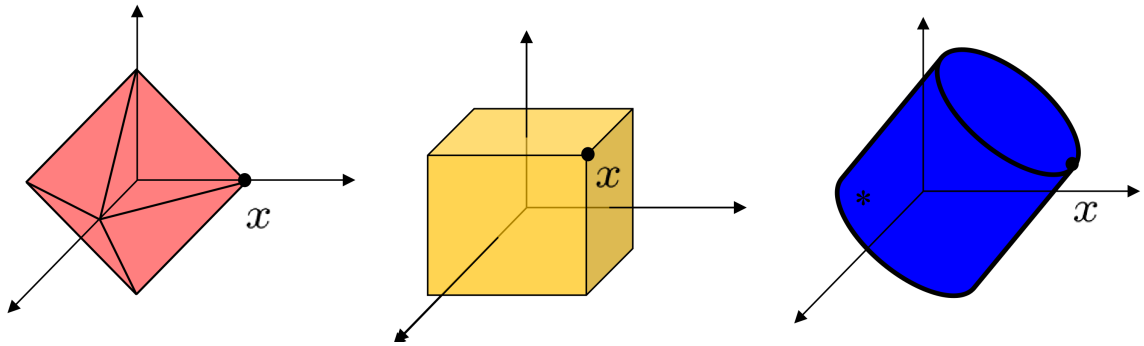


Figure 6.1: (**Left panel**): Cross-polytope as atomic norm for sparse signals (**Middle panel**): Hypercube as atomic norm for binary signals (**Right panel**): Nuclear norm ball as atomic norm for low rank matrices.

formalism: Let $\mathcal{A} := \{\mathbf{a}_1, \mathbf{a}_2, \cdots : \mathbf{a}_i \in \mathbb{R}^n, \ \forall i\}$ be the *atomic set* of signals that can be synthesized as a k-sparse positive linear combination of atoms $\mathbf{a}_i$ in $\mathcal{A}$; for example in the vector case,

$$\mathbf{x} = \sum_{i=1}^{k} c_i \mathbf{a}_i, \qquad c_i \geq 0, \ \mathbf{a}_i \in \mathcal{A} \ \text{ and } \ \|\mathbf{c}\|_0 \leq k. \tag{6.1}$$

Given an atomic set $\mathcal{A}$, we define its convex hull $\text{conv}(\mathcal{A})$ as the set of points within the convex hull of the atoms $\mathbf{a}_i \in \mathcal{A}, \ \forall i$. Given $\mathcal{A}$ and a signal $\mathbf{x} \in \mathbb{R}^n$, we define the atomic norm as:

$$\|\mathbf{x}\|_{\mathcal{A}} = \inf \left\{ \sum_{i=1}^{|\mathcal{A}|} c_i \ \Big| \ \mathbf{x} = \sum_{i=1}^{|\mathcal{A}|} c_i \mathbf{a}_i, \ c_i \geq 0, \ \forall \mathbf{a}_i \in \mathcal{A} \right\}. \tag{6.2}$$

To showcase the omnipresence of this formulation, we present a few representative examples that fall under the same "umbrella"; see Figure 6.1:

- (i) *Sparse signals*, i.e., $\mathbf{\Psi} = \mathbf{I}$. In this case, $\mathcal{A} = \{\pm \mathbf{e}_i, \ \forall i \in \mathcal{N}\}$ where $\mathbf{e}_i$ is the canonical vector in n-dimensions with 1 in the i-th position and the rest are equal to 0. The $\text{conv}(\mathcal{A})$ is the cross-polytope in n-dimensions, i.e., the closed unit ball in the ℓ_1-norm on $\mathbb{R}^n$. The atomic norm $\|\mathbf{x}\|_{\mathcal{A}}$ is the ℓ_1-norm $\|\mathbf{x}\|_1$.
- (ii) *Binary signals*. The atomic set is defined as $\mathcal{A} = \{\pm 1\}^n$ with $\text{conv}(\mathcal{A})$ the hypercube in n-dimensions. The atomic norm $\|\mathbf{x}\|_{\mathcal{A}}$ is the ℓ_∞-norm $\|\mathbf{x}\|_\infty$.
- (ii) *Low-rank signals*. The atomic set is defined as the set of 1-rank orthonormal matrices with $\text{conv}(\mathcal{A})$ the nuclear norm ball $(p \times n)$-dimensions. The atomic norm $\|\mathbf{x}\|_{\mathcal{A}}$ is the nuclear norm $\|\mathbf{X}\|_\star$.

The notion of atomic norm decomposition extends to multiway arrays: permutation matrices, etc.

6.3 Projection and proximity operations

Projection operations: Given an anchor point $\mathbf{x} \in \mathbb{R}^n$, the Euclidean distance of an arbitrary $\mathbf{w} \in \mathbb{R}^n$ to $\mathbf{x}$ is given by their ℓ_2-norm difference $\|\mathbf{x} - \mathbf{w}\|_2$. An interesting question regarding Euclidean distances is

finding point(s) $\mathbf{w}$ with the minimum Euclidean distance to $\mathbf{x}$ that satisfies(-y) additional constraints.

In this thesis, we will be mostly interested in the following abstract problem: Given a set $\mathcal{M}$ and an anchor point $\mathbf{x} \in \mathbb{R}^n$, a key operation in our subsequent discussions is the following projection problem:

$$\mathcal{P}_{\mathcal{M}}(\mathbf{x}) \in \underset{\mathbf{w}\in\mathbb{R}^n}{\arg\min} \left\{ \|\mathbf{w}-\mathbf{x}\|_2^2 \;\middle|\; \mathbf{w} \in \mathcal{M} \right\}. \tag{6.3}$$

Depending on the nature of $\mathcal{M}$, the above problem might have a closed form solution. However, in this thesis, we mostly focus on hard projection problems where $\mathcal{M}$ contains both combinatorial and convex constrants to be satisfied; more information is given in Chapter 1.

Proximity operations: Consider $g : \mathbb{R}^n \to \mathbb{R}$ is a function. We define the proximity operator of g as [CW05b, eq. (2.13)]:

$$\text{prox}_{\lambda}^{g}(\mathbf{x}) := \underset{\mathbf{w}\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|\mathbf{w}-\mathbf{x}\|_2^2 + \lambda \cdot g(\mathbf{w}) \right\}, \tag{6.4}$$

where g can be considered as a regularizer for the Euclidean distance metric with $\lambda > 0$ as the regularizer weight. One can easily observe the connection between the proximity operator in (6.4) and the projection operation in (6.3): assuming g is defined such that well-represents a predefined model[1], proximity operator regularizes the Euclidean distance objective function by incorporating the constraint of (6.3) in the objective. Moreover, it is known that we can obtain equivalent solutions by appropriately selecting λ.

6.4 Optimization basics

Existing algorithmic solutions invariably rely on two structural assumptions on the objective function that particularly stand out among many others: the *Lipschitz continuous gradient* assumption and the *strong regularity* condition.

Definition 18. *(Lipschitz gradient continuity) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex, smooth differentiable function. Then, f is a smooth Lipschitz continuous gradient function if and only if for any $\mathbf{v},\ \mathbf{w} \in dom(f)$:*

$$\|\nabla f(\mathbf{v}) - \nabla f(\mathbf{w})\|_2 \leq L\|\mathbf{v}-\mathbf{w}\|_2,$$

for some global constant $L > 0$.

Definition 19. *(Strong regularity condition) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a L-Lipschitz convex, twice differentiable function. Then, f is strongly convex if and only if:*

$$\mu\mathbf{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbf{I}, \quad \forall \mathbf{x} \in dom(f),$$

for some global constant $\mu > 0$.

In order to use the previous structures in practice, one needs efficient optimization solutions that scale up in high-dimensional settings. In our discussions next, it will be apparent that the key actors for this

[1] As we explain in the next chapter, consider the case where $g(\mathbf{x}) := \|\mathbf{x}\|_0$ is the ℓ_0-norm that well-approximates the discrete simple sparsity model.

purpose are projection and proximity operations over restricted sets that go beyond simple selection heuristics, with provable solution quality as well as low-complexity runtime/space bounds.

Projection operations faithfully follow the underlying combinatorial model but, in most cases, result in hard-to-solve or even combinatorial optimization problems. Furthermore, model misspecification often results in wildly inaccurate solutions.

Proximity operators of (often convex) model-structured functions often can only partially describe the true underlying model and might lead to "rules-of-thumb" in problem solving (e.g., how to set up the regularization parameter). However, such approaches work quite well in practice and are more robust to deviations from the model, leading to satisfactory solutions.

Here, our intention is to present an overview of the dominant approaches followed in practice. We consider the following three general optimization formulations:

- *Projection formulation:* Given a signal model $\mathcal{M}$ known a priori, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed data fidelity/loss function; e.g., in the case of linear regression with measurement matrix $\boldsymbol{\Phi}$ and measurements $\boldsymbol{y} = \boldsymbol{\Phi}\mathbf{x}^\star$, f usually represents the least-squares metric $f(\mathbf{x}) := \|\boldsymbol{y} - \boldsymbol{\Phi}\mathbf{x}\|_2^2$. In the chapters next, we consider the *projected* minimization problem:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) \quad \text{subject to} \quad \mathbf{x} \in \mathcal{M}. \tag{6.5}$$

- *Proximity formulation:* Given a signal model $\mathcal{M}$, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed data fidelity/loss function, $g : \mathbb{R}^n \to \mathbb{R}$ a closed regularization term, possibly non-smooth, that "faithfully" models $\mathcal{M}$ and $\lambda > 0$. In some cases, we use the following composite minimization formulation as solution to the problem at hand:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}). \tag{6.6}$$

- *Model-structured function minimization:* Given a signal model $\mathcal{M}$, let $g : \mathbb{R}^n \to \mathbb{R}$ be a closed regularization term, possibly non-smooth, that "faithfully" models $\mathcal{M}$. Moreover, let $f : \mathbb{R}^n \to \mathbb{R}$ be a closed data fidelity/loss function and $\sigma > 0$. Consider the following minimization problem:

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} \quad g(\mathbf{x}) \quad \text{subject to} \quad f(\mathbf{x}) \leq \sigma. \tag{6.7}$$

In most of the cases above, we assume that f is a convex, twice differential function, $\mathcal{M}$ usually describes a *non-convex* set and g represents its tightest convex function modeling $\mathcal{M}$. The above will be apparent from the context in each chapter.

6.4.1 Projected gradient descent method

Iterative "greedy" algorithms solving (6.5) greedily refine a current LDM solution, using only "local" information available at the current iteration. Most of the algorithmic solutions so far concentrate on the *projected gradient descent algorithm*: a popular method, known for its simplicity and ease of implementation. Per iteration, the total computational complexity is determined by the calculation of the gradient and the projection operation on $\mathcal{M}$ as in (6.3). The above lead to the following simple recursion:

$$\mathbf{x}^{i+1} = \mathcal{P}_{\mathcal{M}}\left(\mathbf{x}^i - \frac{\mu}{2}\nabla f(\mathbf{x}^i)\right), \tag{6.8}$$

where μ is a step size and $\mathcal{P}_{\mathcal{M}}(\cdot)$ is the projection onto the model $\mathcal{M}$.

Representative examples within the LDM framework are hard thresholding methods over simple sparse sets [BD09a, NT09a, Fou11, KC11, KPC12], as we describe in Chapters 1 and 2.

6.4.2 Proximity methods

Proximity gradient methods for (6.6) are iterative processes that rely on two key structural assumptions: (i) f has Lipschitz continuous gradient[2] (see Definition 18) and (ii) the regularizing term g is endowed with a *tractable* proximity operator. As often happens in practice, we will further focus on the convex case, where f is convex and g is proper, lower semicontinuous and possibly nonsmooth convex function.

By the Lipschitz gradient continuity and given a putative solution $\mathbf{x}^i \in \mathrm{dom}(f+g)$, one can locally approximate f around $\mathbf{x}^i$ using a quadratic function as:

$$f(\mathbf{x}) \leq Q(\mathbf{x}, \mathbf{x}^i) := f(\mathbf{x}^i) + \nabla f(\mathbf{x}^i)^T(\mathbf{x} - \mathbf{x}^i) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^i\|_2^2, \quad \forall \mathbf{x} \in \mathrm{dom}(f+g).$$

The special structure of this upper-bound allows us to consider a majorization-minimization approach: instead of solving (6.6) directly, we solve a sequence of simpler composite quadratic problems:

$$\mathbf{x}^{i+1} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ Q(\mathbf{x}, \mathbf{x}^i) + g(\mathbf{x}) \right\}. \tag{6.9}$$

In particular, we observe that (6.9) is equivalent to the following iterative *proximity* operation, similar to (6.4):

$$\mathbf{x}^{i+1} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ \frac{1}{2} \left\| \mathbf{x} - \left(\mathbf{x}^i - \frac{1}{L}\nabla f(\mathbf{x}^i) \right) \right\|_2^2 + \frac{1}{2L} g(\mathbf{x}) \right\}. \tag{6.10}$$

Here, the anchor point $\mathbf{w}$ in (6.4) is the gradient descent step: $\mathbf{w} := \mathbf{x}^i - \frac{1}{L}\nabla f(\mathbf{x}^i)$.

Instances of (6.10) have convergence rate:

$$f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}) - \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}) \right\} \leq \mathcal{O}\left(\frac{1}{T} \right),$$

where T is the total number of iterations.

Iterative algorithms can use memory to provide momentum in convergence. Based on Nesterov's optimal gradient methods [Nes83], [BT09b] proves the universality of such acceleration in the composite convex minimization case of (3.22), where $g(\mathbf{x})$ can be any convex norm with tractable proximity operator, with convergence rate:

$$f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}) - \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \lambda \cdot g(\mathbf{x}) \right\} \leq \mathcal{O}\left(\frac{1}{T^2} \right),$$

However, the resulting optimization criterion in (6.10) is more challenging when g stands for more elaborate LDMs. Within this context, [SRB11, VSBV13] present a new convergence analysis for proximity

[2] In [TDKC13a], we consider a more general class of functions with no *global* Lipschitz constant L over their domain. The description of this material is provided in Chapter 5.

(accelerated) gradient problems, under the assumption of *inexact proximity evaluations* and study how these errors propagate into the convergence rate.

An emerging direction for solving composite minimization problems of the form (6.6) is based on the proximity-Newton method [FM81]. The origins of this method can be traced back to the work of [FM81, Bon94], which relies on the concept of *strong regularity* introduced by [Rob80] for generalized equations–see Definition (19). This method has been recently studied by several authors such as [BF12, LSS12, SRB11]. The convergence analysis of these methods is encouraged by standard Newton methods and requires the strong regularity of the Hessian of f near the optimal solution (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$). In this case, we identify that the basic optimization framework above can be easily adjusted to second-order Newton gradient and quasi-Newton approaches:

$$\mathbf{x}^{i+1} \in \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2} \left\| \mathbf{x} - \left(\mathbf{x}^i - \mathbf{H}_i^{-1} \nabla f(\mathbf{x}^i) \right) \right\|_{\mathbf{H}_i}^2 + \frac{1}{2} g(\mathbf{x}) \right\}. \tag{6.11}$$

where $\mathbf{H}_i$ represents either the actual Hessian of f at $\mathbf{x}^i$ (i.e., $\nabla^2 f(\mathbf{x}^i)$) or a symmetric positive definite matrix approximating $\nabla^2 f(\mathbf{x}^i)$. Given a computationally efficient Newton direction, one can re-use the model-based proximity solutions presented in the previous subsection along with a second order *variable metric* gradient descent scheme, as presented in (6.11) [TDKC13a].

Bibliography

[ABDF10] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356, 2010.

[ABDF11] M. Afonso, J. Bioucas-Dias, and M. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *Image Processing, IEEE Transactions on*, 20(3):681–695, 2011.

[ABRS10] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[ADF11] E. Andersen, J. Dahl, and H. Friberg. Markowitz portfolio optimization using MOSEK. Technical report, MOSEK technical report: TR-2009-2, 2011.

[AGM12] K.-J. Ahn, S. Guha, and A. McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 5–14. ACM, 2012.

[AKMZ02] F. Alqallaf, K. Konis, R. Martin, and R. Zamar. Scalable robust covariance and correlation estimates for data mining. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 14–23. ACM, 2002.

[AMP+11] A. Argyriou, C. Micchelli, M. Pontil, L. Shen, and Y. Xu. Efficient first order methods for linear composite regularizers. *arXiv preprint arXiv:1104.1436*, 2011.

[AMS96] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

[And58] T. Anderson. *An introduction to multivariate statistical analysis*. Wiley, New York, 1958.

[ARS07] H. Attouch, P. Redont, and A. Soubeyran. A new class of alternating proximal minimization algorithms with costs-to-move. *SIAM Journal on Optimization*, 18(3):1061–1081, 2007.

[BA11] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In *NIPS*, pages 406–414, 2011.

[Bac10] F. Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, pages 118–126, 2010.

[BAC11] L. Briceno-Arias and P. Combettes. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim.*, 21(4):1230–1250, 2011.

[BAd10] F. Bach, S. Ahipasaoglu, and A. d'Aspremont. Convex relaxations for subset selection. *arXiv preprint arXiv:1006.3601*, 2010.

[Bar99] R. Baraniuk. Optimal tree approximation with wavelets. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 196–207. International Society for Optics and Photonics, 1999.

[BB08] P. Boufounos and R. Baraniuk. 1-bit compressive sensing. In *Information Sciences and Systems, 2008. CISS 2008. 42nd Annual Conference on*, pages 16–21. IEEE, 2008.

[BBC13] N. Bhan, L. Baldassarre, and V. Cevher. Tractability of interpretability via selection of group-sparse models. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, 2013.

[BBC14] B. Bah, L. Baldassarre, and V. Cevher. Model-based sketching and recovery with expanders. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, number EPFL-CONF-187484, 2014.

[BBCK13] L. Baldassarre, N. Bhan, V. Cevher, and A. Kyrillidis. Group-sparse model selection: Hardness and relaxations. *arXiv preprint arXiv:1303.3207*, 2013.

[BCDH10] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *Information Theory, IEEE Transactions on*, 56(4):1982–2001, 2010.

[BCG11] S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

[BCK13] S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In *10th International Conference on Sampling Theory and Applications (Sampta)*, 2013.

[BCT11] J. Blanchard, C. Cartis, and J. Tanner. Compressed sensing: How sharp is the restricted isometry property? *SIAM review*, 53(1):105–125, 2011.

[BCW10] R. Baraniuk, V. Cevher, and M. Wakin. Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of the IEEE*, 98(6):959–971, 2010.

[BD09a] T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[BD09b] T. Blumensath and M. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *Information Theory, IEEE Transactions on*, 55(4):1872–1882, 2009.

[BD10] T. Blumensath and M. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):298–309, 2010.

[BDDM+09] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272, 2009.

[BDF07] J. Bioucas-Dias and M. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *Image Processing, IEEE Transactions on*, 16(12):2992–3004, 2007.

[BDKY02] R. Baraniuk, R. DeVore, G. Kyriazis, and X. Yu. Near-best tree approximation. *Advances in Computational Mathematics*, 16(4):357–373, 2002.

[BEGd08] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

[Ben09] Y. Bengio. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[Ber82] D. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.

[Ber95] D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1995.

[BF12] S. Becker and M. Fadili. A quasi-Newton proximal splitting method. In *Proceedings of Neutral Information Processing Systems Foundation*, 2012.

[BGI+08] R. Berinde, A. Gilbert, P. Indyk, H. Karloff, and M. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 798–805. IEEE, 2008.

[BH14] A. Beck and N. Hallak. On the minimization over sparse symmetric sets. 2014.

[BJMO11] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

[BL07] J. Bennett and S. Lanning. The netflix prize. In *KDD Cup and Workshop in conjunction with KDD*, 2007.

[BL08] P. Bickel and E. Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227, 2008.

[Blu12] T. Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752–756, 2012.

[BM03] S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (series B)*, 95(2):329–357, 2003.

[BNR10] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

[Bon94] J. Bonnans. Local analysis of Newton-type methods for variational inequalities and nonlinear programming. *Appl. Math. Optim*, 29:161–186, 1994.

[Bow84] A. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[BRT09] P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of LASSO and Dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[BS09] D. Bertsimas and R. Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22, 2009.

[BT89] D. Bertsekas and J. Tsitsiklis. *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[BT09a] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[BT09b] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *Image Processing, IEEE Transactions on*, 18(11):2419–2434, 2009.

[BT11] J. Bien and R. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4):807–820, 2011.

[BT13] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[BT14] B. Bah and J. Tanner. Bounds of restricted isometry constants in extreme asymptotics: formulae for Gaussian matrices. *Linear Algebra and its Applications*, 441:88–109, 2014.

[BTN01] A. Ben-Tal and A.K. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

[BTW14] J. Blanchard, J. Tanner, and K. Wei. CGIHT: Conjugate gradient iterative hard thresholding for compressed sensing and matrix completion. 2014.

[BTWB10] F. Bunea, A. Tsybakov, M. Wegkamp, and A. Barbu. Spades and mixture models. *The Annals of Statistics*, 38(4):2525–2558, 2010.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Can06] E. Candes. Compressive sampling. In *Proceedings oh the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 1433–1452, 2006.

[CCS10] J.-F. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20:1956–1982, March 2010.

[CD00] E. Candes and D. Donoho. Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document, 2000.

[CDS98] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[CGM01] R. Coifman, F. Geshwind, and Y. Meyer. Noiselets. *Applied and Computational Harmonic Analysis*, 10(1):27–44, 2001.

[CHDB09] V. Cevher, C. Hegde, M. Duarte, and R. Baraniuk. Sparse signal recovery using Markov random fields. In *NIPS*, 2009.

[CIHB09] V. Cevher, P. Indyk, C. Hegde, and R. Baraniuk. Recovery of clustered sparse signals from compressive measurements. Technical report, DTIC Document, 2009.

[CIM11] E. Chouzenoux, J. Idier, and S. Moussaoui. A majorize–minimize strategy for subspace optimization applied to image restoration. *Image Processing, IEEE Transactions on*, 20(6):1517–1528, 2011.

[CL12] E. Candes and X. Li. Solving quadratic equations via PhaseLift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, pages 1–10, 2012.

[CLMW11] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[CM87] P. Calamai and J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39(1):93–116, 1987.

[CMBS00] T.-J. Chang, N. Meade, J. Beasley, and Y. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[CNB98] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden markov models. *Signal Processing, IEEE Transactions on*, 46(4):886–902, 1998.

[CP11] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[CPR13] E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Tech. Report.*, xx:1–22, 2013.

[CPS11] M. Cucuringu, J. Puente, and D. Shue. Model selection in undirected graphical models with the elastic net. *arXiv preprint arXiv:1111.0559*, 2011.

[CR02] S. Cotter and B. Rao. Sparse channel estimation via matching pursuit with application to equalization. *Communications, IEEE Transactions on*, 50(3):374–377, 2002.

[CR09] E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[CRPW12] V. Chandrasekaran, B. Recht, P.A. Parrilo, and A.S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12:805–849, 2012.

[CRT06] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Information Theory*, 52(2):489 – 509, February 2006.

[CST11] F. Cesarone, A. Scozzari, and F. Tardella. Portfolio selection problems in practice: a comparison between linear and quadratic optimization models. *arXiv preprint arXiv:1105.3594*, 2011.

[CT06] E. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Info. Theory*, 2006.

[CT07] E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[CT13] C. Cartis and A. Thompson. An exact tree projection algorithm for wavelets. *arXiv preprint arXiv:1304.4570*, 2013.

[CW05a] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4:1168–1200, 2005.

[CW05b] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[CW08] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21 – 30, March 2008.

[CWLX14] X. Chang, Y. Wang, R. Li, and Z. Xu. Sparse K-means with ℓ_∞/ℓ_0 penalty for high-dimensional data clustering. *arXiv preprint arXiv:1403.7890*, 2014.

[DAK00] S. Douglas, S. Amari, and S.-Y. Kung. On gradient adaptation with unit-norm constraints. *IEEE Transactions on Signal Processing*, 48(6):1843–1847, 2000.

[DDDM04] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[DDT+08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):83–91, 2008.

[DE03] D. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[dEGJL04] A. d'Aspremont, L. El Ghaoui, M. Jordan, and G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. In *NIPS*, volume 16, pages 41–48, 2004.

[Dem72] A. Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972.

[DFK+04] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1):9–33, 2004.

[DGNU09] V. DeMiguel, L. Garlappi, F. Nogales, and R. Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812, 2009.

[DGU09] V. DeMiguel, L. Garlappi, and R. Uppal. Optimal versus naive diversification: How inefficient is the $1/n$ portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953, 2009.

[DHMS13] A. Dalalyan, M. Hebiri, K. Meziani, and J. Salmon. Learning heteroscedastic models by convex programming under group sparsity. *Proc. of the International conference on Machine Learning*, pages 1–8, 2013.

[Dir14] S. Dirksen. Dimensionality reduction with subgaussian matrices: a unified theory. *arXiv preprint arXiv:1402.3973*, 2014.

[DKM06] P. Drineas, R. Kannan, and M. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM J. Comput.*, 36:158–183, July 2006.

[DM09] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *Information Theory, IEEE Transactions on*, 55(5):2230–2249, 2009.

[Don06] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[DRVW06] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, pages 1117–1126, New York, NY, USA, 2006. ACM.

[DSSSC08] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[DT05] D. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9452–9457, 2005.

[DV06] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(042), 2006.

[DVR08] J. Dahl, L. Vandenberghe, and V. Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods & Software*, 23(4):501–520, 2008.

[EB92] J. Eckstein and D. Bertsekas. On the Douglas - Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55:293–318, 1992.

[EHJT04] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[EM09] Y.C. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *Information Theory, IEEE Transactions on*, 55(11):5302–5316, 2009.

[EV03] C. Estan and G. Varghese. New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. *ACM Transactions on Computer Systems (TOCS)*, 21(3):270–313, 2003.

[Fac09] International Neuroinformatics Coordinating Faculty. Spike time prediction - challenge c, 2009.

[FBD10] M. Figueiredo and J. Bioucas-Dias. Restoration of Poissonian images using alternating direction optimization. *Image Processing, IEEE Transactions on*, 19(12):3133–3145, 2010.

[FGLE12] S. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: Error bounds, sample complexity and efficient estimators. *New Journal of Physics*, 14(9):095022, 2012.

[FH11] P. Ferreira and R. Higgins. The establishment of sampling as a scientific principle-a striking case of multiple discovery. *Notices of the AMS*, 58(10), 2011.

[FHT08] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, 2008.

[FHT10] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group LASSO and a sparse group LASSO. *arXiv preprint arXiv:1001.0736*, 2010.

[FM81] M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8):989–1000, 1981.

[FNW07] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):586–597, 2007.

[Fou11] S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[Fou12] S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77. Springer, 2012.

[FP03] F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 1-2. Springer-Verlag, 2003.

[FRP10] M. Fazel, B. Recht, and P. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[GB11] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, April 2011.

[GBY06] M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006.

[GFO06] M. Gomez, C. Flores, and M. Osorio. Hybrid search for cardinality constrained portfolio optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1865–1866. ACM, 2006.

[GJY11] D. Ge, X. Jiang, and Y. Ye. A note on the complexity of ℓ_p minimization. *Mathematical programming*, 129(2):285–299, 2011.

[GK02] W Gerstner and W. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[GK09a] R. Garg and R. Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 337–344. ACM, 2009.

[GK09b] A. Gramfort and M. Kowalski. Improving M/EEG source localizationwith an inter-condition sparse prior. In *IEEE International Symposium on Biomedical Imaging*, 2009.

[GLF+10] D. Gross, Y.-K. Liu, S. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Physical review letters*, 105(15):150401, 2010.

[GM11] D. Goldfarb and S. Ma. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Found. Comput. Math.*, 11(2):183–210, 2011.

[GM12] D. Goldfarb and S. Ma. Fast alternating linearization methods of minimization of the sum of two convex functions. *Math. Program., Ser. A*, pages 1–34, 2012.

[GM13] A. Gittens and M. Mahoney. Revisiting the Nystrom method for improved large-scale machine learning. In *Proceedings of The 30th International Conference on Machine Learning*, pages 567–575, 2013.

[GO09] T. Goldstein and S. Osher. The split Bregman method for ℓ_1-pegularized problems. *SIAM J. Imaging Sciences*, 2(2):323–343, 2009.

[Gol64] A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70(5):709–710, 1964.

[GOS12] T. Goldstein, B. O'Donoghue, and S. Setzer. Fast alternating direction optimization methods. Tech. report., Department of Mathematics, University of California, Los Angeles, USA, May 2012.

[GR02] M. Gutknecht and S. Röllin. The Chebyshev iteration revisited. *Parallel Computing*, 28(2):263–283, 2002.

[GR12] D. Guillot and B. Rajaratnam. Retaining positive definiteness in thresholded matrices. *Linear Algebra and its Applications*, 436(11):4143–4160, 2012.

[GR13] D. Guillot and B. Rajaratnam. Functions preserving positive definiteness for sparse matrices. *arXiv preprint arXiv:1210.3894*, 2013.

[HBL11] J. He, L. Balzano, and J. Lui. Online robust subspace tracking from partial information. *arXiv:1109.3827*, 2011.

[HC09] L. He and L. Carin. Exploiting structure in wavelet-based bayesian compressive sensing. *Signal Processing, IEEE Transactions on*, 57(9):3488–3497, 2009.

[HDC09] C. Hegde, M. Duarte, and V. Cevher. Compressive sensing recovery of spike trains using a structured sparsity model. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[HJ90] R. Horn and C. Johnson. *Matrix analysis*. Cambridge Univ. Press, 1990.

[HMT11] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53:217–288, May 2011.

[HMW12] Z. Harmany, R. Marcia, and R. Willett. This is SPIRAL-TAP: Sparse poisson intensity reconstruction algorithms – theory and practice,. *IEEE Transactions on Image Processing*, Submitted:1–13, 2012.

[HR11] A. Hero and B. Rajaratnam. Large-scale correlation screening. *Journal of the American Statistical Association*, 106(496):1540–1552, 2011.

[HS52] M. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

[HS09] M. Herman and T. Strohmer. High-resolution radar via compressed sensing. *Signal Processing, IEEE Transactions on*, 57(6):2275–2284, 2009.

[HSDR11] C. Hsieh, M. Sustik, I. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in Neutral Information Processing Systems (NIPS)*, 24:1–18, 2011.

[Huo99] X. Huo. *Sparse image representation via combined transforms*. PhD thesis, stanford university, 1999.

[HYZ08] E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[HZ10] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

[HZM11] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[Ind07] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at*, 2007.

[IR13] P. Indyk and I. Razenshteyn. On model-based RIP-1 matrices. In *Automata, Languages, and Programming*, pages 564–575. Springer, 2013.

[Izh03] E. Izhikevich. Simple model of spiking neurons. *IEEE Transactions on neural networks*, 14(6):1569–1572, 2003.

[JAB11] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.

[JGM+11] R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, F. Bach, and B. Thirion. Multi-scale mining of fmri data with hierarchical structured sparsity. In *Pattern Recognition in NeuroImaging (PRNI)*, 2011.

[JMD10] P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, volume 23, pages 937–945, 2010.

[JMOB11] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Reasearch*, 12:2297–2334, 2011.

[JNS13] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 665–674. ACM, 2013.

[Joh01] I. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Ann. Statist*, 29(2):295–327, 2001.

[Jol05] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[JOV09] L. Jacob, G. Obozinski, and J.-P. Vert. Group LASSO with overlap and graph LASSO. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440. ACM, 2009.

[KBCK13] A. Kyrillidis, S. Becker, V. Cevher, and C. Koch. Sparse projections onto the simplex. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 235–243, 2013.

[KC11] A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In *4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, number EPFL-CONF-183059, 2011.

[KC12a] A. Kyrillidis and V. Cevher. Combinatorial selection and least absolute shrinkage via the CLASH algorithm. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 2216–2220. Ieee, 2012.

[KC12b] A. Kyrillidis and V. Cevher. Matrix ALPS: Accelerated low rank and sparse matrix reconstruction. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 185–188. Ieee, 2012.

[KC13] A. Kyrillidis and V. Cevher. Fast proximal algorithms for self-concordant function minimization with application to sparse graph selection. *Proc. of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, 2013.

[KC14] A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. *Journal of Mathematical Imaging and Vision*, 48(2):235–265, 2014.

[Kim95] D. Kim. *Least squares mixture decomposition estimation*. PhD thesis, 1995.

[KMO10] R. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. on Information Theory*, 56(6):2980–2998, 2010.

[KMTDC14] A. Kyrillidis, R. Karimi Mahabadi, Q. Tran-Dinh, and V. Cevher. Scalable sparse covariance estimation via self-concordance. In *Proc. of the 28th AAAI International Conference on Artificial Intelligence (AAAI-14)*, pages 1–9. 2014.

[KP10] J. Kim and H. Park. Fast active-set-type algorithms for ℓ_1-regularized linear regression. In *Proceedings of the 13th International Conference on Artificial Intelligience and Statistics (AISTATS)*, volume 9, pages 397–404, Sardinia, Italy, 2010.

[KPC12] A. Kyrillidis, G. Puy, and V. Cevher. Hard thresholding with norm constraints. In *2012 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*, pages 3645–3648. IEEE, 2012.

[KSB09] N. Krämer, J. Schäfer, and A. Boulesteix. Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC bioinformatics*, 10(1):384, 2009.

[KVZ14] A. Kyrillidis, M. Vlachos, and A. Zouzias. Approximate matrix multiplication with application to linear embeddings. *arXiv preprint arXiv:1403.7683*, 2014.

[KX10] S. Kim and E. Xing. Tree-guided group LASSO for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 543–550, 2010.

[LÖ4] J. Löefberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[Lar] R. M. Larsen. PROPACK: Software for large and sparse SVD calculations. http://soi.stanford.edu/řmunk/PROPACK.

[Lau12] S. Laue. A hybrid algorithm for convex semidefinite optimization. In *ICML*, 2012.

[LB10] K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. *IEEE Transactions on Information Theory*, 56(9):4402–4416, 2010.

[LCM10] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[LDP07] M. Lustig, D. Donoho, and J. Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.

[LHA+07] E. Lein, M. Hawrylycz, N. Ao, M. Ayres, A. Bensinger, A. Bernard, A. Boe, M. Boguski, K. Brockway, and E. Byrnes. Genome-wide atlas of gene expression in the adult mouse brain. *Nature*, 445(7124):168–176, 2007.

[Liu11] Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In *NIPS*, pages 1638–1646, 2011.

[LP66] E. Levitin and B. Polyak. Constrained minimization methods. *USSR Computational mathematics and mathematical physics*, 6(5):1–50, 1966.

[LRS+10] J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. Tropp. Practical large-scale optimization for max-norm regularization. In *NIPS*, pages 1297–1305, 2010.

[LSDP06] M. Lustig, J. Santos, D. Donoho, and J. Pauly. *kt*-SPARSE: High frame rate dynamic MRI exploiting spatio-temporal sparsity. In *Proceedings of the 13th Annual Meeting of ISMRM, Seattle*, volume 2420, 2006.

[LSS12] J.D. Lee, Y. Sun, and M.A. Saunders. Proximal newton-type methods for convex optimization. *Tech. Report.*, pages 1–25, 2012.

[LT13] R. Luss and M. Teboulle. Conditional gradient algorithms for rank-one matrix approximations with a sparsity constraint. *SIAM Review*, 55(1):65–98, 2013.

[Lu10] Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

[LY09] J. Liu and J. Ye. Efficient Euclidean projections in linear time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 657–664. ACM, 2009.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[Mal99] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[Mar52] H. Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952. Wiley Library.

[ME11] M. Mishali and Y. Eldar. Sub-nyquist sampling. *Signal Processing Magazine, IEEE*, 28(6):98–124, 2011.

[MF81] H. Mine and M. Fukushima. A minimization method for the sum of a convex function and a continuously differentiable function. *J. Optim. Theory Appl.*, 33:9–23, 1981.

[MGV+11] V. Michel, A. Gramfort, G. Varoquaux, E. Eger, and B. Thirion. Total variation regularization for fMRI-based prediction of behavior. *Medical Imaging, IEEE Transactions on*, 30(7):1328 –1340, july 2011.

[MJD10] R. Meka, P. Jain, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, 2010.

[Mor62] J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899, 1962.

[Mut05] S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

[MVVR10] S. Mosci, S. Villa, A. Verri, and L. Rosasco. A primal-dual algorithm for group ℓ_1 regularization with overlapping groups. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[Nat95] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[NC10] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

[Nes83] Y. Nesterov. A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[Nes04] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[Nes05a] Y. Nesterov. Excessive gap technique in non-smooth convex minimization. *SIAM J. Optim.*, 16(1):235–249, 2005.

[Nes05b] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005.

[Nes11] Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.

[Nes13] Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2013.

[Ng04] A. Ng. Feature selection, ℓ_1 vs. ℓ_2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[NN94] Y. Nesterov and A. Nemirovski. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial Mathematics, 1994.

[NT97] Y. Nesterov and M. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Research*, 22(1):1–42, 1997.

[NT09a] D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[NT09b] A. Nemirovski and M. Todd. Interior-point methods for optimization. *Acta Numerica*, pages 191–234, 2009.

[NW88] G. Nemhauser and L. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[NW06] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[NWF78] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions — I. *Mathematical Programming*, 14(1):265–294, 1978.

[OB12] G. Obozinski and F. Bach. Convex relaxation for combinatorial penalties. *arXiv preprint arXiv:1205.1240*, 2012.

[OJV11] G. Obozinski, L. Jacob, and J.P. Vert. Group LASSO with overlaps: The latent group LASSO approach. *arXiv preprint arXiv:1110.0413*, 2011.

[OONR12] P. Olsen, F. Oztoprak, J. Nocedal, and S. Rennie. Newton-like methods for sparse inverse covariance estimation. *Optimization Online*, 2012.

[Par62] E. Parzen. On estimation of a probability density function and mode. *Annals of mathematical statistics*, 33(3):1065–1076, 1962.

[PEGC12] M. Pilanci, L. El Ghaoui, and V. Chandrasekaran. Recovery of sparse probability measures via convex programming. In *NIPS*, pages 2429–2437, 2012.

[PKB14] D. Papapailiopoulos, A. Kyrillidis, and C. Boutsidis. Provable deterministic leverage score sampling. *In 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2014.

[PMG+12] G. Puy, J. Marques, R. Gruetter, J. Thiran, D. Van De Ville, P. Vandergheynst, and Y. Wiaux. Spread spectrum magnetic resonance imaging. *Medical Imaging, IEEE Transactions on*, 31(3):586–598, 2012.

[Pol12] I. Pollak. Covariance estimation and related problems in portfolio optimization. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th*, pages 369–372. IEEE, 2012.

[Pri11] E. Price. Efficient sketches for the set query problem. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 41–56. SIAM, 2011.

[RBV08] F. Rapaport, E. Barillot, and J.P. Vert. Classification of arraycgh data using fused svm. *Bioinformatics*, 24(13):i375–i382, 2008.

[RFP10] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[RLZ09] A. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

[RNWK11] N. Rao, R. Nowak, S. Wright, and N. Kingsbury. Convex approaches to model wavelet sparsity patterns. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1917–1920, 2011.

[Rob80] S. M. Robinson. Strongly Regular Generalized Equations. *Mathematics of Operations Research, Vol. 5, No. 1 (Feb., 1980), pp. 43-62*, 5:43–62, 1980.

[Roc70] R. Rockafellar. *Convex analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

[Rot12] A. Rothman. Positive definite estimators of large covariance matrices. *Biometrika*, 99(3):733–740, 2012.

[RR13] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.

[RRG+12] B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pages 1583–1591, 2012.

[RRN12] N. Rao, B. Recht, and R. Nowak. Signal recovery in unions of subspaces with applications to compressive imaging. *arXiv preprint arXiv:1209.3079*, 2012.

[RSS+00] G. Rätsch, B. Schölkopf, A. Smola, S. Mika, T. Onoda, and K. Müller. Robust ensemble learning for data mining. In *PAKDD*, pages 341–344, 2000.

[RSV08] H. Rauhut, K. Schnass, and P. Vandergheynst. Compressed sensing and redundant dictionaries. *Information Theory, IEEE Transactions on*, 54(5):2210–2219, 2008.

[Rud82] M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, pages 65–78, 1982.

[RWRY11] P. Ravikumar, M. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electron. J. Statist.*, 5:935–988, 2011.

[RZMC11] M. Rohrdanz, W. Zheng, M. Maggioni, and C. Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of chemical physics*, 134(12):124116, 2011.

[sap] Galaxy entertainment group: Enabling rapid growth with SAP ERP HCM and SAP ERP financials. http://download.sap.com/.

[SBdG12] N. Städler, P. Bülmann, and S. Van de Geer. ℓ_1-penalization for mixture regression models. *Tech. Report.*, pages 1–35, 2012.

[Sha49] C. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[Sha93] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12):3445–3462, 1993.

[SMG10] K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. *arXiv preprint arXiv:1011.0097*, 2010.

[SPH09] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. *Signal Processing, IEEE Transactions on*, 57(8):3075–3085, 2009.

[SR09] K. Scheinberg and I. Rish. SINCO-a greedy coordinate ascent method for sparse inverse covariance selection problem. *preprint*, 2009.

[SRB11] M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS, Granada, Spain*, 2011.

[SS05] J. Schäfer and K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.

[Ste56] H. Steinhaus. Sur la division des corp materiels en parties. *Bulletin L' Académie Polonaise des Science*, 1:801–804, 1956.

[STM+05] A. Subramanian, P. Tamayo, V. Mootha, S. Mukherjee, B. Ebert, M. Gillette, A. Paulovich, S. Pomeroy, T. Golub, and E. Lander. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[SXH08] M. Stojnic, W. Xu, and B. Hassibi. Compressed sensing-probabilistic analysis of a null-space characterization. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3377–3380. IEEE, 2008.

[TC12a] H. Tyagi and V. Cevher. Active learning of multi-index function models. In *Advances in Neural Information Processing Systems 25*, pages 1475–1483, 2012.

[TC12b] H. Tyagi and V. Cevher. Learning ridge functions with randomized sampling in high dimensions. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2025–2028. IEEE, 2012.

[TDC14] Q. Tran-Dinh and V. Cevher. An optimal primal-dual decomposition framework. Technical report, LIONS - EPFL, 2014.

[TDKC13a] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *arXiv preprint arXiv:1308.2867*, 2013.

[TDKC13b] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. A path following method for composite self-concordant barrier minimization. Technical report, LIONS, EPFL, 2013.

[TDKC13c] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. A proximal newton framework for composite minimization: Graph learning without cholesky decompositions and matrix inversions. *JMLR W&CP*, 28(2):271–279, 2013.

[TDNSD13] Q. Tran-Dinh, I. Necoara, C. Savorgnan, and M. Diehl. An inexact perturbed path-following method for Lagrangian decomposition in large-scale separable convex optimization. *SIAM J. Optim.*, 23(1):95–125, 2013.

[TDSD13] Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Compt. Optim. Appl.*, 55(1):75–211, 2013.

[TG07] J Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the LASSO. *J. Royal. Statist. Soc B*, 58(1):267–288, 1996.

[TLD+10] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *Information Theory, IEEE Transactions on*, 56(1):520–544, 2010.

[TSR+05] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused LASSO. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[TW13] J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. *SIAM Journal on Scientific Computing*, 35(5):S104–S125, 2013.

[VDBF08] E. Van Den Berg and M. Friedlander. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[VFK] M. Vlachos, N. Freris, and A. Kyrillidis. Compressive mining: Fast and optimal data mining in the compressed domain.

[VGPT10] G. Varoquaux, A. Gramfort, J.-B. Poline, and B. Thirion. Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in Neural Information Processing Systems (NIPS)*, volume 23, pages 2334–2342. 2010.

[VRMV12] S. Villa, L. Rosasco, S. Mosci, and A. Verri. Proximal methods for the latent group LASSO penalty. *Computational Optimization and Applications*, pages 1–27, 2012.

[VSBV13] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

[Wai09] M. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (LASSO). *Information Theory, IEEE Transactions on*, 55(5):2183–2202, 2009.

[Wan12] H. Wang. Two new algorithms for solving covariance graphical LASSO based on coordinate descent and ECM. *arXiv preprint arXiv:1205.4120*, 2012.

[War09] R. Ward. Compressed sensing with cross validation. *Information Theory, IEEE Transactions on*, 55(12):5773–5782, 2009.

[WJ63] J. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[WN99] SJ Wright and J Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[WNF09] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

[WPSB08] A. Wagadarikar, N. Pitsianis, X. Sun, and D. Brady. Spectral image estimation for coded aperture snapshot spectral imagers. In *Optical Engineering+ Applications*. International Society for Optics and Photonics, 2008.

[WSB11] A. Waters, A. Sankaranarayanan, and R. Baraniuk. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. In *NIPS*, pages 1089–1097, 2011.

[WT10] D. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490), 2010.

[WYZ12] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Prog. Comp.*, 4:333–361, 2012.

[XMZ12] L. Xue, S. Ma, and H. Zou. Positive definite ℓ_1 penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 2012.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[YLY11] L. Yuan, J. Liu, and J. Ye. Efficient methods for overlapping group LASSO. In *NIPS*, volume 35, pages 352–360, 2011.

[Yua12] X. Yuan. Alternating direction method for covariance selection models. *Journal of Scientific Computing*, 51(2):261–273, 2012.

[ZH05] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[ZJH10] Y. Zhou, R. Jin, and S. Hoi. Exclusive LASSO for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010.

[ZRY09] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.

[ZSSL10] H. Zhou, M. Sehl, J. Sinsheimer, and K. Lange. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, 26(19):2375, 2010.

[ZSY13] Z. Zhang, Y. Shi, and B. Yin. MR images reconstruction based on TV-group sparse model. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[ZT11] T. Zhou and D. Tao. GODEC: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 33–40, 2011.

[ZY06] P. Zhao and B. Yu. On model selection consistency of LASSO. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.