Structured Tensor Recovery and Decomposition

Cun Mu

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Structured Tensor Recovery and Decomposition

Cun Mu

Tensors, a.k.a. multi-dimensional arrays, arise naturally when modeling higher-order objects and relations. Among ubiquitous applications including image processing, collaborative filtering, demand forecasting and higher-order statistics, there are two recurring themes in general: tensor recovery and tensor decomposition. The first one aims to recover the underlying tensor from incomplete information; the second one is to study a variety of tensor decompositions to represent the array more concisely and moreover to capture the salient characteristics of the underlying data. Both topics are respectively addressed in this thesis.

Chapter 2 and Chapter 3 focus on low-rank tensor recovery (LRTR) from both theoretical and algorithmic perspectives. In Chapter 2, we first provide a negative result to the sum of nuclear norms (SNN) model—an existing convex model widely used for LRTR; then we propose a novel convex model and prove this new model is better than the SNN model in terms of the number of measurements required to recover the underlying low-rank tensor. In Chapter 3, we first build up the connection between robust low-rank tensor recovery and the compressive principle component pursuit (CPCP), a convex model for robust low-rank matrix recovery. Then we focus on developing convergent and scalable optimization methods to solve the CPCP problem. In specific, our convergent method, proposed by combining classical ideas from Frank-Wolfe and proximal methods, achieves scalability with linear per-iteration cost.

Chapter 4 generalizes the successive rank-one approximation (SROA) scheme for matrix eigen-decomposition to a special class of tensors called symmetric and orthogonally decomposable (SOD) tensor. We prove that the SROA scheme can robustly recover the symmetric canonical decomposition of the underlying SOD tensor even in the presence of noise. Perturbation bounds, which can be regarded as a higher-order generalization of the Davis-Kahan theorem, are provided in terms of the noise magnitude.

Table of Contents

List of Figures

Notation

$\mathbb{R}^n$	n-dimensional real space
$\boldsymbol{x}$	bold small letters as vectors
x_i	the i-th entry of vector $\boldsymbol{x}$
$\boldsymbol{e}_i$	the i-th standard basis
$\|\boldsymbol{x}\|_p$	p-norm of the vector $\boldsymbol{x}$
$\|\boldsymbol{x}\|$	ℓ_2-norm of the vector $\boldsymbol{x}$
$\boldsymbol{X}$	bold capital letters as matrices
$X_{i:}$	i-th row of $\boldsymbol{X}$ as column vector
$X_{:j}$	j-th column of $\boldsymbol{X}$ as column vector
X_{ij}	the (i,j)-entry of the matrix $\boldsymbol{X}$
$\|\boldsymbol{X}\|$	matrix operator norm
$\|\boldsymbol{X}\|_F$	matrix Frobenius norm
$\|\boldsymbol{X}\|_*$	matrix nuclear norm
$\mathrm{rank}(\boldsymbol{X})$	rank of a matrix
$\mathrm{null}\,(\boldsymbol{X})$	nullspace of a matrix
$\otimes$	outer product
$\bigotimes_{j=1}^{K} \mathbb{R}^{i_j}$	$\mathbb{R}^{i_1 \times i_2 \times \cdots \times i_K}$
$\bigotimes^{K} \mathbb{R}^{n}$	$\mathbb{R}^{\overbrace{n \times n \times \cdots \times n}^{K\ times}}$
$\boldsymbol{\mathcal{X}}$	bold Euler script capital letters as tensors
$\mathcal{X}_{i_1,i_2,\ldots,i_K}$	the $(i_1, i_2, \ldots, i_K)$-entry of the tensor $\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{X}}_{(k)}$	mode-k matricization
$\boldsymbol{\mathcal{X}}_{(\mathcal{R},\mathcal{C})}$	mode-$(\mathcal{R},\mathcal{C})$ matricization
$\|\boldsymbol{\mathcal{X}}\|_F$	tensor Frobenius norm
$\|\boldsymbol{\mathcal{X}}\|$	tensor operator norm

$\mathrm{rank}_{\mathrm{cp}}(\boldsymbol{\mathcal{X}})$	tensor CP rank
$\mathrm{rank}_{\mathrm{tc}}(\boldsymbol{\mathcal{X}})$	tensor Tucker rank
$\delta(\mathcal{C})$	statistical dimension of convex cone $\mathcal{C}$
$\mathrm{proj}_{\mathcal{S}}[\boldsymbol{x}]$	projection of $\boldsymbol{x}$ onto the set $\mathcal{S}$
$(\cdot)^\top$	transposition without conjugation
$(\cdot)^*$	conjugate transposition, equivalent to $(\cdot)^\top$ for real vectors/matrices
$[k]$	the integer set $\{1, \ldots, k\}$
$X \sim \mathcal{L}$	random variable X distributed by the law $\mathcal{L}$
$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$	standard Gaussian distribution in $\mathbb{R}^n$
$\mathrm{Ber}(\theta)$	standard Bernoulli distribution with parameter θ
$X \sim_{i.i.d.} \mathcal{L}$	elements in (vector- or matrix-valued) X independent, identically distributed by the law $\mathcal{L}$
w.h.p.	short for "with high probability"
i.i.d.	short for "independent, identically distributed"
w.l.o.g.	short for "without loss of generality"
w.r.t.	short for "with respect to"

Acknowledgments

I would like to give my foremost thanks to my advisor, Professor Donald Goldfarb. It is truly a wonderful experience working with him. His great sense of humor, enthusiasm in research and inspirational words of wisdom have not only made my PhD journey exceptionally enriching and enjoyable, but will also be my most valuable life assets.

I am especially grateful to my co-advisor, Professor John Wright, who sheds the light to me on the exciting and possible interaction between fundamental theoretical progress and practical impact. His high intellectual standard, intrepid attitudes towards research difficulties and thoughtful care over the students set me a perfect role model for being a researcher, an educator and a responsible person.

I would also express special thanks to my collaborator, Professor Daniel Hsu, who also taught me the class Advanced Machine Learning. His expertise in the area of tensor and great patience on working out details have substantially helped with the thesis preparation.

I would like to give my sincere gratitude to my other dissertation committee members, Professor Garud Iyengar and Professor Daniel Bienstock. I have been greatly influenced by their courses: Convex Optimization by Garud and Integer Programming by Dan. These are the very first optimization courses that inspired my interests in the field, and set me foundations for research thereafter.

I am fortunate to have had great colleagues and friends throughout my Columbian years. During the past six years at Columbia, I have met so many good friends, which become my integral part of life. They have made my life more colorful and the office always feel like home. I have enjoyed close-knit friendship with my fellow Columbia friends: Chen Chen, Ningyuan Chen, Xingyun Chen, Yupeng Chen, Zhengyu Chen, Antoine Desir, Jing Dong, Itai Feigebaum, Jing Guo, Wenqi Hu, Bo Huang, Jingfei Jia, Henry Kuo, Yenson Lau, Anran Li, Fengpei Li, Juan Li, Xin Li, Zhige Li, Yan Liu, Zhipeng Liu, Yin Brian Lu, Carlos Lopez, Wei Liu, Zhe Liu, Lijian Lu, Ni Ma, Shiqian Ma, Gonzalo Munoz, Yanan Pei, Zhiwei Tony Qin, Qing Qu, Ju Sun, Yunjie Sun, Chun Wang, Xinshang Wang, Zheng Wang, Ji Wen, Zaiwen Wen, Linan Yang, Shuoguang Yang, Chun Ye, Wotao Yin, Chaoxu Zhou, Fan Zhang, Xiaopei Zhang, Yuqian Zhang, ...

Last but not least, my heartfelt gratitude goes to my family, particularly my parents, Weimin Mu and Xiuli Xu, and my wife, Hao Cher Han. Their unconditional love, trust and support always grant me the courage to

conquer the tough times in life.

Cun Mu

Apr 21, 2017

New York

To my parents and wife

Chapter 1

Introduction

Multidimensional arrays, a.k.a. tensors, generalize vectors (i.e. one-dimensional arrays) and matrices (i.e. two-dimensional arrays). They arise naturally as a flexible and integral approach to data representation and modelling, especially in problems where the underlying objects are multi-dimensional with entries indexed by several continuous and discrete variables. For instance, in collaborative filtering [KABO10] and demand forecasting [LX10, HQB15], historical ratings and sales data are often organized with indices in user ID, product ID and contextual variables including time, location and so on; in computer vision and graphics [LMWY09], visual data are naturally indexed by the specifications in space, frequency channel, time point, etc.; in statistics, higher order moments and cumulants for multivariate distributions are tensors with equal length indexed by the variables along each dimension [McC87]. Across ubiquitous tensor applications over different areas, there are two recurring challenges in general. The first one, known as *tensor recovery*, is to recover the underlying tensor from incomplete information. For example, the ultimate goal of collaborative filtering is to figure out the missing ratings from the sparsely observed ones and thus make more precise and personalized recommendations to customers. The second challenge is on how to extract useful information from these multidimensional data, which normally relies on various *tensor decompositions* to provide a concise representation of the original tensor and moreover to capture the salient features of the underlying data.

Both challenges, *tensor recovery* and *tensor decomposition*, are respectively addressed in this thesis. In specific, Chapter 2 and Chapter 3, based on our previous works [MHWG14] and [MZWG16], focus on tensor recovery from both theoretical and algorithmic aspects, and Chapter 4, based on our previous work [MHG15], discusses topics in tensor decomposition. In the remaining part of this chapter, the nomenclature used in the thesis will be established following an overview of each chapter.

1.1 Overview

In Chapter 2, we focus on recovering low-rank tensors from incomplete information, which is a recurring problem in signal processing and machine learning. The most popular convex relaxation of this problem minimizes the *sum of the nuclear norms (SNN)* of the unfolding matrices of the tensor. We show that this approach can be *substantially suboptimal*: reliably recovering a K-way $n \times n \times \cdots \times n$ tensor of Tucker rank $(r, r, \ldots, r)$ from Gaussian measurements requires $\Omega(rn^{K-1})$ observations. In contrast, a certain (intractable) nonconvex formulation needs only $O(r^K + nrK)$ observations. We introduce a *simple and new convex relaxation*, which partially bridges this gap. Our new formulation succeeds with $O(r^{\lfloor K/2 \rfloor} n^{\lceil K/2 \rceil})$ observations. The *lower bound* for the SNN model follows from our new result on *recovering signals with multiple structures* (e.g. sparse, low rank), which indicates the significant suboptimality of the common approach of *minimizing the sum of individual sparsity inducing norms* (e.g. ℓ_1, nuclear norm). Our new tractable formulation for low-rank tensor recovery shows how the sample complexity can be reduced by designing convex regularizers that exploit several structures jointly.

Chapter 3 is more about an algorithmic exploration. We first build up the connection between the robust low-rank tensor recovery problem and the robust low-rank matrix recovery problem, and then focus on developing scalable optimization methods to solve the latter problem. Recovering matrices from compressive and grossly corrupted observations is a fundamental problem in robust statistics, with rich applications in computer vision and machine learning. In theory, under certain conditions, this problem can be solved in polynomial time via a natural convex relaxation, known as *Compressive Principal Component Pursuit (CPCP)*. However, many existing provably convergent algorithms for CPCP suffer from *superlinear per-iteration cost*, which severely limits their applicability to large-scale problems. In this chapter, we propose provably convergent, scalable and practical methods to solve CPCP with *linear per-iteration cost*. Our method combines classical ideas from *Frank-Wolfe* and *proximal methods*. In each iteration, we exploit Frank-Wolfe to *update the low-rank component with rank-one SVD* and *exploit a proximal gradient step for the sparse term*. Convergence results and implementation details are discussed. We also demonstrate the practicability and scalability of our approach with numerical experiments on visual data.

In Chapter 4, we study a particular tensor decomposition with a wide range of applications in signal processing, machine learning and statistics. In specific, many idealized problems in higher-order statistical estimation [McC87], independent component analysis [Com94, CJ10] and parameter estimation for latent variable models [AGH+14] can be reduced to the problem of finding the symmetric canonical decomposition

of an underlying *symmetric and orthogonally decomposable (SOD)* tensor. Drawing inspiration from the matrix case, the *successive rank-one approximations (SROA)* scheme has been proposed and shown to yield this tensor decomposition exactly, and a plethora of numerical methods have thus been developed for the tensor rank-one approximation problem. In practice, however, the inevitable errors—e.g., from estimation, computation, and modeling, necessitate that the input tensor can only be assumed to be a nearly SOD tensor—i.e., a symmetric tensor slightly perturbed from the underlying SOD tensor. Chapter 4 proves that even in the presence of perturbation, SROA can still robustly recover the symmetric canonical decomposition of the underlying tensor. It is shown that when the perturbation error is small enough, the approximation errors do not accumulate with the iteration number. Numerical results are presented to support the theoretical findings.

1.2 Notations and Preliminaries

The notations, used throughout the thesis, are largely borrowed from [Kie00, Lim05, KB09].

The *order* of a tensor is referred to as the number of dimensions, also known as *modes* or *ways*. Some trivial examples of tensors are scalars, vectors and matrices. *Scalars* (tensors of order zero) are denoted by lowercase letters, e.g., x. *Vectors* (tensors of order one) are denoted by boldface lowercase letters, e.g., $\boldsymbol{x}$. Matrices (tensor of order two) are denoted by boldface capital letters, e.g., $\boldsymbol{X}$. *High-order tensors* (order three or higher) are denoted by boldface Euler script letters, e.g., $\boldsymbol{\mathcal{X}}$.

For a tensor $\boldsymbol{\mathcal{X}}$ of order K, its $(i_1, i_2, \ldots, i_K)$-th entry is denoted as $\mathcal{X}_{i_1,i_2,\ldots,i_K}$. The i-th entry of a vector $\boldsymbol{x}$ is denoted as x_i, the (i,j)-th entry of a matrix $\boldsymbol{X}$ is denoted as X_{ij}.

A *fiber* of a tensor $\boldsymbol{\mathcal{X}}$ is a column vector defined by fixing each index of $\boldsymbol{\mathcal{X}}$ except one. The i-th column of a matrix $\boldsymbol{X}$, denoted by $X_{:i}$, is a mode-1 fiber, and the i-th row of $\boldsymbol{X}$, denoted by $X_{i:}$, is a mode-2 fiber, where a colon adapted from many numerical computing languages, e.g. MATLAB, is commonly used to indicate all elements of one particular mode. Third-order tensors have *column, row and tube fibers*, respectively, denoted as $\mathcal{X}_{:jk}$, $\mathcal{X}_{i:k}$ and $\mathcal{X}_{ij:}$, which by convention are all considered as column vectors when extracted from $\boldsymbol{\mathcal{X}}$.

A *slice* of a tensor $\boldsymbol{\mathcal{X}}$ is a two-dimensional section defined by fixing all indices except two ones. A third-order tensor has *horizontal, lateral and frontal slices*, respectively, denoted as $\mathcal{X}_{i::}$, $\mathcal{X}_{:j:}$ and $\mathcal{X}_{::k}$.

The set of K-way $I_1 \times I_2 \times \cdots \times I_K$ tensor, $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, in short, is denoted by $\bigotimes_{j=1}^{K} \mathbb{R}^{I_j}$. For any tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \bigotimes_{j=1}^{K} \mathbb{R}^{I_j}$, their inner product is defined as the sum of all the element-wise products, i.e.

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle := \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \mathcal{X}_{i_1 i_2 \cdots i_K} \mathcal{Y}_{i_1 i_2 \cdots i_K}. \tag{1.2.1}$$

Tensor as multilinear map. In addition to being considered as a multi-way array, a tensor $\boldsymbol{\mathcal{X}} \in \bigotimes_{j=1}^{K} \mathbb{R}^{I_j}$ can also be interpreted as a *multilinear map* in the following sense: for any matrices $\boldsymbol{V}_i \in \mathbb{R}^{I_i \times m_i}$ for $i \in [K]$, we interpret $\boldsymbol{\mathcal{X}}(\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_K)$ as a tensor in $\mathbb{R}^{m_1 \times m_2 \times \cdots \times m_p}$ whose $(i_1, i_2, \ldots, i_K)$-th entry is

$$\Big(\boldsymbol{\mathcal{X}}\left(\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_p\right)\Big)_{i_1, i_2, \ldots, i_K} := \sum_{j_i \in [I_1]} \sum_{j_2 \in [I_2]} \cdots \sum_{j_K \in [I_K]} \mathcal{X}_{j_1, j_2, \ldots, j_p} (V_1)_{j_1 i_1} (V_2)_{j_2 i_2} \cdots (V_K)_{j_K i_K}. \tag{1.2.2}$$

This multilinear interpretation is a powerful tool to conceptually simplify and visualize the notion of tensor, and will be frequently exploited throughout the thesis.

Example 1.1 *To better understand this interpretation, we provide several examples below.*

▷ $K = 2$ *(namely, $\boldsymbol{\mathcal{X}}$ is a matrix of size n_1 by n_2):*

$$\boldsymbol{\mathcal{X}}(\boldsymbol{V}_1, \boldsymbol{V}_2) = \boldsymbol{V}_1^\top \boldsymbol{\mathcal{X}} \boldsymbol{V}_2 \in \mathbb{R}^{m_1 \times m_2}. \tag{1.2.3}$$

▷ *Each entry of the tensor can also be expressed as the scalar returned by applying the multilinear map defined by the tensor on standard basis vectors correspondingly. In specific, for any $i_1 \in [I_1], i_2 \in [I_2], \ldots,$ and $i_K \in [I_K]$,*

$$\boldsymbol{\mathcal{X}}(\boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \ldots, \boldsymbol{e}_{i_K}) = \mathcal{X}_{i_1, i_2, \ldots, i_K}, \tag{1.2.4}$$

where $\boldsymbol{e}_i$ denotes the i-th standard basis.

▷ $i_1 = i_2 = \cdots = i_K = n$ *and* $\boldsymbol{V}_i = \boldsymbol{x} \in \mathbb{R}^n$ *for all* $i \in [K]$*:*

$$\boldsymbol{\mathcal{X}} \boldsymbol{x}^{\otimes K} := \boldsymbol{\mathcal{X}}(\underbrace{\boldsymbol{x}, \boldsymbol{x}, \ldots, \boldsymbol{x}}_{K \text{ times}}) = \sum_{i_1, i_2, \ldots, i_K \in [n]} \mathcal{X}_{i_1, i_2, \ldots, i_K} x_{i_1} x_{i_2} \cdots x_{i_K}, \tag{1.2.5}$$

which defines a homogeneous polynomial of degree K.

Tensor norms. Two tensor norms will be frequently visited in this thesis. For a tensor $\boldsymbol{\mathcal{X}} \in \bigotimes_{j=1}^{K} \mathbb{R}^{I_j}$, its *Frobenius norm* is defined as

$$\|\boldsymbol{\mathcal{X}}\|_F := \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \mathcal{X}_{i_1 i_2 \cdots i_K}^2}; \tag{1.2.6}$$

and its *operator norm* is defined as

$$\|\boldsymbol{\mathcal{X}}\| := \max_{\|\boldsymbol{x}_i\|=1} \boldsymbol{\mathcal{X}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K). \tag{1.2.7}$$

Matricization. *Matricization*, also known as *unfolding* or *flattening*, is the procedure of rearranging the elements from a tensor into a matrix. This can be a useful trick to simplify the problem from the tensor domain to the matrix one, which might be well studied in the literature. Admittedly, there are tons of ways to put the elements of a multi-way array into a matrix. We are particularly interested in matricizations that can preserve certain algebraic structures of the original tensor. Several ones most relevant to the thesis are described below.

The *mode-k matricization* of a tensor $\boldsymbol{\mathcal{X}} \in \bigotimes_{i\in[K]} \mathbb{R}^{I_i}$ yields a matrix denoted by $\boldsymbol{\mathcal{X}}_{(k)} \in \mathbb{R}^{I_k \times \Pi_{i\neq k} I_i}$, whose columns are the mode-k fibers arranged via certain lexicographical order of the indices except for the k-th index. More rigorously, the $(i_1, i_2, \ldots, i_K)$-th element of $\boldsymbol{\mathcal{X}}$ is mapped to the (i_n, j)-th element in $\boldsymbol{\mathcal{X}}_{(k)}$, where

$$j = 1 + \sum_{k\neq l\in[K]} \left[\left(i_l - 1 \right) \cdot \prod_{n\neq l'\in[l-1]} I_{l'} \right]. \tag{1.2.8}$$

There are also multiple ways we can stack the tensor into a vector. In this thesis, we specifically define

$$\text{vec}\left(\boldsymbol{\mathcal{X}}\right) := \text{vec}\left(\boldsymbol{\mathcal{X}}_{(1)}\right). \tag{1.2.9}$$

Example 1.2 *Consider a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{3\times4\times2}$, whose frontal slices are*

$$\boldsymbol{\mathcal{X}}_{::1} = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \quad \textit{and} \quad \boldsymbol{\mathcal{X}}_{::2} = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}. \tag{1.2.10}$$

Then we can matricize $\boldsymbol{\mathcal{X}}$ along the first, the second and the third modes respectively, which yields

$$\boldsymbol{\mathcal{X}}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}, \tag{1.2.11}$$

$$\boldsymbol{\mathcal{X}}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix}, \tag{1.2.12}$$

$$\boldsymbol{\mathcal{X}}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & \cdots & 22 & 23 & 24 \end{bmatrix}. \tag{1.2.13}$$

Vectorization will yield a column vector in $\mathbb{R}^{24}$*:*

$$\mathrm{vec}(\boldsymbol{\mathcal{X}}) = \mathrm{vec}(\boldsymbol{\mathcal{X}}_{(1)}) = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}^\top. \tag{1.2.14}$$

The mode-k matricization can be considered as we partition the set of modes $[K] = \{1, 2, \cdots, K\}$ into $\{k\}$ and $[K]/\{k\}$. Once thinking about mode-k matricization in this direction, it is natural to enrich the class of matricization by considering more general partitions over the set $[K]$.

Let the ordered sets $\mathcal{R} = \{r_1, r_2, \ldots, r_L\}$ and $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ be a partitioning of $[K] = \{1, 2, \ldots, K\}$. The matricized tensor induced by this partition $\{\mathcal{R}, \mathcal{C}\}$ is a matrix denoted by $\boldsymbol{\mathcal{X}}_{(\mathcal{R}\times\mathcal{C})} \in \mathbb{R}^{J\times K}$ where

$$J = \prod_{k\in\mathcal{R}} I_k, \quad K = \prod_{k\in\mathcal{C}} I_k, \quad \text{and}$$

the $(i_1, i_2, \ldots, i_K)$-th element of $\boldsymbol{\mathcal{X}}$ is mapped to the (i, j)-th entry in this mode-$(\mathcal{R}, \mathcal{C})$ matricization $\boldsymbol{\mathcal{X}}_{\mathcal{R}\times\mathcal{C}}$ with

$$i = 1 + \sum_{l\in[L]} \left[\left(i_{r_L} - 1 \right) \cdot \prod_{l'\in[l-1]} I_{r_{l'}} \right], \tag{1.2.15}$$

$$j = 1 + \sum_{m\in[M]} \left[\left(i_{r_m} - 1 \right) \cdot \prod_{m'\in[m-1]} I_{r_{m'}} \right]. \tag{1.2.16}$$

Remark 1.3 *For the mode-k matricization* $\boldsymbol{X}_{(k)}$*, we can regard it as the matricized tensor induced by* $\mathcal{R} = \{k\}$ *and* $\mathcal{C} = \{1, 2, \ldots, k-1, k+1, \ldots, K\}$*, i.e.*

$$\boldsymbol{\mathcal{X}}_{(k)} = \boldsymbol{\mathcal{X}}_{(\{k\}\times\{1,2,\ldots,k-1,k+1,\ldots,K\})}. \tag{1.2.17}$$

For the vectorization $\mathrm{vec}(\boldsymbol{\mathcal{X}})$*, conventionally, we can define it as the matricized tensor induced by* $\mathcal{R} = [K]$ *and* $\mathcal{C} = \emptyset$*, i.e.* $vec(\boldsymbol{\mathcal{X}}) = \boldsymbol{\mathcal{X}}_{([K]\times\emptyset)}$.

This more general treatment of matricization is first formally introduced by Kolda [Kol06]. The concept might not be much useful and appreciated for tensors of order three. In contrast, it will provide substantially more freedom in the choice of tensor flattening once the fourth dimension and beyond come on the stage, and is frequently revisited recently in different contexts including the low-rank tensor recovery [MHWG14,

JMZ15], and the semidefinite programming approach to relax the best tensor rank-one approximation [JMZ14, NW14, HJLW16].

Example 1.4 *Consider a four-way tensor* $\mathcal{X} \in \mathbb{R}^{2\times 2\times 2\times 2}$ *with elements specified as*

$$\mathcal{X}_{::11} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad \mathcal{X}_{::21} = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}, \quad \mathcal{X}_{::12} = \begin{bmatrix} 9 & 11 \\ 10 & 12 \end{bmatrix}, \quad \mathcal{X}_{::22} = \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix}. \tag{1.2.18}$$

All the mode-k unfoldings will yield a matrix in $\mathbb{R}^{2\times 8}$*, e.g.,*

$$\mathcal{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \end{bmatrix}. \tag{1.2.19}$$

However, the matricized tensor induced by $\mathcal{R} = \{1, 2\}$ *and* $\mathcal{C} = \{3, 4\}$ *yields more balanced square matrix:*

$$\mathcal{X}_{(\{1,2\}\times\{3,4\})} = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{1.2.20}$$

A sharp observation may lead to the following property that

$$\mathcal{X}_{(\{1,2\}\times\{3,4\})} = \mathsf{reshape}(\mathcal{X}_{(1)}, 4, 4).$$

It turns out the above equality holds more universally in tensor unfolding.

MATLAB implementations for matricization. The folding and unfolding procedures discussed above can be implemented in surprisingly simple MATLAB instructions. The code below is adapted from Kolda [Kol06].

```
size = [2,2,2,2];
X = reshape(1:16, size); % the four-way tensor in Example 1.4

% mode-1 matricization
R = [1]; C = [2,3,4];
J = prod(size(R)); K = prod(size(C));
Y = reshape(permute(X,[R,C]), J,K); % mode-1 unfolding
Z = ipermute(reshape(Y,[size(R), size(C)]), [R C]); % convert back to the original tensor
```

```
9
10 % mode-{R,C} matricization
11 R = [1,2]; C = [3,4];
12 J = prod(size(R)); K = prod(size(C));
13 Y = reshape(permute(X,[R,C]), J,K); % matricized tensor induced by R and C
14 Z = ipermute(reshape(Y,[size(R), size(C)]), [R C]); % convert back to the original tensor
```

Tensor-Matrix Multiplication. The *mode-k matrix product* between a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{J \times I_k}$, denoted by $\boldsymbol{\mathcal{X}} \times_k \boldsymbol{U}$, returns a new tensor of size $I_1 \times I_2 \times \cdots \times I_{k-1} \times J \times I_{k+1} \times I_{k+2} \times \cdots \times I_N$, with elements specified as

$$(\boldsymbol{\mathcal{X}} \times_k \boldsymbol{U})_{i_1 \cdots i_{k-1} j i_{k+1} \cdots i_K} := \sum_{i_n \in I_k} \mathcal{X}_{i_1 i_2 \cdots i_K} U_{j i_n}. \tag{1.2.21}$$

Two equivalent definitions, using mulilinear map and mode-k matricization, are also available and may provide more insights into this tensor-matrix multiplication.

First, it can be verified by checking definition that

$$\boldsymbol{\mathcal{X}} \times_k \boldsymbol{U} = \boldsymbol{\mathcal{X}}\Big(\underbrace{\boldsymbol{I}, \cdots, \boldsymbol{I}}_{k-1 \text{ times}}, \boldsymbol{U}^\top, \underbrace{\boldsymbol{I}, \cdots, \boldsymbol{I}}_{n-k \text{ times}}\Big). \tag{1.2.22}$$

Moreover, the k-mode matrix product can be considered as each mode-k fiber is multiplied by the matrix $\boldsymbol{U}$, which can be precisely expressed as

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_k \boldsymbol{U} \iff \boldsymbol{\mathcal{Y}}_{(k)} = \boldsymbol{U}\boldsymbol{\mathcal{X}}_{(k)}. \tag{1.2.23}$$

Example 1.5 *Consider the product along the first mode between the tensor $\boldsymbol{\mathcal{X}}$ in Example (1.2) and*

$$\boldsymbol{U} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}. \tag{1.2.24}$$

Then the frontal slices of $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U} \in \mathbb{R}^{2 \times 4 \times 2}$ are

$$\boldsymbol{\mathcal{Y}}_{::1} = \boldsymbol{U}\boldsymbol{\mathcal{X}}_{::1} = \begin{bmatrix} 22 & 49 & 76 & 103 \\ 28 & 64 & 100 & 136 \end{bmatrix} \quad \textit{and} \quad \boldsymbol{\mathcal{Y}}_{::2} = \boldsymbol{U}\boldsymbol{\mathcal{X}}_{::2} = \begin{bmatrix} 130 & 157 & 184 & 211 \\ 172 & 208 & 244 & 280 \end{bmatrix}. \tag{1.2.25}$$

For a series of tensor-matrix multiplication, the following property is most useful and relevant with our

discussions in later chapters:

$$\boldsymbol{\mathcal{X}} \times_m \boldsymbol{A} \times_n \boldsymbol{B} = \begin{cases} \boldsymbol{\mathcal{X}} \times_n \boldsymbol{B} \times_m \boldsymbol{A} & \text{if } \ m \neq n \\ \boldsymbol{\mathcal{X}} \times_n (\boldsymbol{B}\boldsymbol{A}) & \text{if } \ m = n. \end{cases} \tag{1.2.26}$$

Literally, $\times_k$ is commutative when they are applied on different modes. But when they are applied on the same mode, $\times_k$ is no longer commutative as matrix multiplication is not commutative.

Rank-one tensors. The *vector outer product* is denoted by the symbol $\otimes$. The outer product of K vectors, $\{\boldsymbol{v}_i\}_{i\in[K]} \in \bigtimes_{k=1}^{K} \mathbb{R}^{I_k}$ is defined as

$$(\boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \cdots \otimes \boldsymbol{v}_K)_{i_1 i_2 \cdots i_K} := (v_1)_{i_1}(v_2)_{i_2} \cdots (v_K)_{i_K} \qquad \forall\ i_k \in [I_k] \ \text{ and } \ k \in [K]. \tag{1.2.27}$$

This means that each element of the tensor is the product of vector elements at the corresponding entries. The definition (1.2.27) also extrapolates the concept of rank-one matrix to rank-one tensor. A K-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is rank one if there exists $\{\boldsymbol{v}_i\}_{i\in[K]} \in \bigtimes_{k=1}^{K} \mathbb{R}^{I_k}$ such that $\boldsymbol{\mathcal{X}}$ can be expressed as $\boldsymbol{v}_1 \otimes \boldsymbol{v}_2 \otimes \cdots \otimes \boldsymbol{v}_K$. Rank-one tensors play fundamental roles in tensor decomposition where different ways to express the tensor into the sum of rank-one tensors are pursued.

Symmetric tensors. A tensor is called *cubical* if it has the same size along each mode. The set of real order-K cubical tensors with dimension n along each mode is denoted by $\bigotimes^K \mathbb{R}^n$. A cubical tensor $\boldsymbol{\mathcal{X}} \in \bigotimes^K \mathbb{R}^n$ is called *symmetric* if its entries are invariant under any permutation of their indices: for any $i_1, i_2, \ldots, i_K \in [n]$:

$$\mathcal{X}_{i_1 i_2 \ldots i_K} = \mathcal{X}_{i_{\pi(1)} i_{\pi(2)} \ldots i_{\pi(K)}} \tag{1.2.28}$$

for any permutation mapping π on $[K]$. This definition naturally extends the concept of symmetry from matrices to tensors of higher order, and will be the mathematical object of our main focus in Chapter 4.

A three-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n\times n\times n}$, for example, is symmetric if

$$\boldsymbol{\mathcal{X}}_{ijk} = \boldsymbol{\mathcal{X}}_{ikj} = \boldsymbol{\mathcal{X}}_{jik} = \boldsymbol{\mathcal{X}}_{jki} = \boldsymbol{\mathcal{X}}_{kij} = \boldsymbol{\mathcal{X}}_{kji}, \qquad \forall\ i, j, k \in [n]. \tag{1.2.29}$$

A tensor $\boldsymbol{\mathcal{X}} \in \bigotimes^K \mathbb{R}^n$ is *diagonal* if $\boldsymbol{\mathcal{X}}_{i_1 i_2 \cdots i_K}$ is zero unless $i_1 = i_2 = \cdots = i_K$. Literally speaking, this describes a tensor with nonzero elements possible only on the superdiagonal entries, which is a higher order analogue of diagonal matrices. Clearly, a diagonal tensor is symmetric.

Then rank-one tensor

$$\boldsymbol{v}^{\otimes K} := \underbrace{\boldsymbol{v} \otimes \boldsymbol{v} \otimes \cdots \otimes \boldsymbol{v}}_{K \text{ times}} \in \bigotimes^{K} \mathbb{R}^n$$

is another commonly encountered example of symmetric tensors. The converse is also true that a symmetric tensor $\boldsymbol{\mathcal{X}}$ of rank-one can always be written as $\boldsymbol{\mathcal{X}} = \boldsymbol{v}^{\otimes K}$.

When the tensor $\boldsymbol{\mathcal{X}} \in \bigotimes^{K} \mathbb{R}^n$ is symmetric, its operator norm originally defined as

$$\|\boldsymbol{\mathcal{X}}\| \quad = \max_{\|\boldsymbol{x}_i\|=1} \boldsymbol{\mathcal{X}}(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K) \tag{1.2.30}$$

can be simplified by restricting $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_K$ without loss of generality (see, e.g., [CHLZ12, ZLQ12]), namely

$$\|\boldsymbol{\mathcal{X}}\| \quad = \quad \max_{\|\boldsymbol{x}\|=1} \left|\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K}\right| \quad = \quad \max_{x_1^2+x_2^2+\cdots+x_n^2=1} \left| \sum_{i_1 \in [n]} \sum_{i_2 \in [n]} \cdots \sum_{i_K \in [n]} \mathcal{X}_{i_1 i_2 \cdots i_K} x_{i_1} x_{i_2} \cdots x_{i_K} \right|. \tag{1.2.31}$$

The definition of symmetric tensors also immediately yield the following property:

$$\mathrm{vec}\Big(\boldsymbol{\mathcal{X}}\big(\boldsymbol{I}, \underbrace{\boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}, \boldsymbol{x}}_{K-1 \text{ times}}\big)\Big) \;=\; \mathrm{vec}\Big(\boldsymbol{\mathcal{X}}\big(\boldsymbol{x}, \boldsymbol{I}, \boldsymbol{x}, \cdots, \boldsymbol{x}, \boldsymbol{x}\big)\Big) \;= \cdots = \; \mathrm{vec}\Big(\boldsymbol{\mathcal{X}}\big(\boldsymbol{x}, \boldsymbol{x}, \boldsymbol{x}, \cdots, \boldsymbol{x}, \boldsymbol{I}\big)\Big).$$

In order to refer to the above quantity more conveniently, we define

$$\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K-1} := \boldsymbol{\mathcal{X}}(\boldsymbol{x}, \ldots, \boldsymbol{x}, \boldsymbol{I}) \in \mathbb{R}^n, \tag{1.2.32}$$

$$\left(\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K-1}\right)_i \;=\; \sum_{i_1 i_2 \ldots i_{K-1} \in [n]} \mathcal{X}_{i_1 i_2 \ldots i_{K-1} i}\, x_{i_1} x_{i_2} \cdots x_{i_{K-1}}. \tag{1.2.33}$$

It can be also verified that the vector $\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K-1}$ is aligned with the gradient $\nabla_{\boldsymbol{x}}\left(\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K}\right)$ as

$$\nabla_{\boldsymbol{x}}\left(\boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K}\right) = K \cdot \boldsymbol{\mathcal{X}}\boldsymbol{x}^{\otimes K-1}. \tag{1.2.34}$$

Tensor decompositions and ranks. The *CANDECOMP/PARAFAC (CP) decomposition* [CC70, Har70] factorizes a tensor into a sum of rank-one tensor components. Mathematically, the CP decomposition of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$ is given by

$$\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\lambda}; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \ldots, \boldsymbol{A}^{(K)}]\!] := \sum_{r \in [R]} \lambda_r \cdot \boldsymbol{a}_r^{(1)} \otimes \boldsymbol{a}_r^{(2)} \otimes \cdots \otimes \boldsymbol{a}_r^{(K)}. \tag{1.2.35}$$

Here $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_R]^\top \in \mathbb{R}^R_+$, $\left\|\boldsymbol{a}_r^{(k)}\right\| = 1$ for all $(r,k) \in [R] \times [K]$ and $\boldsymbol{A}^{(k)} = [\boldsymbol{a}_1^{(k)}, \boldsymbol{a}_2^{(k)}, \cdots, \boldsymbol{a}_R^{(k)}] \in \mathbb{R}^{I_k \times R}$ for all $k \in [K]$. With the CP decomposition provided in (1.2.35), the tensor element $\mathcal{X}_{i_1,i_2,\ldots,i_K}$ can be concisely expressed as

$$\mathcal{X}_{i_1,i_2,\ldots,i_K} = \sum_{r\in[R]} \lambda_r \prod_{k\in[K]} a^{(k)}_{r,i_k}. \tag{1.2.36}$$

The *CP-rank* of the tensor $\boldsymbol{\mathcal{X}}$ is aligned with concept of matrix ranks. Recall that the rank of a matrix $\mathcal{X} \in^{m\times n}$ is defined as smallest number such that $\boldsymbol{X}$ can be written as the sum of rank-one matrices. Similarly, the CP-rank of the tensor $\boldsymbol{\mathcal{X}}$ is the smallest number R such that (1.2.35) holds.

The *Tucker decomposition* [Tuc66] searches for the following pattern:

$$\boldsymbol{\mathcal{X}} = [\![\boldsymbol{G}; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \ldots, \boldsymbol{A}^{(K)}]\!] \tag{1.2.37}$$

$$= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(1)} \times_3 \cdots \times_K \boldsymbol{A}^{(K)} \tag{1.2.38}$$

$$= \sum_{r_1\in[R_1]} \cdots \sum_{r_K\in[R_K]} \boldsymbol{\mathcal{G}}_{r_1 r_2 \cdots r_K} \boldsymbol{a}^{(1)}_{r_1} \otimes \boldsymbol{a}^{(2)}_{r_2} \otimes \cdots \otimes \boldsymbol{a}^{(K)}_{r_K} \tag{1.2.39}$$

Here, $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_K}$ is called *core tensor* and the orthogonal matrix $\boldsymbol{A}^{(k)} = [\boldsymbol{a}_1^{(k)}, \boldsymbol{a}_2^{(k)}, \cdots, \boldsymbol{a}_r^{(k)}] \in \mathbb{R}^{I_k \times r}$ is called *factor matrix*. The *Tucker-rank* of $\boldsymbol{\mathcal{X}}$, denoted by $\operatorname{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}})$, is a K-tuple, describing the rank of each mode-k unfolding matrix, i.e.

$$\operatorname{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}}) := \left(\operatorname{rank}(\boldsymbol{\mathcal{X}}_{(1)}), \operatorname{rank}(\boldsymbol{\mathcal{X}}_{(2)}), \ldots, \operatorname{rank}(\boldsymbol{\mathcal{X}}_{(K)})\right). \tag{1.2.40}$$

There are also a number of other tensor decompositions [CC70, Har72, CPK80, HL96] as variants of the above CP and Tucker ones by imposing more constraints over the factors. These decompositions are not that relevant with the thesis and thus will not be discussed in details.

Chapter 2

Low-Rank Tensor Recovery

2.1 Introduction

Tensors arise naturally in problems where the goal is to estimate a multi-dimensional object whose entries are indexed by several continuous or discrete variables. For example, a video is indexed by two spatial variables and one temporal variable; a hyperspectral datacube is indexed by two spatial variables and a frequency/wavelength variable. While tensors often reside in extremely high-dimensional data spaces, in many applications, the tensor of interest is *low-rank*, or approximately so [KB09], and hence has much lower-dimensional structure. The general problem of estimating a low-rank tensor has applications in many different areas, both theoretical and applied: e.g., estimating latent variable graphical models [AGH+14], classifying audio [MSS06], mining text [CC12], processing radar signals [NS10], multilinear multitask learning [RPABBP13] , to name a few.

In this chapter, we consider the problem of recovering a K-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$ from linear measurements $\boldsymbol{z} = \mathcal{G}[\boldsymbol{\mathcal{X}}] \in \mathbb{R}^m$. Typically, $m \ll N = \prod_{i=1}^K n_i$, and so the problem of recovering $\boldsymbol{\mathcal{X}}$ from $\boldsymbol{z}$ is *ill-posed*. In the past few years, tremendous progress has been made in understanding how to exploit structural assumptions such as sparsity for vectors [CRT06] or low-rankness for matrices [RFP10] to develop computationally tractable methods for tackling ill-posed inverse problems. In many situations, convex optimization can estimate a structured object from near-minimal sets of observations [NRWY12, CRPW12, ALMT14]. For example, an $n \times n$ matrix of rank r can, with high probability, be exactly recovered from Cnr generic linear measurements, by minimizing the nuclear norm $\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X})$. Since a rank r matrix has $r(2n - r)$ degrees of freedom, this is nearly optimal.

In contrast, the correct generalization of these results to low-rank tensors is not obvious. The numerical algebra of tensors is fraught with hardness results [HL13]. For example, even computing a tensor's (CP) rank [CC70, Har70],

$$\operatorname{rank}_{\mathrm{cp}}(\boldsymbol{\mathcal{X}}) = \min\{r \mid \boldsymbol{\mathcal{X}} = \sum\nolimits_{i=1}^{r} \boldsymbol{a}_1^{(i)} \otimes \cdots \otimes \boldsymbol{a}_K^{(i)}\}, \tag{2.1.1}$$

is NP-hard in general. The nuclear norm of a tensor is also intractable, and so we cannot simply follow the formula that has worked for vectors and matrices.

With an eye towards numerical computation, many researchers have studied how to recover tensors of small *Tucker rank* [Tuc66]. The Tucker rank, also known as n-rank, of a K-way tensor $\boldsymbol{\mathcal{X}}$ is a K-dimensional vector whose i-th entry is the (matrix) rank of the mode-i unfolding $\boldsymbol{\mathcal{X}}_{(i)}$ of $\boldsymbol{\mathcal{X}}$:

$$\operatorname{rank}_{\mathrm{tc}}(\boldsymbol{\mathcal{X}}) = \big(\operatorname{rank}(\boldsymbol{\mathcal{X}}_{(1)}), \cdots, \operatorname{rank}(\boldsymbol{\mathcal{X}}_{(K)})\big). \tag{2.1.2}$$

Here, the matrix $\boldsymbol{\mathcal{X}}_{(i)} \in \mathbb{R}^{n_i \times \prod_{j \neq i} n_j}$ is obtained by concatenating all the mode-i fibers of $\boldsymbol{\mathcal{X}}$ as column vectors. Each *mode-i fiber* is an n_i-dimensional vector obtained by fixing every index of $\boldsymbol{\mathcal{X}}$ but the i-th one. The Tucker rank of $\boldsymbol{\mathcal{X}}$ can be computed efficiently using the (matrix) singular value decomposition. For this reason, we focus on tensors of low Tucker rank. However, we will see that our proposed regularization strategy also automatically adapts to recover tensors of low CP rank, with reduction in the required number of measurements.

The definition (2.1.2) suggests a natural, tractable convex approach to recovering low-rank tensors: seek the $\boldsymbol{\mathcal{X}}$ that minimizes $\sum_i \lambda_i \left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_*$ out of all $\boldsymbol{\mathcal{X}}$ satisfying $\mathcal{G}[\boldsymbol{\mathcal{X}}] = \boldsymbol{z}$. We will refer to this as the *sum-of-nuclear-norms (SNN)* model. Originally proposed in [LMWY09], this approach has been widely studied [GRY11, SDS10, STDLS13, TSHK11] and applied to various datasets in imaging [SHKM14, KS13, LL10, LYZY10].

Perhaps surprisingly, we show that this natural approach can be substantially suboptimal. Moreover, we will suggest a simple new convex regularizer with provably better performance. Suppose $n_1 = \cdots = n_K = n$, and $\operatorname{rank}_{\mathrm{tc}}(\boldsymbol{\mathcal{X}}) \preceq (r, r, \ldots, r)$. Let $\mathfrak{T}_r$ denote the set of all such tensors,[1] namely

$$\mathfrak{T}_r := \left\{\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times n \times n \times \cdots \times n} \mid \operatorname{rank}_{\mathrm{tc}}(\boldsymbol{\mathcal{X}}) \preceq (r, r, \ldots, r)\right\}. \tag{2.1.3}$$

We will consider the problem of estimating an element $\boldsymbol{\mathcal{X}}$ of $\mathfrak{T}_r$ from Gaussian measurements $\mathcal{G}$ (i.e., $\boldsymbol{z}_i = \langle \boldsymbol{\mathcal{G}}_i, \boldsymbol{\mathcal{X}} \rangle$, where $\boldsymbol{\mathcal{G}}_i$ has i.i.d. standard normal entries). To describe a generic tensor in $\mathfrak{T}_r$, we need at most $r^K + rnK$ parameters. In Section 2.2, we show that a certain nonconvex strategy can recover all $\boldsymbol{\mathcal{X}} \in \mathfrak{T}_r$

[1] To keep the presentation in this chapter compact, we state most of our results regarding tensors in $\mathfrak{T}_r$, although it is not difficult to modify them for general tensors.

exactly when $m > (2r)^K + 2nrK$. In contrast, the best known theoretical guarantee for SNN minimization, due to Tomioka et al. [TSHK11], shows that $\boldsymbol{\mathcal{X}} \in \mathfrak{T}_r$ can be recovered (or accurately estimated) from Gaussian measurements $\mathcal{G}$, provided $m = \Omega(rn^{K-1})$. In Section 2.3, we prove that this number of measurements is also *necessary*: accurate recovery is unlikely unless $m = \Omega(rn^{K-1})$. Thus, there is a substantial gap between an ideal nonconvex approach and the best known tractable surrogate. In Section 2.4, we introduce a simple alternative, which we call the *square reshaping* model, which reduces the required number of measurements to $O(r^{\lfloor K/2 \rfloor} n^{\lceil K/2 \rceil})$. For $K > 3$, we obtain an improvement of a multiplicative factor polynomial in n.

Our theoretical results pertain to Gaussian operators $\mathcal{G}$. The motivation for studying Gaussian measurements is threefold. First, Gaussian measurements may be of interest for compressed sensing recovery [Don06], either directly as a measurement strategy, or indirectly due to universality phenomena [BLM12]. Moreover, the available theoretical tools for Gaussian measurements are very sharp, allowing us to rigorously investigate the efficacy of various regularization schemes, and prove both upper and lower bounds on the number of observations required. Furthermore, the results with respect to Gaussian measurements have direct implications to the minimax risk for denoising [OH16, ALMT14]. In Section 2.4, we demonstrate that our qualitative conclusions carry over to more realistic measurement models, such as random subsampling [LMWY09]. We expect our results to be of great interest for a wide range of problems in tensor completion [LMWY09], robust tensor recovery/decomposition [LYZY10, GQ14] and sensing.

Our technical approach draws on, and enriches, the literature on general structured model recovery. The surprisingly poor behavior of the SNN model is an example of a phenomenon first discovered by Oymak et al. [OJF+12]: for recovering objects with multiple structures, a combination of structure-inducing norms is often not significantly more powerful than the best individual structure-inducing norm. Our lower bound for the SNN model follows from a general result of this nature, which we prove using the novel geometric framework of [ALMT14]. Compared to [OJF+12], our result pertains to a more general family of regularizers, and gives sharper constants. In addition, for low-rank tensor recovery problem, we demonstrate the possibility to reduce the number of generic measurements through a new convex regularizer that exploits several sparse structures jointly.

2.2 Bounds for Non-Convex Recovery

In this section, we introduce a non-convex model for tensor recovery, and show that it recovers low-rank tensors from near-minimal numbers of measurements. While our nonconvex formulation is computationally

intractable, it gives a baseline for evaluating tractable (convex) approaches.

For a tensor of low Tucker rank, the matrix unfolding along each mode is low-rank. Suppose we observe $\mathcal{G}[\boldsymbol{\mathcal{X}}_0] \in \mathbb{R}^m$. We would like to attempt to recover $\boldsymbol{\mathcal{X}}_0$ by minimizing some combination of the ranks of the unfoldings, over all tensors $\boldsymbol{\mathcal{X}}$ that are consistent with our observations. This suggests a *vector optimization* problem [BV04, Chap. 4.7]:

$$\text{minimize}_{(\text{w.r.t. } \mathbb{R}^K_+)} \quad \text{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}}) \quad \text{subject to} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0]. \tag{2.2.1}$$

In vector optimization, a feasible point is called *Pareto optimal* if no other feasible point dominates it in every criterion. In a similar vein, we say that (2.2.1) recovers $\boldsymbol{\mathcal{X}}_0$ if there does not exist any other tensor $\boldsymbol{\mathcal{X}}$ that is consistent with the observations and has no larger rank along each mode:

Definition 2.1 *We call $\boldsymbol{\mathcal{X}}_0$ recoverable by (2.2.1) if the set*

$$\{\boldsymbol{\mathcal{X}}' \neq \boldsymbol{\mathcal{X}}_0 \mid \mathcal{G}[\boldsymbol{\mathcal{X}}'] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0],\ \text{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}}') \preceq_{\mathbb{R}^K_+} \text{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}}_0)\} = \emptyset.$$

This is equivalent to saying that $\boldsymbol{\mathcal{X}}_0$ is the unique optimal solution to the *scalar* optimization:

$$\text{minimize}_{\boldsymbol{\mathcal{X}}} \quad \max_i \left\{ \frac{\text{rank}(\boldsymbol{\mathcal{X}}_{(i)})}{\text{rank}(\boldsymbol{\mathcal{X}}_{0(i)})} \right\} \qquad \text{subject to} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0]. \tag{2.2.2}$$

The problems (2.2.1)-(2.2.2) are not tractable. However, they do serve as a baseline for understanding how many generic measurements are required to recover $\boldsymbol{\mathcal{X}}_0$ from an information theoretic perspective.

The recovery performance of program (2.2.1) depends heavily on the properties of $\mathcal{G}$. Suppose (2.2.1) fails to recover $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$. Then there exists another $\boldsymbol{\mathcal{X}}' \in \mathfrak{T}_r$ such that $\mathcal{G}[\boldsymbol{\mathcal{X}}'] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0]$. So, to guarantee that (2.2.1) recovers *any* $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$, a necessary and sufficient condition is that $\mathcal{G}$ is injective on $\mathfrak{T}_r$, which can be implied by the condition $\text{null}(\mathcal{G}) \cap \mathfrak{T}_{2r} = \{\mathbf{0}\}$. Consequently, if $\text{null}(\mathcal{G}) \cap \mathfrak{T}_{2r} = \{\mathbf{0}\}$, (2.2.1) will recover any $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$. We expect this to occur when the number of measurements significantly exceeds the number of intrinsic degrees of freedom of a generic element of $\mathfrak{T}_r$, which is $O(r^K + nrK)$. The following theorem shows that when m is approximately twice this number, with probability one, $\mathcal{G}$ is injective on $\mathfrak{T}_r$:

Theorem 2.2 *Whenever $m \geq (2r)^K + 2nrK + 1$, with probability one, $\text{null}(\mathcal{G}) \cap \mathfrak{T}_{2r} = \{0\}$, and hence (2.2.1) recovers every $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$.*

The proof of Theorem 2.2 follows from a covering argument, which we establish in several steps. Let

$$\mathfrak{S}_{2r} = \{\boldsymbol{\mathcal{D}} \mid \boldsymbol{\mathcal{D}} \in \mathfrak{T}_{2r}, \|\boldsymbol{\mathcal{D}}\|_F = 1\}. \tag{2.2.3}$$

The following lemma shows that the required number of measurements can be bounded in terms of the exponent of the covering number for $\mathfrak{S}_{2r}$, which can be considered as a proxy for dimensionality:

Lemma 2.3 *Suppose that the covering number for $\mathfrak{S}_{2r}$ with respect to Frobenius norm, satisfies*

$$N(\mathfrak{S}_{2r}, \|\cdot\|_F, \varepsilon) \leq (\beta/\varepsilon)^d, \tag{2.2.4}$$

for some integer d and scalar β that does not depend on ε. Then if $m \geq d+1$, with probability one $\operatorname{null}(\mathcal{G}) \cap \mathfrak{S}_{2r} = \emptyset$, which implies that $\operatorname{null}(\mathcal{G}) \cap \mathfrak{T}_{2r} = \{\mathbf{0}\}$.

It just remains to find the covering number of $\mathfrak{S}_{2r}$. We use the following lemma, which uses the triangle inequality to control the effect of perturbations in the factors of the Tucker decomposition

$$[[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K]] := \boldsymbol{\mathcal{C}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_K \boldsymbol{U}_K, \tag{2.2.5}$$

where the *mode-i (matrix) product* of tensor $\boldsymbol{\mathcal{A}}$ with matrix $\boldsymbol{B}$ of compatible size, denoted as $\boldsymbol{\mathcal{A}} \times_i \boldsymbol{B}$, outputs a tensor $\boldsymbol{\mathcal{C}}$ such that $\boldsymbol{\mathcal{C}}_{(i)} = \boldsymbol{B}\boldsymbol{\mathcal{A}}_{(i)}$.

Lemma 2.4 *Let $\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}}' \in \mathbb{R}^{r_1,\ldots,r_K}$, and $\boldsymbol{U}_1, \boldsymbol{U}_1' \in \mathbb{R}^{n_1 \times r_1}, \ldots, \boldsymbol{U}_K, \boldsymbol{U}_K' \in \mathbb{R}^{n_K \times r_K}$ with $\boldsymbol{U}_i^* \boldsymbol{U}_i = \boldsymbol{U}_i'^* \boldsymbol{U}_i' = \boldsymbol{I}$, and $\|\boldsymbol{\mathcal{C}}\|_F = \|\boldsymbol{\mathcal{C}}'\|_F = 1$. Then*

$$\|[[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]] - [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}_1', \ldots, \boldsymbol{U}_K']]\|_F \leq \|\boldsymbol{\mathcal{C}} - \boldsymbol{\mathcal{C}}'\|_F + \sum_{i=1}^{K} \|\boldsymbol{U}_i - \boldsymbol{U}_i'\|. \tag{2.2.6}$$

Using this result, we construct an ε-net for $\mathfrak{S}_{2r}$ by building $\varepsilon/(K+1)$-nets for each of the $K+1$ factors $\boldsymbol{\mathcal{C}}$ and $\{\boldsymbol{U}_i\}$. The total size of the resulting ε net is thus bounded by the following lemma:

Lemma 2.5 $N(\mathfrak{S}_{2r}, \|\cdot\|_F, \varepsilon) \leq (3(K+1)/\varepsilon)^{(2r)^K + 2nrK}$

With these observations in hand, Theorem 2.2 follows immediately.

2.3 Convexification: Sum of Nuclear Norms?

Since the nonconvex problem (2.2.1) is NP-hard for general $\mathcal{G}$, it is tempting to seek a convex surrogate. In matrix recovery problems, the nuclear norm is often an excellent convex surrogate for the rank [Faz02, RFP10, Gro11]. It seems natural, then, to replace the ranks in (2.2.1) with nuclear norms. Due to convexity, the

resulting vector optimization problem can be solved by the following scalar optimization:

$$\min_{\boldsymbol{\mathcal{X}}} \quad \sum_{i=1}^{K} \lambda_i \|\boldsymbol{\mathcal{X}}_{(i)}\|_* \quad \text{s.t.} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0], \tag{2.3.1}$$

where $\boldsymbol{\lambda} \geq \mathbf{0}$. The optimization (2.3.1) was first introduced by [LMWY09] and has been used successfully in applications in imaging [SHKM14, KS13, LL10, EAHK13, LYZY10]. Similar convex relaxations have been considered in a number of theoretical and algorithmic works [GRY11, SDS10, TSHK11, STDLS13]. It is not too surprising, then, that (2.3.1) provably recovers the underlying tensor $\boldsymbol{\mathcal{X}}_0$, when the number of measurements m is sufficiently large. The following is a (simplified) corollary of results of Tomioka et. al. [TSHK11] [2]:

> **Corollary 2.6 (of [TSHK11], Theorem 3)** *Suppose that $\boldsymbol{\mathcal{X}}_0$ has Tucker rank $(r, \ldots, r)$, and $m \geq Crn^{K-1}$, where C is a constant. Then with high probability, $\boldsymbol{\mathcal{X}}_0$ is the optimal solution to (2.3.1), with each $\lambda_i = 1$.*

This result shows that there *is* a range in which (2.3.1) succeeds: loosely, when we undersample by at most a factor of $m/N \sim r/n$. However, the number of observations $m \sim rn^{K-1}$ is significantly larger than the number of degrees of freedom in $\boldsymbol{\mathcal{X}}_0$, which is on the order of $r^K + nrK$. Is it possible to prove a better bound for this model? Unfortunately, we show that in general $O(rn^{K-1})$ measurements are also *necessary* for reliable recovery using (2.3.1):

> **Theorem 2.7** *Let $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$ be nonzero. Set $\kappa = \min_i \left\{ \left\| (\boldsymbol{\mathcal{X}}_0)_{(i)} \right\|_*^2 / \|\boldsymbol{\mathcal{X}}_0\|_F^2 \right\} \times n^{K-1}$. Then if the number of measurements $m \leq \kappa - 2$, $\boldsymbol{\mathcal{X}}_0$ is not the unique solution to (2.3.1), with probability at least $1 - 4\exp(-\frac{(\kappa-m-2)^2}{16(\kappa-2)})$. Moreover, there exists $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$ for which $\kappa = rn^{K-1}$.*

This implies that Corollary 2.6 (as well as some other results of [TSHK11]) is essentially tight. Unfortunately, it has negative implications for the efficacy of the SNN model in (2.3.1): although a generic element $\boldsymbol{\mathcal{X}}_0$ of $\mathfrak{T}_r$ can be described using at most $r^K + nrK$ real numbers, we require $\Omega(rn^{K-1})$ observations to recover it using (2.3.1). Theorem 2.7 is a direct consequence of a much more general principle underlying multi-structured recovery, which is elaborated next. After that, in Section 2.4, we show that for low-rank tensor recovery, better convexifying schemes are available.

2.3.1 General lower bound for multiple structures

The poor behavior of (2.3.1) is an instance of a much more general phenomenon, first discovered by Oymak et. al. [OJF+12]. Our target tensor $\boldsymbol{\mathcal{X}}_0$ has *multiple* low-dimensional structures simultaneously: it is low-rank

[2] Tomioka et. al. also show noise stability when $m = \Omega(rn^{K-1})$ and give extensions to the case where the $\text{rank}_{\text{tc}}(\boldsymbol{\mathcal{X}}_0) = (r_1, \ldots, r_K)$ differs from mode to mode.

along *each* of the K modes. In practical applications, many other such *simultaneously structured* objects could also be of interest. For sparse phase retrieval problems in signal processing [OJF+12], the task can be rephrased to infer a block sparse matrix, which implies both sparse and low-rank structures. In robust metric learning [LML13], the goal is to estimate a matrix that is column sparse and low rank concurrently. In computer vision, many signals of interest are both low-rank and sparse in an appropriate basis [LRZM12]. To recover such simultaneously structured objects, it is tempting to build a convex relaxation by combining the convex relaxations for each of the individual structures. In the tensor case, this yields (2.3.1). Surprisingly, this combination is often not significantly more powerful than the best single regularizer [OJF+12]. We obtain Theorem 2.7 as a consquence of a new, general result of this nature, using a geometric framework introduced in [ALMT14]. Compared to [OJF+12], this approach has a clearer geometric intuition, covers a more general class of regularizers[3] and yields sharper bounds.

Setup. In general, we are interested in recovering a signal $\boldsymbol{x}_0$ with several low-dimensional structures simultaneously, based on generic measurements with respect to $\boldsymbol{x}_0$. Here the target signal $\boldsymbol{x}_0$ could lie in any finite dimensional Hilbert space (e.g. a vector in $\mathbb{R}^n$, a matrix in $\mathbb{R}^{n_1 \times n_2}$, a tensor in $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$), but without loss of generality, we will consider $\boldsymbol{x}_0 \in \mathbb{R}^n$. Let $\|\cdot\|_{(i)}$ be the penalty norm corresponding to the i-th structure (e.g. ℓ_1, nuclear norm). Consider the following *sum-of-norms (SoN)* model,

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad f(x) := \lambda_1 \|\boldsymbol{x}\|_{(1)} + \lambda_2 \|\boldsymbol{x}\|_{(2)} + \cdots + \lambda_K \|\boldsymbol{x}\|_{(K)} \qquad \text{subject to} \quad \mathcal{G}[\boldsymbol{x}] = \mathcal{G}[\boldsymbol{x}_0], \tag{2.3.2}$$

where $\mathcal{G}[\cdot]$ is a Gaussian measurement operator, and $\boldsymbol{\lambda} > \mathbf{0}$. In the subsequent analysis, we will evaluate the performance of (2.3.2) in terms of recovering $\boldsymbol{x}_0$, where the only assumption we require is:

Assumption 2.8 *The target signal $\boldsymbol{x}_0$ is nonzero.*

Optimality condition. Is $\boldsymbol{x}_0$ the unique optimal solution to (2.3.2)? Recall that the descent cone of a function f at a point $\boldsymbol{x}_0$ is defined as

$$\mathcal{C}(f, \boldsymbol{x}_0) := \operatorname{cone}\{\boldsymbol{v} \mid f(\boldsymbol{x}_0 + \boldsymbol{v}) \leq f(\boldsymbol{x}_0)\}, \tag{2.3.3}$$

which, in short, will be denoted as $\mathcal{C}$. Then $\boldsymbol{x}_0$ is the unique optimal solution if and only if

$$\operatorname{null}(\mathcal{G}) \cap \mathcal{C} = \{\mathbf{0}\}. \tag{2.3.4}$$

Conversely, recovery fails if $\operatorname{null}(\mathcal{G})$ has nontrivial intersection with $\mathcal{C}$.

[3] [OJF+12] studies decomposable norms, with some additional assumptions. Our result holds for arbitrary norms.

Since $\mathcal{G}$ is a Gaussian operator, $\mathrm{null}(\mathcal{G})$ is a uniformly oriented random subspace of dimension $(n-m)$. This random subspace is more likely to have nontrivial intersection with $\mathcal{C}$ if $\mathcal{C}$ is *large*, in a sense we will make precise.

Denote the polar cone of $\mathcal{C}$ as $\mathcal{C}^\circ$, i.e.

$$\mathcal{C}^\circ := \left\{ \boldsymbol{u} \in \mathbb{R}^n \mid \sup_{x \in \mathcal{C}} \langle \boldsymbol{u}, \boldsymbol{x} \rangle \leq 0 \right\}. \tag{2.3.5}$$

Because polarity reverses inclusion, we expect that $\mathcal{C}$ will be *large* whenever $\mathcal{C}^\circ$ is *small*, which leads us to control the size of $\mathcal{C}^\circ$.

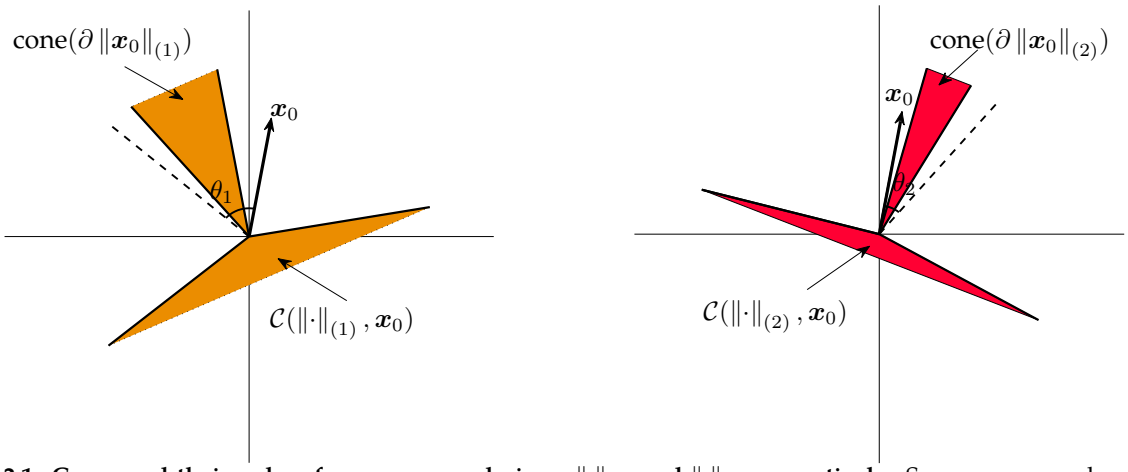


Figure 2.1: Cones and their polars for convex regularizers $\|\cdot\|_{(1)}$ and $\|\cdot\|_{(2)}$ respectively. Suppose our $\boldsymbol{x}_0$ has two sparse structures simultaneously. Regularizer $\|\cdot\|_{(1)}$ has a larger conic hull of subdifferential at $\boldsymbol{x}_0$, i.e. $\mathrm{cone}(\partial \|\boldsymbol{x}_0\|_{(1)})$, which results in a smaller descent cone. Thus minimizing $\|\cdot\|_{(1)}$ is more likely to recover $\boldsymbol{x}_0$ than minimizing $\|\cdot\|_{(2)}$. Consider convex regularizer $f(\boldsymbol{x}) = \|\boldsymbol{x}_0\|_{(1)} + \|\boldsymbol{x}_0\|_{(2)}$. Suppose as depicted, $\theta_1 \geq \theta_2$. Then both $\mathrm{cone}(\partial \|\boldsymbol{x}_0\|_{(1)})$ and $\mathrm{cone}(\partial \|\boldsymbol{x}_0\|_{(2)})$ are in the circular cone $\mathrm{circ}(\boldsymbol{x}_0, \theta_1)$. Thus we have: $\mathrm{cone}\big(\partial f(\boldsymbol{x}_0)\big) = \mathrm{cone}(\partial \|\boldsymbol{x}_0\|_{(1)} + \partial \|\boldsymbol{x}_0\|_{(2)}) \subseteq \mathrm{conv}\big\{\mathrm{circ}(\boldsymbol{x}_0, \theta_1), \mathrm{circ}(\boldsymbol{x}_0, \theta_2)\big\} = \mathrm{circ}(\boldsymbol{x}_0, \theta_1)$.

As $f(\boldsymbol{x}_0) \neq 0 = \min_{\boldsymbol{x} \in \mathbb{R}^n} f(x)$, it can be verified that [Roc97, Thm. 23.7]

$$\mathcal{C}^\circ = \mathrm{cone}\left(\partial f\left(\boldsymbol{x}_0\right)\right) = \mathrm{cone}\left(\sum_{i \in [K]} \lambda_i \partial f_i(\boldsymbol{x}_0)\right), \tag{2.3.6}$$

where the sum is made in Minkowski sense.

In order to control the size of $\mathcal{C}^o$ based on (2.3.6), we will next establish some basic geometric properties for each single norm.

Properties for each single norm. Consider a general single norm $\|\cdot\|_\diamond$ and denote its dual norm (a.k.a. polar function) as $\|\cdot\|_\diamond^\circ$, i.e. for any $\boldsymbol{u} \in \mathbb{R}^n$,

$$\|\boldsymbol{u}\|_\diamond^\circ := \sup_{\|\boldsymbol{x}\|_\diamond \le 1} \langle \boldsymbol{x}, \boldsymbol{u} \rangle . \tag{2.3.7}$$

Define $L := \sup_{\boldsymbol{x} \neq \boldsymbol{0}} \|\boldsymbol{x}\|_\diamond / \|\boldsymbol{x}\|$, which implies that $\|\cdot\|_\diamond$ is L-Lipschitz: $\|\boldsymbol{x}\|_\diamond \le L \|\boldsymbol{x}\|$ for all $\boldsymbol{x}$. Then we also have $\|\boldsymbol{u}\| \le L \|\boldsymbol{u}\|_\diamond^\circ$ for all $\boldsymbol{u}$ as

$$\|\boldsymbol{u}\|_\diamond^\circ = \sup_{\|\boldsymbol{x}\|_\diamond \le 1} \langle \boldsymbol{x}, \boldsymbol{u} \rangle \ge \sup_{L\|\boldsymbol{x}\| \le 1} \langle \boldsymbol{x}, \boldsymbol{u} \rangle = \sup_{\|\boldsymbol{x}\| \le 1/L} \langle \boldsymbol{x}, \boldsymbol{u} \rangle = \frac{1}{L} \|\boldsymbol{u}\| . \tag{2.3.8}$$

In addition, noting that

$$\partial \|\cdot\|_\diamond (\boldsymbol{x}) = \left\{ \boldsymbol{u} \mid \langle \boldsymbol{u}, \boldsymbol{x} \rangle = \|\boldsymbol{x}\|_\diamond , \ \|\boldsymbol{u}\|_\diamond^\circ \le 1 \right\}, \tag{2.3.9}$$

for any $\boldsymbol{u} \in \partial \|\cdot\|_\diamond (\boldsymbol{x}_0)$, we have

$$\cos\left(\angle(\boldsymbol{u}, \boldsymbol{x}_0)\right) := \frac{\langle \boldsymbol{u}, \boldsymbol{x}_0 \rangle}{\|\boldsymbol{u}\| \, \|\boldsymbol{x}_0\|} \ \ge \ \frac{\|\boldsymbol{x}_0\|_\diamond}{L \, \|\boldsymbol{u}\|_\diamond^\circ \, \|\boldsymbol{x}_0\|} \ \ge \ \frac{\|\boldsymbol{x}_0\|_\diamond}{L \, \|\boldsymbol{x}_0\|}. \tag{2.3.10}$$

A more geometric way of summarizing this fact is as follows: for $\boldsymbol{x} \neq \boldsymbol{0}$, let

$$\mathrm{circ}(\boldsymbol{x}, \theta) = \{ \boldsymbol{z} \mid \angle(\boldsymbol{z}, \boldsymbol{x}) \le \theta \}, \tag{2.3.11}$$

denote the *circular cone* with axis $\boldsymbol{x}$ and angle θ. Then with $\theta := \cos^{-1}(\|\boldsymbol{x}_0\|_\diamond / L \|\boldsymbol{x}_0\|)$,

$$\partial \|\cdot\|_\diamond (\boldsymbol{x}_0) \subseteq \mathrm{circ}\left(\boldsymbol{x}_0, \theta\right). \tag{2.3.12}$$

Table 2.1 describes the angle parameters θ for various structure inducing norms. Notice that in general, more complicated $\boldsymbol{x}_0$ leads to smaller angles θ. For example, if $\boldsymbol{x}_0$ is a k-sparse vectors with entries all of the same magnitude, and $\|\cdot\|_\diamond$ the ℓ^1 norm, $\cos^2\theta = k/n$. As $\boldsymbol{x}_0$ becomes more dense, $\partial \|\cdot\|_\diamond$ is contained in smaller and smaller circular cones.

Polar cone $\subseteq$ circular cone. For $f = \sum_i \lambda_i \|\cdot\|_{(i)}$, notice that every element of $\partial f(\boldsymbol{x}_0)$ is a conic combination of elements of the $\partial \|\cdot\|_{(i)} (\boldsymbol{x}_0)$. Since each of the $\partial \|\cdot\|_{(i)} (\boldsymbol{x}_0)$ is contained in a circular cone with axis $\boldsymbol{x}_0$, $\partial f(\boldsymbol{x}_0)$ itself is also contained in a circular cone, and thus based on (2.3.6), we have

Lemma 2.9 *For $\boldsymbol{x}_0 \neq \mathbf{0}$, set $\theta_i = \cos^{-1}\left(\|\boldsymbol{x}_0\|_{(i)} / L_i \|\boldsymbol{x}_0\|\right)$, where $L_i = \sup_{\boldsymbol{x}\neq\mathbf{0}} \|\boldsymbol{x}\|_{(i)} / \|\boldsymbol{x}\|$. Then*

$$\mathcal{C}^\circ = \operatorname{cone}\left(\partial f\left(\boldsymbol{x}_0\right)\right) \subseteq \operatorname{circ}\left(\boldsymbol{x}_0, \max_{i\in[K]} \theta_i\right). \tag{2.3.13}$$

So, the subdifferential of our combined regularizer f is contained in a circular cone whose angle is given by the largest of the θ_i. Figure 2.1 visualizes this geometry.

Statistical Dimension. How does this behavior affect the recoverability of $\boldsymbol{x}_0$ via (2.3.2)? The informal reasoning above suggests that as θ becomes smaller, the descent cone $\mathcal{C}$ becomes larger, and we require more measurements to recover $\boldsymbol{x}_0$. This can be made precise using the elegant framework introduced by Amelunxen et al. [ALMT14]. They define the *statistical dimension* of the convex cone $\mathcal{C}$ to be the expected norm square of the projection of a standard Gaussian vector onto C:

$$\delta(\mathcal{C}) := \mathbb{E}_{\boldsymbol{g}\sim_{\text{i.i.d.}}\mathcal{N}(0,1)}\left[\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{g})\|^2\right] \tag{2.3.14}$$

Using tools from spherical integral geometry, Amelunxen et al. [ALMT14] show that for linear inverse problems with Gaussian measurements, a sharp phase transition in recoverability occurs around $m = \delta(\mathcal{C})$. Since we attempt to derive a necessary condition for the success of (2.3.2), we need only one side of their result with slight modifications:

Corollary 2.10 *Let $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^m$ be a Gaussian operator, and $\mathcal{C}$ a convex cone. Then if $m \leq \delta(\mathcal{C})$,*

$$\mathbb{P}\left[\mathcal{C} \cap \operatorname{null}(\mathcal{G}) = \{\mathbf{0}\}\right] \;\leq\; 4\exp\left(-\frac{(\delta(\mathcal{C}) - m)^2}{16\delta(\mathcal{C})}\right). \tag{2.3.15}$$

Table 2.1: Concise models and their surrogates. For each norm $\|\cdot\|_\diamond$, the third column describes the range of achievable angles θ. Larger $\cos\theta$ corresponds to a smaller C^o, a larger C, and hence a larger number of measurements required for reliable recovery.

Object	**Complexity Measure**	**Relaxation**	$\cos^2\theta$	$\kappa = n\cos^2\theta$
Sparse $\boldsymbol{x} \in \mathbb{R}^n$	$k = \|\boldsymbol{x}\|_0$	$\|\boldsymbol{x}\|_1$	$[\frac{1}{n}, \frac{k}{n}]$	$[1, k]$
Column-sparse $\boldsymbol{x} \in \mathbb{R}^{n_1\times n_2}$	$c = \#\{j \mid \boldsymbol{x}\boldsymbol{e}_j \neq \mathbf{0}\}$	$\sum_j \|\boldsymbol{x}\boldsymbol{e}_j\|$	$[\frac{1}{n_2}, \frac{c}{n_2}]$	$[n_1, cn_1]$
Low-rank $\boldsymbol{x} \in \mathbb{R}^{n_1\times n_2}$ $(n_1 \geq n_2)$	$r = \operatorname{rank}(\boldsymbol{x})$	$\|\boldsymbol{x}\|_*$	$[\frac{1}{n_2}, \frac{r}{n_2}]$	$[n_1, rn_1]$

To apply this result to our problem, we need to have a lower bound on the statistical dimension $\delta(\mathcal{C})$, of the descent cone $\mathcal{C}$ of f at $\boldsymbol{x}_0$. Using the Pythagorean theorem, monotonicity of $\delta(\cdot)$, and Lemma 2.9, we calculate

$$\delta(\mathcal{C}) \;=\; n - \delta(\mathcal{C}^\circ) \;=\; n - \delta\left(\operatorname{cone}(\partial f(\boldsymbol{x}_0))\right) \;\geq\; n - \delta(\operatorname{circ}(\boldsymbol{x}_0, \max_i \theta_i)). \tag{2.3.16}$$

Moreover, using the properties of statistical dimension, we are able to prove an upper bound for the statistical dimension of circular cone, which improves the constant in existing results [ALMT14, McC13].

Lemma 2.11 $\delta(\operatorname{circ}(\boldsymbol{x}_0, \theta)) \leq n \sin^2\theta + 2$.

Finally, by combining (2.3.16) and Lemma 2.11, we have $\delta(\mathcal{C}) \geq n \min_i \cos^2\theta_i - 2$. Using Corollary 2.10, we obtain:

Theorem 2.12 (SoN model.) *Suppose the target signal $\boldsymbol{x}_0 \neq \mathbf{0}$. For each i-th norm ($i \in [K]$), define $L_i := \sup_{\boldsymbol{x}\neq\mathbf{0}} \|\boldsymbol{x}\|_{(i)} / \|\boldsymbol{x}\|$. Set*

$$\kappa_i \;=\; \frac{n\,\|\boldsymbol{x}_0\|_{(i)}^2}{L_i^2\,\|\boldsymbol{x}_0\|^2} \;=\; n\cos^2(\theta_i), \quad \textit{and} \quad \kappa = \min_i \kappa_i.$$

Then the statistical dimension of the descent cone of f at the point $\boldsymbol{x}_0$: $\delta\left(\mathcal{C}\left(f, \boldsymbol{x}_0\right)\right) \geq \kappa - 2$, and thus if the number of generic measurements $m \leq \kappa - 2$,

$$\mathbb{P}\left[\boldsymbol{x}_0 \text{ is the unique optimal solution to (2.3.2)}\right] \leq 4\exp\left(-\frac{(\kappa - m - 2)^2}{16\,(\kappa - 2)}\right). \tag{2.3.17}$$

Consequently, for reliable recovery, the number of measurements needs to be at least proportional to κ.[4] Notice that $\kappa = \min_i \kappa_i$ is determined by only the best of the structures. Per Table 2.1, κ_i is often on the order of the number of degrees of freedom in a generic object of the i-th structure. For example, for a k-sparse vector whose nonzeros are all of the same magnitude, $\kappa = k$.

Theorem 2.12 together with Table 2.1 leads us to the phenomenon that recently discovered by Oymak et al. [OJF+12]: for recovering objects with multiple structures, a combination of structure-inducing norms tends to be not significantly more powerful than the best individual structure-inducing norm. As we demonstrate, this general behavior follows a clear geometric interpretation that the subdifferential of a norm at $\boldsymbol{x}_0$ is contained in a relatively small circular cone with central axis $\boldsymbol{x}_0$.

Extension. Here we consider a slightly more general setup: a signal $\boldsymbol{x}_0 \in \mathbb{R}^n$, *after appropriate linear transforms*, has K low-dimensional structures simultaneously. These linear transforms can be quite general, and could

[4] E.g., if $m = (\kappa - 2)/2$, the probability of success is at most $4\exp(-(\kappa - 2)/64)$.

be either prescribed by experts or adaptively learned from training data.

In specific, for any i in $[K]$, there exists an appropriate linear transform $\mathcal{A}_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ such that $\mathcal{A}_i[\boldsymbol{x}_0]$ follows a parsimonious model in $\mathbb{R}^{m_i}$ (e.g. sparsity, low-rank). Let $\|\cdot\|_{(i)}$ be the penalty norms corresponding to the i-th structure (e.g. ℓ_1, nuclear norm). Based on generic measurements collected, it is natural to recover $\boldsymbol{x}_0$ using the following *sum-of-composite-norms (SoCN)* formulation

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \; f(x) := \lambda_1 \|\mathcal{A}_1[\boldsymbol{x}]\|_{(1)} + \lambda_2 \|\mathcal{A}_2[\boldsymbol{x}]\|_{(2)} + \cdots + \lambda_K \|\mathcal{A}_K[\boldsymbol{x}]\|_{(K)} \qquad \text{s.t.} \quad \mathcal{G}[\boldsymbol{x}] = \mathcal{G}[\boldsymbol{x}_0], \tag{2.3.18}$$

where $\mathcal{G}[\cdot]$ is a Gaussian measurement operator, and $\boldsymbol{\lambda} > \boldsymbol{0}$. Essentially following the same reasoning as above, a result similar to Theorem 2.12, stating a lower bound on the number of generic measurements required, can be achieved:

Theorem 2.13 (SoCN model) *Suppose the target signal $\boldsymbol{x}_0 \notin \cap_{i\in[K]} null(\mathcal{A}_i)$. For each $i \in [K]$, define $L_i = \sup_{\boldsymbol{x}\in\mathbb{R}^{m_i}\setminus\{\boldsymbol{0}\}} \|\boldsymbol{x}\|_{(i)} / \|\boldsymbol{x}\|$. Set*

$$\kappa_i \;=\; \frac{n\,\|\mathcal{A}_i\boldsymbol{x}_0\|_{(i)}^2}{L_i^2\,\|\mathcal{A}_i\|^2\,\|\boldsymbol{x}_0\|^2}, \quad \textit{and} \quad \kappa = \min_i \kappa_i.$$

Then if $m \le \kappa - 2$,

$$\mathbb{P}\left[\boldsymbol{x}_0 \text{ is the unique optimal solution to (2.3.18)}\right] \le 4\exp\left(-\frac{(\kappa-m-2)^2}{16\,(\kappa-2)}\right). \tag{2.3.19}$$

Remark 2.14 *Clearly, Theorem 2.12 can be regarded as a special case of Theorem 2.13, where $\mathcal{A}_i'$s are all identity operators.*

2.3.2 Low-rank tensors

We can specialize Theorem 2.12 to low-rank tensors as follows: if the target signal $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$, i.e. a K-mode $n\times n\times\cdots\times n$ tensor of Tucker rank $(r, r, \ldots, r)$, then for each $i \in [K]$, $\|\cdot\|_{(i)} := \left\|(\cdot)_{(i)}\right\|_*$ is $L_i = \sqrt{n}$-Lipschitz. Hence

$$\kappa = \min_i \left\{ \left\|(\boldsymbol{\mathcal{X}}_0)_{(i)}\right\|_*^2 / \|\boldsymbol{\mathcal{X}}_0\|_F^2 \right\} n^{K-1}. \tag{2.3.20}$$

The term $\min_i \left\{ \left\|(\boldsymbol{\mathcal{X}}_0)_{(i)}\right\|_*^2 / \|\boldsymbol{\mathcal{X}}_0\|_F^2 \right\}$ lies between 1 and r, inclusively. For example, if $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_1$, then that term is equal to 1; if $\boldsymbol{\mathcal{X}}_0 = [[\boldsymbol{\mathcal{C}}, \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]]$ with $\boldsymbol{U}_i^*\boldsymbol{U}_i = \boldsymbol{I}$ and $\boldsymbol{\mathcal{C}}$ (super)diagonal ($\boldsymbol{\mathcal{C}}_{i_1\ldots i_r} = \boldsymbol{1}_{\{i_1=i_2=\cdots=i_r\}}$), then that term is equal to r. That exactly yields Theorem 2.7.

Empirical estimates of the statistical dimension. As noted in Theorem 2.12, the statistical dimension of the descent cone $\delta(\mathcal{C})$ plays a crucial role in deriving our lower bound for the number of generic measurements. In the following, we will numerically justify our theoretical result for $\delta(\mathcal{C})$ under the setting of our interest, low-rank tensors.

Consider $\boldsymbol{\mathcal{X}}_0$ as a K-mode $n \times n \times \cdots \times n$ (super)diagonal tensor with only the first r diagonal entries as 1 and 0 elsewhere. Clearly, $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$, and Corollary 2.6, Theorem 2.12 and expression (2.3.20) yield

$$\delta(\mathcal{C}) := \delta\left(\mathcal{C}\left(\sum_{i=1}^{K}\left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_*, \boldsymbol{\mathcal{X}}_0\right)\right) \geq rn^{K-1} - 2, \quad \text{and} \quad \delta(\mathcal{C}) = \Theta(rn^{K-1}). \tag{2.3.21}$$

In the following, we will numerically corroborate (2.3.21) based on recent results developed in statistical decision theory.

In order to estimate $\delta(\mathcal{C})$, we construct a perturbed observation $\boldsymbol{\mathcal{Z}}_0 = \boldsymbol{\mathcal{X}}_0 + \sigma\boldsymbol{\mathcal{G}}$, where $\mathrm{vec}(\boldsymbol{\mathcal{G}})$ is a standard normal vector and σ is the standard deviation parameter. Then

$$\hat{\boldsymbol{\mathcal{X}}} := \arg\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{Z}}_0 - \boldsymbol{\mathcal{X}}\|_F \quad \text{s.t.} \quad \sum_{i=1}^{K}\left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_* \leq Kr = \sum_{i=1}^{K}\left\|(\boldsymbol{\mathcal{X}}_0)_{(i)}\right\|_*, \tag{2.3.22}$$

can be computed as an estimate of $\boldsymbol{\mathcal{X}}_0$. Due to the recent results from Oymak and Hassibi [OH16], the normalized mean-squared error (NMSE), defined as

$$\mathsf{NMSE}(\sigma) := \frac{\mathbb{E}\left[\left\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_0\right\|_F^2\right]}{\sigma^2}, \tag{2.3.23}$$

is a decreasing function over $\sigma > 0$ and

$$\delta(\mathcal{C}) := \lim_{\sigma \to 0^+} \mathsf{NMSE}(\sigma). \tag{2.3.24}$$

Therefore, for small σ, NMSE serves a good estimator for $\delta(\mathcal{C})$. For more discussions on related tensor denoising problems, see Section 2.5.

In our experiment, we set $\sigma = 10^{-8}$ and for different triples of (K, r, n), we measure the empirical NMSE averaged over 10 repeats. Dykstra's Algorithm (see Section 2.8.1) is exploited to solve the convex problem (2.3.22). Numerical outputs are presented in Figure 2.2, which firmly conforms to our theoretical results displayed in (2.3.21).

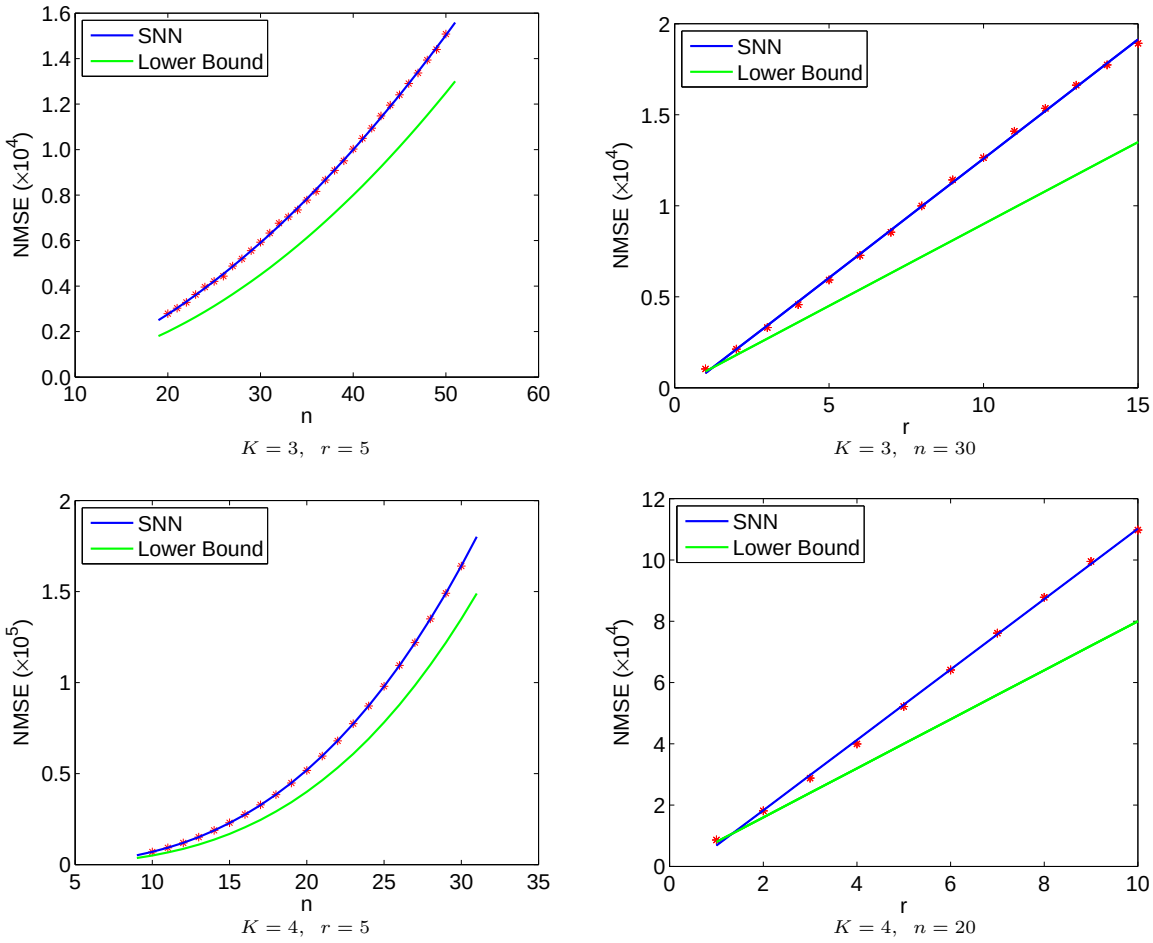


Figure 2.2: Lower bound for statistical dimension. Each red cross represents the empirical estimate of $\delta(\mathcal{C})$ for one particular triple (K, r, n). The blue curves fit the red dots based on the relationship $\delta(\mathcal{C}) = \Theta(rn^{K-1})$. In specific, in the left top (resp. left bottom) figure, we fit the red crosses with a quadratic (resp. cubic) curve; and in the right figures, we fit the red crosses with linear curves. Note that the red crosses fit pretty well with the blue curves, which is consistent with our result that $\delta(\mathcal{C}) = \Theta(rn^{K-1})$. The blue curves correspond to our lower bound $rn^{K-1} - 2$, which tightly lie below the red crosses. This empirically corroborates the lower bound result $\delta(\mathcal{C}) \geq rn^{K-1} - 2$.

2.4 A Better Convexification: Square Deal

The number of measurements promised by Corollary 2.6 and Theorem 2.7 is actually the same (up to constants) as the number of measurements required to recover a tensor $\boldsymbol{\mathcal{X}}_0$ which is low-rank along just one mode. Since matrix nuclear norm minimization correctly recovers a $n_1 \times n_2$ matrix of rank r when $m \geq Cr(n_1 + n_2)$ [CRPW12], solving

$$\text{minimize } \|\boldsymbol{\mathcal{X}}_{(1)}\|_* \quad \text{subject to} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0] \tag{2.4.1}$$

also recovers $\boldsymbol{\mathcal{X}}_0$ w.h.p. when $m \geq Crn^{K-1}$.

This suggests a more mundane explanation for the difficulty with (2.3.1): the term rn^{K-1} comes from the need to reconstruct the right singular vectors of the $n \times n^{K-1}$ matrix $\boldsymbol{\mathcal{X}}_{(1)}$. If we had some way of matricizing a tensor that *produced a more balanced (square) matrix* and also *preserved the low-rank property*, we could remedy this effect, and reduce the overall sampling requirement. In fact, this is possible when the order K of $\boldsymbol{\mathcal{X}}_0$ is four or larger.

Square reshaping. For $\boldsymbol{A} \in \mathbb{R}^{m_1 \times n_1}$, and integers m_2 and n_2 satisfying $m_1 n_1 = m_2 n_2$, the reshaping operator $\mathsf{reshape}(\boldsymbol{A}, m_2, n_2)$ returns an $m_2 \times n_2$ matrix whose elements are taken columnwise from $\boldsymbol{A}$. This operator rearranges elements in $\boldsymbol{A}$ and leads to a matrix of different shape. In the following, we reshape matrix $\boldsymbol{\mathcal{X}}_{(1)}$ to a more square matrix while preserving the low-rank property. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_K}$. Select some $j \in [K]$. Then we define matrix $\boldsymbol{\mathcal{X}}_{[j]}$ as[5]

$$\boldsymbol{\mathcal{X}}_{[j]} = \mathsf{reshape}\Big(\boldsymbol{\mathcal{X}}_{(1)}, \prod_{i=1}^{j} n_i, \prod_{i=j+1}^{K} n_i\Big). \tag{2.4.2}$$

We can view $\boldsymbol{\mathcal{X}}_{[j]}$ as a natural generalization of the standard tensor matricization. When $j = 1$, $\boldsymbol{\mathcal{X}}_{[j]}$ is nothing but $\boldsymbol{\mathcal{X}}_{(1)}$. However, when some $j > 1$ is selected, $\boldsymbol{\mathcal{X}}_{[j]}$ could become a more balanced matrix. This reshaping also preserves some of the algebraic structures of $\boldsymbol{\mathcal{X}}$. In particular, the following lemma can be easily obtained based on Prop. 3.7 of [Kol06]:

Lemma 2.15 *Let* $\mathrm{rank_{tc}}(\boldsymbol{\mathcal{X}}) = (r_1, r_2, \cdots, r_K)$, *and* $\mathrm{rank_{cp}}(\boldsymbol{\mathcal{X}}) = r_{\mathrm{cp}}$. *Then* $\mathrm{rank}(\boldsymbol{\mathcal{X}}_{[j]}) \leq r_{\mathrm{cp}}$, *and* $\mathrm{rank}(\boldsymbol{\mathcal{X}}_{[j]}) \leq \min\Big\{ \prod_{i=1}^{j} r_i, \prod_{i=j+1}^{K} r_i \Big\}$.

Thus, $\boldsymbol{\mathcal{X}}_{[j]}$ is not only more balanced but also maintains the low-rank property of the tensor $\boldsymbol{\mathcal{X}}$, which motivates us to recover $\boldsymbol{\mathcal{X}}_0$ by solving

$$\text{minimize} \quad \big\|\boldsymbol{\mathcal{X}}_{[j]}\big\|_* \quad \text{subject to} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0]. \tag{2.4.3}$$

[5] One can also think of (2.4.2) as embedding the tensor $\boldsymbol{\mathcal{X}}$ into the matrix $\boldsymbol{\mathcal{X}}_{[j]}$ as follows: $\boldsymbol{\mathcal{X}}_{i_1, i_2, \cdots, i_K} = \big(\boldsymbol{\mathcal{X}}_{[j]}\big)_{a,b}$, where

$$\begin{aligned} a &= 1 + \sum_{m=1}^{j} \left((i_m - 1) \prod_{l=1}^{m-1} n_l \right) \\ b &= 1 + \sum_{m=j+1}^{K} \left((i_m - 1) \prod_{l=j+1}^{m-1} n_l \right). \end{aligned}$$

Using Lemma 2.15 and [CRPW12], we can prove that this relaxation exactly recovers $\boldsymbol{\mathcal{X}}_0$, when the number of measurements is sufficiently large:

Theorem 2.16 *Consider a K-way tensor with the same length (say n) along each mode. (1) If $\boldsymbol{\mathcal{X}}_0$ has CP rank r, using*

$$\text{minimize} \quad \|\boldsymbol{\mathcal{X}}_{\square}\|_* \quad \text{subject to} \quad \mathcal{G}[\boldsymbol{\mathcal{X}}] = \mathcal{G}[\boldsymbol{\mathcal{X}}_0], \tag{2.4.4}$$

with $\boldsymbol{\mathcal{X}}_{\square} = \boldsymbol{\mathcal{X}}_{[\lceil \frac{K}{2} \rceil]}$, $m \geq Crn^{\lceil \frac{K}{2} \rceil}$ is sufficient to recover $\boldsymbol{\mathcal{X}}_0$ with high probability. (2) If $\boldsymbol{\mathcal{X}}_0$ has Tucker rank $(r, r, \cdots, r)$, using (2.4.4), $m \geq Cr^{\lfloor \frac{K}{2} \rfloor} n^{\lceil \frac{K}{2} \rceil}$ is sufficient to recover $\boldsymbol{\mathcal{X}}_0$ with high probability.

The number of measurements $O(r^{\lfloor \frac{K}{2} \rfloor} n^{\lceil \frac{K}{2} \rceil})$ required to recover $\boldsymbol{\mathcal{X}}$ with square reshaping (2.4.4), is always within a constant of the number $O(rn^{K-1})$ with the sum-of-nuclear-norms model, and is significantly smaller when r is small and $K \geq 4$. E.g., we obtain an improvement of a multiplicative factor of $n^{\lfloor K/2 \rfloor - 1}$ when r is a constant. This is a significant improvement.

Remark 2.17 *Recall the generalized tensor matricization discussed in Chapter 1. $\boldsymbol{\mathcal{X}}_{[j]}$ is a special case of $\boldsymbol{\mathcal{X}}_{(\mathcal{R} \times \mathcal{C})}$ with $\mathcal{R} = [j]$ and $\mathcal{C} = [K] \backslash [j]$. In general, we can pick up the partition $\{\mathcal{R}, \mathcal{C}\}$ of $[K]$ flexibly based on the prior information and physical meaning of the underlying tensor to achieve the best recovery performance.*

Low-rank tensor completion. We corroborate the improvement of square reshaping with numerical experiments on *low-rank tensor completion (LRTC)*. LRTC attempts to reconstruct the low-rank tensor $\boldsymbol{\mathcal{X}}_0$ from a subset Ω of its entries. By imposing appropriate incoherence conditions, it is possible to prove exact recovery guarantees for both our square model [Gro11] and the SNN model [HMGW14] for LRTC. However, unlike the recovery problem under Gaussian random measurements, due to the lack of sharp bounds, it is more difficult to establish a negative result for the SNN model (like Theorem 2.7). Nonetheless, numerical results below clearly indicate the advantage of our square approach, complementing our theoretical results established in previous sections.

We generate our four-way tensors $\boldsymbol{\mathcal{X}}_0 \in \mathbb{R}^{n \times n \times n \times n}$ as $\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mathcal{C}}_0 \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3 \times_4 \boldsymbol{U}_4$, where $\boldsymbol{\mathcal{C}}_0 \in \mathbb{R}^{r_1 \times r_2 \times r_3 \times r_4}$ and $\boldsymbol{U}_i \in \mathbb{R}^{n_i \times r_i}$ for each $i \in [4]$ are constructed under the random Gaussian models (by MATLAB command): each entry of $\boldsymbol{C}_0$, $\boldsymbol{U}_1$, $\boldsymbol{U}_2$, $\boldsymbol{U}_3$ and $\boldsymbol{U}_4$ is generated using `randn()`. The observed entries are chosen uniformly with ratio ρ. We compare the recovery performances between

$$\text{minimize}_{\boldsymbol{\mathcal{X}}} \quad \sum_{i=1}^{K} \|\boldsymbol{\mathcal{X}}_{(i)}\|_* \quad \text{subject to} \quad \mathcal{P}_{\Omega}[\boldsymbol{\mathcal{X}}] = \mathcal{P}_{\Omega}[\boldsymbol{\mathcal{X}}_0], \quad \text{and} \tag{2.4.5}$$

$$\text{minimize}_{\boldsymbol{\mathcal{X}}} \qquad \|\boldsymbol{\mathcal{X}}_{\square}\|_* \qquad \text{subject to} \qquad \mathcal{P}_{\Omega}[\boldsymbol{\mathcal{X}}] = \mathcal{P}_{\Omega}[\boldsymbol{\mathcal{X}}_0]. \tag{2.4.6}$$

We fix (r_1, r_2, r_3, r_4) as $(1, 1, 1, 1)$ and $(1, 1, 2, 2)$ respectively. For each choice of (r_1, r_2, r_3, r_4), we increase the problem size n from 10 to 30 with increment 1, and the observation ratio ρ from 0.01 to 0.2 with increment 0.01. For each (ρ, n)-pair, we simulate 10 test instances and declare a trial to be successful if the recovered $\boldsymbol{\mathcal{X}}^\star$ satisfies $\|\boldsymbol{\mathcal{X}}^\star - \boldsymbol{\mathcal{X}}_0\|_F / \|\boldsymbol{\mathcal{X}}_0\|_F \leq 10^{-2}$.

The optimization problems are solved using efficient first-order methods. Since (2.4.6) is equivalent to standard matrix completion, we use the existing solver ALM [LCM10]. For the sum of nuclear norms minimization (2.4.5), we implement the Douglas-Rachford algorithm (see Appendix 2.8.2 for details).

Figure 2.3 plots the fraction of correct recovery for each pair. Clearly, the square approach succeeds in a much larger region.

2.5 Tensor denoising

A classical problem in statistical inference is to estimate the target signal with Gaussian perturbed observations. Here, we briefly discuss this denoising problem under the context of low-rank tensors.

In specific, the target signal is a low-rank tensor (in terms of Tucker rank), say $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$, and we observe $\boldsymbol{\mathcal{Z}}_0 = \boldsymbol{\mathcal{X}}_0 + \sigma\boldsymbol{\mathcal{G}}$, where $\text{vec}(\boldsymbol{\mathcal{G}})$ is a standard norm vector and σ is an unknown standard deviation parameter. To estimate $\boldsymbol{\mathcal{X}}_0$, a natural way is to solve the following convex optimization problem[6]

$$\widehat{\boldsymbol{\mathcal{X}}}_\tau \quad := \quad \arg\min_{\boldsymbol{\mathcal{X}}} \ \|\boldsymbol{\mathcal{Z}}_0 - \boldsymbol{\mathcal{X}}\|_F \quad \text{s.t.} \quad f(\boldsymbol{\mathcal{X}}_0) \leq \tau, \tag{2.5.2}$$

where f is a convex function promoting the low-rank tensor structure, and $\tau > 0$ balances the structural penalty and the date fidelity term.

One way to evaluate the denoising performance of this convex regularizer f is to measure the minimax normalized mean-squared-error (NMSE) risks, defined as

$$R_{mm}\left(f, f(\boldsymbol{\mathcal{X}}_0)\right) \quad := \quad \sup_{\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r, \sigma > 0} \ \frac{1}{\sigma^2}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{X}}}_{f(\boldsymbol{\mathcal{X}}_0)} - \boldsymbol{\mathcal{X}}_0\right\|_F^2\right], \tag{2.5.3}$$

$$R_{mm}\left(f\right) \quad := \quad \sup_{\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r, \sigma > 0} \ \inf_{\tau > 0} \ \frac{1}{\sigma^2}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{X}}}_{\tau} - \boldsymbol{\mathcal{X}}_0\right\|_F^2\right], \tag{2.5.4}$$

[6] Problem (2.5.2) is equivalent to its Lagrangian formulation

$$\min_{\boldsymbol{\mathcal{X}}} \ \frac{1}{2}\|\boldsymbol{\mathcal{Z}}_0 - \boldsymbol{\mathcal{X}}\|_F^2 + \lambda f(\boldsymbol{\mathcal{X}}_0) \tag{2.5.1}$$

with a proper choice of $\lambda \geq 0$.

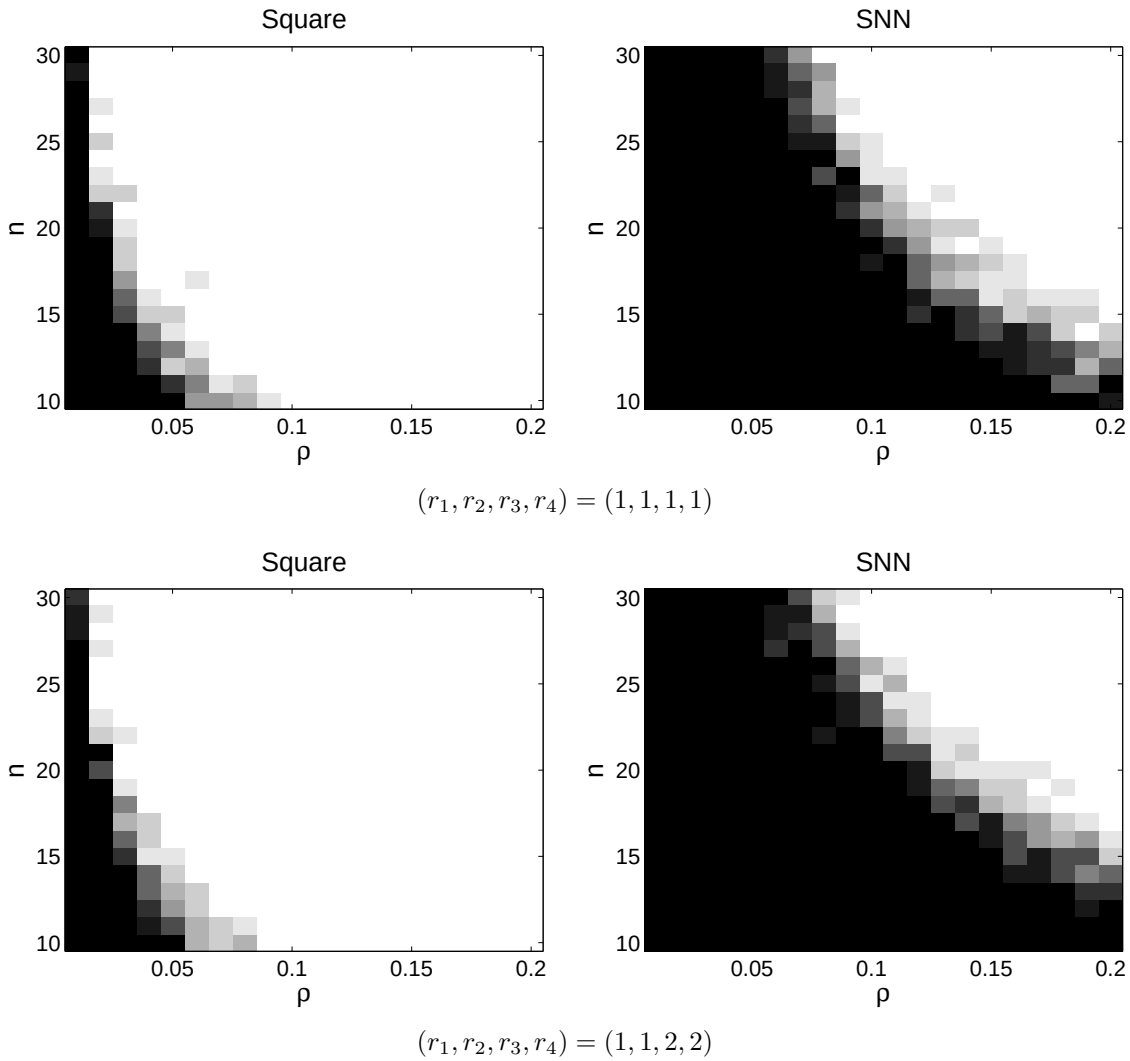


Figure 2.3: Tensor completion with Gaussian random data. The colormap indicates the fraction of instances that are correctly recovered for each (ρ, n)-pair, which increases with brightness from 100% failure (black) to 100% success (white).

i.e. the risk corresponds to the normalized mean-squared error (NMSE) for either the fixed oracle value $\tau = f(\boldsymbol{\mathcal{X}}_0)$ or the best tuned $\boldsymbol{\tau}$, at worst choices of the underlying signal $\boldsymbol{\mathcal{X}}_0$ and the noise level σ. Due to the general result proved by Oymak and Hassibi [OH16, Theorem 3.1], quantities in (2.5.3) and (2.5.4) are closely related with statistical dimension,

$$R_{mm}\left(f, f(\boldsymbol{\mathcal{X}}_0)\right) = \sup_{\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r} \delta\left(\mathcal{C}\left(f, \boldsymbol{\mathcal{X}}_0\right)\right) \quad \text{and} \quad R_{mm}\left(f\right) = \sup_{\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r} \delta\left(\mathcal{C}\left(f, \boldsymbol{\mathcal{X}}_0\right)\right) - O(n^{K/2}), \tag{2.5.5}$$

where we recall that $\mathcal{C}\left(f, \boldsymbol{\mathcal{X}}_0\right)$ denotes the descent cone of f at the point $\boldsymbol{\mathcal{X}}_0$. Based on this result, we can

Table 2.2: Minimax NMSE risks of different convex regularizers for the low-rank tensor estimation. Note that the risks for the Single Norm model and the SNN model are essentially on the same order, which is substantially higher than the one for the Square model. This can be viewed as a dual phenomenon of our results (Theorem 2.7 and Theorem 2.16) regarding the exact low-rank tensor recovery using generic measurements. Both of these two results arise from the study on the statistical dimension of the descent cone of certain convex function f at the target signal $\boldsymbol{x}_0$.

Model	**Convex regularizer** $f(\cdot)$	$R_{mm,f(\boldsymbol{\mathcal{X}}_0)}(f)$	$R_{mm}(f)$
Single Norm	$\left\|\boldsymbol{\mathcal{X}}_{(1)}\right\|_*$	$\Theta(rn^{K-1})$	$\Theta(rn^{K-1})$
SNN	$\sum_{i\in[K]}\lambda_i\left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_*$	$\Theta(rn^{K-1})$	$\Theta(rn^{K-1})$
Square	$\left\|\boldsymbol{\mathcal{X}}_{\square}\right\|_*$	$\Theta(r^{\lfloor\frac{K}{2}\rfloor}n^{\lceil\frac{K}{2}\rceil})$	$\Theta(r^{\lfloor\frac{K}{2}\rfloor}n^{\lceil\frac{K}{2}\rceil})$

easily characterize the Minimax MSE risks of several convex functions f discussed in this chapter (see Table 2.2). [7]

To empirically verify the results in Table 2.2, we construct $\boldsymbol{\mathcal{X}}_0$ as a 4-mode $n\times n\times n\times n$ (super)diagonal tensor with only the first r diagonal entries as 1 and 0 elsewhere, and choose $\sigma=10^{-8}$. Convex regularizers $f(\cdot)$ listed in Table 2.2: $\left\|\boldsymbol{\mathcal{X}}_{(1)}\right\|_*$, $\sum_{i\in[K]}\left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_*$, and $\|\boldsymbol{\mathcal{X}}_{\square}\|_*$, are respectively tested. For different pairs (r,n), we compute the empirical NMSE by averaging $\frac{1}{\sigma^2}\left\|\widehat{\boldsymbol{\mathcal{X}}}_{f(\boldsymbol{\mathcal{X}}_0)}-\boldsymbol{\mathcal{X}}_0\right\|_F^2$ over 10 repeats. Curves are fitted based on the complexities displayed in Table 2.2. It can be clearly observed that the obtained curves fit the empirical NMSE quite tightly.

2.6 Proofs for Section 2.2

Proof of Lemma 2.3. The arguments we used below are primarily adapted from [ENP12], where their interest is to establish the number of Gaussian measurements required to recover a low rank matrix by rank minimization.

Notice that every $\boldsymbol{\mathcal{D}}\in\mathfrak{S}_{2r}$, and every i, $\langle\boldsymbol{\mathcal{G}}_i,\boldsymbol{\mathcal{D}}\rangle$ is a standard Gaussian random variable, and so

$$\forall\, t>0,\quad \mathbb{P}\left[|\langle\boldsymbol{\mathcal{G}}_i,\boldsymbol{\mathcal{D}}\rangle|<t\right]<2t\cdot\frac{1}{\sqrt{2\pi}}=t\sqrt{\frac{2}{\pi}}. \tag{2.6.1}$$

[7] For the SNN model ($f=\sum_{i\in[K]}\lambda_i\left\|\boldsymbol{\mathcal{X}}_{(i)}\right\|_*$), we also have λ_i's to choose, and so $R_{mm}\left(f,f(\boldsymbol{\mathcal{X}}_0)\right)$ and $R_{mm}\left(f\right)$ are instead naturally defined as

$$R_{mm}\left(f,f(\boldsymbol{\mathcal{X}}_0)\right) := \sup_{\boldsymbol{\mathcal{X}}_0\in\mathfrak{T}_r,\sigma>0}\ \inf_{\boldsymbol{\lambda}>\mathbf{0}}\ \frac{1}{\sigma^2}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{X}}}_{f(\boldsymbol{\mathcal{X}}_0)}-\boldsymbol{\mathcal{X}}_0\right\|_F^2\right], \tag{2.5.6}$$

$$R_{mm}\left(f\right) := \sup_{\boldsymbol{\mathcal{X}}_0\in\mathfrak{T}_r,\sigma>0}\ \inf_{\tau>0,\lambda>0}\ \frac{1}{\sigma^2}\mathbb{E}\left[\left\|\widehat{\boldsymbol{\mathcal{X}}}_{\tau}-\boldsymbol{\mathcal{X}}_0\right\|_F^2\right]. \tag{2.5.7}$$

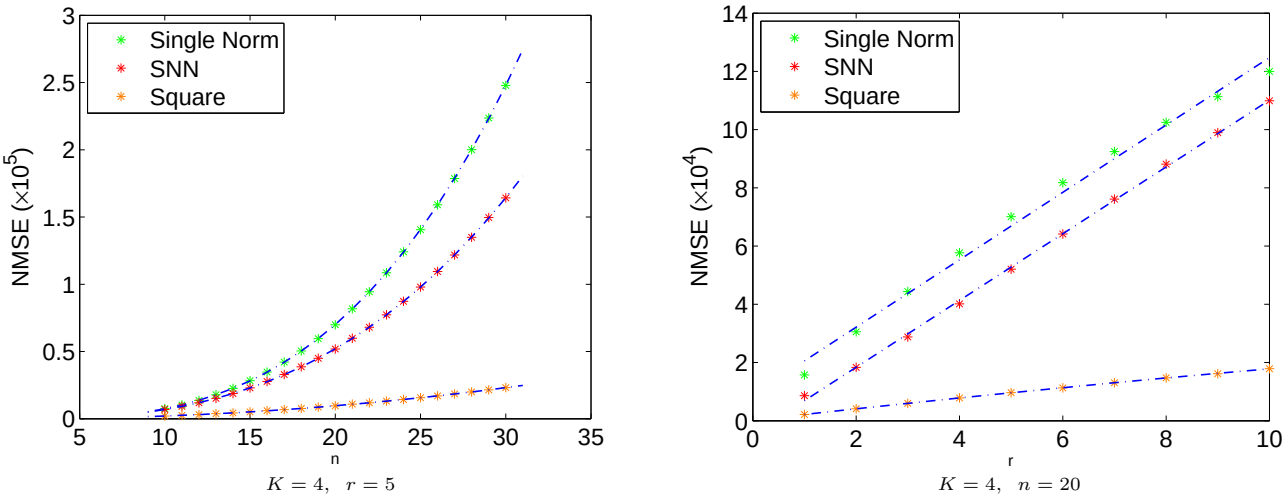


Figure 2.4: Tensor denoising. Each cross corresponds to the empirical estimate of NMSE for a (r, n)-pair. Different colors are used to indicate different convex models. The blue dashed lines are fitted using polynomials consistent with the complexities displayed in Table 2.2.

Let $\mathfrak{N}$ be an ε-net for $\mathfrak{S}_{2r}$ in terms of $\|\cdot\|_F$. Because the measurements are independent, for any fixed $\bar{\boldsymbol{\mathcal{D}}} \in \mathfrak{S}_{2r}$,

$$\mathbb{P}\left[\left\|\mathcal{G}[\bar{\boldsymbol{\mathcal{D}}}]\right\|_\infty < t\right] \;<\; \left(t\sqrt{2/\pi}\right)^m. \tag{2.6.2}$$

Moreover, for any $\boldsymbol{\mathcal{D}} \in \mathfrak{S}_{2r}$, we have

$$\begin{aligned} \|\mathcal{G}[\boldsymbol{\mathcal{D}}]\|_\infty &\geq \max_{\bar{\boldsymbol{\mathcal{D}}}\in\mathfrak{N}} \left\{ \left\|\mathcal{G}[\bar{\boldsymbol{\mathcal{D}}}]\right\|_\infty - \|\mathcal{G}\|_{F\to\infty} \left\|\bar{\boldsymbol{\mathcal{D}}} - \boldsymbol{\mathcal{D}}\right\|_F \right\} && (2.6.3)\\ &\geq \min_{\bar{\boldsymbol{\mathcal{D}}}\in\mathfrak{N}} \left\{\left\|\mathcal{G}[\bar{\boldsymbol{\mathcal{D}}}]\right\|_\infty\right\} - \varepsilon \|\mathcal{G}\|_{F\to\infty}. && (2.6.4) \end{aligned}$$

Hence,

$$\begin{aligned} &\mathbb{P}\left[\inf_{\boldsymbol{\mathcal{D}}\in\mathfrak{S}_{2r}} \|\mathcal{G}[\boldsymbol{\mathcal{D}}]\|_\infty < \varepsilon\log(1/\varepsilon)\right]\\ &\quad\leq \mathbb{P}\left[\min_{\boldsymbol{\mathcal{D}}\in\mathfrak{N}} \|\mathcal{G}[\boldsymbol{\mathcal{D}}]\|_\infty < 2\varepsilon\log(1/\varepsilon)\right] + \mathbb{P}\left[\|\mathcal{G}\|_{F\to\infty} > \log(1/\varepsilon)\right]\\ &\quad\leq \#\mathfrak{N} \times \left(2\sqrt{2/\pi}\times\varepsilon\log(1/\varepsilon)\right)^m + \mathbb{P}\left[\|\mathcal{G}\|_{F\to\infty} > \log(1/\varepsilon)\right]\\ &\quad\leq \beta^d (2\sqrt{2/\pi})^m \varepsilon^{m-d}\log(1/\varepsilon)^m + \mathbb{P}\left[\|\mathcal{G}\|_{F\to\infty} > \log(1/\varepsilon)\right]. \end{aligned} \tag{2.6.5}$$

Since $m \geq d+1$, (2.6.5) goes to zero as $\varepsilon \searrow 0$. Hence, taking a sequence of decreasing ε, we can show that $\mathbb{P}\left[\inf_{\boldsymbol{\mathcal{D}}\in\mathfrak{S}_{2r}} \|\mathcal{G}[\boldsymbol{\mathcal{D}}]\|_\infty = 0\right] \leq t$ for every positive t, establishing the result.

Proof of Lemma 3.8. This follows from the basic fact that for any tensor $\boldsymbol{\mathcal{X}}$ and matrix $\boldsymbol{U}$ of compatible size,

$$\|\boldsymbol{\mathcal{X}} \times_k \boldsymbol{U}\|_F = \left\|\boldsymbol{U}\boldsymbol{\mathcal{X}}_{(k)}\right\|_F \leq \|\boldsymbol{U}\| \left\|\boldsymbol{\mathcal{X}}_{(k)}\right\|_F = \|\boldsymbol{U}\| \|\boldsymbol{\mathcal{X}}\|_F . \tag{2.6.6}$$

Write

$$\begin{aligned}
&\left\|[[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]] - [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \ldots, \boldsymbol{U}'_K]]\right\|_F \\
&\quad \leq \quad \left\|[[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]] - [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]]\right\|_F \\
&\qquad + \quad \left\|\sum_{i=1}^{K} [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \ldots, \boldsymbol{U}'_i, \boldsymbol{U}_{i+1}, \ldots \boldsymbol{U}_k]] - [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \ldots, \boldsymbol{U}'_{i-1}, \boldsymbol{U}_i, \ldots \boldsymbol{U}_K]]\right\|_F \\
&\quad \leq \quad \left\|\boldsymbol{\mathcal{C}} - \boldsymbol{\mathcal{C}}'\right\|_F + \sum_{i=1}^{K} \|\boldsymbol{U}_i - \boldsymbol{U}'_i\|,
\end{aligned}$$

where the first inequality follows from triangle inequality and the second inequality follows from the fact that $\|\boldsymbol{\mathcal{C}}\|_F = 1$, $\|\boldsymbol{U}_j\| = 1$, $\boldsymbol{U}_i^* \boldsymbol{U}_i = \boldsymbol{I}$ and $\boldsymbol{U}_i'^* \boldsymbol{U}'_i = \boldsymbol{I}$.

Proof of Lemma 2.5. The idea of this proof is to construct a net for each component of the Tucker decomposition and then combine those nets to form a *compound* net with the desired cardinality.

Denote $\mathfrak{C} = \{\boldsymbol{\mathcal{C}} \in \mathbb{R}^{2r \times 2r \times \cdots \times 2r} \mid \|\boldsymbol{\mathcal{C}}\|_F = 1\}$ and $\mathcal{O} = \{\boldsymbol{U} \in \mathbb{R}^{n \times r} \mid \boldsymbol{U}^* \boldsymbol{U} = \boldsymbol{I}\}$. Clearly, for any $\boldsymbol{\mathcal{C}} \in \mathfrak{C}$, $\|\boldsymbol{\mathcal{C}}\|_F = 1$, and for any $\boldsymbol{U} \in \mathcal{O}$, $\|\boldsymbol{U}\| = 1$. Thus by [Ver07, Prop. 4] and [Ver12, Lemma 5.2], there exists an $\frac{\varepsilon}{K+1}$-net $\mathfrak{C}'$ covering $\mathfrak{C}$ with respect to the Frobenius norm such that $\#\mathfrak{C}' \leq (\frac{3(K+1)}{\varepsilon})^{(2r)^K}$, and there exists an $\frac{\varepsilon}{K+1}$-net $\mathcal{O}'$ covering $\mathcal{O}$ with respect to the operator norm such that $\#\mathcal{O}' \leq (\frac{3(K+1)}{\varepsilon})^{2nr}$. Construct

$$\mathfrak{S}'_{2r} = \{[[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \ldots, \boldsymbol{U}'_K]] \mid \boldsymbol{\mathcal{C}}' \in \mathfrak{C}', \ \boldsymbol{U}'_i \in \mathcal{O}'\}. \tag{2.6.7}$$

Clearly $\#\mathfrak{S}'_{2r} \leq \left(\frac{3(K+1)}{\varepsilon}\right)^{(2r)^K + 2nrK}$. The rest is to show that $\mathfrak{S}'_{2r}$ is indeed an ε-net covering $\mathfrak{S}_{2r}$ with respect to the Frobenius norm.

For any fixed $\boldsymbol{\mathcal{D}} = [[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \cdots, \boldsymbol{U}_K]] \in \mathfrak{S}_{2r}$ where $\boldsymbol{\mathcal{C}} \in \mathfrak{C}$ and $\boldsymbol{U}_i \in \mathcal{O}$, by our constructions above, there exist $\boldsymbol{\mathcal{C}}' \in \mathfrak{C}'$ and $\boldsymbol{U}'_i \in \mathcal{O}'$ such that $\left\|\boldsymbol{\mathcal{C}} - \boldsymbol{\mathcal{C}}'\right\|_F \leq \frac{3(K+1)}{\varepsilon}$ and $\|\boldsymbol{U}_i - \boldsymbol{U}'_i\| \leq \frac{3(K+1)}{\varepsilon}$. Then $\boldsymbol{\mathcal{D}}' = [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \cdots, \boldsymbol{U}'_K]] \in \mathfrak{S}'_{2r}$ is within ε-distance from $\boldsymbol{\mathcal{D}}$, since by the triangle inequality derived in Lemma 2, we have

$$\left\|\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{D}}'\right\|_F = \left\|[[\boldsymbol{\mathcal{C}}; \boldsymbol{U}_1, \ldots, \boldsymbol{U}_K]] - [[\boldsymbol{\mathcal{C}}'; \boldsymbol{U}'_1, \ldots, \boldsymbol{U}'_K]]\right\|_F \leq \left\|\boldsymbol{\mathcal{C}} - \boldsymbol{\mathcal{C}}'\right\|_F + \sum_{i=1}^{K} \|\boldsymbol{U}_i - \boldsymbol{U}'_i\| \leq \varepsilon. \tag{2.6.8}$$

This completes the proof.

2.7 Proofs for Section 2.3

Proof of Corollary 2.10. Denote $\lambda = \delta(\mathcal{C}) - m$. Then following the result derived by Amelunxen et al. [ALMT14, Theorem 7.2], we have

$$\begin{aligned}\mathbb{P}\left[\mathcal{C} \cap \mathrm{null}(\mathcal{G}) = \{\mathbf{0}\}\right] &\leq 4\exp\left(-\frac{\lambda^2/8}{\min\{\delta(\mathcal{C}), \delta(\mathcal{C}^\circ)\} + \lambda}\right) \\ &\leq 4\exp\left(-\frac{\lambda^2/8}{\delta(\mathcal{C}) + \lambda}\right) \\ &\leq 4\exp\left(-\frac{(\delta(\mathcal{C}) - m)^2}{16\delta(\mathcal{C})}\right).\end{aligned} \tag{2.7.1}$$

Proof of Lemma 2.11. Denote $\mathrm{circ}(\boldsymbol{e}_n, \theta)$ as $\mathrm{circ}_n(\theta)$, where $\boldsymbol{e}_n$ is the nth standard basis for $\mathbb{R}^n$. Since $\delta\big(\mathrm{circ}(\boldsymbol{x}_0, \theta)\big) = \delta\big(\mathrm{circ}(\boldsymbol{e}_n, \theta)\big)$, it is sufficient to prove $\delta\big(\mathrm{circ}_n(\theta)\big) \leq n\sin^2\theta + 2$.

Let us first consider the case where n is *even*. Define a discrete random variable V supported on $\{0, 1, 2, \cdots, n\}$ with probability mass function $\mathbb{P}[V = k] = v_k$. Here v_k denotes the k-th intrinsic volumes of $\mathrm{circ}_n(\theta)$. Then it can be verified [see Ame11, Ex. 4.4.8]

$$v_k = \frac{1}{2}\binom{\frac{1}{2}(n-2)}{\frac{1}{2}(k-1)}\sin^{k-1}(\theta)\cos^{n-k-1}(\theta) \qquad \text{for } k = 1, 2, \cdots, n-1. \tag{2.7.2}$$

From Prop. 5.11 of [ALMT14], we know that

$$\delta\big(\mathrm{circ}_n(\theta)\big) = \mathbb{E}[V] = \sum_{k=1}^{n}\mathbb{P}[V \geq k]. \tag{2.7.3}$$

Moreover, by the interlacing result [ALMT14, Prop. 5.6] and the fact that $\mathbb{P}[V \geq 2k] = \mathbb{P}[V \geq 2k-1] - \mathbb{P}[V = 2k-1]$, we have

$$\begin{aligned}
\mathbb{P}[V \geq 1] &\leq 2\mathbb{P}[V = 1] + 2\mathbb{P}[V = 3] + \cdots + 2\mathbb{P}[V = n-1], \\
\mathbb{P}[V \geq 2] &\leq \mathbb{P}[V = 1] + 2\mathbb{P}[V = 3] + \cdots + 2\mathbb{P}[V = n-1]; \\
\\
\mathbb{P}[V \geq 3] &\leq 2\mathbb{P}[V = 3] + 2\mathbb{P}[V = 5] + \cdots + 2\mathbb{P}[V = n-1], \\
\mathbb{P}[V \geq 4] &\leq \mathbb{P}[V = 3] + 2\mathbb{P}[V = 5] + \cdots + 2\mathbb{P}[V = n-1]; \\
\\
\vdots \quad & \quad \vdots \quad \vdots \\
\\
\mathbb{P}[V \geq n-1] &\leq 2\mathbb{P}[V = n-1], \\
\mathbb{P}[V \geq n] &\leq \mathbb{P}[V = n-1].
\end{aligned}$$

Summing up the above inequalities, we have

$$\begin{aligned}\mathbb{E}[V] &= \sum_{k=1}^{n}\mathbb{P}[V\geq k] \\ &\leq \sum_{k=1,3,\cdots,n-1} 2(k-1)v_k + \sum_{k=1,3,\cdots,n-1} 3v_k \\ &\leq (n-2)\sin^2\theta + \frac{3}{2}\sum_{k=0}^{n} v_k \\ &\leq (n-2)\sin^2\theta + \frac{3}{2} \;=\; n\sin^2\theta + 2\cos^2\theta - \frac{1}{2},\end{aligned} \tag{2.7.4}$$

where the second last inequality follows from the observations that $\sum_{k=1,3,\cdots,n-1}\frac{k-1}{2}\cdot(2v_k) = \mathbb{E}\left[Bin(\frac{n-2}{2},\sin^2\theta)\right]$ and $\sum_{k=0}^{n} v_k \geq \sum_{k=1,3,\cdots,n-1} 2v_k$ again by the interlacing result [ALMT14, Prop. 5.6].

Suppose n is *odd*. Since the intersection of $\mathrm{circ}_{n+1}(\theta)$ with any n-dimensional linear subspace containing $\boldsymbol{e}_{n+1}$ is an isometric image of $\mathrm{circ}_n(\theta)$, by Prop. 4.1 of [ALMT14], we have

$$\delta(\mathrm{circ}_n(\theta)) = \delta(\mathrm{circ}_n(\theta)\times\{\mathbf{0}\}) \leq \delta(\mathrm{circ}_{n+1}(\theta)) \leq (n+1)\sin^2\theta + 2\cos^2\theta - \frac{1}{2} \leq n\sin^2\theta + \cos^2\theta + \frac{1}{2}. \tag{2.7.5}$$

Thus, taking both cases (n is even and n is odd) into consideration, we have

$$\delta\big(\mathrm{circ}_n(\theta)\big) \leq n\sin^2\theta + \cos^2\theta + \frac{1}{2} < n\sin^2\theta + 2. \tag{2.7.6}$$

Proof of Theorem 2.12. Notice that for any fixed $m>0$, the function $f:\ t\to 4\exp\left(-\frac{(t-m)^2}{16t}\right)$ is decreasing for $t\geq m$. Then due to Corollary 2.10 and the fact that $\delta(\mathcal{C})\geq\kappa-2\geq m$, we have

$$\begin{aligned}\mathbb{P}\left[\boldsymbol{x}_0 \text{ is the unique optimal solution to (2.3.2)}\right] &= \mathbb{P}\left[\mathcal{C}\cap\mathrm{null}(\mathcal{G})=\{\mathbf{0}\}\right] \\ &\leq 4\exp\left(-\frac{(\delta(\mathcal{C})-m)^2}{16\delta(\mathcal{C})}\right) \\ &\leq 4\exp\left(-\frac{(\kappa-m-2)^2}{16\,(\kappa-2)}\right).\end{aligned} \tag{2.7.7}$$

Proof of Theorem 2.13. The argument for Theorem 2.12 can be easily adapted to prove Theorem 2.13, with the following additional observation regarding the function $\|\mathcal{A}[\cdot]\|_\diamond$, where $\|\cdot\|_\diamond$ is a norm in $\mathbb{R}^m$ with dual norm $\|\cdot\|_\diamond^\circ$, and $\mathcal{A}:\mathbb{R}^n\to\mathbb{R}^m$ is a linear operator satisfying $\mathcal{A}[\boldsymbol{x}_0]\neq\mathbf{0}$. Essentially, we will next prove that $\partial\|\mathcal{A}[\cdot]\|_\diamond(\boldsymbol{x}_0)$ is contained in a circular cone, which is analogous to (2.3.12).

For any $\boldsymbol{u}\in\partial\|\mathcal{A}[\cdot]\|_\diamond(\boldsymbol{x}_0)$, there exists a $\boldsymbol{v}\in\partial\|\cdot\|_\diamond(\mathcal{A}[\boldsymbol{x}_0])$ such that $\boldsymbol{u}=\mathcal{A}^\star\boldsymbol{v}$. Thus we have

$$\cos(\angle(\boldsymbol{u},\boldsymbol{x}_0)) = \frac{\langle\boldsymbol{u},\boldsymbol{x}_0\rangle}{\|\boldsymbol{u}\|\,\|\boldsymbol{x}_0\|} = \frac{\langle\mathcal{A}^*\boldsymbol{v},\boldsymbol{x}_0\rangle}{\|\mathcal{A}^*\boldsymbol{v}\|\,\|\boldsymbol{x}_0\|} \geq \frac{\langle\boldsymbol{v},\mathcal{A}\boldsymbol{x}_0\rangle}{\|\mathcal{A}\|\,\|\boldsymbol{v}\|\,\|\boldsymbol{x}_0\|}. \tag{2.7.8}$$

Define $L := \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \|\boldsymbol{x}\|_\diamond / \|\boldsymbol{x}\|_2$, which implies that $\|\cdot\|_\diamond$ is L-Lipschitz: $\|\boldsymbol{x}\|_\diamond \leq L\|\boldsymbol{x}\|$ for all $\boldsymbol{x}$. Then $\|\boldsymbol{v}\| \leq L\|\boldsymbol{v}\|_\diamond^\circ$ for all $\boldsymbol{v}$ as well. Thus, we have

$$\cos(\angle(\boldsymbol{u},\boldsymbol{x}_0)) \geq \frac{\langle \boldsymbol{v}, \mathcal{A}\boldsymbol{x}_0\rangle}{L\|\mathcal{A}\|\,\|\boldsymbol{v}\|_\diamond^\circ\,\|\boldsymbol{x}_0\|}. \tag{2.7.9}$$

Recall that

$$\partial \|\cdot\|_\diamond(\boldsymbol{x}) = \left\{\boldsymbol{v} \mid \langle \boldsymbol{v},\boldsymbol{x}\rangle = \|\boldsymbol{x}\|_\diamond,\ \|\boldsymbol{v}\|_\diamond^\circ \leq 1\right\}. \tag{2.7.10}$$

We can therefore further simplify

$$\cos(\angle(\boldsymbol{u},\boldsymbol{x}_0)) \geq \frac{\|\mathcal{A}\boldsymbol{x}_0\|_\diamond}{L\|\mathcal{A}\|\,\|\boldsymbol{x}_0\|}, \tag{2.7.11}$$

which is equivalent to saying

$$\partial \|\mathcal{A}[\cdot]\|_\diamond(\boldsymbol{x}_0) \subseteq \mathrm{circ}(\boldsymbol{x}_0,\theta), \tag{2.7.12}$$

with $\theta := \cos^{-1}\left(\frac{\|\mathcal{A}\boldsymbol{x}_0\|_\diamond}{L\|\mathcal{A}\|\|\boldsymbol{x}_0\|}\right)$.

2.8 First-Order Methods for Problems (2.3.22) and (2.4.5)

In the previous sections, we have proposed and analyzed several convex problems related with tensor recovery and denoising. Though these problems are computationally tractable in general, off-the-shelf convex software packages, e.g. CVX [GB14, GB08], may not tame the large-scale monster in the tensor domain. For example, a four-way tensor with length 30 along each mode, has amounted to nearly one million elements in store. Consequently, we leverage more scalable first-order methods to solve convex problems involved in the chapter. [8]

2.8.1 Dykstra's Algorithm for problem (2.3.22)

By splitting $\boldsymbol{\mathcal{X}}$ into $\{\boldsymbol{\mathcal{X}}_i\}_{i\in[K]}$, problem (2.3.22) can be reformulated as

$$\begin{aligned} \min_{\{\boldsymbol{\mathcal{X}}_i\}_{i\in[K]}} \quad & \sum_{i\in[K]} \left\|\boldsymbol{\mathcal{Z}}_0 - \boldsymbol{\mathcal{X}}_{(i)}\right\|_F^2 \\ \text{s.t.} \quad & \sum_{i\in[K]} \left\|(\boldsymbol{\mathcal{X}}_i)_{(i)}\right\|_* \leq \tau := Kr \end{aligned} \tag{2.8.1}$$

[8] MATLAB codes are available on CM's personal website: https://sites.google.com/site/mucun1988/. MATLAB Tensor Toolbox [BK15] has been utilized in our implementation.

$$\boldsymbol{\mathcal{X}}_1 = \boldsymbol{\mathcal{X}}_2 = \cdots = \boldsymbol{\mathcal{X}}_K \in \otimes_{i\in[K]}\mathbb{R}^n.$$

This is essentially to compute $\mathcal{P}_{\mathcal{C}_1\cap\mathcal{C}_2}[\boldsymbol{z}_0]$: the projection of $\boldsymbol{z}_0 := \underbrace{(\boldsymbol{\mathcal{Z}}_0, \boldsymbol{\mathcal{Z}}_0, \cdots, \boldsymbol{\mathcal{Z}}_0)}_{K\ times}$ onto the intersection of two closed convex sets $\mathcal{C}_1$ and $\mathcal{C}_2$ in the Hilbert space $\mathcal{H}$, where $\mathcal{H} := \bigtimes_{i\in[K]}\mathbb{R}^{n\times n\times\cdots\times n}$ and

$$\mathcal{C}_1 := \left\{ (\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \cdots, \boldsymbol{\mathcal{X}}_K) \in \mathcal{H} \;\middle|\; \sum_{i\in[K]} \left\|(\boldsymbol{\mathcal{X}}_i)_{(i)}\right\|_* \le \tau \right\}, \tag{2.8.2}$$

$$\mathcal{C}_2 := \{(\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \cdots, \boldsymbol{\mathcal{X}}_K) \in \mathcal{H} \mid \boldsymbol{\mathcal{X}}_1 = \boldsymbol{\mathcal{X}}_2 = \cdots = \boldsymbol{\mathcal{X}}_K\}. \tag{2.8.3}$$

As both $\mathcal{P}_{\mathcal{C}_1}[\cdot]$ and$\mathcal{P}_{\mathcal{C}_2}[\cdot]$ have closed form solutions that can be easily computed, we apply *Dykstra's algorithm* [see BC11, Chap. 29.1] to tackle problem (2.3.22).

Algorithm 1 Dykstra's algorithm for problem (2.3.22)

Initialization: $\boldsymbol{z}^{(0)} \leftarrow (\boldsymbol{\mathcal{Z}}_0, \boldsymbol{\mathcal{Z}}_0, \cdots, \boldsymbol{\mathcal{Z}}_0) \in \mathcal{H}$, $\boldsymbol{q}^{(-1)} \leftarrow \boldsymbol{0} \in \mathcal{H}$, $\boldsymbol{q}^{(0)} \leftarrow \boldsymbol{0} \in \mathcal{H}$;
for $n \leftarrow 1, 2, \ldots$ **do**
  if $2 \mid n$ **then**
    $\boldsymbol{z}^{(n)} \leftarrow \mathcal{P}_{\mathcal{C}_2}[\boldsymbol{z}^{(n-1)} + \boldsymbol{q}^{(n-2)}]$;
    $\boldsymbol{q}^{(n)} \leftarrow \boldsymbol{z}^{(n-1)} + \boldsymbol{q}^{(n-2)} - \boldsymbol{z}^{(n)}$;
  else
    $\boldsymbol{z}^{(n)} \leftarrow \mathcal{P}_{\mathcal{C}_1}[\boldsymbol{z}^{(n-1)} + \boldsymbol{q}^{(n-2)}]$;
    $\boldsymbol{q}^{(n)} \leftarrow \boldsymbol{z}^{(n-1)} + \boldsymbol{q}^{(n-2)} - \boldsymbol{z}^{(n)}$;
  end if
end for

For the sequence $\{\boldsymbol{z}^{(n)}\}$ generated by Algorithm 1, its convergence to the optimal solution of problem (2.3.22) follows directly from Theorem 29.2 of the book by Bauschke and Combettes [BC11].

2.8.2 Douglas-Rachford Algorithm for problem (2.4.5)

By splitting $\boldsymbol{\mathcal{X}}$ into $\{\boldsymbol{\mathcal{X}}_i\}_{i\in[K+1]}$, problem (2.4.5) can be reformulated as

$$\begin{aligned} \min_{\{\boldsymbol{\mathcal{X}}_i\}_{i\in[K+1]}} \quad & \sum_{i\in[K]} \left\|(\boldsymbol{\mathcal{X}}_i)_{(i)}\right\|_* \\ \text{s.t.} \quad & \mathcal{P}_\Omega[\boldsymbol{\mathcal{X}}_{K+1}] = \boldsymbol{\mathcal{M}} \\ & \boldsymbol{\mathcal{X}}_1 = \boldsymbol{\mathcal{X}}_2 = \cdots = \boldsymbol{\mathcal{X}}_{K+1} \in \mathbb{R}^{n\times n\times\cdots\times n}. \end{aligned} \tag{2.8.4}$$

If we denote $\boldsymbol{x} := (\boldsymbol{\mathcal{X}}_1, \boldsymbol{\mathcal{X}}_2, \cdots, \boldsymbol{\mathcal{X}}_{K+1}) \in \mathcal{H} := \bigtimes_{i\in[K+1]}\mathbb{R}^{n\times n\times\cdots\times n}$, then problem (2.8.4) can be com-

pactly expressed as

$$\min_{\boldsymbol{x}\in\mathcal{H}} \quad F(\boldsymbol{x}) + G(\boldsymbol{x}), \tag{2.8.5}$$

where

$$F(\boldsymbol{x}) := \sum_{i\in[K]} \left\|(\boldsymbol{\mathcal{X}}_i)_{(i)}\right\|_* + \mathbb{1}_{\{\mathcal{P}_\Omega[\boldsymbol{\mathcal{X}}_{K+1}]=\boldsymbol{\mathcal{M}}\}}, \tag{2.8.6}$$

$$G(\boldsymbol{x}) := \mathbb{1}_{\{\boldsymbol{\mathcal{X}}_1=\boldsymbol{\mathcal{X}}_2=\cdots=\boldsymbol{\mathcal{X}}_{K+1}\}}, \quad \text{and} \tag{2.8.7}$$

here the indicator function for a set C, $\mathbb{1}_C(\boldsymbol{x})$, equals 0 if $\boldsymbol{x}\in C$ and $+\infty$ otherwise. Note that the proximity operators of F and G, i.e.

$$\text{prox}_F(\boldsymbol{x}) \;:=\; \arg\min_{\boldsymbol{y}\in\mathcal{H}} \; F(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \quad \text{and} \tag{2.8.8}$$

$$\text{prox}_G(\boldsymbol{x}) \;:=\; \arg\min_{\boldsymbol{y}\in\mathcal{H}} \; G(\boldsymbol{y}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|^2 \tag{2.8.9}$$

can be both easily computed. Therefore, it is quite suitable to apply the *Douglas-Rachford algorithm* here [see CP11, for more details].

Algorithm 2 Douglas-Rachford algorithm for problem (2.4.5)

1: **Initialization:** $\boldsymbol{x}^{(0)} \leftarrow \boldsymbol{0} \in \mathcal{H}$;
2: **for** $n \leftarrow 0, 1, 2, \ldots$ **do**
3: $\quad \boldsymbol{y}^{(n)} \leftarrow \text{prox}_G(\boldsymbol{x}^{(n)})$;
4: $\quad \boldsymbol{x}^{(n+1)} \leftarrow \text{prox}_F(2\boldsymbol{y}^{(n)} - \boldsymbol{x}^{(n)}) + \boldsymbol{x}^{(n)} - \boldsymbol{y}^{(n)}$;
5: **end for**
6: **Output** $\text{prox}_G(\boldsymbol{x}^{(n+1)})$;

2.9 Conclusion

In this chapter, we establish several theoretical bounds for the problem of low-rank tensor recovery using random Gaussian measurements. For the nonconvex model (2.2.1), we show that $(2r)^K+2nrK+1$ measurements are sufficient to recover any $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$ almost surely. For the conventional convex surrogate sum-of-nuclear-norms (SNN) model (2.3.1), we prove a necessary condition that $\Omega(rn^{K-1})$ Gaussian measurements are required for reliable recovery. This lower bound is derived from our study of multi-structured object recovery in a very general setting, which can be applied to many other scenarios (e.g. signal processing, metric learning, computer vision). To narrow the apparent gap between the non-convex model and the SNN model,

we unfold the tensor into a more balanced matrix while preserving its low-rank property, leading to our square reshaping model (2.4.4). We then prove that $O(r^{\lfloor \frac{K}{2} \rfloor} n^{\lceil \frac{K}{2} \rceil})$ measurements are sufficient to recover a tensor $\boldsymbol{\mathcal{X}}_0 \in \mathfrak{T}_r$ with high probability. Though the theoretical results only pertain to Gaussian measurements, our numerical experiments still suggest the square reshaping model outperforms the SNN model in other settings.

Compared with $\Omega(rn^{K-1})$ measurements required by the SNN model, the sample complexity, $O(r^{\lfloor \frac{K}{2} \rfloor} n^{\lceil \frac{K}{2} \rceil})$, required by the square reshaping (2.4.4), is always within a constant of it, and is much better for small r and $K \geq 4$. Although this is a significant improvement, in contrast with the nonconvex model (2.2.1), the improved sample complexity achieved by the square model is still suboptimal. It remains an open and intriguing problem to obtain near-optimal tractable convex relaxations for all $K > 2$.

Since the release of our work [MHWG13] online, our proposed square model for low-rank tensor recovery has been successfully applied under different contexts, including seismic reconstruction [GCS15], traffic data recovery [TWW$^+$] and video recovery [JMZ15, BPTD17], to name a few. Moreover, we have also noted that several interesting models and algorithms have been proposed and analyzed, focusing on the low-rank tensor completion (LRTC) problem. Yuan and Zhang [OJF$^+$15] extended the negative result for the SNN model to more general sampling schemes. Yuan and Zhang [YZ15] analyzed the tensor nuclear norm model (though not computationally tractable) and established better sampling complexity result. Several other works – e.g. [JO14, Asw14], achieved better sample complexity using tractable methods by considering special subclasses of low-rank tensors. In addition, many works in the field of numerical optimization have designed efficient methods to solve LRTC related non-convex models, e.g. alternating minimization [RPABBP13, XHYS13], Riemannian optimization [KSV14], where empirical successes have been greatly witnessed. Further analyzing these methods is an interesting problem for future research.

Putting our work in a broader setting, to recover objects with multiple structures, regularizing with a combination of individual structure-inducing norms is proven to be substantially suboptimal (Theorem 2.12 and also [OJF$^+$12]). The resulting sample requirements tend to be much larger than the intrinsic degrees of freedom of the low-dimensional manifold in which the structured signal lies. Our square model for low-rank tensor recovery demonstrates the possibility that a better exploitation of those structures can significantly reduce this sample complexity (see also [RBV13, ROV14] for ideas in this direction). However, there are still no clear clues on how to intelligently utilize several simultaneous structures generally, and moreover how to design tractable methods to recover multi-structured objects with near minimal numbers of measurements. These problems are definitely worth future study.

Chapter 3

Robust Low-Rank Tensor Recovery

The robust low-rank tensor recovery problem aims to recover the underlying low-rank tensor from both sparse corruptions and dense small ones. More formally, we are trying to (robustly) recover the low-rank tensor $\boldsymbol{\mathcal{L}}_0$ from the corrupted observations $\boldsymbol{\mathcal{T}}$:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{L}}_0 + \boldsymbol{\mathcal{S}}_0 + \boldsymbol{\mathcal{N}}_0, \tag{3.0.1}$$

where $\boldsymbol{\mathcal{S}}_0$ is a sparse tensor and $\boldsymbol{\mathcal{N}}_0$ is a dense tensor with small magnitudes.

In the previous chapter, we have proposed and proved the convex function $\left\|(\cdot)_{(\mathcal{R}\times\mathcal{C})}\right\|_*$ as an appropriate convex surrogate to encourage the low-rankness of tensors, with a proper choice of $\mathcal{R}, \mathcal{C}$ partitioning the index set $[K]$. Moreover, the ℓ_1 norm, $\|\cdot\|_1$, has been well known as a convex replacement for the sparsity. Therefore, it is natural to solve the following convex program to tackle the robust low-rank tensor recovery task:

$$\min_{\boldsymbol{\mathcal{L}},\, \boldsymbol{\mathcal{S}}} \quad \frac{1}{2}\left\|\boldsymbol{\mathcal{L}}_{(\mathcal{R}\times\mathcal{C})} + \boldsymbol{\mathcal{S}}_{(\mathcal{R}\times\mathcal{C})} - \boldsymbol{\mathcal{T}}_{(\mathcal{R}\times\mathcal{C})}\right\|_F^2 + \lambda_L \left\|\boldsymbol{\mathcal{L}}_{(\mathcal{R}\times\mathcal{C})}\right\|_* + \lambda_S \left\|\boldsymbol{\mathcal{S}}_{(\mathcal{R}\times\mathcal{C})}\right\|_1, \tag{3.0.2}$$

where $\lambda_L, \lambda_S \geq 0$ are regularization parameters.

Problem (3.0.2) is essentially equivalent to the *stable principal component pursuit (SPCP)* problem [CLMW11, ZLW+10], which is originally proposed for the low-rank matrix recovery. In the rest of this chapter, we will focus on developing scalable optimization methods to solve a convex model (more general than stable principal component pursuit) called *compressive principal component pursuit (CPCP)* .

3.1 Robust Low-Rank Matrix Recovery

Suppose that a matrix $\boldsymbol{M}_0 \in \mathbb{R}^{m\times n}$ is of the form $\boldsymbol{M}_0 = \boldsymbol{L}_0 + \boldsymbol{S}_0 + \boldsymbol{N}_0$, where $\boldsymbol{L}_0$ is a low-rank matrix, $\boldsymbol{S}_0$ is a sparse error matrix, and $\boldsymbol{N}_0$ is a dense noise matrix. Linear measurements

$$\boldsymbol{b} \;=\; \mathcal{A}[\boldsymbol{M}_0] \;=\; \big(\, \langle \boldsymbol{A}_1, \boldsymbol{M}_0\rangle, \langle \boldsymbol{A}_2, \boldsymbol{M}_0\rangle, \ldots, \langle \boldsymbol{A}_p, \boldsymbol{M}_0\rangle \,\big)^\top \in \mathbb{R}^p \tag{3.1.1}$$

are collected, where $\mathcal{A}:\ \mathbb{R}^{m\times n} \to \mathbb{R}^p$ is the sensing operator, $\boldsymbol{A}_k$ is the sensing matrix for the k-th measurement and $\langle \boldsymbol{A}_k, \boldsymbol{M}_0\rangle \doteq \mathrm{Tr}(\boldsymbol{M}_0^\top \boldsymbol{A}_k)$. *Can we, in a tractable way, recover $\boldsymbol{L}_0$ and $\boldsymbol{S}_0$ from $\boldsymbol{b}$, given $\mathcal{A}$?*

One natural approach is to solve the optimization combining the fidelity term and the structural terms:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \;\; \frac{1}{2}\left\|\boldsymbol{b} - \mathcal{A}[\boldsymbol{L}+\boldsymbol{S}]\right\|_2^2 + \lambda_L \mathrm{rank}(\boldsymbol{L}) + \lambda_S \left\|\boldsymbol{S}\right\|_0 . \tag{3.1.2}$$

Here, λ_L and λ_S are regularization parameters, and $\|\boldsymbol{S}\|_0$ denotes the number of nonzero entries in $\boldsymbol{S}$.

Unfortunately, problem (3.1.2) is nonconvex, and hence is not directly tractable. However, by replacing the ℓ_0 norm $\|\boldsymbol{S}\|_0$ with the ℓ_1 norm $\|\boldsymbol{S}\|_1 \doteq \sum_{i=1}^m \sum_{j=1}^n |S_{ij}|$, and replacing the rank $\mathrm{rank}(\boldsymbol{L})$ with the nuclear norm $\|\boldsymbol{L}\|_*$ (defined as the sum of the singular values of $\boldsymbol{L}$), we obtain a natural, tractable, convex relaxation of (3.1.2),

$$\min_{\boldsymbol{L},\boldsymbol{S}} \;\; \frac{1}{2}\left\|\boldsymbol{b} - \mathcal{A}[\boldsymbol{L}+\boldsymbol{S}]\right\|_2^2 + \lambda_L \left\|\boldsymbol{L}\right\|_* + \lambda_S \left\|\boldsymbol{S}\right\|_1 . \tag{3.1.3}$$

This convex surrogate is sometimes referred to as *compressive principal component pursuit (CPCP)* [WGMM13]. Equivalently, since

$$\left\{\, \boldsymbol{M} \in \mathbb{R}^{m\times n} \;\mid\; \boldsymbol{b} = \mathcal{A}[\boldsymbol{M}] \,\right\} \;=\; \left\{\, \boldsymbol{M} \in \mathbb{R}^{m\times n} \;\mid\; \mathcal{P}_{\mathcal{Q}}[\boldsymbol{M}] = \mathcal{P}_{\mathcal{Q}}[\boldsymbol{M}_0] \,\right\},$$

where $\mathcal{Q} \subseteq \mathbb{R}^{m\times n}$ is a linear subspace spanned by the set of sensing matrices $\{\boldsymbol{A}_i\}_{i=1}^p$, and $\mathcal{P}_{\mathcal{Q}}$ denotes the projection operator onto that subspace, we can rewrite problem (3.1.3) in the (possibly) more compact form, [1]

$$\min_{\boldsymbol{L},\boldsymbol{S}} \;\; f(\boldsymbol{L},\boldsymbol{S}) \doteq \frac{1}{2}\left\|\mathcal{P}_{\mathcal{Q}}[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}_0]\right\|_F^2 + \lambda_L \left\|\boldsymbol{L}\right\|_* + \lambda_S \left\|\boldsymbol{S}\right\|_1 . \tag{3.1.4}$$

Recently, CPCP and its close variants have been studied for different sensing operators $\mathcal{A}[\cdot]$ (or equivalently different subspaces $\mathcal{Q}$). In specific, [CSPW11, CLMW11, ZLW+10, HKZ11, ANW12] consider the case where

[1] Despite being equivalent, one formulation might be preferred over the other in practice, depending on the specifications of the sensing operator $\mathcal{A}[\cdot]$. In this chapter, we will mainly focus on solving problem (3.1.4) and its variants. Our methods, however, are not restrictive to (3.1.4) and can be easily extended to problem (3.1.3).

a subset $\Omega \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ of the entries of $\boldsymbol{M}_0$ is observed. Then CPCP can be reduced to

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad \frac{1}{2} \|\mathcal{P}_\Omega[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}_0]\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* + \lambda_S \|\boldsymbol{S}\|_1 , \tag{3.1.5}$$

where $\mathcal{P}_\Omega[\cdot]$ denotes the orthogonal projection onto the linear space of matrices supported on Ω, i.e., $\mathcal{P}_\Omega[\boldsymbol{M}_0](i,j) = (\boldsymbol{M}_0)_{ij}$ if $(i,j) \in \Omega$ and $\mathcal{P}_\Omega[\boldsymbol{M}_0](i,j) = 0$ otherwise. [WGMM13] studies the case where each $\mathcal{A}_k$ is an i.i.d. $\mathcal{N}(0,1)$ matrix, which is equivalent (in distribution) to saying that we choose a linear subspace $\mathcal{Q}$ uniformly at random from the set of all p-dimensional subspaces of $\mathbb{R}^{m\times n}$ and observe $\mathcal{P}_Q[\boldsymbol{M}_0]$. Accordingly, all the above provide theoretical guarantees for CPCP, under fairly mild conditions, to produce accurate estimates of $\boldsymbol{L}_0$ and $\mathcal{P}_\Omega[\boldsymbol{S}_0]$ (or $\boldsymbol{S}_0$), even when the number of measurements p is substantially less than mn.

Inspired by these theoretical results, researchers from different fields have leveraged CPCP to solve many practical problems, including video background modeling [CLMW11], batch image alignment [PGW+12], face verification [ZMKW13], photometric stereo [WGS+11], dynamic MRI [OCS14], topic modeling [MZWM10], latent variable graphical model learning [CPW12] and outlier detection and robust Principal Component Analysis [CLMW11], to name just a few.

Living in the era of *big data*, most of these applications involve large datasets and high dimensional data spaces. Therefore, to fully realize the benefit of the theory, we need *provably convergent* and *scalable* algorithms for CPCP. This has motivated much research into the development of first-order methods for problem (3.1.4) and its variants; e.g., see [LGW+09, LCM10, YY13b, TY11, AGM12, AI15]. These methods, in essence, all exploit a closed-form expression for the proximal operator of the nuclear norm, which involves the singular value decomposition (SVD). Hence, the dominant cost in each iteration is computing an SVD of the same size as the input data. This is substantially more scalable than off-the-shelf interior point solvers such as SDPT3 [TTT03]. Nevertheless, the superlinear cost of each iteration has limited the practical applicability of these first-order methods to problems involving several thousands of data points and several thousands of dimensions. The need to compute a sequence of full or partial SVDs is a serious bottleneck for truly large-scale applications.

As a remedy, in this chapter, we design more scalable algorithms to solve CPCP that compute only a rank-one SVD in each iteration. Our approach leverages two classical and widely studied ideas – Frank-Wolfe iterations to handle the nuclear norm, and proximal steps to handle the ℓ_1 norm. This turns out to be an ideal combination of techniques to solve large-scale CPCP problems. In particular, it yields algorithms that are substantially *more scalable* than prox-based first-order methods such as ISTA and FISTA [BT09], and converge

much faster in practice than a straightforward application of Frank-Wolfe.

The remainder of this chapter is organized as follows. Section 3.2 reviews the general properties of the Frank-Wolfe algorithm, and describes several basic building blocks that we will use in our algorithms. Section 3.3 and Section 3.4 respectively describe how to modify the Frank-Wolfe algorithm to solve CPCP's *norm constrained* version

$$\min_{\boldsymbol{L},\boldsymbol{S}} \; l(\boldsymbol{L},\boldsymbol{S}) \doteq \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}_0]\right\|_F^2 \quad \text{s.t. } \|\boldsymbol{L}\|_* \le \tau_L,\ \|\boldsymbol{S}\|_1 \le \tau_S, \tag{3.1.6}$$

and the penalized version, i.e. problem (3.1.4), by incorporating proximal regularization to more effectively handle the ℓ_1 norm. Convergence results and our implementation details are also discussed. Section 3.5 presents numerical experiments on large datasets that demonstrate the scalability of our proposed algorithms. In Section 3.6, we summarize our contributions and discuss potential future works.

3.2 Preliminaries on Frank-Wolfe method

3.2.1 Frank-Wolfe method

The Frank-Wolfe (FW) method [FW56], also known as the conditional gradient method [LP66], applies to the general problem of minimizing a differentiable convex function h over a compact, convex domain $\mathcal{D} \subseteq \mathbb{R}^n$:

$$\text{minimize} \quad h(\boldsymbol{x}) \qquad \text{subject to} \quad \boldsymbol{x} \in \mathcal{D} \subseteq \mathbb{R}^n. \tag{3.2.1}$$

Here, ∇h is assumed to be L-Lipschitz:

$$\forall\, \boldsymbol{x},\, \boldsymbol{y} \in \mathcal{D}, \qquad \|\nabla h(\boldsymbol{x}) - \nabla h(\boldsymbol{y})\| \le L\,\|\boldsymbol{x}-\boldsymbol{y}\|\,. \tag{3.2.2}$$

Throughout, we let $D = \max_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}} \|\boldsymbol{x}-\boldsymbol{y}\|$ denote the diameter of the feasible set $\mathcal{D}$.

In its simplest form, the Frank-Wolfe algorithm proceeds as follows. At each iteration k, we linearize the objective function h about the current point $\boldsymbol{x}^k$:

$$h(\boldsymbol{v}) \approx h(\boldsymbol{x}^k) + \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}-\boldsymbol{x}^k \right\rangle. \tag{3.2.3}$$

We minimize the linearization over the feasible set $\mathcal{D}$ to obtain

$$\boldsymbol{v}^k \in \arg\min_{\boldsymbol{v}\in\mathcal{D}} \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v} \right\rangle, \tag{3.2.4}$$

Algorithm 3 Frank-Wolfe method for problem (3.2.1)

1: **Initialization:** $\boldsymbol{x}^0 \in \mathcal{D}$;
2: **for** $k = 0,\ 1,\ 2,\ \ldots$ **do**
3: $\quad \boldsymbol{v}^k \in \operatorname{argmin}_{\boldsymbol{v}\in\mathcal{D}} \langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k)\rangle$;
4: $\quad \gamma = \frac{2}{k+2}$;
5: $\quad \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)$;
6: **end for**

and then take a step in the feasible descent direction $\boldsymbol{v}^k - \boldsymbol{x}^k$:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \frac{2}{k+2}(\boldsymbol{v}^k - \boldsymbol{x}^k). \tag{3.2.5}$$

This yields a very simple procedure, which we summarize as Algorithm 3. The particular step size, $\frac{2}{k+2}$, comes from the convergence analysis of the algorithm, which we discuss in more details below.

First proposed in [FW56], FW-type methods have been frequently revisited in different fields. Recently, they have experienced a resurgence in statistics, machine learning and signal processing, due to their ability to yield highly scalable algorithms for optimization with structure-encouraging norms such as the ℓ_1 norm and nuclear norm. In particular, if $\boldsymbol{x}$ is a matrix and $\mathcal{D} = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_* \leq \beta\}$ is a nuclear norm ball, the subproblem

$$\min_{\boldsymbol{v}\in\mathcal{D}} \ \langle \boldsymbol{v}, \nabla h(\boldsymbol{x})\rangle \tag{3.2.6}$$

can be solved using only the singular vector pair corresponding to the single leading singular value of the matrix $\nabla h(\boldsymbol{x})$. Thus, at each iteration, we only have to compute a rank-one partial SVD. This is substantially cheaper than the full/partial SVD exploited in proximal methods [JS10, HJN14]. We recommend [Jag13] as a comprehensive survey of the latest developments in FW-type methods.

Algorithm 4 Frank-Wolfe method for problem (3.2.1) with general updating scheme

1: **Initialization:** $\boldsymbol{x}^0 \in \mathcal{D}$;
2: **for** $k = 0,\ 1,\ 2,\ \ldots$ **do**
3: $\quad \boldsymbol{v}^k \in \operatorname{argmin}_{\boldsymbol{v}\in\mathcal{D}} \langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k)\rangle$;
4: $\quad \gamma = \frac{2}{k+2}$;
5: $\quad$ Update $\boldsymbol{x}^{k+1}$ to some point in $\mathcal{D}$ such that $h(\boldsymbol{x}^{k+1}) \leq h(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k))$;
6: **end for**

In the past five decades, numerous variants of Algorithm 3 have been proposed and implemented. Many modify Algorithm 3 by replacing the simple updating rule (3.2.5) with more sophisticated schemes, e.g.,

$$\boldsymbol{x}^{k+1} \ \in \ \arg\min_{\boldsymbol{x}} \ h(\boldsymbol{x}) \quad \text{s.t.} \ \ \boldsymbol{x} \in \operatorname{conv}\{\boldsymbol{x}^k,\ \boldsymbol{v}^k\} \tag{3.2.7}$$

or

$$\boldsymbol{x}^{k+1} \in \arg\min_{\boldsymbol{x}} h(\boldsymbol{x}) \quad \text{s.t. } \boldsymbol{x} \in \text{conv}\{\boldsymbol{x}^k, \boldsymbol{v}^k, \boldsymbol{v}^{k-1}, \ldots, \boldsymbol{v}^{k-j}\}. \tag{3.2.8}$$

The convergence of these schemes can be analyzed simultaneously, using the fact that they produce iterates $\boldsymbol{x}^{k+1}$ whose objective is no greater than that produced by the original Frank-Wolfe update scheme:

$$h(\boldsymbol{x}^{k+1}) \leq h(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)).$$

Algorithm 4 states a general version of Frank-Wolfe, whose update is only required to satisfy this relationship. It includes as special cases the updating rules (3.2.5), (3.2.7) and (3.2.8). This flexibility will be crucial for effectively handling the sparse structure in the CPCP problems (3.1.4) and (3.1.6).

The convergence of Algorithm 4 can be proved using well-established techniques [HJN14, Jag13, DR70, DH78, Pat93, Zha03, Cla10, FG16]. Using these ideas, one can show that it converges at a rate of $O(1/k)$ in function value:

Theorem 3.1 *Let $\boldsymbol{x}^\star$ be an optimal solution to (3.2.1). For $\{\boldsymbol{x}^k\}$ generated by Algorithm 4, we have for $k = 0, 1, 2, \ldots,$*

$$h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) \leq \frac{2LD^2}{k+2}. \tag{3.2.9}$$

Proof For $k = 0, 1, 2, \ldots,$ we have

$$\begin{aligned}
h(\boldsymbol{x}^{k+1}) &\leq h(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)) \\
&\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \frac{L\gamma^2}{2} \left\| \boldsymbol{v}^k - \boldsymbol{x}^k \right\|^2 \\
&\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \frac{\gamma^2 LD^2}{2} \qquad (3.2.10) \\
&\leq h(\boldsymbol{x}^k) + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{x}^\star - \boldsymbol{x}^k \right\rangle + \frac{\gamma^2 LD^2}{2} \\
&\leq h(\boldsymbol{x}^k) + \gamma(h(\boldsymbol{x}^\star) - h(\boldsymbol{x}^k)) + \frac{\gamma^2 LD^2}{2}, \qquad (3.2.11)
\end{aligned}$$

where the second inequality holds since $\nabla h(\cdot)$ is L-Lipschitz continuous; the third line follows because D is the diameter for the feasible set $\mathcal{D}$; the fourth inequality follows from $\boldsymbol{v}^k \in \text{argmin}_{\boldsymbol{v}\in\mathcal{D}} \left\langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k) \right\rangle$ and $\boldsymbol{x}^\star \in \mathcal{D}$; the last one holds since $h(\cdot)$ is convex.

Rearranging terms in (3.2.11), one obtains that for $k = 0, 1, 2, \ldots,$

$$h(\boldsymbol{x}^{k+1}) - h(\boldsymbol{x}^\star) \leq (1-\gamma)\left(h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)\right) + \frac{\gamma^2 LD^2}{2}. \tag{3.2.12}$$

Therefore, by mathematical induction, it can be verified that

$$h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) \leq \frac{2LD^2}{k+2}, \quad \text{for } k = 1,\ 2,\ 3,\ \ldots.$$

■

Remark 3.2 *Note that the constant in the rate of convergence depends on the Lipschitz constant L of h and the diameter $\mathcal{D}$.*

While Theorem 3.1 guarantees that Algorithm 4 converges at a rate of $O(1/k)$, in practice it is useful to have a more precise bound on the suboptimality at iterate k. The surrogate duality gap

$$d(\boldsymbol{x}^k) = \left\langle \boldsymbol{x}^k - \boldsymbol{v}^k, \nabla h(\boldsymbol{x}^k) \right\rangle, \tag{3.2.13}$$

provides a useful upper bound on the suboptimality $h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)$:

$$\begin{aligned} h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star) &\leq -\left\langle \boldsymbol{x}^\star - \boldsymbol{x}^k, \nabla h(\boldsymbol{x}^k) \right\rangle \\ &\leq -\min_{\boldsymbol{v}} \left\langle \boldsymbol{v} - \boldsymbol{x}^k, \nabla h(\boldsymbol{x}^k) \right\rangle = \left\langle \boldsymbol{x}^k - \boldsymbol{v}^k, \nabla h(\boldsymbol{x}^k) \right\rangle = d(\boldsymbol{x}^k). \end{aligned} \tag{3.2.14}$$

This was first proposed in [FW56] and later [Jag13] showed that $d(\boldsymbol{x}^k) = O(1/k)$. Next, we provide a refinement of this result, using ideas from [Jag13, Cla10]:

Theorem 3.3 *Let $\{\boldsymbol{x}^k\}$ be the sequence generated by Algorithm 4. Then for any $K \geq 1$, there exists $1 \leq \tilde{k} \leq K$ such that*

$$d(\boldsymbol{x}^{\tilde{k}}) \leq \frac{6LD^2}{K+2}. \tag{3.2.15}$$

Proof For notational convenience, we denote $h^k \doteq h(\boldsymbol{x}^k)$, $\Delta^k \doteq h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)$, $d^k \doteq d(\boldsymbol{x}^k)$, $C \doteq 2LD^2$, $B \doteq K+2$, $\hat{k} \doteq \lceil \frac{1}{2}B \rceil - 1$, $\mu \doteq \lceil \frac{1}{2}B \rceil / B$.

Suppose on the contrary that

$$d^k > \frac{3C}{B}, \quad \text{for all } k \in \left\{ \lceil \frac{1}{2}B \rceil - 1,\ \lceil \frac{1}{2}B \rceil,\ \ldots,\ K \right\}. \tag{3.2.16}$$

From (3.2.10), we know that for any $k \geq 1$

$$\Delta^{k+1} \leq \Delta^k + \gamma \left\langle \nabla h(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \frac{\gamma^2 LD^2}{2} = \Delta^k - \frac{2d^k}{k+2} + \frac{C}{(k+2)^2}. \tag{3.2.17}$$

Therefore, by using (3.2.17) repeatedly, one has

$$\begin{aligned}
\Delta^{K+1} &\leq \Delta^{\hat{k}} - \sum_{k=\hat{k}}^{K} \frac{2d^k}{k+2} + \sum_{k=\hat{k}}^{K} \frac{C}{(k+2)^2} \\
&< \Delta^{\hat{k}} - \frac{6C}{B} \sum_{k=\hat{k}}^{K} \frac{1}{k+2} + C \sum_{k=\hat{k}}^{K} \frac{1}{(k+2)^2} \\
&= \Delta^{\hat{k}} - \frac{6C}{B} \sum_{k=\hat{k}+2}^{B} \frac{1}{k} + C \sum_{k=\hat{k}+2}^{B} \frac{1}{k^2} \\
&\leq \frac{C}{\mu B} - \frac{6C}{B} \cdot \frac{B-\hat{k}-1}{B} + C \cdot \frac{B-\hat{k}-1}{B(\hat{k}+1)} \\
&= \frac{C}{\mu B} - \frac{6C}{B}(1-\mu) + \frac{C}{B}\frac{1-\mu}{\mu} \\
&= \frac{C}{\mu B}\left(2 - 6\mu(1-\mu) - \mu\right) \qquad (3.2.18)
\end{aligned}$$

where the second line is due to our assumption (3.2.16); the fourth line holds since $\Delta^{\hat{k}} \leq \frac{C}{\hat{k}+2}$ by Theorem 1, and $\sum_{k=a}^{b} \frac{1}{k^2} \leq \frac{b-a+1}{b(a-1)}$ for any $b \geq a > 1$.

Now define $\phi(x) = 2 - 6x(1-x) - x$. Clearly $\phi(\cdot)$ is convex. Since $\phi(\frac{1}{2}) = \phi(\frac{2}{3}) = 0$, we have $\phi(x) \leq 0$ for any $x \in [\frac{1}{2}, \frac{2}{3}]$. As $\mu = \lceil \frac{1}{2}B \rceil / B \in [\frac{1}{2}, \frac{2}{3}]$, from (3.2.18), we have

$$\Delta^{K+1} = h(\boldsymbol{x}^{K+1}) - h(\boldsymbol{x}^\star) < \frac{C}{\mu B}\phi(\mu) \leq 0,$$

which is a contradiction. ■

> **Remark 3.4** *The convergence rate for the duality gap matches the one for $h(\boldsymbol{x}^k) - h(\boldsymbol{x}^\star)$ (see (3.2.9)), which suggests that the upper bound $d(\boldsymbol{x}^k)$ can serve as a practical stopping criterion.*

For our problem, the main computational burden in Algorithms 3 and 4 will be solving the linear subproblem $\min_{\boldsymbol{v}\in\mathcal{D}} \langle \boldsymbol{v}, \nabla h(\boldsymbol{x}^k) \rangle$,[2] i.e. minimizing linear functions over the unit balls for $\|\cdot\|_*$ and $\|\cdot\|_1$. Fortunately, both of these operations have simple closed-form solutions, which we will describe in the next section.

3.2.2 Optimization oracles

We now describe several optimization oracles involving the ℓ_1 norm and the nuclear norm, which serve as the main building blocks for our methods. These oracles have computational costs that are (essentially) linear

[2] In some situations, we can significantly reduce this cost by solving this problem inexactly [DH78, Jag13]. Our algorithms and results can also tolerate inexact step calculations; we omit the discussion here for simplicity.

in the size of the input.

Minimizing a linear function over the nuclear norm ball Since the dual norm of the nuclear norm is the operator norm, i.e., $\|\boldsymbol{Y}\| = \max_{\|\boldsymbol{X}\|_* \leq 1} \langle \boldsymbol{Y}, \boldsymbol{X} \rangle$, the optimization problem

$$\text{minimize}_{\boldsymbol{X}} \ \langle \boldsymbol{Y}, \boldsymbol{X} \rangle \qquad \text{subject to } \|\boldsymbol{X}\|_* \leq 1 \tag{3.2.19}$$

has optimal value $-\|\boldsymbol{Y}\|$. One minimizer is the rank-one matrix $\boldsymbol{X}^\star = -\boldsymbol{u}\boldsymbol{v}^\top$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the left- and right- singular vectors corresponding to the top singular value of $\boldsymbol{Y}$, and can be efficiently computed (e.g. using power method).

Minimizing a linear function over the ℓ_1 ball Since the dual norm of the ℓ_1 norm is the ℓ_∞ norm, i.e., $\|\boldsymbol{Y}\|_\infty := \max_{(i,j)} |Y_{ij}| = \max_{\|\boldsymbol{X}\|_1 \leq 1} \langle \boldsymbol{Y}, \boldsymbol{X} \rangle$, the optimization problem

$$\text{minimize}_{\boldsymbol{X}} \ \langle \boldsymbol{Y}, \boldsymbol{X} \rangle \qquad \text{subject to } \|\boldsymbol{X}\|_1 \leq 1 \tag{3.2.20}$$

has optimal value $-\|\boldsymbol{Y}\|_\infty$. One minimizer is the one-sparse matrix $\boldsymbol{X}^\star = -\text{sgn}(Y_{i^\star j^\star})\boldsymbol{e}_{i^\star}\boldsymbol{e}_{j^\star}^\top$, where $(i^\star, j^\star) \in \arg\max_{(i,j)} |Y_{ij}|$; i.e. $\boldsymbol{X}^\star$ has exactly one nonzero element.

Projection onto the ℓ_1 ball To effectively handle the sparse term in the norm constrained problem (3.1.6), we will need to modify the Frank-Wolfe algorithm by incorporating additional projection steps. For any $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$ and $\beta > 0$, the projection onto the ℓ_1-ball:

$$\mathcal{P}_{\|\cdot\|_1 \leq \beta}[\boldsymbol{Y}] = \arg \min_{\|\boldsymbol{X}\|_1 \leq \beta} \ \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 , \tag{3.2.21}$$

can be easily solved with $O\left(mn(\log m + \log n)\right)$ cost [DSSSC08]. Moreover, a divide and conquer algorithm, achieving linear cost in expectation to solve (3.2.21), has also been proposed in [DSSSC08].

Proximal mapping of ℓ_1 norm To effectively handle the sparse term arising in problem (3.1.4), we will need to modify the Frank-Wolfe algorithm by incorporating additional proximal steps. For any $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$ and $\lambda > 0$, the proximal mapping of ℓ_1 norm has the following closed-form expression

$$\mathcal{T}_\lambda[\boldsymbol{Y}] \ = \ \arg \min_{\boldsymbol{X} \in \mathbb{R}^{m \times n}} \ \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{X}\|_1 , \tag{3.2.22}$$

where $\mathcal{T}_\lambda : \mathbb{R} \to \mathbb{R}$ denotes the soft-thresholding operator $\mathcal{T}_\lambda(x) = \text{sgn}(x) \max\{|x| - \lambda, 0\}$, and extension to matrices is obtained by applying the scalar operator $\mathcal{T}_\lambda(\cdot)$ to each element.

3.3 Frank-Wolfe-Projection Method for Norm Constrained Problem

In this section, we develop scalable algorithms for the norm-constrained compressive principal component pursuit problem,

$$\min_{\boldsymbol{L},\boldsymbol{S}} \; l(\boldsymbol{L},\boldsymbol{S}) = \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 \quad \text{s.t.} \quad \|\boldsymbol{L}\|_* \le \tau_L,\ \|\boldsymbol{S}\|_1 \le \tau_S. \tag{3.3.1}$$

We first describe a straightforward application of the Frank-Wolfe method to this problem. We will see that although it has relatively cheap iterations, it converges very slowly on typical numerical examples, because it only makes a one-sparse update to the sparse term $\boldsymbol{S}$ at a time. We will show how to remedy this problem by augmenting the FW iteration with an additional proximal step (essentially a projected gradient step) in each iteration, yielding a new algorithm which updates $\boldsymbol{S}$ much more efficiently. Because it combines Frank-Wolfe and projection steps, we will call this new algorithm Frank-Wolfe-Projection (FW-P).

Properties of the objective and constraints. To apply Frank-Wolfe to (3.3.1), we first note that the objective $l(\boldsymbol{L},\boldsymbol{S})$ in (3.3.1) is differentiable, with

$$\nabla_{\boldsymbol{L}} l(\boldsymbol{L},\boldsymbol{S}) = \mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}] \tag{3.3.2}$$

$$\nabla_{\boldsymbol{S}} l(\boldsymbol{L},\boldsymbol{S}) = \mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]. \tag{3.3.3}$$

Moreover, the following lemma shows that the gradient map $\nabla l(\boldsymbol{L},\boldsymbol{S}) = (\nabla_{\boldsymbol{L}} l, \nabla_{\boldsymbol{S}} l)$ is 2-Lipschitz:

Lemma 3.5 *For all $(\boldsymbol{L},\boldsymbol{S})$ and $(\boldsymbol{L}',\boldsymbol{S}')$, we have $\|\nabla l(\boldsymbol{L},\boldsymbol{S}) - \nabla l(\boldsymbol{L}',\boldsymbol{S}')\|_F \le 2\,\|(\boldsymbol{L},\boldsymbol{S}) - (\boldsymbol{L}',\boldsymbol{S}')\|_F$.*

Proof From (3.3.2) and (3.3.3), we have

$$\begin{aligned}
\|\nabla l(\boldsymbol{L},\boldsymbol{S}) - \nabla l(\boldsymbol{L}',\boldsymbol{S}')\|_F^2 &= 2\,\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}] - \mathcal{P}_Q[\boldsymbol{L}'+\boldsymbol{S}'-\boldsymbol{M}]\|_F^2 \\
&= 2\,\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}] - \mathcal{P}_Q[\boldsymbol{L}'+\boldsymbol{S}']\|_F^2 \\
&\le 2\,\|\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{L}'-\boldsymbol{S}'\|_F^2 \\
&\le 4\,\|\boldsymbol{L}-\boldsymbol{L}'\|_F^2 + 4\,\|\boldsymbol{S}-\boldsymbol{S}'\|_F^2 \\
&= 4\,\|(\boldsymbol{L},\boldsymbol{S}) - (\boldsymbol{L}',\boldsymbol{S}')\|_F^2,
\end{aligned}$$

which implies the result. ∎

The feasible set in (3.3.1) is compact. The following lemma bounds its diameter D:

Lemma 3.6 *The feasible set* $\mathcal{D} = \{(\boldsymbol{L}, \boldsymbol{S}) \mid \|\boldsymbol{L}\|_* \leq \tau_L,\ \|\boldsymbol{S}\|_1 \leq \tau_S\}$ *has diameter* $D \leq 2\sqrt{\tau_L^2 + \tau_S^2}$.

Proof For any $\boldsymbol{Z} = (\boldsymbol{L}, \boldsymbol{S})$ and $\boldsymbol{Z}' = (\boldsymbol{L}', \boldsymbol{S}') \in \mathcal{D}$,

$$\begin{aligned}\|\boldsymbol{Z} - \boldsymbol{Z}'\|_F^2 = \|\boldsymbol{L} - \boldsymbol{L}'\|_F^2 + \|\boldsymbol{S} - \boldsymbol{S}'\|_F^2 &\leq (\|\boldsymbol{L}\|_F + \|\boldsymbol{L}'\|_F)^2 + (\|\boldsymbol{S}\|_F + \|\boldsymbol{S}'\|_F)^2 \\ &\leq (\|\boldsymbol{L}\|_* + \|\boldsymbol{L}'\|_*)^2 + (\|\boldsymbol{S}\|_1 + \|\boldsymbol{S}'\|_1)^2 \leq 4\tau_L^2 + 4\tau_S^2. \end{aligned} \tag{3.3.4}$$

■

3.3.1 Frank-Wolfe for problem (3.3.1)

Since (3.3.1) asks us to minimize a convex, differentiable function with Lipschitz gradient over a compact convex domain, the Frank-Wolfe method in Algorithm 3 applies. It generates a sequence of iterates $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k)$. Using the expression for the gradient in (3.3.2)-(3.3.3), at each iteration, the step direction $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k)$ is generated by solving the linearized subproblem

$$\begin{aligned}\begin{pmatrix} \boldsymbol{V}_L^k \\ \boldsymbol{V}_S^k \end{pmatrix} \in \arg\min \quad & \left\langle \begin{pmatrix} \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \\ \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_L \\ \boldsymbol{V}_S \end{pmatrix} \right\rangle \\ \text{s.t.} \quad & \|\boldsymbol{V}_L\|_* \leq \tau_L,\ \ \|\boldsymbol{V}_S\|_1 \leq \tau_S, \end{aligned} \tag{3.3.5}$$

which decouples into two independent subproblems:

$$\begin{aligned}\boldsymbol{V}_L^k &\in \arg\min_{\|\boldsymbol{V}_L\|_* \leq \tau_L} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{V}_L \rangle, \\ \boldsymbol{V}_S^k &\in \arg\min_{\|\boldsymbol{V}_S\|_1 \leq \tau_S} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{V}_S \rangle. \end{aligned}$$

These subproblems can be easily solved by exploiting the linear optimization oracles introduced in Section 3.2.2. In particular,

$$\boldsymbol{V}_L^k = -\tau_L \boldsymbol{u}^k (\boldsymbol{v}^k)^\top, \tag{3.3.6}$$

$$\boldsymbol{V}_S^k = -\tau_S \cdot \delta_{i^\star j^\star}^k \cdot \boldsymbol{e}_{i^\star}^k (\boldsymbol{e}_{j^\star}^k)^\top, \tag{3.3.7}$$

where $\boldsymbol{u}^k$ and $\boldsymbol{v}^k$ are leading left- and right- singular vectors of $\mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$, $(i^\star, j^\star)$ is the index of the largest element of $\mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$ in magnitude and $\delta_{ij}^k = \operatorname{sgn}\left[\left(\mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]\right)_{ij}\right]$. Algorithm 5 gives the Frank-Wolfe method specialized to problem (3.3.1).

The major advantage of Algorithm 5 lies in the simplicity of the update rules (3.3.6)-(3.3.7). Both have

Algorithm 5 Frank-Wolfe method for problem (3.3.1)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$; $\boldsymbol{V}_L^k = \tau_L \boldsymbol{D}_L^k$;
  $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$; $\boldsymbol{V}_S^k = \tau_S \boldsymbol{D}_S^k$;
  $\gamma = \frac{2}{k+2}$;
  $\boldsymbol{L}^{k+1} = \boldsymbol{L}^k + \gamma(\boldsymbol{V}_L^k - \boldsymbol{L}^k)$;
  $\boldsymbol{S}^{k+1} = \boldsymbol{S}^k + \gamma(\boldsymbol{V}_S^k - \boldsymbol{S}^k)$;
end for

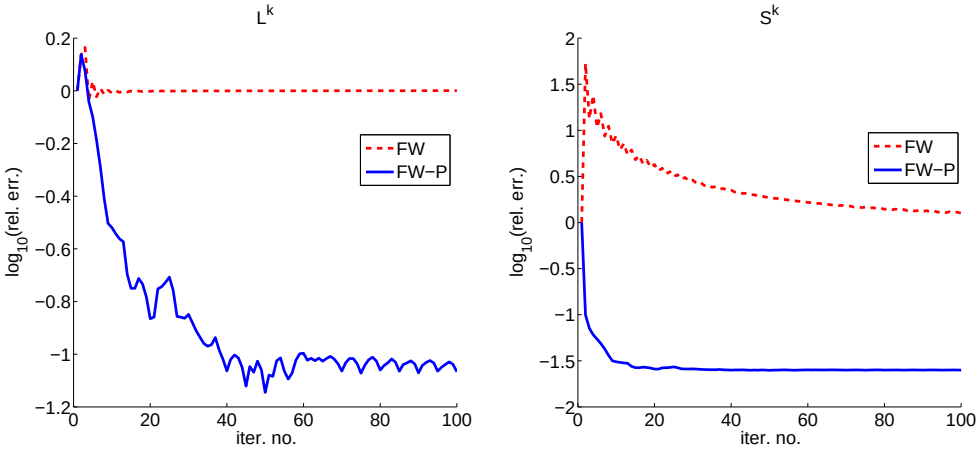


Figure 3.1: Comparisons between Algorithms 5 and 6 for problem (3.3.1) on synthetic data. The data are generated in MATLAB as $m = 1000;\ n = 1000;\ r = 5;\ L_0 = randn(m,r) * randn(r,n);\ Omega = ones(m,n);\ S_0 = 100 * randn(m,n). * (rand(m,n) < 0.01);\ M = L_0 + S_0 + randn(m,n);\ \tau_L = norm\_nuc(L_0);\ \tau_S = norm(vec(S_0),1)$; The left figure plots $\log_{10}(\left\|\boldsymbol{L}^k - \boldsymbol{L}_0\right\|_F / \|\boldsymbol{L}_0\|_F)$ versus the iteration number k. The right figure plots $\log_{10}(\left\|\boldsymbol{S}^k - \boldsymbol{S}_0\right\|_F / \|\boldsymbol{S}_0\|_F)$ versus k. The FW-P method is clearly more efficient than the straightforward FW method in recovering $\boldsymbol{L}_0$ and $\boldsymbol{S}_0$.

closed form, and both can be computed in time (essentially) linear in the size of the input. Because $\boldsymbol{V}_L^k$ is rank-one, the algorithm can be viewed as performing a sequence of rank one updates.

The major disadvantage of Algorithm 5 is that $\boldsymbol{S}$ has only a one-sparse update at each iteration, since $\boldsymbol{V}_S^k = -\tau_S \cdot \delta_{i^\star j^\star} \cdot \boldsymbol{e}_{i^\star}^k (\boldsymbol{e}_{j^\star}^k)^\top$ has only one nonzero entry. This is a significant disadvantage in practice, as the optimal $\boldsymbol{S}^\star$ may have a relatively large number of nonzero entries. Indeed, in theory, the CPCP relaxation works even when a constant fraction of the entries in $\boldsymbol{S}_0$ are nonzero. In applications such as foreground-background separation, the number of nonzero entries in the target sparse term can be quite large. The dashed curves in Figure 3.1 show the effect of this on the practical convergence of the algorithm, on a simulated example of size $1,000 \times 1,000$, in which about 1% of the entries in the target sparse matrix $\boldsymbol{S}_0$ are nonzero. As shown, the progress is quite slow.

3.3.2 FW-P algorithm: combining Frank-Wolfe and projected gradient

To overcome the drawback of the naive Frank-Wolfe algorithm described above, we propose incorporating an additional gradient projection step after each Frank-Wolfe update. This additional step updates the sparse term $\boldsymbol{S}$ only, with the goal of accelerating convergence in these variables. At iteration k, let $(\boldsymbol{L}^{k+1/2}, \boldsymbol{S}^{k+1/2})$ be the result produced by Frank-Wolfe. To produce the next iterate, we retain the low rank term $\boldsymbol{L}^{k+1/2}$, but set

$$\begin{aligned} \boldsymbol{S}^{k+1} &= \mathcal{P}_{\|\cdot\|_1 \leq \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}})\right] && (3.3.8)\\ &= \mathcal{P}_{\|\cdot\|_1 \leq \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right]; && (3.3.9)\end{aligned}$$

i.e. we simply take an additional projected gradient step in the sparse term $\boldsymbol{S}$. The resulting algorithm is presented as Algorithm 6 below. We call this method the FW-P algorithm, as it combines Frank-Wolfe steps and projections. In Figure 3.1, we compare Algorithms 5 and 6 on synthetic data. In this example, the FW-P method is clearly more efficient in recovering $\boldsymbol{L}_0$ and $\boldsymbol{S}_0$.

Algorithm 6 FW-P method for problem (3.3.1)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \leq 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L\rangle$; $\boldsymbol{V}_L^k = \tau_L \boldsymbol{D}_L^k$;
  $\boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \leq 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S\rangle$; $\boldsymbol{V}_S^k = \tau_S \boldsymbol{D}_S^k$;
  $\gamma = \frac{2}{k+2}$;
  $\boldsymbol{L}^{k+\frac{1}{2}} = \boldsymbol{L}^k + \gamma(\boldsymbol{V}_L^k - \boldsymbol{L}^k)$;
  $\boldsymbol{S}^{k+\frac{1}{2}} = \boldsymbol{S}^k + \gamma(\boldsymbol{V}_S^k - \boldsymbol{S}^k)$;
  $\boldsymbol{S}^{k+1} = \mathcal{P}_{\|\cdot\|_1 \leq \tau_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right]$;
  $\boldsymbol{L}^{k+1} = \boldsymbol{L}^{k+\frac{1}{2}}$;
end for

The convergence of Algorithm 6 can be analyzed by recognizing it as a specific instance of the generalized Frank-Wolfe iteration in Algorithm 4. This projection step (3.3.9) can be regarded as a proximal step to set $\boldsymbol{S}^{k+1}$ as

$$\begin{aligned}\arg\min_{\|\boldsymbol{S}\|_1 \leq \tau_S} \quad & \hat{l}^{k+\frac{1}{2}}(\boldsymbol{S}) := l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}) + \\ & \langle \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}), \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\rangle + \frac{1}{2}\left\|\boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\right\|_F^2.\end{aligned}$$

It can then be easily verified that

$$\hat{l}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+\frac{1}{2}}) = l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}), \quad \text{and} \quad \hat{l}^{k+\frac{1}{2}}(\boldsymbol{S}) \geq l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}) \ \text{ for any } \boldsymbol{S}, \tag{3.3.10}$$

since $\nabla_{\boldsymbol{S}} l(\boldsymbol{L}, \boldsymbol{S})$ is 1-Lipschitz. This implies that the FW-P algorithm chooses a next iterate whose objective is no worse than that produced by the Frank-Wolfe step:

$$l(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}) = l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+1}) \leq \hat{l}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+1}) \leq \hat{l}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+\frac{1}{2}}) = l(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}).$$

This is precisely the property that is required to invoke Algorithm 4 and Theorems 3.1 and 3.3. Using Lemmas 3.8 and 3.9 to estimate the Lipschitz constant of ∇l and the diameter of $\mathcal{D}$, we obtain the following result, which shows that FW-P retains the $O(1/k)$ convergence rate of the original FW method:

Theorem 3.7 *Let $l^\star$ be the optimal value to problem (3.3.1), $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k)$ and $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k)$ be the sequence produced by Algorithm 6. Then we have*

$$l(\boldsymbol{L}^k, \boldsymbol{S}^k) - l^\star \leq \frac{16(\tau_L^2 + \tau_S^2)}{k+2}. \tag{3.3.11}$$

Moreover, for any $K \geq 1$, there exists $1 \leq \tilde{k} \leq K$ such that the surrogate duality gap (defined in (3.2.13)) satisfies

$$d(\boldsymbol{x}^{\tilde{k}}) = \left\langle \boldsymbol{x}^{\tilde{k}} - \boldsymbol{v}^{\tilde{k}}, \nabla l(\boldsymbol{x}^{\tilde{k}}) \right\rangle \leq \frac{48(\tau_L^2 + \tau_S^2)}{K+2}. \tag{3.3.12}$$

Proof Substituting $L = 2$ (Lemma 3.5) and $D \leq 2\sqrt{\tau_L^2 + \tau_S^2}$ (Lemma 3.6) into Theorems 3.1 and 3.3, we can easily obtain the above result. ∎

3.4 Frank-Wolfe-Thresholding Method for Penalized Problem

In this section, we develop a scalable algorithm for the penalized version of the CPCP problem,

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad f(\boldsymbol{L}, \boldsymbol{S}) \doteq \frac{1}{2} \|\mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}]\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* + \lambda_S \|\boldsymbol{S}\|_1 . \tag{3.4.1}$$

In Section 3.4.1, we reformulate problem (3.4.1) into the form of (3.2.1) so that the Frank-Wolfe method can be applied. In Section 3.4.2, we apply the Frank-Wolfe method directly to the reformulated problem, achieving linear per-iteration cost and $O(1/k)$ convergence in function value. However, because it updates the sparse term one element at a time, it converges very slowly on typical numerical examples. In Section 3.4, we introduce our FW-T method, which resolves this issue. Our FW-T method essentially exploits *the Frank-Wolfe step to handle the nuclear norm* and *a proximal gradient step to handle the ℓ_1-norm*, while *keeping iteration cost low* and *retaining convergence guarantees*.

3.4.1 Reformulation as smooth, constrained optimization

Note that problem (3.4.1) has a non-differentiable objective function and an unbounded feasible set. To apply the Frank-Wolfe method, we exploit a two-step reformulation to transform (3.4.1) into the form of (3.2.1). First, we borrow ideas from [HJN14] and work with the epigraph reformulation of (3.4.1),

$$\begin{aligned} \min \quad & g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) \doteq \frac{1}{2} \|\mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}]\|_F^2 + \lambda_L t_L + \lambda_S t_S \\ \text{s.t.} \quad & \|\boldsymbol{L}\|_* \leq t_L, \ \|\boldsymbol{S}\|_1 \leq t_S, \end{aligned} \tag{3.4.2}$$

obtained by introducing auxiliary variables t_L and t_S. Now the objective function $g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S)$ is differentiable, with

$$\nabla_L g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) = \nabla_S g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) = \mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}], \tag{3.4.3}$$

$$\nabla_{t_L} g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) = \lambda_L, \quad \nabla_{t_S} g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) = \lambda_S. \tag{3.4.4}$$

A calculation, which we summarize in the following lemma, shows that the gradient $\nabla g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) = (\nabla_L g, \nabla_S g, \nabla_{t_L} g, \nabla_{t_S} g)$ is 2-Lipschitz:

Lemma 3.8 *For all* $(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S)$ *and* $(\boldsymbol{L}', \boldsymbol{S}', t_L', t_S')$ *feasible to* (3.4.2),

$$\|\nabla g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) - \nabla g(\boldsymbol{L}', \boldsymbol{S}', t_L', t_S')\|_F \leq 2 \|(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) - (\boldsymbol{L}', \boldsymbol{S}', t_L', t_S')\|_F . \tag{3.4.5}$$

Proof Based on (3.4.3) and (3.4.4), it follows directly that

$$\begin{aligned} \|\nabla g(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) - \nabla g(\boldsymbol{L}', \boldsymbol{S}', t_L', t_S')\|_F^2 &\leq 4 \|\boldsymbol{L} - \boldsymbol{L}'\|_F^2 + 4 \|\boldsymbol{S} - \boldsymbol{S}'\|_F^2 \\ &\leq 4 \|(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) - (\boldsymbol{L}', \boldsymbol{S}', t_L', t_S')\|_F^2 , \end{aligned}$$

which implies the result. ∎

However, the Frank-Wolfe method still cannot deal with (3.4.2), since its feasible region is unbounded. If we could somehow obtain upper bounds on the optimal values of t_L and t_S: $U_L \geq t_L^\star$ and $U_S \geq t_S^\star$, then we could solve the equivalent problem

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathcal{P}_Q[\boldsymbol{L} + \boldsymbol{S} - \boldsymbol{M}]\|_F^2 + \lambda_L t_L + \lambda_S t_S \\ \text{s.t.} \quad & \|\boldsymbol{L}\|_* \leq t_L \leq U_L, \ \|\boldsymbol{S}\|_1 \leq t_S \leq U_S, \end{aligned} \tag{3.4.6}$$

which now has a compact and convex feasible set. One simple way to obtain such U_L, U_S is as follows. One trivial feasible solution to problem (3.4.2) is $\boldsymbol{L} = \boldsymbol{0}$, $\boldsymbol{S} = \boldsymbol{0}$, $t_L = 0$, $t_S = 0$. This solution has objective value

$\frac{1}{2}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2$. Hence, the optimal objective value is no larger than this. This implies that for any optimal $t_L^\star, t_S^\star$,

$$t_L^\star \;\le\; \frac{1}{2\lambda_L}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2\,, \qquad t_S^\star \;\le\; \frac{1}{2\lambda_S}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2\,. \tag{3.4.7}$$

Hence, we can always choose

$$U_L = \frac{1}{2\lambda_L}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2\,, \quad U_S = \frac{1}{2\lambda_S}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2 \tag{3.4.8}$$

to produce a valid, bounded feasible region. The following lemma bounds its diameter D:

Lemma 3.9 *The feasible set* $\mathcal{D} = \{(\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) \mid \|\boldsymbol{L}\|_* \le t_L \le U_L,\ \|\boldsymbol{S}\|_1 \le t_S \le U_S\}$ *has diameter* $D \le \sqrt{5}\cdot\sqrt{U_L^2 + U_S^2}$.

Proof Since for any $\boldsymbol{Z} = (\boldsymbol{L}, \boldsymbol{S}, t_L, t_S)$, $\boldsymbol{Z}' = (\boldsymbol{L}', \boldsymbol{S}', t_L', t_S') \in \mathcal{D}$, we have

$$\begin{aligned}
\|\boldsymbol{Z} - \boldsymbol{Z}'\|_F^2 &= \|\boldsymbol{L} - \boldsymbol{L}'\|_F^2 + \|\boldsymbol{S} - \boldsymbol{S}'\|_F^2 + (t_L - t_L')^2 + (t_S - t_S')^2 \\
&\le (\|\boldsymbol{L}\|_F + \|\boldsymbol{L}'\|_F)^2 + (\|\boldsymbol{S}\|_F + \|\boldsymbol{S}'\|_F)^2 + (t_L - t_L')^2 + (t_S - t_S')^2 \\
&\le (\|\boldsymbol{L}\|_* + \|\boldsymbol{L}'\|_*)^2 + (\|\boldsymbol{S}\|_1 + \|\boldsymbol{S}'\|_1)^2 + (t_L - t_L')^2 + (t_S - t_S')^2 \\
&\le (U_L + U_L)^2 + (U_S + U_S)^2 + U_L^2 + U_S^2 \\
&= 5(U_L^2 + U_S^2),
\end{aligned}$$

which implies the result. ∎

With these modifications, we can apply Frank-Wolfe directly to obtain a solution $(\widehat{\boldsymbol{L}}, \widehat{\boldsymbol{S}}, \widehat{t_L}, \widehat{t_S})$ to (3.4.6), and hence to produce a solution $(\widehat{\boldsymbol{L}}, \widehat{\boldsymbol{S}})$ to the original problem (3.4.1). In subsection 3.4.2, we describe how to do this. Unfortunately, this straightforward solution has two main disadvantages. First, as in the norm constrained case, it produces only one-sparse updates to $\boldsymbol{S}$, which results in slow convergence. Second, the exact primal convergence rate in Theorem 3.1 depends on the diameter of the feasible set, which in turn depends on the accuracy of our (crude) upper bounds U_L and U_S. In subsection 3.4.3, we show how to remedy both issues, yielding a Frank-Wolfe-Thresholding method that performs significantly better in practice.

3.4.2 Frank-Wolfe for problem (3.4.6)

Applying the Frank-Wolfe method in Algorithm 3 generates a sequence of iterates $\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$. Using the expressions for the gradient in (3.4.3) and (3.4.4), at each iteration, $\boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k, V_{t_L}^k, V_{t_S}^k)$ is generated by solving the linearized subproblem

$$\boldsymbol{v}^k \in \arg\min_{\boldsymbol{v}\in\mathcal{D}} \quad \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_L + \boldsymbol{V}_S \right\rangle + \lambda_L V_{t_L} + \lambda_S V_{t_S}, \tag{3.4.9}$$

which can be decoupled into two independent subproblems,

$$(\boldsymbol{V}_L^k, V_{t_L}^k) \in \arg\min_{\|\boldsymbol{V}_L\|_* \le V_{t_L} \le U_L} g_L(\boldsymbol{V}_L, V_{t_L}) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_L \right\rangle + \lambda_L V_{t_L} \tag{3.4.10}$$

$$(\boldsymbol{V}_S^k, V_{t_S}^k) \in \arg\min_{\|\boldsymbol{V}_S\|_1 \le V_{t_S} \le U_S} g_S(\boldsymbol{V}_S, V_{t_S}) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{V}_S \right\rangle + \lambda_S V_{t_S}. \tag{3.4.11}$$

Let us consider problem (3.4.10) first. Set

$$\boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \quad \hat{g}_L(\boldsymbol{D}_L) \doteq \left\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}], \boldsymbol{D}_L \right\rangle + \lambda_L. \tag{3.4.12}$$

Because $g_L(\boldsymbol{V}_L, V_{t_L})$ is a homogeneous function, i.e., $g_L(\alpha \boldsymbol{V}_L, \alpha V_{t_L}) = \alpha g_L(\boldsymbol{V}_L, V_{t_L})$, for any $\alpha \in \mathbb{R}$, its optimal value $g_L(\boldsymbol{V}_L^k, V_{t_L}^k) = V_{t_L}^k \hat{g}_L(\boldsymbol{D}_L^k)$. Hence $V_{t_L}^k = U_L$ if $\hat{g}_L(\boldsymbol{D}_L^k) < 0$, and $V_{t_L}^k = 0$ if $\hat{g}_L(\boldsymbol{D}_L^k) > 0$. From this observation, it can be easily verified (see also [HJN14, Lemma 1] for a more general treatment) that

$$(\boldsymbol{V}_L^k, V_{t_L}^k) \in \begin{cases} \{(\boldsymbol{0}, 0)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) > 0 \\ \operatorname{conv}\{(\boldsymbol{0}, 0), U_L(\boldsymbol{D}_L^k, 1)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) = 0 \\ \{U_L(\boldsymbol{D}_L^k, 1)\} & \text{if } \hat{g}_L(\boldsymbol{D}_L^k) < 0. \end{cases} \tag{3.4.13}$$

In a similar manner, we can update $(\boldsymbol{V}_S^k, V_{t_S}^k)$. This leads fairly directly to the implementation of the Frank-Wolfe method for problem (3.4.6), described in Algorithm 7. As a direct corollary of Theorem 3.1, using parameters calculated in Lemmas 3.8 and 3.9, we have

Corollary 3.10 *Let $\boldsymbol{x}^\star = (\boldsymbol{L}^\star, \boldsymbol{S}^\star, t_L^\star, t_S^\star)$ be an optimal solution to (3.4.6). For $\{\boldsymbol{x}^k\}$ generated by Algorithm 7, we have for $k = 0,\ 1,\ 2,\ \ldots,$*

$$g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{20(U_L^2 + U_S^2)}{k+2}. \tag{3.4.14}$$

Proof Applying Theorem 3.1 with parameters calculated in Lemmas 3.8 and 3.9, we directly have

$$g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{2 \cdot 2 \cdot \left(\sqrt{5(U_L^2 + U_S^2)}\right)^2}{k+2} = \frac{20(U_L^2 + U_S^2)}{k+2}. \tag{3.4.15}$$

Algorithm 7 Frank-Wolfe method for problem (3.4.6)

1: **Initialization:** $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$; $t_L^0 = t_S^0 = 0$;
2: **for** $k = 0,\ 1,\ 2,\ \ldots$ **do**
3: $\quad \boldsymbol{D}_L^k \in \arg\min_{\|\boldsymbol{D}_L\|_* \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L \rangle$;
4: $\quad \boldsymbol{D}_S^k \in \arg\min_{\|\boldsymbol{D}_S\|_1 \le 1} \langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S \rangle$;
5: $\quad$ **if** $\lambda_L \ge -\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_L^k \rangle$ **then**
6: $\quad\quad \boldsymbol{V}_L^k = \boldsymbol{0}$; $V_{t_L}^k = 0$
7: $\quad$ **else**
8: $\quad\quad \boldsymbol{V}_L^k = U_L \boldsymbol{D}_L^k$, $V_{t_L}^k = U_L$;
9: $\quad$ **end if**
10: $\quad$ **if** $\lambda_S \ge -\langle \mathcal{P}_Q[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}],\ \boldsymbol{D}_S^k \rangle$ **then**
11: $\quad\quad \boldsymbol{V}_S^k = \boldsymbol{0}$; $V_{t_S}^k = 0$;
12: $\quad$ **else**
13: $\quad\quad \boldsymbol{V}_S^k = U_S \boldsymbol{D}_S^k$, $V_{t_S}^k = U_S$;
14: $\quad$ **end if**
15: $\quad \gamma = \frac{2}{k+2}$;
16: $\quad \boldsymbol{L}^{k+1} = (1-\gamma)\boldsymbol{L}^k + \gamma \boldsymbol{V}_L^k$, $t_L^{k+1} = (1-\gamma) t_L^k + \gamma V_{t_L}^k$;
17: $\quad \boldsymbol{S}^{k+1} = (1-\gamma)\boldsymbol{S}^k + \gamma \boldsymbol{V}_S^k$, $t_S^{k+1} = (1-\gamma) t_S^k + \gamma V_{t_S}^k$;
18: **end for**

A more careful calculation below slightly improves the constant in (3.4.15).

$$\begin{aligned} g(\boldsymbol{x}^{k+1}) &= g(\boldsymbol{x}^k + \gamma(\boldsymbol{v}^k - \boldsymbol{x}^k)) \\ &\le g(\boldsymbol{x}^k) + \gamma \left\langle \nabla g(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + \gamma^2 \left\| \boldsymbol{V}_L^k - \boldsymbol{L}^k \right\|_F^2 + \gamma^2 \left\| \boldsymbol{V}_S^k - \boldsymbol{S}^k \right\|_F^2 \\ &\le g(\boldsymbol{x}^k) + \gamma \left\langle \nabla g(\boldsymbol{x}^k), \boldsymbol{v}^k - \boldsymbol{x}^k \right\rangle + 4\gamma^2 (U_L^2 + U_S^2), \end{aligned} \tag{3.4.16}$$

where the second line holds by noting that g is only linear in t_L and t_S; the last line holds as

$$\begin{aligned} \left\| \boldsymbol{V}_L^k - \boldsymbol{L}^k \right\|_F^2 &\le (\left\| \boldsymbol{V}_L^k \right\|_F + \left\| \boldsymbol{L}^k \right\|_F)^2 \le (U_L + U_L)^2 = 4U_L^2, \quad \text{and} \\ \left\| \boldsymbol{V}_S^k - \boldsymbol{S}^k \right\|_F^2 &\le (\left\| \boldsymbol{V}_S^k \right\|_F + \left\| \boldsymbol{S}^k \right\|_F)^2 \le (U_S + U_S)^2 = 4U_S^2. \end{aligned}$$

Following the arguments in the proof of Theorem 1 with (3.2.10) replaced by (3.4.16), we can easily obtain that

$$g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{16(U_L^2 + U_S^2)}{k+2}.$$

■

In addition to the above convergence result, another major advantage of Algorithm 7 is the simplicity of the update rules (lines 3-4 in Algorithm 7). Both have closed-form solutions that can be computed in time (essentially) linearly dependent on the size of the input.

However, two clear limitations substantially hinder Algorithm 7's efficiency. First, as in the norm con-

strained case, $\boldsymbol{V}_S^k$ has only one nonzero entry, so $\boldsymbol{S}$ has a one-sparse update in each iteration. Second, the exact rate of convergence relies on our (crude) guesses of U_L and U_S (Corollary 3.10). In the next subsection, we present remedies to resolve both issues.

3.4.3 FW-T algorithm: combining Frank-Wolfe and proximal methods

To alleviate the difficulties faced by Algorithm 7, we propose a new algorithm called Frank-Wolfe-Thresholding (FW-T) (Algorithm 8), that combines a modified FW step with a proximal gradient step. Below we highlight the key features of FW-T.

Proximal gradient step for $\boldsymbol{S}$ To update $\boldsymbol{S}$ in a more efficient way, we incorporate an additional proximal gradient step for $\boldsymbol{S}$. At iteration k, let $(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}})$ be the result produced by Frank-Wolfe step. To produce the next iterate, we retain the low-rank term $\boldsymbol{L}^{k+\frac{1}{2}}$, but execute a proximal gradient step for the function $f(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S})$ at the point $\boldsymbol{S}^{k+\frac{1}{2}}$, i.e.

$$\begin{aligned}\boldsymbol{S}^{k+1} &\in \arg\min_{\boldsymbol{S}} \left\langle \nabla_{\boldsymbol{S}} f(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}),\ \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\rangle + \frac{1}{2}\left\|\boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1 \\ &= \arg\min_{\boldsymbol{S}} \left\langle \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}], \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}} \right\rangle + \frac{1}{2}\left\|\boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1 \end{aligned} \tag{3.4.17}$$

which can be easily computed using the soft-thresholding operator:

$$\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right]. \tag{3.4.18}$$

Exact line search For the Frank-Wolfe step, instead of choosing the fixed step length $\frac{2}{k+2}$, we implement an exact line search by solving a two-dimensional quadratic problem (3.4.20), as in [HJN14]. This modification turns out to be crucial to achieve a primal convergence result that only weakly depends on the tightness of our guesses U_L and U_S.

Adaptive updates of U_L and U_S We initialize U_L and U_S using the crude bound (3.4.8). Then, at the end of the k-iteration, we respectively update

$$U_L^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_L, \quad U_S^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_S. \tag{3.4.19}$$

This scheme maintains the property that $U_L^{k+1} \geq t_L^\star$ and $U_S^{k+1} \geq t_S^\star$. Moreover, we prove (Lemma 3.11) that g is non-increasing through our algorithm, and so this scheme produces a sequence of tighter upper bounds

Algorithm 8 FW-T method for problem (3.4.1)

1: **Input:** data matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$; weights λ_L, $\lambda_S > 0$; max iteration number T;
2: **Initialization:** $\boldsymbol{L}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$; $t_L^0 = t_S^0 = 0$; $U_L^0 = g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0)/\lambda_L$; $U_S^0 = g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0)/\lambda_S$;
3: **for** $k = 0,\ 1,\ 2,\ \cdots,\ T$ **do**
4: *same as lines 3-14 in Algorithm 7;*
5: $\left(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}}\right)$ is computed as an optimizer to

$$\begin{aligned} \min \quad & \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}+\boldsymbol{S}-\boldsymbol{M}]\right\|_F^2 + \lambda_L t_L + \lambda_S t_S \\ \text{s.t.} \quad & \begin{pmatrix} \boldsymbol{L} \\ t_L \end{pmatrix} \in \operatorname{conv}\left\{\begin{pmatrix} \boldsymbol{L}^k \\ t_L^k \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_L^k \\ V_{t_L}^k \end{pmatrix}\right\} \\ & \begin{pmatrix} \boldsymbol{S} \\ t_S \end{pmatrix} \in \operatorname{conv}\left\{\begin{pmatrix} \boldsymbol{S}^k \\ t_S^k \end{pmatrix}, \begin{pmatrix} \boldsymbol{V}_S^k \\ V_{t_S}^k \end{pmatrix}\right\}; \end{aligned} \tag{3.4.20}$$

6: $\boldsymbol{S}^{k+1} = \mathcal{T}\left[\boldsymbol{S}^{k+\frac{1}{2}} - \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}], \lambda_S\right]$;
7: $\boldsymbol{L}^{k+1} = \boldsymbol{L}^{k+\frac{1}{2}}$, $t_L^{k+1} = t_L^{k+\frac{1}{2}}$; $t_S^{k+1} = \left\|\boldsymbol{S}^{k+1}\right\|_1$;
8: $U_L^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_L$;
9: $U_S^{k+1} = g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1})/\lambda_S$;
10: **end for**

for $U_L^\star$ and $U_S^\star$. Although this dynamic scheme does not improve the theoretical convergence result, some acceleration is empirically exhibited.

Convergence analysis Since both the FW step and the proximal gradient step do not increase the objective value, we can easily recognize FW-T method as a descent algorithm:

Lemma 3.11 *Let $\{(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)\}$ be the sequence of iterates produced by the FW-T algorithm. For each $k = 0, 1, 2\cdots$,*

$$g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1}) \le g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}}) \le g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k). \tag{3.4.21}$$

Proof Since $(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$ is always feasible to the quadratic program (3.4.20),

$$g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}}) \le g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k). \tag{3.4.22}$$

Based on (3.4.17), the threshold step (line 6 in Algorithm 3) can be written as

$$\begin{aligned} \boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \quad & \hat{g}^{k+\frac{1}{2}}(\boldsymbol{S}) \doteq \frac{1}{2}\left\|\mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}]\right\|_F^2 + \lambda_L t_L^{k+\frac{1}{2}} + \lambda_S \left\|\boldsymbol{S}\right\|_1 \\ & + \langle \mathcal{P}_Q[\boldsymbol{L}^{k+\frac{1}{2}} + \boldsymbol{S}^{k+\frac{1}{2}} - \boldsymbol{M}],\ \boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\rangle + \frac{1}{2}\left\|\boldsymbol{S} - \boldsymbol{S}^{k+\frac{1}{2}}\right\|_F^2. \end{aligned}$$

The following properties of $\hat{g}^{k+\frac{1}{2}}(\cdot)$ can be easily verified

$$\hat{g}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+\frac{1}{2}}) = g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, \|\boldsymbol{S}^{k+\frac{1}{2}}\|_1) \le g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}});$$

$$\hat{g}^{k+\frac{1}{2}}(\boldsymbol{S}) \ge g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}, t_L^{k+\frac{1}{2}}, \|\boldsymbol{S}\|_1), \quad \text{for any } \boldsymbol{S}.$$

Therefore, we have

$$\begin{aligned} g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1}) &= g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+1}, t_L^{k+\frac{1}{2}}, t_S^{k+1}) \le \hat{g}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+1}) \\ &\le \hat{g}^{k+\frac{1}{2}}(\boldsymbol{S}^{k+\frac{1}{2}}) \le g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}}) \end{aligned} \tag{3.4.23}$$

Combining (3.4.22) and (3.4.23), we obtain

$$g(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}, t_L^{k+1}, t_S^{k+1}) \le g(\boldsymbol{L}^{k+\frac{1}{2}}, \boldsymbol{S}^{k+\frac{1}{2}}, t_L^{k+\frac{1}{2}}, t_S^{k+\frac{1}{2}}) \le g(\boldsymbol{L}^{k}, \boldsymbol{S}^{k}, t_L^{k}, t_S^{k}).$$

■

Moreover, we can establish primal convergence (almost) independent of U_L^0 and U_S^0:

Theorem 3.12 *Let $r_L^\star$ and $r_S^\star$ be the smallest radii such that*

$$\left\{ (\boldsymbol{L}, \boldsymbol{S}) \;\middle|\; f(\boldsymbol{L}, \boldsymbol{S}) \le g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0) = \frac{1}{2}\|\mathcal{P}_Q[\boldsymbol{M}]\|_F^2 \right\} \subseteq \overline{B(r_L^\star)} \times \overline{B(r_S^\star)}, \tag{3.4.24}$$

where $\overline{B(r)} \doteq \{\boldsymbol{X} \in \mathbb{R}^{m\times n} | \, \|\boldsymbol{X}\|_F \le r\}$ for any $r \ge 0$.[a] *Then for the sequence $\{(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)\}$ generated by Algorithm 8, we have*

$$\begin{aligned} &g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k) - g(\boldsymbol{L}^\star, \boldsymbol{S}^\star, t_L^\star, t_S^\star) \\ &\qquad \le \frac{\min\{4(t_L^\star + r_L^\star)^2 + 4(t_S^\star + r_S^\star)^2,\ 16(U_L^0)^2 + 16(U_S^0)^2\}}{k+2}. \end{aligned} \tag{3.4.25}$$

[a] Since the objective function in problem (3.4.1) is coercive, i.e. $\lim_{k\to+\infty} f(\boldsymbol{L}^k, \boldsymbol{S}^k) = +\infty$ for any sequence $(\boldsymbol{L}^k, \boldsymbol{S}^k)$ such that $\lim_{k\to+\infty} \left\|(\boldsymbol{L}^k, \boldsymbol{S}^k)\right\|_F = +\infty$, clearly $r_L^\star \ge 0$ and $r_S^\star \ge 0$ exist.

Proof For notational convenience, we denote

$$\boldsymbol{x}^k = (\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k),\ \boldsymbol{x}^\star = (\boldsymbol{L}^\star, \boldsymbol{S}^\star, t_L^\star, t_S^\star) \text{ and } \boldsymbol{v}^k = (\boldsymbol{V}_L^k, \boldsymbol{V}_S^k, \boldsymbol{V}_{t_L}^k, \boldsymbol{V}_{t_S}^k).$$

For any point $\boldsymbol{x} = (\boldsymbol{L}, \boldsymbol{S}, t_L, t_S) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \times \mathbb{R} \times \mathbb{R}$, we adopt the notation that $\boldsymbol{L}[\boldsymbol{x}] = \boldsymbol{L}$, $\boldsymbol{S}[\boldsymbol{x}] = \boldsymbol{S}$, $t_L[\boldsymbol{x}] = t_L$ and $t_S[\boldsymbol{x}] = t_S$.

Since $g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{16(U_L^0)^2 + 16(U_S^0)^2}{k+2}$ can be easily established following the proof of Corollary 3.10, below we will focus on the other part that $g(\boldsymbol{x}^k) - g(\boldsymbol{x}^\star) \le \frac{4(t_L^\star + r_L^\star)^2 + 4(t_S^\star + r_S^\star)^2}{k+2}$.

Let us first make two simple observations.

Since $f(\boldsymbol{L}^\star, \boldsymbol{S}^\star) \le g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$, we have

$$U_L^k = g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)/\lambda_L \ge t_L^\star \quad \text{and} \quad U_S^k = g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)/\lambda_S \ge t_S^\star. \tag{3.4.26}$$

Therefore, our U_L^k and U_S^k always bound $t_L^\star$ and $t_S^\star$ from above.

From Lemma 3.11, $g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k)$ is non-increasing,

$$f(\boldsymbol{L}^k, \boldsymbol{S}^k) \le g(\boldsymbol{L}^k, \boldsymbol{S}^k, t_L^k, t_S^k) \le g(\boldsymbol{L}^0, \boldsymbol{S}^0, t_L^0, t_S^0),$$

which implies that $(\boldsymbol{L}^k, \boldsymbol{S}^k) \subseteq \overline{B(r_L^\star)} \times \overline{B(r_S^\star)}$, i.e. $\|\boldsymbol{L}^k\|_F \le r_L^\star$ and $\|\boldsymbol{S}^k\|_F \le r_S^\star$.

Let us now consider the k-th iteration. Similar to the proof in [HJN14], we introduce the auxiliary point $\boldsymbol{v}_+^k = (\frac{t_L^\star}{U_L^k}\boldsymbol{V}_L^k, \frac{t_S^\star}{U_S^k}\boldsymbol{V}_S^k, \frac{t_L^\star}{U_L^k}\boldsymbol{V}_{t_L}^k, \frac{t_S^\star}{U_S^k}\boldsymbol{V}_{t_S}^k)$. Then based on our argument for (3.4.13), it can be easily verified that

$$(\boldsymbol{L}[\boldsymbol{v}_+^k], t_L[\boldsymbol{v}_+^k]) \in \arg\min_{\|\boldsymbol{V}_L\|_* \le V_{t_L} \le t_L^\star} g_L(\boldsymbol{V}_L, V_{t_L}) \tag{3.4.27}$$

$$(\boldsymbol{S}[\boldsymbol{v}_+^k], t_S[\boldsymbol{v}_+^k]) \in \arg\min_{\|\boldsymbol{V}_S\|_1 \le V_{t_S} \le t_S^\star} g_S(\boldsymbol{V}_S, V_{t_S}). \tag{3.4.28}$$

Recall $\gamma = \frac{2}{k+2}$. We have

$$\begin{aligned}
& g(\boldsymbol{x}^{k+\frac{1}{2}}) \\
\le\ & g(\boldsymbol{x}^k + \gamma(\boldsymbol{v}_+^k - \boldsymbol{x}^k)) \\
\le\ & g(\boldsymbol{x}^k) + \gamma\langle \nabla g(\boldsymbol{x}_k),\ \boldsymbol{v}_+^k - \boldsymbol{x}^k\rangle + \gamma^2\left(\|\boldsymbol{L}[\boldsymbol{v}_+^k] - \boldsymbol{L}[\boldsymbol{x}^k]\|_F^2 + \|\boldsymbol{S}[\boldsymbol{v}_+^k] - \boldsymbol{S}[\boldsymbol{x}^k]\|_F^2\right) \\
\le\ & g(\boldsymbol{x}^k) + \gamma\left(g_L(\boldsymbol{L}[\boldsymbol{v}_+^k - \boldsymbol{x}^k], t_L[\boldsymbol{v}_+^k - \boldsymbol{x}^k]) + g_S(\boldsymbol{S}[\boldsymbol{v}_+^k - \boldsymbol{x}^k], t_S[\boldsymbol{v}_+^k - \boldsymbol{x}^k])\right) \\
& \qquad\qquad +\gamma^2\left((t_L^\star + r_L^\star)^2 + (t_S^\star + r_S^\star)^2\right) \\
\le\ & g(\boldsymbol{x}^k) + \gamma\left(g_L(\boldsymbol{L}[\boldsymbol{x}^\star - \boldsymbol{x}^k], t_L[\boldsymbol{x}^\star - \boldsymbol{x}^k]) + g_S(\boldsymbol{S}[\boldsymbol{x}^\star - \boldsymbol{x}^k], t_S[\boldsymbol{x}^\star - \boldsymbol{x}^k])\right) \\
& \qquad\qquad +\gamma^2\left((t_L^\star + r_L^\star)^2 + (t_S^\star + r_S^\star)^2\right) \\
=\ & g(\boldsymbol{x}^k) + \gamma\langle \nabla g(\boldsymbol{x}^k),\ \boldsymbol{x}^\star - \boldsymbol{x}^k\rangle + \gamma^2\left((t_L^\star + r_L^\star)^2 + (t_S^\star + r_S^\star)^2\right) \\
\le\ & g(\boldsymbol{x}^k) + \gamma\left(g(\boldsymbol{x}^\star) - g(\boldsymbol{x}^k)\right) + \gamma^2\left((t_L^\star + r_L^\star)^2 + (t_S^\star + r_S^\star)^2\right),
\end{aligned}$$

where the first inequality holds since $\boldsymbol{x}^k + \gamma(\boldsymbol{v}_+^k - \boldsymbol{x}^k)$ is feasible to the quadratic program (3.4.20) while $\boldsymbol{x}^{k+\frac{1}{2}}$ minimizes it; the third inequality is due to the facts that

$$\|\boldsymbol{L}[\boldsymbol{v}_+^k] - \boldsymbol{L}[\boldsymbol{x}^k]\|_F \le \|\boldsymbol{L}[\boldsymbol{v}_+^k]\|_F + \|\boldsymbol{L}[\boldsymbol{x}^k]\|_F \le \|\boldsymbol{L}[\boldsymbol{v}_+^k]\|_* + \|\boldsymbol{L}[\boldsymbol{x}^k]\|_F \le t_L^\star + r_L^\star$$

$$\left\|\boldsymbol{S}[\boldsymbol{v}_+^k]-\boldsymbol{S}[\boldsymbol{x}^k]\right\|_F \quad\leq\quad \left\|\boldsymbol{S}[\boldsymbol{v}_+^k]\right\|_F+\left\|\boldsymbol{S}[\boldsymbol{x}^k]\right\|_F\leq\left\|\boldsymbol{S}[\boldsymbol{v}_+^k]\right\|_1+\left\|\boldsymbol{S}[\boldsymbol{x}^k]\right\|_F\leq t_S^\star+r_S^\star;$$

the fourth inequality holds as $(\boldsymbol{L}[\boldsymbol{x}^\star],t_L[\boldsymbol{x}^\star])$ and $(\boldsymbol{S}[\boldsymbol{x}^\star],t_S[\boldsymbol{x}^\star])$ are respectively feasible to (3.4.27) and (3.4.28) while $(\boldsymbol{L}[\boldsymbol{v}_+^k],t_L[\boldsymbol{v}_+^k])$ and $(\boldsymbol{S}[\boldsymbol{v}_+^k],t_S[\boldsymbol{v}_+^k])$ respectively minimize (3.4.27) and (3.4.28);

Therefore, we obtain

$$g(\boldsymbol{x}^{k+\frac{1}{2}})-g(\boldsymbol{x}^\star)\leq(1-\gamma)\left(g(\boldsymbol{x}^k)-g(\boldsymbol{x}^\star)\right)+\gamma^2\left((t_L^\star+r_L^\star)^2+(t_S^\star+r_S^\star)^2\right).$$

Moreover, by Lemma 3.11, we have

$$g(\boldsymbol{x}^{k+1})\leq g(\boldsymbol{x}^{k+\frac{1}{2}}).$$

Thus, we obtain the recurrence

$$g(\boldsymbol{x}^{k+1})-g(\boldsymbol{x}^\star)\leq(1-\gamma)\left(g(\boldsymbol{x}^k)-g(\boldsymbol{x}^\star)\right)+\gamma^2\left((t_L^\star+r_L^\star)^2+(t_S^\star+r_S^\star)^2\right).$$

Applying mathematical induction, one can easily obtain that

$$g(\boldsymbol{L}^k,\boldsymbol{S}^k,t_L^k,t_S^k)-g(\boldsymbol{L}^\star,\boldsymbol{S}^\star,t_L^\star,t_S^\star)\leq\frac{4\left((t_L^\star+r_L^\star)^2+(t_S^\star+r_S^\star)^2\right)}{k+2}.$$

■

Since U_L^0 and U_S^0 are quite crude upper bounds for $t_L^\star$ and $t_S^\star$, $16(U_L^0)^2+16(U_S^0)^2$ could be much larger than $4(t_L^\star+r_L^\star)^2+4(t_S^\star+r_S^\star)^2$. Therefore, this primal convergence results depend on U_L^0 and U_S^0 in a very weak manner.

However, the convergence result of the surrogate duality gap $d(\boldsymbol{x}^k)$ still hinges upon the upper bounds:

Theorem 3.13 *Let $\boldsymbol{x}^k$ denote $(\boldsymbol{L}^k,\boldsymbol{S}^k,t_L^k,t_S^k)$ generated by Algorithm 8. Then for any $K\geq 1$, there exists $1\leq\tilde{k}\leq K$ such that*

$$g(\boldsymbol{x}^{\tilde{k}})-g(\boldsymbol{x}^\star)\leq d(\boldsymbol{x}^{\tilde{k}})\leq\frac{48\left((U_L^0)^2+(U_S^0)^2\right)}{K+2}. \tag{3.4.29}$$

Proof Define $\Delta^k=g(\boldsymbol{x}^k)-g(\boldsymbol{x}^\star)$. Following (3.4.16), we have

$$\Delta^{k+1}\leq\Delta^k+\gamma\left\langle\nabla g(\boldsymbol{x}^k),\boldsymbol{v}^k-\boldsymbol{x}^k\right\rangle+4\gamma^2\left((U_L^0)^2+(U_S^0)^2\right). \tag{3.4.30}$$

Then following the arguments in the proof of Theorem 2 with (3.2.17) replaced by (3.4.30), we can easily obtain the result. ■

Stopping criterion Compared to the convergence of $g(\boldsymbol{x}^k)$ (Theorem 3.12), the convergence result for $d(\boldsymbol{x}^k)$ can be much slower (Theorem 3.13). Therefore, here the surrogate duality gap $d(\cdot)$ is not that suitable to serve as a stopping criterion. Consequently, in our implementation, we terminate Algorithm 8 if

$$|\left(g(\boldsymbol{x}^{k+1}) - g(\boldsymbol{x}^k)\right)|/g(\boldsymbol{x}^k) \leq \varepsilon, \tag{3.4.31}$$

for five consecutive iterations.

3.5 Numerical Experiments

In this section, we report numerical results obtained by applying our FW-T method (Algorithm 8) to problem (3.1.5) with real data arising from applications considered in [CLMW11]: *foreground/background separation in surveillance videos*, and *shadow and specularity removal from face images*.

Given observations $\{\boldsymbol{M}_0(i,j) \mid (i,j) \in \Omega\}$, where $\Omega \subseteq \{1,\ldots,m\} \times \{1,\ldots,n\}$ is the index set of the observable entries in $\boldsymbol{M}_0 \in \mathbb{R}^{m\times n}$, we assigned weights

$$\lambda_L = \delta\rho\,\|\mathcal{P}_\Omega[\boldsymbol{M}_0]\|_F \quad \text{and} \quad \lambda_S = \delta\sqrt{\rho}\,\|\mathcal{P}_\Omega[\boldsymbol{M}_0]\|_F\,/\sqrt{\max(m,n)}$$

to problem (3.1.5), [3] where $\rho = |\Omega|/mn$ and δ is chosen as 0.001 for the surveillance problem and 0.01 for the face problem.

We compared our FW-T method with the popular first-order methods *iterative soft-thresholding algorithm* (ISTA) and *fast iterative soft-thresholding algorithm* (FISTA) [BT09], both of whose implementations used partial *singular value decomposition* (SVD). In subsection 3.5.1, we provided detailed descriptions and implementations of ISTA and FISTA.

We set $\varepsilon = 10^{-3}$ in FW-T's stopping criterion (3.4.31),[4] and terminated ISTA and FISTA whenever they reached the objective value returned by the FW-T method.[5] All the experiments were conducted on a computer with Intel Xeon E5-2630 Processor (12 cores at 2.4 GHz), and 64GB RAM running MATLAB R2012b (64 bits).

[3] The ratio $\lambda_L/\lambda_S = \sqrt{\rho\max(m,n)}$ follows the suggestion in [CLMW11]. For applications in computer vision at least, our choices in λ_L and λ_S seem to be quite robust, although it is possible to improve the performance by making slight adjustments to our current settings of λ_L and λ_S.

[4] As discussed in [YZ11, YY13a], with noisy data, solving optimization problems to high accuracy does not necessarily improve the recovery quality. Consequently, we set ε to a modest value.

[5] All codes are available at: https://sites.google.com/site/mucun1988/publi

3.5.1 ISTA & FISTA for problem (3.1.5)

Iterative soft-thresholding algorithm (ISTA), is an efficient way to tackle unconstrained nonsmooth optimization problem especially at large scale. ISTA follows the general idea by iteratively minimizing an upper bound of the original objective. In particular, when applied to problem (3.1.5) of our interest, ISTA updates $(\boldsymbol{L}, \boldsymbol{S})$ for the k-th iteration by solving

$$(\boldsymbol{L}^{k+1}, \boldsymbol{S}^{k+1}) = \arg\min_{\boldsymbol{L},\boldsymbol{S}} \left\langle \begin{pmatrix} \nabla_{\boldsymbol{L}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) \\ \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) \end{pmatrix}, \begin{pmatrix} \boldsymbol{L} - \boldsymbol{L}^k \\ \boldsymbol{S} - \boldsymbol{S}^k \end{pmatrix} \right\rangle + \frac{L_f}{2} \left\| \begin{pmatrix} \boldsymbol{L} \\ \boldsymbol{S} \end{pmatrix} - \begin{pmatrix} \boldsymbol{L}^k \\ \boldsymbol{S}^k \end{pmatrix} \right\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* + \lambda_S \|\boldsymbol{S}\|_1 . \tag{3.5.1}$$

Here $L_f = 2$ denotes the Lipschitz constant of $\nabla l(\boldsymbol{L}, \boldsymbol{S})$ with respect to $(\boldsymbol{L}, \boldsymbol{S})$, and $\nabla_{\boldsymbol{L}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) = \nabla_{\boldsymbol{S}} l(\boldsymbol{L}^k, \boldsymbol{S}^k) = \mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]$. Since $\boldsymbol{L}$ and $\boldsymbol{S}$ are decoupled in (3.5.1), equivalently we have

$$\boldsymbol{L}^{k+1} = \arg\min_{\boldsymbol{L}} \left\| \boldsymbol{L} - \left(\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right) \right\|_F^2 + \lambda_L \|\boldsymbol{L}\|_* , \tag{3.5.2}$$

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \left\| \boldsymbol{S} - \left(\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right) \right\|_F^2 + \lambda_S \|\boldsymbol{S}\|_1 . \tag{3.5.3}$$

The solution to problem (3.5.3) can be given explicitly in terms of the proximal mapping of $\|\cdot\|_1$ as introduced in Section 2.2, i.e.,

$$\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right].$$

For a matrix $\boldsymbol{X}$ and any $\tau \geq 0$, let $\mathcal{D}_\tau(\boldsymbol{X})$ denote the singular value thresholding operator $\mathcal{D}_\tau(\boldsymbol{X}) = \boldsymbol{U}\mathcal{T}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^\top$, where $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the singular value decomposition of $\boldsymbol{X}$. It is not difficult to show [CCS10, MGC11] that the solution to problem (3.5.2) can be given explicitly by

$$\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}] \right].$$

Algorithm 9 summarizes our ISTA implementation for problem (3.1.5).

Algorithm 9 ISTA for problem (3.1.5)

Initialization: $\boldsymbol{L}^0 = \boldsymbol{0}$, $\boldsymbol{S}^0 = \boldsymbol{0}$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\boldsymbol{L}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]\right]$;
  $\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\boldsymbol{S}^k - \frac{1}{2}\mathcal{P}_\Omega[\boldsymbol{L}^k + \boldsymbol{S}^k - \boldsymbol{M}]\right]$;
end for

Regarding ISTA's speed of convergence, it can be proved that $f(\boldsymbol{L}^k, \boldsymbol{S}^k) - f^\star = O(1/k)$, where $f^\star$ denotes the optimal value of problem (3.1.5).

Fast iterative soft-thresholding algorithm (FISTA) introduced in [BT09], is an accelerated version of ISTA, which incorporate a momentum step borrowed from Nesterov's optimal gradient scheme [Nes83]. For FISTA, a better convergence result, $f(\boldsymbol{L}^k, \boldsymbol{S}^k) - f^\star = O(1/k^2)$, can be achieved with a cost per iteration that is comparable to ISTA. Algorithm 10 summarizes our FISTA implementation for problem (3.1.5).

Algorithm 10 FISTA for problem (3.1.5)

Initialization: $\hat{\boldsymbol{L}}^0 = \boldsymbol{L}^0 = \boldsymbol{0}$, $\hat{\boldsymbol{S}}^0 = \boldsymbol{S}^0 = \boldsymbol{0}$, $t_0 = 1$;
for $k = 0,\ 1,\ 2,\ \cdots$ **do**
  $\boldsymbol{L}^{k+1} = \mathcal{D}_{\lambda_L/2}\left[\hat{\boldsymbol{L}}^k - \frac{1}{2}\mathcal{P}_\Omega[\hat{\boldsymbol{L}}^k + \hat{\boldsymbol{S}}^k - \boldsymbol{M}]\right]$;
  $\boldsymbol{S}^{k+1} = \mathcal{T}_{\lambda_S/2}\left[\hat{\boldsymbol{S}}^k - \frac{1}{2}\mathcal{P}_\Omega[\hat{\boldsymbol{L}}^k + \hat{\boldsymbol{S}}^k - \boldsymbol{M}]\right]$;
  $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$;
  $\hat{\boldsymbol{L}}^{k+1} = \boldsymbol{L}^{k+1} + \frac{t^k-1}{t^{k+1}}(\boldsymbol{L}^{k+1} - \boldsymbol{L}^k)$;
  $\hat{\boldsymbol{S}}^{k+1} = \boldsymbol{S}^{k+1} + \frac{t^k-1}{t^{k+1}}(\boldsymbol{S}^{k+1} - \boldsymbol{S}^k)$;
end for

Partial SVD In each iteration of either ISTA or FISTA, we only need those singular values that are larger than $\lambda_S/2$ and their corresponding singular vectors. Therefore, a partial SVD can be utilized to reduce the computational burden of a full SVD. Since most partial SVD software packages (e.g. PROPACK [Lar04]) require specifying in advance the number of top singular values and singular vectors to compute, we heuristically determine this number (denoted as sv^k at iteration k). Specifically, let $d = \min\{m, n\}$, and svp^k denote the number of computed singular values that were larger than $\lambda_L/2$ in the k-th iteration. Similar to

[TY11], in our implementation, we start with $sv^0 = d/10$, and adjust sv^k dynamically as follows:

$$sv^{k+1} = \begin{cases} \min\{svp^k + 1, d\} & \text{if } svp^k < sv^k \\ \min\{svp^k + \text{round}(0.05d), d\} & \text{otherwise.} \end{cases}$$

3.5.2 Foreground-background separation in surveillance video

In surveillance videos, due to the strong correlation between frames, it is natural to model the background as low rank; while foreground objects, such as cars or pedestrians, that normally occupy only a fraction of the video, can be treated as sparse. So, if we stack each frame as a column in the data matrix $\boldsymbol{M}_0$, it is reasonable to assume that $\boldsymbol{M}_0 \approx \boldsymbol{L}_0 + \boldsymbol{S}_0$, where $\boldsymbol{L}_0$ captures the background and $\boldsymbol{S}_0$ represents the foreground movements. Here, we solved problem (3.1.5) for videos introduced in [LHGT04] and [JRP07]. The observed entries were sampled uniformly with ratio ρ chosen respectively as 1, 0.8 and 0.6.

Table 3.1 summarizes the numerical performances of FW-T, ISTA and FISTA in terms of the iteration number and running time (in seconds). As can be observed, our FW-T method is more efficient than ISTA and FISTA, and the advantage becomes more prominent as the size of the data grows and the observations are more compressed (with smaller sampling ratio ρ). Even though the FW-T method took more iterations than FISTA and in many cases than ISTA, it took less time in many cases but one due to its low per-iteration cost. To illustrate this more clearly, in Figure 3.2, we plot the per-iteration cost of these three methods on the Airport and Square videos as a function of the number of frames. The computational cost of FW-T scales linearly with the size of the data, whereas the cost of the other methods increases superlinearly. Another observation is that as the number of measurements decreases, the iteration numbers of both ISTA and FISTA methods grow substantially, while those of the FW-T method remain quite stable. This explains the more favorable behavior of the FW-T method when ρ is small. In Figure 3.3, frames of the original videos, the backgrounds and the foregrounds produced by the FW-T method are presented, and the separation achieved is quite satisfactory.

3.5.3 Shadow and specularity removal from face images

Images taken under varying illumination can also be modeled as the superposition of low-rank and sparse components. Here, the data matrix $\boldsymbol{M}_0$ is again formed by stacking each image as a column. The low-rank term $\boldsymbol{L}_0$ captures the smooth variations [BJ03], while the sparse term $\boldsymbol{S}_0$ represents cast shadows and specularities [WYG+09, ZMKW13]. CPCP can be used to remove the shadows and specularities [CLMW11,

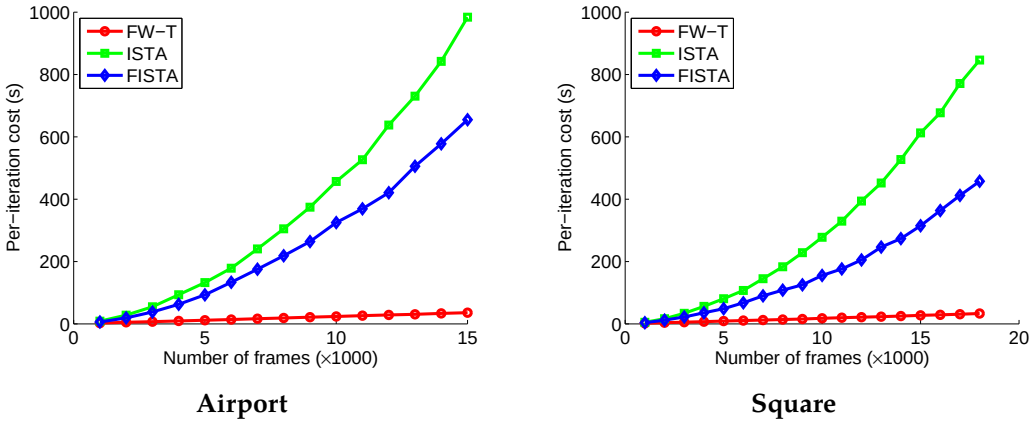


Figure 3.2: Per-iteration cost vs. the number of frames in Airport and Square videos with full observation. The per-iteration cost of our FW-T method grows linearly with the size of data, in contrast with the superlinear per-iteration cost of ISTA and FISTA. That makes the FW-T method more advantageous or may even be the only feasible choice for large problems.

ZMKW13]. Here, we solved problem (3.1.4) for YaleB face images [GBK01]. Table 3.2 summarizes the numerical performances of FW-T, ISTA and FISTA. Similar to the observation made regarding the above surveillance video experiment, the number of iterations required by ISTA and FISTA grows much faster than it does for the FW-T method when ρ decreases. However, unlike in those tests, where the number of frames in each dataset was at least several thousand, the number of frames here is just 65. This prevents the FW-T method from significantly benefiting from its linear per-iteration cost and consequently, while FW-T still outperforms ISTA for values of $\rho \leq 0.7$, the FISTA method is always the fastest. In Figure 3.4, the original images, the low-rank and the sparse parts produced by the FW-T method are presented. Visually, the recovered low-rank component is smoother and better conditioned for face recognition than the original image, while the sparse component corresponds to shadows and specularities.

3.6 Discussion

In this chapter, we have proposed scalable algorithms called Frank-Wolfe-Projection (FW-P) and Frank-Wolfe-Thresholding (FW-T) for norm constrained and penalized versions of CPCP. Essentially, these methods combine classical ideas in Frank-Wolfe and Proximal methods to achieve linear per-iteration cost, $O(1/k)$ convergence in function value and practical efficiency in updating the sparse component. Extensive numerical experiments were conducted on computer vision related applications of CPCP, which demonstrated the great potential of our methods for dealing with problems of very large scale. Moreover, the general idea of

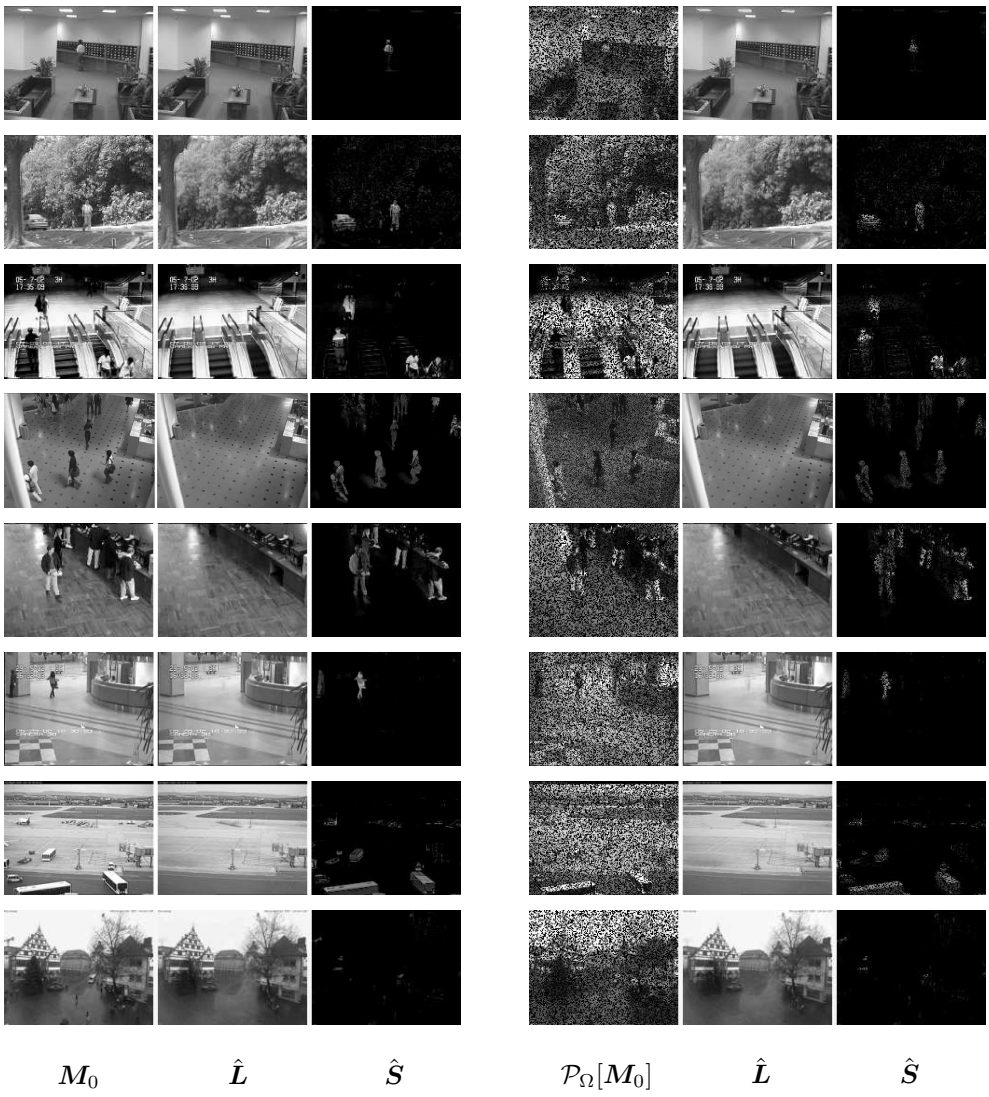


Figure 3.3: Surveillance videos. The videos from top to bottom are respectively Lobby, Campus, Escalator, Mall, Restaurant, Hall, Airport and Square. The left panel presents videos with full observation ($\rho = 1$) and the right one presents videos with partial observation ($\rho = 0.6$). Visually, the low-rank component successfully recovers the background and the sparse one captures the moving objects (e.g. vehicles, pedestrians) in the foreground.

leveraging different methods to deal with different functions may be valuable for other demixing problems.

We are also aware that though our algorithms are extremely efficient in the beginning iterations and quickly arrive at an approximate solution of practical significance, they become less competitive in solutions of very high accuracy, due to the nature of Frank-Wolfe. This suggests further hybridization under our framework (e.g. using nonconvex approaches to handle the nuclear norm) might be utilized in certain applications (see [Lau12] for research in that direction).

Table 3.1: Comparisons of FW-T, ISTA and FISTA on surveillance video data. The advantage of our FW-T method becomes prominent when the data are at large scale and compressed (i.e. the small ρ scenarios).

		FW-T		ISTA		FISTA	
Data	ρ	iter.	time	iter.	time	iter.	time
Lobby	1.0	96	1.94e+02	144	3.64e+02	41	**1.60e+02**
(20480 × 1000)	0.8	104	**2.33e+02**	216	1.03e+03	52	3.55e+02
	0.6	133	**3.12e+02**	380	1.67e+03	74	5.10e+02
Campus	1.0	45	**1.56e+02**	78	1.49e+03	23	4.63e+02
(20480 × 1439)	0.8	44	**1.57e+02**	122	2.34e+03	30	6.45e+02
	0.6	41	**1.39e+02**	218	4.27e+03	43	1.08e+03
Escalator	1.0	81	**7.40e+02**	58	4.19e+03	25	2.18e+03
(20800 × 3417)	0.8	80	**7.35e+02**	90	8.18e+03	32	3.46e+03
	0.6	82	**7.68e+02**	162	1.83e+04	43	5.73e+03
Mall	1.0	38	**4.70e+02**	110	5.03e+03	35	1.73e+03
(81920 × 1286)	0.8	35	**4.58e+02**	171	7.32e+03	44	2.34e+03
	0.6	44	**5.09e+02**	308	1.31e+04	62	3.42e+03
Restaurant	1.0	70	**5.44e+02**	52	3.01e+03	20	1.63e+03
(19200 × 3055)	0.8	74	**5.51e+02**	81	4.84e+03	26	1.82e+03
	0.6	76	**5.73e+02**	144	9.93e+03	38	3.31e+03
Hall	1.0	60	**6.33e+02**	52	2.98e+03	21	1.39e+03
(25344 × 3584)	0.8	62	**6.52e+02**	81	6.45e+03	28	2.90e+03
	0.6	70	**7.43e+02**	144	1.42e+04	39	4.94e+03
Airport	1.0	130	**6.42e+03**	29	2.37e+04	14	1.37e+04
(25344 × 15730)	0.8	136	**6.65e+03**	45	6.92e+04	18	4.27e+04
	0.6	154	**7.72e+03**	77	1.78e+05	24	7.32e+04
Square	1.0	179	**1.24e+04**	29	3.15e+04	13	1.51e+04
(19200 × 28181)	0.8	181	**1.26e+04**	44	1.04e+05	17	6.03e+04
	0.6	191	**1.31e+04**	78	2.63e+05	22	9.88e+05

Table 3.2: Comparisons of FW-T, ISTA and FISTA on YaleB face data. The number of frames, 65, is relatively small for this application. This disables the FW-T method to significantly benefit from its linear per-iteration cost and consequently the FISTA method consistently has a better performance.

		FW-T		ISTA		FISTA	
Data	ρ	iter.	time	iter.	time	iter.	time
YaleB01	1.0	65	34.0	49	21.4	17	**8.69**
(32256×65)	0.9	68	35.6	59	23.9	19	**8.62**
	0.8	79	42.2	76	35.3	22	**10.9**
	0.7	76	39.9	97	44.0	25	**11.1**
	0.6	71	37.5	127	50.2	29	**12.9**
	0.5	80	40.5	182	77.9	35	**15.2**
YaleB02	1.0	64	34.6	51	19.2	18	**7.31**
(32256×65)	0.9	64	26.8	61	22.6	20	**7.32**
	0.8	71	33.9	78	27.7	22	**8.61**
	0.7	71	31.3	99	36.6	26	**11.0**
	0.6	73	36.6	132	53.7	30	**12.4**
	0.5	63	28.0	177	64.6	35	**13.4**
YaleB03	1.0	62	26.0	49	16.6	18	**6.00**
(32256×65)	0.9	71	27.5	62	20.3	20	**6.43**
	0.8	69	30.0	78	26.0	22	**8.32**
	0.7	78	31.5	101	32.9	26	**9.00**
	0.6	73	28.7	132	40.4	30	**10.6**
	0.5	70	28.0	181	60.3	36	**12.8**
YaleB04	1.0	63	28.5	47	16.6	17	**6.35**
(32256×65)	0.9	67	28.7	58	23.1	19	**7.98**
	0.8	68	31.7	72	26.3	23	**9.39**
	0.7	69	30.7	92	35.9	26	**9.84**
	0.6	71	29.4	124	40.0	29	**10.1**
	0.5	74	29.4	174	67.3	36	**14.3**

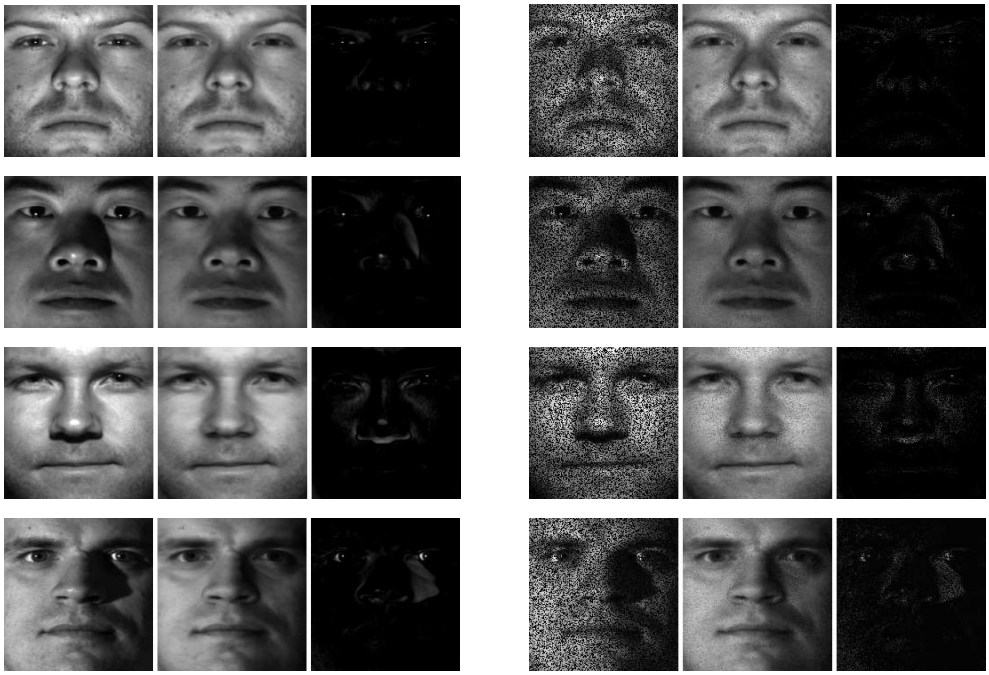


Figure 3.4: Face images. The pictures from top to bottom are respectively YaleB01, YaleB02, YaleB03 and YaleB04 face images. The left panel presents the case with full observation ($\rho = 1$), while the right panel presents the case with partial observation ($\rho = 0.6$). Visually, the recovered low-rank component is smoother and better conditioned for face recognition than the original image, while the sparse component corresponds to shadows and specularities.

Chapter 4

Successive Rank-One Approx. for Nearly Orthogonally Decomposable Symmetric Tensors

4.1 Introduction

The eigenvalue decomposition of symmetric matrices is one of the most important discoveries in mathematics with an abundance of applications across all disciplines of science and engineering. One way to explain such decomposition is to express the symmetric matrix as a minimal sum of rank-one symmetric matrices. It is well known that the eigenvalue decomposition can be simply obtained via *successive rank-one approximation (SROA)*. Specifically, for a symmetric matrix $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ with rank r, one approximates $\boldsymbol{X}$ by a rank-one matrix to minimize the Frobenius norm error:

$$(\lambda_1, \boldsymbol{x}_1) \in \underset{\lambda\in\mathbb{R}, \|\boldsymbol{x}\|=1}{\arg\min} \|\boldsymbol{X} - \lambda \boldsymbol{x}\boldsymbol{x}^\top\|_F ; \tag{4.1.1}$$

then, one approximates the residual $\boldsymbol{X} - \lambda_1 \boldsymbol{x}_1\boldsymbol{x}_1^\top$ by another rank-one matrix $\lambda_2\boldsymbol{x}_2\boldsymbol{x}_2^\top$, and so on. The above procedure continues until one has found r rank-one matrices $\{\lambda_i \boldsymbol{x}_i\boldsymbol{x}_i^\top\}_{i=1}^r$; their summation, $\sum_{i=1}^r \lambda_i \boldsymbol{x}_i\boldsymbol{x}_i^\top$, yields an eigenvalue decomposition of $\boldsymbol{X}$. Moreover, due to the optimal approximation property of the eigen-decomposition, for any positive integer $k \le r$, the best rank-k approximation (in the sense of either the Frobenius norm or the operator norm) to $\boldsymbol{X}$ is simply given by $\sum_{i=1}^k \lambda_i \boldsymbol{x}_i\boldsymbol{x}_i^\top$ [EY36].

In this article, we study decompositions of higher-order symmetric tensors, a natural generalization of symmetric matrices. Many applications in signal processing, machine learning, and statistics, involve higher-order interactions in data; in these cases, higher-order tensors formed from the data are the primary objects of interest. A tensor $\boldsymbol{\mathcal{T}} \in \bigotimes_{i=1}^{p} \mathbb{R}^{n_i} := \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_p}$ of order p is called symmetric if $n_1 = n_2 = \cdots = n_p = n$ and its entries are invariant under any permutation of their indices. Symmetric tensors of order two ($p = 2$) are symmetric matrices. In the sequel, we reserve the term *tensor* (without any further specification) for tensors of order $p \geq 3$. A symmetric rank-one tensor can be naturally defined as a p-fold outer product

$$\boldsymbol{v}^{\otimes p} := \underbrace{\boldsymbol{v} \otimes \boldsymbol{v} \otimes \cdots \otimes \boldsymbol{v}}_{p \text{ times}},$$

where $\boldsymbol{v} \in \mathbb{R}^n$ and $\otimes$ denotes the outer product between vectors.[1] The minimal number of rank-one symmetric tensors whose sum is $\boldsymbol{\mathcal{T}}$ is called the *symmetric tensor rank* in the literature, and any corresponding decomposition is called a *symmetric canonical decomposition* [Har70]. Such decompositions have applications in many scientific and engineering domains [McC87, Com94, SBG04, KB09, CJ10, AGH+14].

By analogy to the matrix case, successive rank-one approximations schemes have been proposed for symmetric tensor decomposition [ZG01, KR02, KBK05, WQ07]. Just as in the matrix case, one first approximates $\boldsymbol{\mathcal{T}}$ by a symmetric rank-one tensor

$$(\lambda_1, \boldsymbol{v}_1) \in \underset{\lambda \in \mathbb{R}, \|\boldsymbol{v}\|=1}{\arg\min} \left\| \boldsymbol{\mathcal{T}} - \lambda \boldsymbol{v}^{\otimes p} \right\|_F, \tag{4.1.2}$$

and then approximate the residual $\boldsymbol{\mathcal{T}} - \lambda_1 \boldsymbol{v}_1^{\otimes p}$ again by another symmetric rank-one tensor, and so on. This process continues until a certain stopping criterion is met. However, different from symmetric matrices, the above procedure for higher order tensors ($p \geq 3$) faces a number of *computational* and *theoretical* challenges.

Unlike problem (4.1.1)—which can be solved efficiently using simple techniques such as power iterations—solving the rank-one approximation to higher order tensors is much more difficult: it is NP-hard, even for symmetric third-order tensors [HL13]. Researchers in numerical linear algebra and numerical optimization have devoted a great amount of effort to solve problem (4.1.2). Broadly speaking, existing methods for problem (4.1.2) can be categorized into three types. First, as problem (4.1.2) is equivalent to finding the extreme value of a homogeneous polynomial over the unit sphere, general-purpose polynomial solvers based on the Sum-of-Squares (SOS) framework [Sho87, Nes00, Par00, Las01, Par03], such as GLOPTIPOLY 3 [HLL09] and SOSTOOLS [PAV+13], can be effectively applied to the rank-one approximation problem. The

[1] For any $1 \leq i_1, i_2, \ldots, i_p \leq n$ and any $\boldsymbol{v} \in \mathbb{R}^n$, the $(i_1, i_2, \ldots, i_p)$-th entry of $\boldsymbol{v}^{\otimes p}$ is $(\boldsymbol{v}^{\otimes p})_{i_1, i_2, \ldots, i_p} = v_{i_1} v_{i_2} \cdots v_{i_p}$.

SOS approach can solve any polynomial problem to any given accuracy through a sequence of semidefinite programs; however, the size of these programs are very large for high-dimensional problems, and hence these techniques are generally limited to relatively small-sized problems. The second approach is to treat problem (4.1.2) as a nonlinear program [Ber99, WN99], and then to exploit and adapt the wealth of ideas from numerical optimization. The resulting methods—which include [DLDMV00, ZG01, KR02, WQ07, KM11, Han13, CHLZ12, ZLQ12, HCD14, GMWZ17] to just name a few—are empirically efficient and scalable, but are only guaranteed to reach a local optimum or stationary point of the objective over the sphere. Therefore, to maximize their performance, these methods need to run with several starting points. The final approach is based on the recent trend of relaxing seemingly intractable optimization problems such as problem (4.1.2) with more tractable convex optimization problems that can be efficiently solved [JMZ14, NW14, YYQ14]. The tensor structure in (4.1.2) has made it possible to design highly-tailored convex relaxations that appear to be very effective. For instance, the semidefinite relaxation approach in [JMZ14] was able to globally solve almost all the randomly generated instances that they tested. Aside from the above solvers, a few algorithms have been specifically designed for the scenario where some side information regarding the solution of (4.1.2) is known. For example, when the signs of the optimizer $\boldsymbol{v}_1$ are revealed, polynomial time approximation schemes for solving (4.1.2) are available [LNQY09].

In contrast to the many active efforts and promising results on the computational side, the theoretical properties of successive rank-one approximations are far less developed. Although SROA is justified for matrix eigenvalue decomposition, it is known to fail for general tensors [SC10]. Indeed, much has been established about the failings of low-rank approximation concepts for tensors that are taken for granted in the matrix setting [Kol01, Kol03, Ste07, Ste08, DSL08]. For instance, the best rank-r approximation to a general tensor is not even guaranteed to exist (though several sufficient conditions for this existence have been recently proposed [LC10, LC14]). Nevertheless, SROA can be still justified for certain classes of symmetric tensors that popularly arise in applications.

Nearly SOD tensors Indeed, in many applications (e.g., higher-order statistical estimation [McC87], independent component analysis [Com94, CJ10], and parameter estimation for latent variable models [AGH+14]), the input tensor $\widehat{\boldsymbol{\mathcal{T}}}$ may be fairly assumed to be a symmetric tensor slightly perturbed from the underlying tensor $\boldsymbol{\mathcal{T}}$, which is *symmetric and orthogonally decomposable (SOD)* [MHG15, WS17, MHG17]. In specific,

$$\widehat{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}},$$

where the underlying SOD tensor can be written as $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{r} \lambda_i \boldsymbol{v}_i^{\otimes p}$ with $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = \delta_{ij}$ for $1 \leq i, j \leq r$, and $\boldsymbol{\mathcal{E}}$ is a perturbation tensor. In these aforementioned applications, we are interested in obtaining the underlying pairs $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^{r}$. When $\boldsymbol{\mathcal{E}}$ vanishes, it is known that $\sum_{i=1}^{r} \lambda_i \boldsymbol{v}_i^{\otimes p}$ is the unique symmetric canonical decomposition [Har70, Kru77], and moreover, successive rank-one approximation exactly recovers $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^{r}$ [ZG01]. However, because of the inevitable perturbation term $\boldsymbol{\mathcal{E}}$ arising from sampling errors, noisy measurements, model misspecification, numerical errors, and so on, it is crucial to understand *the behavior of SROA when* $\boldsymbol{\mathcal{E}} \neq \mathbf{0}$. In particular, one may ask *whether SROA provides an accurate approximation to* $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^{r}$. If the answer is affirmative, then we can indeed take advantage of those sophisticated numerical approaches to solving (4.1.2) mentioned above for many practical problems. This is the gist of this chapter.

Algorithm-independent analysis. The recent work of [AGH+14] proposes a randomized algorithm for approximating SROA based on the power method of [DLDMV00]. There, an error analysis specific to the proposed randomized algorithm (for the case $p = 3$) shows that the decomposition $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^{r}$ of $\boldsymbol{\mathcal{T}}$ can be approximately recovered from $\widehat{\boldsymbol{\mathcal{T}}}$ in polynomial time with high probability—provided that the perturbation $\boldsymbol{\mathcal{E}}$ is sufficiently small (roughly on the order of $1/n$ under a natural measure). Our present aim is to provide a general analysis that is *independent* of the specific approach used to obtain rank-one approximations and it seems to be beneficial. Our analysis shows that the general SROA scheme in fact allows for perturbations to be of the order $1/\sqrt[p-1]{n}$, suggesting advantages of using more sophisticated optimization procedures and potentially more computational resources to solve each rank-one approximation step.

As motivation, we describe a simple and typical example of latent variable models where perturbations of SOD tensors naturally arise.

A motivating example. To illustrate why we are particularly interested in nearly SOD tensors, we now consider the following simple probabilistic model for characterizing the topics of text documents. (We follow the description from [AGH+14].) Let n be the number of distinct topics in the corpus, d be the number of distinct words in the vocabulary, and $t \geq p$ be the number of words in each document. We identify the sets of distinct topics and words, respectively, by $[n]$ and $[d]$. The topic model posits the following generative process for a document. The document's topic $h \in [n]$ is first randomly drawn according to a discrete probability distribution specified by $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ (where we assume $w_i > 0$ for each $i \in [n]$ and $\sum_{i \in [n]} w_i = 1$):

$$\mathbb{P}[h = i] = w_i \quad \text{for all } i \in [n].$$

Given the topic h, each of the document's t words is then drawn independently from the vocabulary according to the discrete distribution specified by the probability vector $\boldsymbol{\mu}_h \in \mathbb{R}^d$; we assume that the probability vectors $\{\boldsymbol{\mu}_i\}_{i\in[n]}$ are linearly independent (and, in particular, $d \geq n$). The task here is to estimate these probability vectors $\boldsymbol{w}$ and $\{\boldsymbol{\mu}_i\}_{i\in[n]}$ based on a corpus of documents.

Denote by $\boldsymbol{M}_2 \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\mathcal{M}}_p \in \bigotimes^p \mathbb{R}^d$, respectively, the pairs probability matrix and p-tuples probability tensor, defined as follows: for all $i_1, i_2, \ldots, i_p \in [d]$, ,

$$(\boldsymbol{M}_2)_{i_1,i_2} = \mathbb{P}\left[\text{1st word} = i_1,\ \text{2nd word} = i_2\right]$$
$$(\boldsymbol{\mathcal{M}}_p)_{i_1,i_2,\ldots,i_p} = \mathbb{P}\left[\text{1st word} = i_1,\ \text{2nd word} = i_2, \cdots,\ p\text{th word} = i_p\right].$$

It can be shown that $\boldsymbol{M}_2$ and $\boldsymbol{\mathcal{M}}_p$ can be precisely represented using $\boldsymbol{w}$ and $\{\boldsymbol{\mu}_i\}_{i\in[n]}$:

$$\boldsymbol{M}_2 = \sum_{i\in[n]} w_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top \quad \text{and} \quad \boldsymbol{\mathcal{M}}_p = \sum_{i\in[n]} w_i \boldsymbol{\mu}_i^{\otimes p}.$$

Since $\boldsymbol{M}_2$ is positive semidefinite and $\operatorname{rank}(\boldsymbol{M}_2) = n$, $\boldsymbol{M}_2 = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ is its reduced eigenvalue decomposition. Here, $\boldsymbol{U} \in \mathbb{R}^{d\times n}$ satisfies $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_n$, and $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\operatorname{diag} \boldsymbol{D} \succ \boldsymbol{0}$. Now define

$$\boldsymbol{W} := \boldsymbol{U}\boldsymbol{D}^{-1/2}, \quad \lambda_i := w_i^{1-p/2}, \quad \text{and} \quad \boldsymbol{v}_i := \sqrt{w_i}\boldsymbol{W}^\top\boldsymbol{\mu}_i \in \mathbb{R}^n \quad \text{for each } i \in [n].$$

Then

$$\boldsymbol{W}^\top\boldsymbol{M}_2\boldsymbol{W} = \boldsymbol{I} = \boldsymbol{M}_2(\boldsymbol{W},\boldsymbol{W}) = \sum_{i\in[n]} w_i(\boldsymbol{W}^\top\boldsymbol{\mu}_i)(\boldsymbol{W}^\top\boldsymbol{\mu}_i)^\top = \sum_{i\in[n]} \boldsymbol{v}_i\boldsymbol{v}_i^\top,$$

which implies that $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ are orthogonal. Moreover,

$$\boldsymbol{\mathcal{T}} := \boldsymbol{\mathcal{M}}_p(\boldsymbol{W},\boldsymbol{W},\ldots,\boldsymbol{W}) = \sum_{i\in[n]} w_i\,(\boldsymbol{W}^\top\boldsymbol{\mu}_i)^{\otimes p} = \sum_{i\in[n]} \lambda_i \boldsymbol{v}_i^{\otimes p}. \tag{4.1.3}$$

Therefore, we can obtain $\{(\lambda_i, \boldsymbol{v}_i)\}_{i\in[n]}$ (and subsequently $\{(w_i, \boldsymbol{\mu}_i)\}_{i\in[n]}$ [2]) by computing the (unique) symmetric canonical decomposition of tensor $\boldsymbol{\mathcal{T}}$, which can be perfectly achieved by SROA [ZG01].

In order to obtain the tensor $\boldsymbol{\mathcal{T}}$, we need $\boldsymbol{M}_2$ and $\boldsymbol{\mathcal{M}}_p$, both of which can be estimated from a collection of documents.[3] However, the quantities $\boldsymbol{M}_2$ and $\boldsymbol{\mathcal{M}}_p$ are only known up to sampling errors, and hence, we are only able to construct a symmetric tensor $\widehat{\boldsymbol{\mathcal{T}}}$ that is, at best, only close to the one in (4.1.3). A critical question is whether we can still use SROA (Algorithm 11) to obtain an approximate decomposition and

[2] After obtaining $\{(\lambda_i, \boldsymbol{v}_i)\}_{i\in[n]}$, it is possible to obtain $\{(w_i, \boldsymbol{\mu}_i)\}_{i\in[n]}$ because for each $i \in [n]$, there exists $j \in [n]$ such that $w_i = \lambda_j^{2/(2-p)}$ and $\boldsymbol{\mu}_i = \lambda_j(\boldsymbol{W}^\top)^\dagger\boldsymbol{v}_j$, where $(\boldsymbol{W}^\top)^\dagger$ denotes the Moore-Penrose pseudoinverse of $\boldsymbol{W}^\top$.

[3] Due to their independence, all pairs (resp., p-tuples) of words in a document can be used in forming estimates of $\boldsymbol{M}_2$ (resp., $\boldsymbol{\mathcal{M}}_p$).

robustly estimate the model parameters.

Algorithm 11 Successive Rank-One Approximation (SROA)

input symmetric tensor $\widehat{\boldsymbol{\mathcal{T}}} \in \bigotimes^p \mathbb{R}^n$.
1: **initialize** $\widehat{\boldsymbol{\mathcal{T}}}_0 := \widehat{\boldsymbol{\mathcal{T}}}$
2: **for** $i = 1$ to n **do**
3: $\quad (\hat{\lambda}_i, \hat{\boldsymbol{v}}_i) \in \arg\min_{\lambda \in \mathbb{R}, \|\boldsymbol{v}\|=1} \left\| \widehat{\boldsymbol{\mathcal{T}}}_{i-1} - \lambda \boldsymbol{v}^{\otimes p} \right\|_F$.
4: $\quad \widehat{\boldsymbol{\mathcal{T}}}_i := \widehat{\boldsymbol{\mathcal{T}}}_{i-1} - \hat{\lambda}_i \hat{\boldsymbol{v}}_i^{\otimes p}$.
5: **end for**
6: **return** $\{(\hat{\lambda}_i, \hat{\boldsymbol{v}}_i)\}_{i \in [n]}$.

Setting Following the notation in the above example, in the sequel, we denote $\widehat{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \bigotimes^p \mathbb{R}^n$. Here $\boldsymbol{\mathcal{T}}$ is a symmetric tensor that is orthogonally decomposable, i.e., $\boldsymbol{\mathcal{T}} = \sum_{i=1}^n \lambda_i \boldsymbol{v}_i^{\otimes p}$ with all $\lambda_i \neq 0$, $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ forming an orthonormal basis of $\mathbb{R}^n$, and $\boldsymbol{\mathcal{E}}$ is a symmetric perturbation tensor with operator norm $\varepsilon := \|\boldsymbol{\mathcal{E}}\|$. Note that in some applications, we might instead have $\boldsymbol{\mathcal{T}} = \sum_{i=1}^r \lambda_i \boldsymbol{v}_i^{\otimes p}$ for some $r < n$. Our results nevertheless can be applied in that setting as well with little modification.

For simplicity, we also assume p is odd and treat it as a constant in big-O notations. (We discuss the even case in Section 4.3.4). Without loss of generality, we can assume $\lambda_i > 0$ for all $i \in [n]$, as we can always change the sign of $\boldsymbol{v}_i$ to make it hold. Moreover, line 3 in Algorithm 11 simply becomes

$$\hat{\boldsymbol{v}}_i \in \arg\max_{\|\boldsymbol{v}\|=1} \widehat{\boldsymbol{\mathcal{T}}}_{i-1} \boldsymbol{v}^{\otimes p}, \quad \hat{\lambda}_i = \widehat{\boldsymbol{\mathcal{T}}}_{i-1} \hat{\boldsymbol{v}}_i^{\otimes p}.$$

Organization Section 4.2 analyzes the first iteration of Algorithm 11 and proves that $(\hat{\lambda}_1, \hat{\boldsymbol{v}}_1)$ is a robust estimate of a pair $(\lambda_i, \boldsymbol{v}_i)$ for some $i \in [n]$. A full decomposition analysis is provided in Section 4.3, in which we establish the following property of tensors: when $\|\boldsymbol{\mathcal{E}}\|$ is small enough, the approximation errors do not accumulate as the iteration number grows; in contrast, the use of deflation is generally not advised in the matrix setting for finding more than a handful of matrix eigenvectors due to potential instability. Numerical experiments are also reported to confirm our theoretical results.

4.2 Rank-One Approximation

In this section, we provide an analysis of the first step of SROA (Algorithm 11), which yield a rank-one approximation to $\widehat{\boldsymbol{\mathcal{T}}}$.

4.2.1 Review of matrix perturbation analysis

We first state a well-known result about perturbations of the eigenvalue decomposition for symmetric matrices; this result serves as a point of comparison for our study of higher-order tensors. The result is stated just for rank-one approximations, and in a form analogous to what we are able to show for the tensor case (Theorem 4.2 below).

Theorem 4.1 ([Wey12, DK70]) *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalue decomposition $\sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^\top$, where $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ are orthonormal. Let $\widehat{\boldsymbol{M}} = \boldsymbol{M} + \boldsymbol{E} \in \mathbb{R}^{n\times n}$ for some symmetric matrix $\boldsymbol{E}$ with $\varepsilon := \|\boldsymbol{E}\|$, and let*

$$(\hat{\lambda}, \hat{\boldsymbol{x}}) \in \operatorname*{arg\,min}_{\lambda\in\mathbb{R}, \|\boldsymbol{x}\|=1} \left\| \widehat{\boldsymbol{M}} - \lambda \boldsymbol{x}\boldsymbol{x}^\top \right\|_F .$$

The following holds.

- *(Perturbation of leading eigenvalue.) $|\hat{\lambda} - \lambda_1| \leq \varepsilon$.*
- *(Perturbation of leading eigenvector.) Suppose $\gamma := \min_{i\neq 1} |\lambda_1 - \lambda_i| > 0$. Then $\langle \hat{\boldsymbol{x}}, \boldsymbol{v}_1 \rangle^2 \geq 1 - (2\varepsilon/\gamma)^2$. This implies that if $2\varepsilon/\gamma \leq 1$, then $\min\{\|\boldsymbol{v}_1 - \hat{\boldsymbol{x}}\|, \|\boldsymbol{v}_1 + \hat{\boldsymbol{x}}\|\} \leq O(\varepsilon/\gamma)$.*

For completeness, we give a proof of the eigenvector perturbation bound below since it is not directly implied by results in [DK70] but essentially uses the same argument.

Proof Since $\widehat{\boldsymbol{M}}$ is symmetric, it has an eigenvalue decomposition $\sum_{i=1}^{n} \hat{\lambda}_i \hat{\boldsymbol{v}}_i \hat{\boldsymbol{v}}_i^\top$, where $|\hat{\lambda}_1| \geq |\hat{\lambda}_2| \geq \cdots \geq |\hat{\lambda}_n|$ and $\{\hat{\boldsymbol{v}}_1, \hat{\boldsymbol{v}}_2, \ldots, \hat{\boldsymbol{v}}_n\}$ are orthonormal. It is straightforward to obtain:

$$\hat{\lambda} = \hat{\lambda}_1 \quad \text{and} \quad \widehat{\boldsymbol{M}}\hat{\boldsymbol{x}} = \hat{\lambda}\hat{\boldsymbol{x}}.$$

By Weyl's inequality [Wey12],

$$|\hat{\lambda} - \lambda_1| \leq \|\boldsymbol{E}\| = \varepsilon.$$

To bound $\langle \hat{\boldsymbol{x}}, \boldsymbol{v}_1 \rangle^2$, we employ an argument very similar to one from [DK70]. Observe that

$$\|\boldsymbol{M}\hat{\boldsymbol{x}} - \lambda_1 \hat{\boldsymbol{x}}\|^2 = \left\|(\hat{\lambda} - \lambda_1)\hat{\boldsymbol{x}} - \boldsymbol{E}\hat{\boldsymbol{x}}\right\|^2 \leq (|\hat{\lambda} - \lambda_1| \, \|\hat{\boldsymbol{x}}\| + \|\boldsymbol{E}\hat{\boldsymbol{x}}\|)^2 \leq 4\varepsilon^2.$$

Moreover,

$$\boldsymbol{M}\hat{\boldsymbol{x}} - \lambda_1 \hat{\boldsymbol{x}} = \sum_{i=1}^{n} (\lambda_i - \lambda_1) \langle \boldsymbol{v}_i, \hat{\boldsymbol{x}} \rangle \boldsymbol{v}_i = \sum_{i=2}^{n} (\lambda_i - \lambda_1) \langle \boldsymbol{v}_i, \hat{\boldsymbol{x}} \rangle \boldsymbol{v}_i,$$

and therefore

$$\|\boldsymbol{M}\hat{\boldsymbol{x}} - \lambda_1\hat{\boldsymbol{x}}\|^2 = \sum_{i=2}^{n}(\lambda_i - \lambda_1)^2\langle \boldsymbol{v}_i, \hat{\boldsymbol{x}}\rangle^2 \geq \gamma^2\sum_{i=2}^{n}\langle \boldsymbol{v}_i, \hat{\boldsymbol{x}}\rangle^2 = \gamma^2(1 - \langle \boldsymbol{v}_1, \hat{\boldsymbol{x}}\rangle^2).$$

Combining the upper and lower bounds on $\|\boldsymbol{M}\hat{\boldsymbol{x}} - \lambda_1\hat{\boldsymbol{x}}\|^2$ gives $\langle \hat{\boldsymbol{x}}, \boldsymbol{v}_1\rangle^2 \geq 1 - (2\varepsilon/\gamma)^2$ as claimed. ■

4.2.2 Single rank-one approximation

The main result of this section concerns the first step of SROA (Algorithm 11) and establishes a perturbation result for nearly SOD tensors.

Theorem 4.2 *For any odd positive integer $p \geq 3$, let $\widehat{\boldsymbol{\mathcal{T}}} := \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \bigotimes^p \mathbb{R}^n$, where $\boldsymbol{\mathcal{T}}$ is a symmetric tensor with orthogonal decomposition $\boldsymbol{\mathcal{T}} = \sum_{i=1}^n \lambda_i \boldsymbol{v}_i^{\otimes p}$, $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ is an orthonormal basis of $\mathbb{R}^n$, $\lambda_i > 0$ for all $i \in [n]$, and $\boldsymbol{\mathcal{E}}$ is a symmetric tensor with operator norm $\varepsilon := \|\boldsymbol{\mathcal{E}}\|$. Let $\hat{\boldsymbol{x}} \in \arg\max_{\|\boldsymbol{x}\|_2=1} \widehat{\boldsymbol{\mathcal{T}}}\boldsymbol{x}^{\otimes p}$ and $\hat{\lambda} := \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{x}}^{\otimes p}$. Then there exists $j \in [n]$ such that*

$$|\hat{\lambda} - \lambda_j| \leq \varepsilon, \quad \|\hat{\boldsymbol{x}} - \boldsymbol{v}_j\|_2 \leq 10\left(\frac{\varepsilon}{\lambda_j} + \left(\frac{\varepsilon}{\lambda_j}\right)^2\right). \tag{4.2.1}$$

To prove Theorem 4.2, we first establish an intermediate result. Let $x_i := \langle \boldsymbol{v}_i, \hat{\boldsymbol{x}}\rangle$, so $\hat{\boldsymbol{x}} = \sum_{i=1}^n x_i\boldsymbol{v}_i$ and $\sum_{i=1}^n x_i^2 = 1$ since $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ is an orthonormal basis for $\mathbb{R}^n$ and $\|\hat{\boldsymbol{x}}\| = 1$. We reorder the indices $[n]$ so that

$$\lambda_1|x_1|^{p-2} \geq \lambda_2|x_2|^{p-2} \geq \cdots \geq \lambda_n|x_n|^{p-2}. \tag{4.2.2}$$

Our intermediate result, derived by simply bounding $\hat{\lambda}$ from both above and below, is as follows.

Lemma 4.3 *In the notation from above,*

$$\lambda_1 \geq \lambda_{\max} - 2\varepsilon, \quad |x_1| \geq 1 - 2\varepsilon/\lambda_1, \quad x_1^2 \geq x_1^{p-1} \geq 1 - 4\varepsilon/\lambda_1, \quad \textit{and} \quad |\hat{\lambda} - \lambda_1| \leq \varepsilon. \tag{4.2.3}$$

where $\lambda_{\max} = \max_{i\in[n]} \lambda_i$.

Proof To show (4.2.3), we will bound $\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{x}}^{\otimes p}$ from both above and below.

For the upper bound, we have

$$\begin{aligned}\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{x}}^{\otimes p} &= \boldsymbol{\mathcal{T}}\hat{\boldsymbol{x}}^{\otimes p} + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p} \\ &= \sum_{i=1}^{n}\lambda_i x_i^p + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p}\end{aligned}$$

$$\leq \sum_{i=1}^{n} \lambda_i |x_i|^{p-2} x_i^2 + \varepsilon \leq \lambda_1 |x_1|^{p-2} + \varepsilon, \tag{4.2.4}$$

where the last inequality follows since $\sum_{i=1}^{n} x_i^2 = 1$.

On the other hand,

$$\hat{\lambda} \geq \max_{i \in [n]} \boldsymbol{\mathcal{T}} \boldsymbol{v}_i^{\otimes p} - \|\boldsymbol{\mathcal{E}}\| = \lambda_{\max} - \varepsilon \geq \lambda_1 - \varepsilon. \tag{4.2.5}$$

Combining (4.2.4) and (4.2.5), it can be easily verified that

$$\lambda_1 \geq \lambda_{\max} - 2\varepsilon, \quad |\lambda_1 - \hat{\lambda}| \leq \varepsilon.$$

and moreover,

$$|x_1| \geq |x_1|^{p-2} \geq \frac{\lambda_1 - 2\varepsilon}{\lambda_1} = 1 - \frac{2\varepsilon}{\lambda_1}$$

which implies that $x_1^{p-1} = |x_1|^{p-2} \cdot |x_1| \geq 1 - 4\varepsilon/\lambda_1$. ■

> **Remark 4.4** *The higher-order requirement, $p \geq 3$, is crucial in the analysis. Specifically we can bound $|x_1|$ below by the lower bound of $|x_1|^{p-2}$, which can be done by bounding $\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}} \hat{\boldsymbol{x}}^{\otimes p}$ from both above and below. This essentially explains why Lemma 4.3, different from the matrix case ($p = 2$), does not rely on the spectral gap condition.*

The bound $|\hat{\lambda} - \lambda_1| \leq \varepsilon$ proved in Lemma 4.3 is comparable to the matrix counterpart in Theorem 4.1, and is optimal in the worst case. Consider $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{n} \lambda_i \boldsymbol{e}_i^{\otimes p}$ with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ and $\boldsymbol{\mathcal{E}} = \varepsilon \boldsymbol{e}_1^{\otimes p}$, for some $\varepsilon > 0$. Then clearly $\hat{\lambda} = \lambda_1 + \varepsilon$ and $|\hat{\lambda} - \lambda_1| = \varepsilon$.

Moreover, when $\boldsymbol{\mathcal{E}}$ vanishes, Lemma 4.3 leads directly to the following result given in [ZG01].

> **Corollary 4.5 ([ZG01])** *Suppose $\boldsymbol{\mathcal{E}} = \boldsymbol{0}$ (i.e., $\widehat{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} = \sum_{i=1}^{n} \lambda_i \boldsymbol{v}_i^{\otimes p}$ is orthogonally decomposable). Then Algorithm 11 computes $\{(\lambda_i, \boldsymbol{v}_i)\}_{i=1}^{n}$ exactly.*

However, compared to Theorem 4.1, the bound $|x_1| \geq 1 - 2\varepsilon/\lambda_1$ appears to be suboptimal; this is because the bound only implies $\|\hat{\boldsymbol{x}} - \boldsymbol{v}_1\| = O(\sqrt{\varepsilon/\lambda_1})$. In the following, we will proceed to improve this result to $\|\hat{\boldsymbol{x}} - \boldsymbol{v}_1\| = O(\varepsilon/\lambda_1)$ by using the first-order optimality condition [WN99]. See [Lim05] for a discussion in the present context.

Consider the Lagrangian function corresponding to the optimization problem $\max_{\|\boldsymbol{x}\|=1} \widehat{\boldsymbol{\mathcal{T}}} \boldsymbol{x}^{\otimes p}$,

$$\mathcal{L}(\boldsymbol{x}, \lambda) := \frac{1}{p} \widehat{\boldsymbol{\mathcal{T}}} \boldsymbol{x}^{\otimes p} - \frac{\lambda}{2} \left(\|\boldsymbol{x}\|^2 - 1 \right), \tag{4.2.6}$$

where $\lambda \in \mathbb{R}$ corresponds to the (scaled) Lagrange multiplier for the equality constraint. As $\hat{\boldsymbol{x}} \in \arg\max_{\|\boldsymbol{x}\|_2=1} \widehat{\boldsymbol{\mathcal{T}}}\boldsymbol{x}^{\otimes p}$ (and the linear independent constraint qualification [WN99] can be easily verified), there exists $\bar{\lambda} \in \mathbb{R}$ such that

$$\nabla \boldsymbol{\mathcal{L}}(\hat{\boldsymbol{x}}, \bar{\lambda}) = \widehat{\boldsymbol{\mathcal{T}}}\boldsymbol{x}^{\otimes p-1} - \bar{\lambda}\boldsymbol{x} = 0. \tag{4.2.7}$$

Moreover, as $\|\hat{\boldsymbol{x}}\| = 1$, $\bar{\lambda} = \bar{\lambda}\langle \boldsymbol{x}, \boldsymbol{x}\rangle = \widehat{\boldsymbol{\mathcal{T}}}\boldsymbol{x}^{\otimes p} = \hat{\lambda}$. Thus we have $\hat{\lambda}\hat{\boldsymbol{x}} = \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{x}}^{\otimes p-1}$.

We are now ready to prove Theorem 4.2. **Proof** [Proof of Theorem 4.2] The first inequality in (4.2.1) has been proved in Lemma 4.3, so we are left to prove the second one.

From the first-order optimality condition above, we have

$$\hat{\lambda}\hat{\boldsymbol{x}} = \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{x}}^{\otimes p-1} = \boldsymbol{\mathcal{T}}\hat{\boldsymbol{x}}^{\otimes p-1} + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1} = \lambda_1 x_1^{p-1}\boldsymbol{v}_1 + \sum_{i\geq 2}\lambda_i x_i^{p-1}\boldsymbol{v}_i + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1}.$$

Thus,

$$\begin{aligned}
\|\lambda_1(\hat{\boldsymbol{x}} - \boldsymbol{v}_1)\|_2 &= \left\|(\lambda_1 - \hat{\lambda})\hat{\boldsymbol{x}} + (\hat{\lambda}\hat{\boldsymbol{x}} - \lambda_1\boldsymbol{v}_1)\right\|_2 \\
&= \left\|(\lambda_1 - \hat{\lambda})\hat{\boldsymbol{x}} + \lambda_1(x_1^{p-1} - 1)\boldsymbol{v}_1 + \sum_{i\geq 2}\lambda_i x_i^{p-1}\boldsymbol{v}_i + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2 \\
&\leq |\lambda_1 - \hat{\lambda}| + \lambda_1|x_1^{p-1} - 1| + \left\|\sum_{i\geq 2}\lambda_i x_i^{p-1}\boldsymbol{v}_i\right\|_2 + \left\|\boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2
\end{aligned} \tag{4.2.8}$$

by the triangle inequality. By Lemma 4.3, we have

$$|\lambda_1 - \hat{\lambda}| \leq \varepsilon, \quad |x_1^{p-1} - 1| \leq 4\varepsilon/\lambda_1, \quad \text{and} \quad \left\|\boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2 \leq \varepsilon. \tag{4.2.9}$$

Moreover,

$$\begin{aligned}
\left\|\sum_{i\geq 2}\lambda_i x_i^{p-1}\boldsymbol{v}_i\right\|_2 = \left(\sum_{i\geq 2}\lambda_i^2 x_i^{2p-2}\right)^{1/2} &\leq \lambda_2|x_2|^{p-2}\sqrt{1 - x_1^2} \\
&\leq \lambda_2(1 - x_1^2) \leq \frac{4\varepsilon\lambda_2}{\lambda_1} \leq 4\varepsilon(1 + 2\varepsilon/\lambda_1),
\end{aligned} \tag{4.2.10}$$

where we have used Lemma 4.3 and the fact that $\lambda_2/\lambda_1 \leq \lambda_{\max}/\lambda_1 \leq 1 + 2\varepsilon/\lambda_1$. Substituting (4.2.9) and (4.2.10) back into (4.2.8), we can easily obtain

$$\|\hat{\boldsymbol{x}} - \boldsymbol{v}_1\|_2 \leq 10\left(\frac{\varepsilon}{\lambda_1} + \left(\frac{\varepsilon}{\lambda_1}\right)^2\right). \tag{4.2.11}$$

■

Remark 4.6 *When $p > 3$, we can slightly sharpen (4.2.11) to*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{v}_1\|_2 \leq 8\frac{\varepsilon}{\lambda_1} + 4\left(\frac{\varepsilon}{\lambda_1}\right)^2,$$

by replacing (4.2.10) with

$$\lambda_2(1 - x_1^2) \leq \lambda_2(1 - |x_1|^{p-2}) \leq \lambda_2 \cdot \frac{2\varepsilon}{\lambda_1} \leq 2\varepsilon(1 + 2\varepsilon/\lambda_1).$$

Theorem 4.2 indicates that the first step of SROA for a nearly SOD tensor approximately recovers $(\lambda_j, \boldsymbol{v}_j)$ for some $j \in [n]$. In particular, whenever ε is small enough relative to λ_1 (e.g., $\varepsilon \leq \lambda_1/2$), there always exists $j \in [n]$ such that $|\hat{\lambda} - \lambda_j| \leq \varepsilon$ and $\|\hat{\boldsymbol{x}} - \boldsymbol{v}_j\|_2 \leq 10 \cdot (1 + 1/2)\varepsilon/\lambda_j = 15\varepsilon/\lambda_j$. This is analogous to Theorem 4.1, except that *the spectral gap condition* required in Theorem 4.1 is *not necessary at all* for the perturbation bounds of SOD tensors.

4.2.3 Numerical verifications for Theorem 4.2

We generate nearly symmetric orthogonally decomposable tensors $\widehat{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \mathbb{R}^{10\times 10\times 10}$ in the following manner. We let the underlying symmetric orthogonally decomposable tensor $\boldsymbol{\mathcal{T}}$ be the diagonal tensor with all diagonal entries equal to one, i.e., $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{10} \boldsymbol{e}_i^{\otimes 3}$ (where $\boldsymbol{e}_i$ is the i-th coordinate basis vector). The perturbation tensor $\boldsymbol{\mathcal{E}}$ is generated under the following three random models before symmetrization:

Binary: independent entries $\mathcal{E}_{i,j,k} \in \{\pm\sigma\}$ uniformly at random;

Uniform: independent entries $\mathcal{E}_{i,j,k} \in [-2\sigma, 2\sigma]$ uniformly at random;

Gaussian: independent entries $\mathcal{E}_{i,j,k} \sim \mathcal{N}(0, \sigma^2)$;

where σ is varied from 0.0001 to 0.2 with increment 0.0001, and one instance is generated for each value of σ.

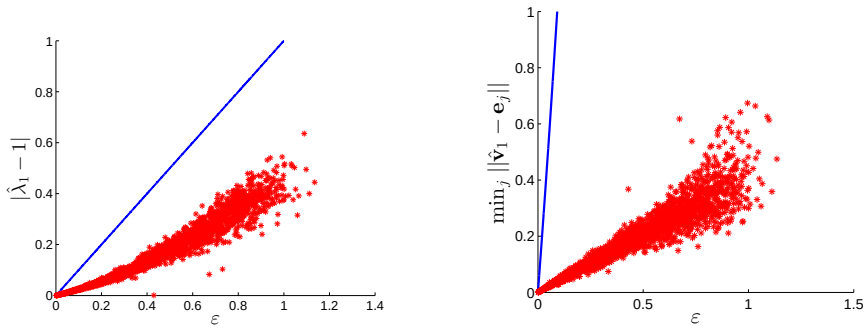
For every randomly generated instance, we solve the polynomial optimization problems

$$\|\boldsymbol{\mathcal{E}}\| = \max_{\|\boldsymbol{x}\|=1} \boldsymbol{\mathcal{E}}\boldsymbol{x}^{\otimes 3} \quad \text{and} \quad \hat{\boldsymbol{v}}_1 \in \arg\max_{\|\boldsymbol{x}\|=1} \widehat{\boldsymbol{\mathcal{T}}}\boldsymbol{x}^{\otimes 3} \tag{4.2.12}$$

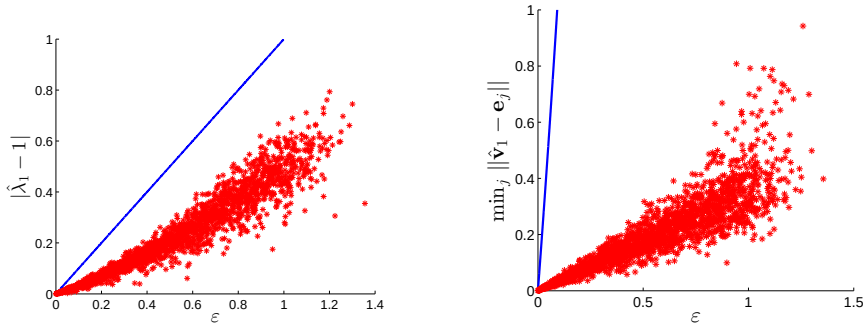
using the general polynomial solver GloptiPoly 3 [HLL09], and set $\hat{\lambda}_1 := \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{v}}_1^{\otimes 3}$. [4]

In Figure 4.1, we plot the approximation error $|\hat{\lambda}_1 - 1|$ and $\min_{j\in[10]} \|\hat{\boldsymbol{v}}_1 - \boldsymbol{e}_j\|$, respectively against the value of norm $\|\boldsymbol{\mathcal{E}}\|$. Each (red) point corresponds to one randomly generated instance, and the (blue) lines are the upper bounds given in Theorem 4.2. We observe no instance violating the theoretical bounds.

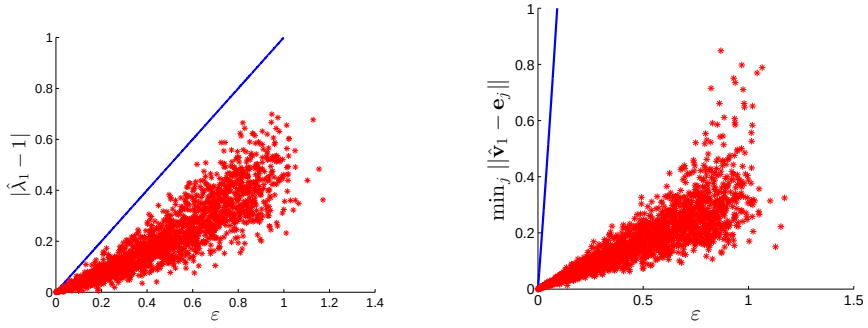
[4] All codes used in this chapter are available on CM's personal website https://sites.google.com/site/mucun1988/.



Perturbation: binary



Perturbation: uniform



Perturbation: Gaussian

Figure 4.1: Approximation Errors of the First Iteration. The approximation errors in $\hat{\lambda}_1$ (resp., $\hat{\boldsymbol{v}}_1$) are plotted on the left (resp., right) as a function of the size of the perturbation ε. Each (red) point corresponds to one randomly generated instance, and the (blue) solid line is the upper bound from Theorem 4.2.

4.3 Full Decomposition Analysis

In the second iteration of Algorithm 11, we have

$$\hat{\boldsymbol{v}}_2 \in \arg \max_{\|\boldsymbol{x}\|_2=1} \widehat{\boldsymbol{\mathcal{T}}}_1 \boldsymbol{x}^{\otimes p}, \quad \hat{\lambda}_2 = \widehat{\boldsymbol{\mathcal{T}}}_1 \hat{\boldsymbol{v}}_2^{\otimes p},$$

where, for some $j \in [n]$,

$$\widehat{\boldsymbol{\mathcal{T}}}_1 = \hat{\boldsymbol{\mathcal{T}}} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1{}^{\otimes p} = \sum_{i \neq j} \lambda_i \boldsymbol{v}_i^{\otimes p} + \widehat{\boldsymbol{\mathcal{E}}} \quad \text{and} \quad \widehat{\boldsymbol{\mathcal{E}}} = \boldsymbol{\mathcal{E}} + (\lambda_j \boldsymbol{v}_j^{\otimes p} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1{}^{\otimes p}).$$

Theorem 4.2 can be directly applied again by bounding the error norm $\|\widehat{\boldsymbol{\mathcal{E}}}\|$. However, since

$$\begin{aligned}
\|\widehat{\boldsymbol{\mathcal{E}}}\| &= \left\| \boldsymbol{\mathcal{E}} + (\lambda_j \boldsymbol{v}_j^{\otimes p} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1{}^{\otimes p}) \right\| \\
&= \left\| \boldsymbol{\mathcal{E}} + (\lambda_j - \hat{\lambda}_1) \boldsymbol{v}_j^{\otimes p} + \hat{\lambda}_1 (\boldsymbol{v}_j^{\otimes p} - \hat{\boldsymbol{v}}_1^{\otimes p}) \right\| \\
&\leq \|\boldsymbol{\mathcal{E}}\| + |\lambda_j - \hat{\lambda}_1| + \hat{\lambda}_1 \left\| \boldsymbol{v}_j^{\otimes p} - \hat{\boldsymbol{v}}_1^{\otimes p} \right\| \\
&\leq (2 + 10\sqrt{p})\varepsilon + O(\varepsilon^2/\lambda_j),
\end{aligned}$$

it appears that the approximation error may increase dramatically with the iteration number.

Fortunately, a more careful analysis shows that approximation error does not in fact accumulate in this way. The high-level reason is that while the operator norm $\|\lambda_j \boldsymbol{v}_j^{\otimes p} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1^{\otimes p}\|$ is of order ε, the relevant quantity is essentially $\left(\lambda_j \boldsymbol{v}_j^{\otimes p} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1^{\otimes p}\right)$ operating on the direction of $\hat{\boldsymbol{v}}_2$, i.e. $|(\lambda_j \boldsymbol{v}_j^{\otimes p} - \hat{\lambda}_1 \hat{\boldsymbol{v}}_1^{\otimes p}) \hat{\boldsymbol{v}}_2^{\otimes p}|$, which only gives rise to a quantity of order ε^2 because $p \geq 3$. This enables us to keep the approximation errors under control.

The main result of this section is as follows.

Theorem 4.7 *Pick any odd positive integer $p \geq 3$. There exists a positive constant $c_0 = c_0(p) > 0$ such that the following holds. Let $\widehat{\boldsymbol{\mathcal{T}}} := \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \bigotimes^p \mathbb{R}^n$, where $\boldsymbol{\mathcal{T}}$ is a symmetric tensor with orthogonal decomposition $\boldsymbol{\mathcal{T}} = \sum_{i=1}^n \lambda_i \boldsymbol{v}_i^{\otimes p}$, $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ is an orthonormal basis of $\mathbb{R}^n$, $\lambda_i > 0$ for all $i \in [n]$, and $\boldsymbol{\mathcal{E}}$ is a symmetric tensor with operator norm $\varepsilon := \|\boldsymbol{\mathcal{E}}\|$. Assume $\varepsilon \leq c_0 \lambda_{\min}/n^{1/(p-1)}$, where $\lambda_{\min} := \min_{i \in [n]} \lambda_i$. Let $\{(\hat{\lambda}_i, \hat{\boldsymbol{v}}_i)\}_{i \in [n]}$ be the output of Algorithm 11 for input $\widehat{\boldsymbol{\mathcal{T}}}$ (where we choose $\hat{\lambda}_i$ to be positive whenever possible). Then there exists a permutation π on $[n]$ such that*

$$|\lambda_{\pi(j)} - \hat{\lambda}_j| \leq 2\varepsilon, \qquad \left\| \boldsymbol{v}_{\pi(j)} - \hat{\boldsymbol{v}}_j \right\| \leq 20\varepsilon/\lambda_{\pi(j)}, \quad \forall j \in [n].$$

4.3.1 Deflation analysis

The proof of Theorem 4.7 is based on the following lemma, which provides a careful analysis of the errors introduced in $\boldsymbol{\mathcal{T}}_i$ from steps $1, 2, \ldots, i$ in Algorithm 11. This lemma is a generalization of a result from [AGH+14] (which only dealt with the $p = 3$ case) and also more transparently reveals the sources of errors that result from deflation.

Lemma 4.8 *Fix a subset $S \subseteq [n]$ and assume that $0 \le \hat{\varepsilon} \le \lambda_i/2$ for each $i \in S$. Choose any $\{(\hat{\lambda}_i, \hat{\boldsymbol{v}}_i)\}_{i\in S} \subset \mathbb{R} \times \mathbb{R}^n$ such that*

$$|\lambda_i - \hat{\lambda}_i| \le \hat{\varepsilon}, \quad \|\hat{\boldsymbol{v}}_i\| = 1, \quad \text{and} \quad \langle \boldsymbol{v}_i, \hat{\boldsymbol{v}}_i \rangle \ge 1 - 2(\hat{\varepsilon}/\lambda_i)^2 > 0,$$

and define $\boldsymbol{\Delta}_i := \lambda_i \boldsymbol{v}_i^{\otimes p} - \hat{\lambda}_i \hat{\boldsymbol{v}}_i^{\otimes p}$ for $i \in S$. Pick any unit vector $\boldsymbol{x} = \sum_{i=1}^n x_i \boldsymbol{v}_i$. Let $S_1 \subseteq S$ be the indices $i \in [n]$ such that $\lambda_i |x_i| \ge 4\hat{\varepsilon}$, and let $S_2 := S \setminus S_1$. Then

$$\left\| \sum_{i\in S_1} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \right\|_2 \le 2^{p+1} p \left(\sum_{i\in S_1} x_i^{2(p-2)} \right)^{1/2} \hat{\varepsilon} + 2^{p+1} \sum_{i\in S_1} |x_i|^{p-1} \hat{\varepsilon}, \tag{4.3.1}$$

$$\left\| \sum_{i\in S_2} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \right\|_2 \le 6^p \left(\sum_{i\in S_2} \left(\frac{\hat{\varepsilon}}{\lambda_i} \right)^{2(p-2)} \right)^{1/2} \hat{\varepsilon} + 6^p \sum_{i\in S_2} \left(\frac{\hat{\varepsilon}}{\lambda_i} \right)^{p-1} \hat{\varepsilon}. \tag{4.3.2}$$

These imply that there exists positive constants $c_1, c_2 > 0$, depending only on p, such that

$$\left\| \sum_{i\in S_1} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \right\|_2 \le c_1 \cdot \left(\sum_{i\in S_1} |x_i|^{p-1} \hat{\varepsilon} \right), \tag{4.3.3}$$

$$\left\| \sum_{i\in S_2} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \right\|_2 \le c_2 \cdot \left(\sum_{i\in S_2} \left(\frac{\hat{\varepsilon}}{\lambda_i} \right)^{p-1} \hat{\varepsilon} \right), \tag{4.3.4}$$

$$\left\| \sum_{i\in S} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \right\|_2 \le c_1 \cdot \left(\sum_{i\in S} |x_i|^{p-1} \hat{\varepsilon} \right) + c_2 \cdot \left(|S| \left(\frac{\hat{\varepsilon}}{\min_{i\in S} \lambda_i} \right)^{p-1} \hat{\varepsilon} \right). \tag{4.3.5}$$

Remark 4.9 *Lemma 4.8 indicates that the accumulating error $\sum_{i\in S} \boldsymbol{\Delta}_i$ much less severely affects vectors that are incoherent with $\{\boldsymbol{v}_i : i \in S\}$. For instance, $\left\| \sum_{i\in S} \boldsymbol{\Delta}_i \boldsymbol{v}_i^{\otimes p-1} \right\| = O(\hat{\varepsilon}^2)$ for $i \in [n] \setminus S$, while $\left\| \sum_{i\in S} \boldsymbol{\Delta}_i \boldsymbol{v}_i^{\otimes p-1} \right\| = O(\hat{\varepsilon})$ for $i \in S$. We leave the proof for 4.8 in Section 4.4.*

4.3.2 Proof of main theorem

We now use Lemma 4.8 to prove the main theorem.

Proof [Proof of Theorem 4.7] It suffices to prove that the following property holds for each $i \in [n]$:

$$\text{there is a permutation } \pi \text{ on } [i] \text{ s.t. for all } j \in [i], \begin{cases} |\lambda_{\pi(j)} - \hat{\lambda}_j| \le 2\varepsilon, \text{ and} \\ \left\| \boldsymbol{v}_{\pi(j)} - \hat{\boldsymbol{v}}_j \right\| \le \frac{20\varepsilon}{\lambda_{\pi(j)}}. \end{cases} \tag{$*$}$$

The proof is by induction. The base case of ($*$) (where $i = 1$) follows directly from by Theorem 4.2.

Assume the induction hypothesis ($*$) is true for some $i \in [n-1]$. We will prove that there exists an

$l \in [n] \setminus \{\pi(j) : j \in [i]\}$ that satisfies

$$|\lambda_l - \hat{\lambda}_{i+1}| \le 2\varepsilon, \quad \|\boldsymbol{v}_l - \hat{\boldsymbol{v}}_{i+1}\| \le 20\varepsilon/\lambda_l. \tag{4.3.6}$$

To simplify notation, we assume without loss of generality (by renumbering indices) that $\pi(j) = j$ for each $j \in [i]$. Let $\hat{\boldsymbol{x}} = \sum_{i\in[n]} x_i \boldsymbol{v}_i := \hat{\boldsymbol{v}}_{i+1}$ and $\hat{\lambda} := \hat{\lambda}_{i+1}$, and further assume without loss of generality (again by renumbering indices) that

$$\lambda_{i+1}|x_{i+1}|^{p-2} \ge \lambda_{i+2}|x_{i+2}|^{p-2} \ge \cdots \ge \lambda_n |x_n|^{p-2}.$$

In the following, we will show that $l = i + 1$ is an index satisfying (4.3.6). We use the assumption that

$$\varepsilon < \min\left\{\frac{1}{8}, \frac{1}{2.5 + 10c_1}, \frac{1}{10(40c_2 n)^{1/(p-1)}}\right\} \cdot \lambda_{\min} \tag{4.3.7}$$

(which holds with a suitable choice of c_0 in the theorem statement). Here, c_1 and c_2 are the constants from Lemma 4.8 when $\hat{\varepsilon} = 10\varepsilon$. It can be verified that ($*$) implies that the conditions for Lemma 4.8 are satisfied with this value of $\hat{\varepsilon}$.

Recall that $\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}_i \hat{\boldsymbol{x}}^{\otimes p}$, where

$$\widehat{\boldsymbol{\mathcal{T}}}_i = \widehat{\boldsymbol{\mathcal{T}}} - \sum_{j=1}^{i} \hat{\lambda}_j \hat{\boldsymbol{v}}_j^{\otimes p} = \sum_{j=i+1}^{n} \lambda_j \boldsymbol{v}_j^{\otimes p} + \boldsymbol{\mathcal{E}} + \sum_{j=1}^{i} \boldsymbol{\Delta}_j.$$

We now bound $\hat{\lambda}$ from above and below. For the lower bound, we use (4.3.4) from Lemma 4.8 to obtain

$$\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}_i \hat{\boldsymbol{x}}^{\otimes p} \ge \max_{j\in[n]\setminus[i]} \widehat{\boldsymbol{\mathcal{T}}}_i \boldsymbol{v}_j{}^{\otimes p} \ge \lambda_{\max,i} - \varepsilon - c_2 n \left(\frac{10\varepsilon}{\lambda_{\min}}\right)^{p-1} \varepsilon \ge \lambda_{\max,i} - 1.25\varepsilon \tag{4.3.8}$$

where $\lambda_{\max,i} := \max_{j\in[n]\setminus[i]} \lambda_j$ and $\lambda_{\min} := \min_{j\in[n]} \lambda_j$; the final inequality uses the conditions on ε in (4.3.7). For the upper bound, we have

$$\begin{aligned}
\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}_i \hat{\boldsymbol{x}}^{\otimes p} &= \sum_{j=i+1}^{n} \lambda_i x_i^p + \boldsymbol{\mathcal{E}} \hat{\boldsymbol{x}}^{\otimes p} + \sum_{j=1}^{i} \boldsymbol{\Delta}_j \hat{\boldsymbol{x}}^{\otimes p} \\
&\le \sum_{j=i+1}^{n} \lambda_j x_j^p + \varepsilon + 10c_1 \sum_{j=1}^{i} |x_j|^{p-1}\varepsilon + 10c_2 n \left(\frac{10\varepsilon}{\lambda_{\min}}\right)^{p-1} \varepsilon \\
&\le \lambda_{i+1}|x_{i+1}|^{p-2} \sum_{j=i+1}^{n} x_j^2 + \varepsilon + 10c_1\varepsilon \sum_{j=1}^{i} x_j^2 + 10c_2 n \left(\frac{10\varepsilon}{\lambda_{\min}}\right)^{p-1} \varepsilon \\
&\le \max\left\{\lambda_{i+1}|x_{i+1}|^{p-2}, 10c_1\varepsilon\right\} + 1.25\varepsilon.
\end{aligned} \tag{4.3.9}$$

The first inequality above follows from (4.3.5) in Lemma 4.8; the third inequality uses the fact that $\sum_{j=1}^{n} x_j^2 =$

1 as well as the conditions on ε in (4.3.7). If the max is achieved by the second argument $10c_1\varepsilon$, then combining (4.3.8) and (4.3.9) gives

$$(2.5 + 10c_1)\varepsilon \geq \lambda_{\max,i} \geq \lambda_{\min},$$

a contradiction of (4.3.7). Therefore the max in (4.3.9) must be achieved by $\lambda_{i+1}|x_{i+1}|^{p-2}$, and hence combining (4.3.8) and (4.3.9) gives

$$\lambda_{i+1}|x_{i+1}|^{p-2} \geq \lambda_{\max,i} - 2.5\varepsilon \quad \text{and} \quad |\hat{\lambda} - \lambda_{i+1}| \leq 1.25\varepsilon.$$

This in turn implies that

$$|x_{i+1}| \geq |x_{i+1}|^{p-2} \geq 1 - \frac{2.5\varepsilon}{\lambda_{i+1}}, \quad \lambda_{i+1} \geq \lambda_{\max,i} - 2.5\varepsilon, \quad \text{and} \quad x_{i+1}^2 \geq x_{i+1}^{p-1} \geq 1 - \frac{5\varepsilon}{\lambda_{i+1}}. \tag{4.3.10}$$

Thus, we have shown that $\hat{\boldsymbol{x}}$ is indeed coherent with $\hat{\boldsymbol{v}}_{i+1}$. Next, we will sharpen the bound for $\|\hat{\boldsymbol{x}} - \hat{\boldsymbol{v}}_{i+1}\|$ by considering the first order optimality condition.

Since $\hat{\boldsymbol{x}} \in \arg\min_{\|\boldsymbol{x}\|_2=1} \widehat{\boldsymbol{\mathcal{T}}}_i \boldsymbol{x}^{\otimes p}$, a first-order optimality condition similar to (4.2.7) implies $\hat{\lambda} = \widehat{\boldsymbol{\mathcal{T}}}_i \hat{\boldsymbol{x}}^{\otimes p}$. Thus

$$\begin{aligned}\hat{\lambda}\hat{\boldsymbol{x}} = \widehat{\boldsymbol{\mathcal{T}}}_i \hat{\boldsymbol{x}}^{\otimes p-1} &= \left(\sum_{j=i+1}^{n} \lambda_j \boldsymbol{v}_j^{\otimes p} + \boldsymbol{\mathcal{E}} + \sum_{j=1}^{i} \boldsymbol{\Delta}_j\right) \hat{\boldsymbol{x}}^{\otimes p-1} \\ &= \lambda_{i+1} x_{i+1}^{p-1} \boldsymbol{v}_{i+1} + \sum_{j=i+2}^{n} \lambda_j x_j^{p-1} \boldsymbol{v}_j + \boldsymbol{\mathcal{E}} \hat{\boldsymbol{x}}^{\otimes p-1} + \sum_{j=1}^{i} \boldsymbol{\Delta}_j \hat{\boldsymbol{x}}^{\otimes p-1}.\end{aligned}$$

Therefore

$$\begin{aligned}&\|\lambda_{i+1}(\hat{\boldsymbol{x}} - \boldsymbol{v}_{i+1})\|_2 \\ &= \left\|(\lambda_{i+1} - \hat{\lambda})\hat{\boldsymbol{x}} + (\hat{\lambda}\hat{\boldsymbol{x}} - \lambda_{i+1}\boldsymbol{v}_{i+1})\right\|_2 \\ &= \left\|(\lambda_{i+1} - \hat{\lambda})\hat{\boldsymbol{x}} + \lambda_{i+1}(x_{i+1}^{p-1} - 1)\boldsymbol{v}_{i+1} + \sum_{j=i+2}^{n} \lambda_j x_j^{p-1} \boldsymbol{v}_j + \boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1} + \sum_{j=1}^{i} \boldsymbol{\Delta}_j \hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2 \\ &\leq |\lambda_{i+1} - \hat{\lambda}| + \lambda_{i+1}|x_{i+1}^{p-1} - 1| + \left\|\sum_{j=i+2}^{n} \lambda_j x_j^{p-1} \boldsymbol{v}_j\right\|_2 + \left\|\boldsymbol{\mathcal{E}}\hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2 + \left\|\sum_{j=1}^{i} \boldsymbol{\Delta}_j \hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2.\end{aligned} \tag{4.3.11}$$

For the third term in (4.3.11), we use the fact that $|x_{i+2}| \leq \sqrt{1 - x_{i+1}^2}$, the bounds from (4.3.10) and the

conditions on ε in (4.3.7) to obtain

$$\begin{aligned}\left\|\sum_{j=i+2}^{n}\lambda_j x_j^{p-1}\boldsymbol{v}_j\right\|_2 &= \left(\sum_{j=i+2}^{n}\lambda_j^2 x_j^{2p-2}\right)^{1/2}\\ &\le \lambda_{i+2}|x_{i+2}|^{p-2}\sqrt{1-x_{i+1}^2}\\ &\le \lambda_{i+2}(1-x_{i+1}^2)\\ &\le \lambda_{\max,i}\frac{5\varepsilon}{\lambda_{i+1}}\\ &\le \frac{5\varepsilon}{1-2.5\varepsilon/\lambda_{\max,i}}\\ &\le 7.5\varepsilon. \end{aligned} \tag{4.3.12}$$

For the last term in (4.3.11), we use (4.3.5) from Lemma 4.8 and the conditions on ε in (4.3.7) to get

$$\begin{aligned}\left\|\sum_{j=1}^{i}\boldsymbol{\Delta}_j\hat{\boldsymbol{x}}^{\otimes p-1}\right\|_2 &\le 10c_1\sum_{j=1}^{n}|x_j|^{p-1}\varepsilon + 10c_2 n\left(\frac{10\varepsilon}{\lambda_{\min}}\right)^{p-1}\varepsilon\\ &\le 10c_1(1-x_{i+1}^2)\varepsilon + 10c_2 n\left(\frac{10\varepsilon}{\lambda_{\min}}\right)^{p-1}\varepsilon\\ &\le \frac{50c_1}{\lambda_{i+1}}\varepsilon^2 + 0.25\varepsilon\\ &\le 5.25\varepsilon. \end{aligned} \tag{4.3.13}$$

Therefore, substituting (4.3.10), (4.3.13) and $\|\boldsymbol{\mathcal{E}}\| \le \varepsilon$ into (4.3.11) gives

$$\|\lambda_{i+1}(\hat{\boldsymbol{x}} - \boldsymbol{v}_{i+1})\|_2 \le 20\varepsilon.$$

■

4.3.3 Stability of full decomposition

Theorem 4.7 states a (perhaps unexpected) phenomenon that the approximation errors do not accumulate with iteration number, whenever the perturbation error is small enough. In this subsection, we numerically corroborate this fact.

We generate nearly symmetric orthogonally decomposable tensors $\widehat{\boldsymbol{\mathcal{T}}} = \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \mathbb{R}^{10\times 10\times 10}$ as follows. We construct the underlying symmetric orthogonally decomposable tensor $\boldsymbol{\mathcal{T}}$ as the diagonal tensor with all diagonal entries equal to one, i.e., $\boldsymbol{\mathcal{T}} = \sum_{i=1}^{10}\boldsymbol{e}_i^{\otimes 3}$ (where $\boldsymbol{e}_i$ is the i-th coordinate basis vector). The

perturbation tensor $\boldsymbol{\mathcal{E}}$ is generated under three random models with $\sigma = 0.01$ before symmetrization:

Binary: independent entries $\mathcal{E}_{i,j,k} \in \{\pm\sigma\}$ uniformly at random;

Uniform: independent entries $\mathcal{E}_{i,j,k} \in [-2\sigma, 2\sigma]$ uniformly at random;

Gaussian: independent entries $\mathcal{E}_{i,j,k} \sim \mathcal{N}(0, \sigma^2)$.

For each random model, we generate 500 random instances, and apply Algorithm 11 to each $\widehat{\boldsymbol{\mathcal{T}}}$ to obtain approximate pairs $\{(\hat{\lambda}_i, \hat{\boldsymbol{v}}_i)\}_{i\in[10]}$. Again, we use GloptiPoly 3 to solve the polynomial optimization problem in Algorithm 11.

In Figure 4.2, we plot the mean and the standard deviation of the approximation errors for $\hat{\lambda}_i$ and $\hat{\boldsymbol{v}}_i$ from the 500 random instances (for each $i \in [10]$). These indeed do not appear to grow or accumulate as the iteration number increases. This is consistent with our results in Theorem 4.7.

4.3.4 When p is even

We now briefly discuss the case where the order of the tensor is even, i.e., $p \geq 4$ is an even integer.

Let $\widehat{\boldsymbol{\mathcal{T}}} := \boldsymbol{\mathcal{T}} + \boldsymbol{\mathcal{E}} \in \bigotimes^p \mathbb{R}^n$, where $\boldsymbol{\mathcal{T}}$ is a symmetric tensor with orthogonal decomposition $\boldsymbol{\mathcal{T}} = \sum_{i=1}^n \lambda_i \boldsymbol{v}_i^{\otimes p}$, where $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\}$ is an orthonormal basis of $\mathbb{R}^n$, $\lambda_i \neq 0$ for all $i \in [n]$, and $\boldsymbol{\mathcal{E}}$ is a symmetric tensor with operator norm $\varepsilon := \|\boldsymbol{\mathcal{E}}\|$. Note that unlike the case when p is odd, we cannot assume $\lambda_i > 0$ for all $i \in [n]$, and correspondingly, line 3 in Algorithm 11 now becomes

$$\hat{\boldsymbol{v}}_i \in \operatorname*{arg\,max}_{\|\boldsymbol{v}\|=1} \left|\widehat{\boldsymbol{\mathcal{T}}}_{i-1}\boldsymbol{v}^{\otimes p}\right| = \operatorname*{arg\,max}_{\|\boldsymbol{v}\|=1} \max\left\{\widehat{\boldsymbol{\mathcal{T}}}_{i-1}\boldsymbol{v}^{\otimes p}, -\widehat{\boldsymbol{\mathcal{T}}}_{i-1}\boldsymbol{v}^{\otimes p}\right\}, \quad \hat{\lambda}_i = \widehat{\boldsymbol{\mathcal{T}}}_{i-1}\hat{\boldsymbol{v}}_i^{\otimes p}.$$

Nevertheless, the pair $(\hat{\lambda}_i, \hat{\boldsymbol{v}}_i)$ still satisfies the first-order optimality condition $\hat{\lambda}_i\hat{\boldsymbol{v}}_i = \widehat{\boldsymbol{\mathcal{T}}}\hat{\boldsymbol{v}}_i^{\otimes p-1}$.

Our proof for Theorem 4.7 can be easily modified and leads to the following result: there exists a positive constant $\hat{c}_0 = \hat{c}_0(p) > 0$ such that whenever $\varepsilon \leq \hat{c}_0 \left(\min_{i\in[n]} |\lambda_i|\right)/n^{1/(p-1)}$, there exists a permutation π on $[n]$ such that

$$|\lambda_{\pi(j)} - \hat{\lambda}_j| \leq 2\varepsilon, \qquad \min\left\{\left\|\boldsymbol{v}_{\pi(j)} - \hat{\boldsymbol{v}}_j\right\|, \left\|\boldsymbol{v}_{\pi(j)} + \hat{\boldsymbol{v}}_j\right\|\right\} \leq 20\varepsilon/|\lambda_{\pi(j)}|, \quad \forall j \in [n].$$

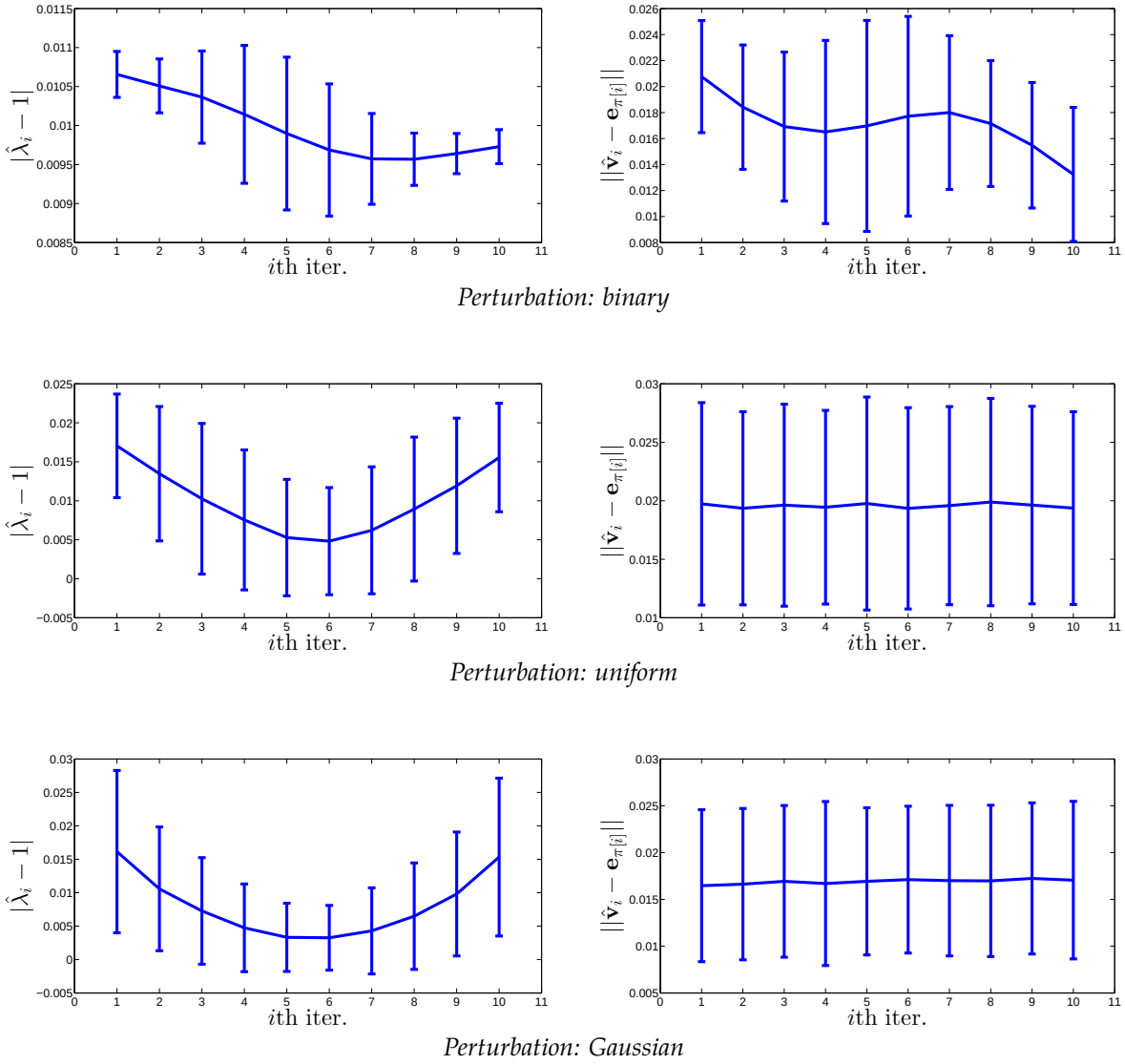


Figure 4.2: Approximation Errors of Algorithm 11. For each vertical bar over the iteration index $i \in [10]$, the midpoint is the mean of the approximation errors of $\hat{\lambda}_i$ (left) and $\hat{\boldsymbol{v}}_i$ (right), computed over 500 randomly generated instances. The error bars extend to two standard deviations above and below the mean.

4.4 Proof of Lemma 4.8

Proof The lemma holds trivially if $\hat{\varepsilon} = 0$. So we may assume $\hat{\varepsilon} > 0$. Therefore, for every $i \in S_1$, we have $|x_i| \geq 4\hat{\varepsilon}/\lambda_i > 0$. Let $c_i := \langle \boldsymbol{v}_i, \hat{\boldsymbol{v}}_i \rangle$, $\boldsymbol{w}_i := (\hat{\boldsymbol{v}}_i - c_i \boldsymbol{v}_i)/\left\|\hat{\boldsymbol{v}}_i - c_i \boldsymbol{v}_i\right\|_2$, and $y_i := \langle \boldsymbol{w}_i, \boldsymbol{x} \rangle$, so

$$\hat{\boldsymbol{v}}_i = c_i \boldsymbol{v}_i + \sqrt{1 - c_i^2}\boldsymbol{w}_i \quad \text{and} \quad \langle \hat{\boldsymbol{v}}_i, \boldsymbol{x} \rangle = c_i x_i + \sqrt{1 - c_i^2} y_i.$$

We first establish a few inequalities that will be frequently used later. Since $|\lambda_i - \hat{\lambda}_i| \le \hat{\varepsilon} \le \lambda_i/2$, one has $\hat{\varepsilon}/\lambda_i \le 1/2$, and $1/2 \le \hat{\lambda}_i/\lambda_i \le 3/2$. Also, since $c_i \ge 1 - 2(\hat{\varepsilon}/\lambda_i)^2 \ge 1/2$,

$$\sqrt{1-c_i^2} \le \sqrt{1-\left(1-2\left(\hat{\varepsilon}/\lambda_i\right)^2\right)^2} = \sqrt{4\left(\hat{\varepsilon}/\lambda_i\right)^2\left(1-\left(\hat{\varepsilon}/\lambda_i\right)^2\right)} \le 2\hat{\varepsilon}/\lambda_i.$$

For each $i \in S$,

$$\begin{aligned}
&\boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1} \\
&= \left(\lambda_i \boldsymbol{v}_i^{\otimes p} - \hat{\lambda}_i \hat{\boldsymbol{v}}_i^{\otimes p}\right)\boldsymbol{x}^{\otimes p-1} \\
&= \lambda_i x_i^{p-1}\boldsymbol{v}_i - \hat{\lambda}_i \left\langle \hat{\boldsymbol{v}}_i, \boldsymbol{x}\right\rangle^{p-1} \hat{\boldsymbol{v}}_i \\
&= \lambda_i x_i^{p-1}\boldsymbol{v}_i - \hat{\lambda}_i \left(c_i x_i + \sqrt{1-c_i^2}\, y_i\right)^{p-1}\left(c_i \boldsymbol{v}_i + \sqrt{1-c_i^2}\,\boldsymbol{w}_i\right) \\
&= \left(\lambda_i x_i^{p-1} - \hat{\lambda}_i c_i \left(c_i x_i + \sqrt{1-c_i^2}\, y_i\right)^{p-1}\right)\boldsymbol{v}_i - \left(\hat{\lambda}_i \sqrt{1-c_i^2}\left(c_i x_i + \sqrt{1-c_i^2}\, y_i\right)^{p-1}\right)\boldsymbol{w}_i.
\end{aligned}$$

Therefore, due to the orthonormality of $\{\boldsymbol{v}_i\}_{i\in[n]}$ and the triangle inequality, for each $j \in \{1,2\}$,

$$\begin{aligned}
\left\|\sum_{i\in S_j} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1}\right\|_2 \le &\left(\sum_{i\in S_j}\left(\lambda_i x_i^{p-1} - \hat{\lambda}_i c_i\left(c_i x_i + \sqrt{1-c_i^2}\, y_i\right)^{p-1}\right)^2\right)^{1/2} \\
&+ \sum_{i\in S_j}\left|\hat{\lambda}_i\sqrt{1-c_i^2}\left(c_i x_i + \sqrt{1-c_i^2}\, y_i\right)^{p-1}\right|.
\end{aligned} \tag{4.4.1}$$

We now prove (4.3.1). For any $i \in S_1$, since $x_i \neq 0$, we may write (4.4.1) as

$$\begin{aligned}
\left\|\sum_{i\in S_1} \boldsymbol{\Delta}_i \boldsymbol{x}^{\otimes p-1}\right\|_2 \le &\left(\sum_{i\in S_1} x_i^{2p-4}\left(\lambda_i x_i - \hat{\lambda}_i x_i c_i^p\left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\,\frac{y_i}{x_i}\right)^{p-1}\right)^2\right)^{1/2} \\
&+ \sum_{i\in S_1}\left|\hat{\lambda}_i x_i^{p-1} c_i^{p-1}\sqrt{1-c_i^2}\left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\,\frac{y_i}{x_i}\right)^{p-1}\right|.
\end{aligned} \tag{4.4.2}$$

Observe that

$$\left|\sqrt{\frac{1-c_i^2}{c_i^2}}\,\frac{y_i}{x_i}\right| \le \frac{\sqrt{1-c_i^2}}{|c_i|}\,\frac{1}{|x_i|} \le \frac{4\hat{\varepsilon}}{\lambda_i |x_i|} \le 1$$

because $|c_i| \ge 1/2$ and $\sqrt{1-c_i^2} \ge 2\hat{\varepsilon}/\lambda_i$. Moreover, since $1 + (p-1)z \le (1+z)^{p-1} \le 1 + (2^{p-1}-1)z$ for any $z \in [0,1]$,

$$\left|\left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\,\frac{y_i}{x_i}\right)^{p-1} - 1\right| \le (2^{p-1}-1)\frac{4\hat{\varepsilon}}{\lambda_i|x_i|} = (2^{p+1}-4)\frac{\hat{\varepsilon}}{\lambda_i|x_i|}. \tag{4.4.3}$$

Therefore,

$$\begin{aligned}\left|\lambda_i x_i - \hat{\lambda}_i x_i c_i^p \left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\frac{y_i}{x_i}\right)^{p-1}\right| &\leq |\lambda_i x_i - \hat{\lambda}_i x_i| + \hat{\lambda}_i |x_i| \left|1 - c_i^p \left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\frac{y_i}{x_i}\right)^{p-1}\right| \\ &\leq \hat{\varepsilon} + \frac{\hat{\lambda}_i |x_i|}{\lambda_i}\left((2^{p+1}-4)\frac{\hat{\varepsilon}}{|x_i|} + p\hat{\varepsilon} + (2^{p+1}-4)\frac{\hat{\varepsilon}}{|x_i|}\frac{p\hat{\varepsilon}}{\lambda_i}\right) \\ &\leq \hat{\varepsilon} + \frac{3}{2}\big((2^{p+1}-4) + p + (2^{p-1}-1)p\big)\hat{\varepsilon} \\ &\leq 2^{p+1} p \hat{\varepsilon}. \end{aligned} \tag{4.4.4}$$

The second inequality above is obtained using the inequality $|(1+a)(1+b)-1| \leq |a| + |b| + |ab|$ for any $a, b \in \mathbb{R}$, together with the inequality from (4.4.3) and the fact $|1 - c_i^p| \leq 2p(\hat{\varepsilon}/\lambda_i)^2 \leq p(\hat{\varepsilon}/\lambda_i)$. Using the resulting inequality in (4.4.4), the first summand in (4.4.2) can be bounded as

$$\left(\sum_{i \in S_1} x_i^{2p-2}\left(\lambda_i - \hat{\lambda}_i c_i^p \left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\frac{y_i}{x_i}\right)^{p-1}\right)^2\right)^{1/2} \leq 2^{p+1} p \left(\sum_{i \in S_1} x_i^{2(p-2)}\right)^{1/2} \hat{\varepsilon}. \tag{4.4.5}$$

To bound the second summand in (4.4.2), we have

$$\begin{aligned}\sum_{i \in S_1}\left|\hat{\lambda}_i x_i^{p-1} c_i^{p-1}\sqrt{1-c_i^2}\left(1 + \sqrt{\frac{1-c_i^2}{c_i^2}}\frac{y_i}{x_i}\right)^{p-1}\right| &\leq \sum_{i \in S_1}\left|\hat{\lambda}_i x_i^{p-1}\sqrt{1-c_i^2}\left(1 + \frac{\sqrt{1-c_i^2}}{c_i}\frac{1}{|x_i|}\right)^{p-1}\right| \\ &\leq \sum_{i \in S_1}\left|\hat{\lambda}_i x_i^{p-1}\frac{2\hat{\varepsilon}}{\lambda_i}\left(1 + \frac{4\hat{\varepsilon}}{\lambda_i |x_i|}\right)^{p-1}\right| \\ &\leq 2^{p+1}\sum_{i \in S_1} |x_i|^{p-1}\hat{\varepsilon}, . \end{aligned} \tag{4.4.6}$$

The second inequality uses the facts $c_i \geq 1/2$ and $\sqrt{1-c_i^2} \leq 2\hat{\varepsilon}/\lambda_i$; the last inequality uses the facts $\hat{\lambda}_i/\lambda_i \leq 3/2$ and $\lambda_i |x_i| \geq 4\hat{\varepsilon}$. Combining (4.4.5) and (4.4.6) gives the claimed inequality in (4.3.1) via (4.4.2).

It remains to prove (4.3.2). For each $i \in S_2$,

$$\begin{aligned}\left|\lambda_i x_i^{p-1} - \hat{\lambda}_i c_i \left(c_i x_i + \sqrt{1-c_i^2} y_i\right)^{p-1}\right| &\leq \lambda_i |x_i|^{p-1} + \hat{\lambda}_i \left(|x_i| + \sqrt{1-c_i^2}\right)^{p-1} \\ &\leq \lambda_i \left(\frac{4\hat{\varepsilon}}{\lambda_i}\right)^{p-1} + \hat{\lambda}_i \left(\frac{4\hat{\varepsilon}}{\lambda_i} + \frac{2\hat{\varepsilon}}{\lambda_i}\right)^{p-1} \\ &\leq \left(4^{p-1} + \frac{3}{2}\cdot 6^{p-1}\right)\left(\frac{\hat{\varepsilon}}{\lambda_i}\right)^{p-2}\hat{\varepsilon} \\ &\leq 6^p \left(\frac{\hat{\varepsilon}}{\lambda_i}\right)^{p-2}\hat{\varepsilon}. \end{aligned}$$

The second inequality uses the facts $\sqrt{1-c_i^2} \leq 2\hat{\varepsilon}/\lambda_i$ and $\lambda_i |x_i| < 4\hat{\varepsilon}$ for all $i \in S_2$; the third inequality uses

the fact $\hat{\lambda}_i/\lambda_i \leq 3/2$. Therefore

$$\left(\sum_{i\in S_2}\left(\lambda_i x_i^{p-1} - \hat{\lambda}_i c_i\left(c_i x_i + \sqrt{1-c_i^2}y_i\right)^{p-1}\right)^2\right)^{1/2} \leq 6^p\left(\sum_{i\in S_2}\left(\frac{\hat{\varepsilon}}{\lambda_i}\right)^{2(p-2)}\right)^{1/2}\hat{\varepsilon}. \tag{4.4.7}$$

Moreover,

$$\begin{aligned}\sum_{i\in S_2}\left|\hat{\lambda}_i\sqrt{1-c_i^2}\left(c_i x_i + \sqrt{1-c_i^2}y_i\right)^{p-1}\right| &\leq \sum_{i\in S_2}\left|\hat{\lambda}_i\sqrt{1-c_i^2}\left(|x_i| + \sqrt{1-c_i^2}\right)^{p-1}\right| \\ &\leq \sum_{i\in S_2}\hat{\lambda}_i\frac{2\hat{\varepsilon}}{\lambda_i}\left(\frac{4\hat{\varepsilon}}{\lambda_i} + \frac{2\hat{\varepsilon}}{\lambda_i}\right)^{p-1} \\ &\leq 3\cdot 6^{p-1}\sum_{i\in S_2}\left(\frac{\hat{\varepsilon}}{\lambda_i}\right)^{p-1}\hat{\varepsilon} \\ &\leq 6^p\sum_{i\in S_2}\left(\frac{\hat{\varepsilon}}{\lambda_i}\right)^{p-1}\hat{\varepsilon}. \end{aligned} \tag{4.4.8}$$

Combining (4.4.7) and (4.4.8) establishes (4.3.2) via (4.4.1) (with $j = 2$). ■

4.5 Conclusion

This chapter sheds light on a problem at the intersection of numerical linear algebra and statistical estimation, and our results draw upon and enrich the literature in both areas.

From the perspective of numerical linear algebra, SROA was previously only known to exactly recover the symmetric canonical decomposition of an orthogonal decomposable tensor. Our results show that it can robustly recover (approximate) orthogonal decompositions even when applied to nearly SOD tensors; this substantially enlarges the applicability of SROA.

Previous work on statistical estimation via orthogonal tensor decompositions considered the specific randomized power iteration algorithm of Anandkumar et al. [AGH+14], which has been successfully applied in a number of contexts [CL13, ZHPA13, ALB13, HS13, AGHK14, DWA14]. Our results provide formal justification for using other rank-one approximation methods in these contexts, and it seems to be quite beneficial, in terms of sample complexity and statistical efficiency, to use more sophisticated methods. Specifically, the perturbation error $\|\boldsymbol{\mathcal{E}}\|$ that can be tolerated is relaxed from power iteration's $O(1/n)$ to $O(1/\sqrt[p-1]{n})$. In future work, we plan to empirically investigate these potential benefits in a number of applications.

We also note that solvers for rank-one tensor approximation often lack rigorous runtime or error analyses,

which is not surprising given the computational difficulty of the problem for general tensors [HL13]. However, tensors that arise in applications are often more structured, such as being nearly SOD. Thus, another promising future research direction is to sidestep computational hardness barriers by developing and analyzing methods for such specially structured tensors (see also [AGH+14, BKS14] for ideas along this line).

Bibliography

Bibliography

[AGH+14] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014.

[AGHK14] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade. A tensor approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15(Jun):2239–2312, 2014.

[AGM12] N. S. Aybat, D. Goldfarb, and S. Ma. Efficient algorithms for robust and stable principal component pursuit problems. *Computational Optimization and Applications*, pages 1–29, 2012.

[AI15] N. S. Aybat and G. Iyengar. An alternating direction method with increasing penalty for stable principal component pursuit. *Computational Optimization and Applications*, 61(3):635–668, 2015.

[ALB13] M. G. Azar, A. Lazaric, and E. Brunskill. Sequential transfer in multi-armed bandit with finite set of models. In *Advances in Neural Information Processing Systems 26*, 2013.

[ALMT14] D. Amelunxen, M. Lotz, M. McCoy, and J. Tropp. Living on the edge: Phase transitions in convex programs with random data. *Inform. Inference*, 3(3):224–294, 2014.

[Ame11] D. Amelunxen. *Geometric analysis of the condition of the convex feasibility problem*. PhD thesis, PhD Thesis, Univ. Paderborn, 2011.

[ANW12] A. Agarwal, S. Negahban, and M. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, 40(2):1171–1197, 2012.

[Asw14] A. Aswani. Positive low-rank tensor completion. *arXiv preprint arXiv:1412.0620*, 2014.

[BC11] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011.

[Ber99] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[BJ03] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[BK15] B. W. Bader and T. G. Kolda. Matlab tensor toolbox version 2.6. *http://www.sandia.gov/ tgkolda/TensorToolbox/index-2.6.html*, 2015.

[BKS14] B. Barak, J. A. Kelner, and D. Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. *arXiv preprint arXiv:1407.1543*, 2014.

[BLM12] M. Bayati, M. Lelarge, and A. Montanari. Universality in polytope phase transitions and message passing algorithms. *arXiv preprint arXiv:1207.7321*, 2012.

[BPTD17] J. A. Bengua, H. N. Phiem, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 2017.

[BT09] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[CC70] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[CC12] S. B. Cohen and M. Collins. Tensor decomposition for fast parsing with latent-variable PCFGs. In *NIPS*, 2012.

[CCS10] J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[CHLZ12] B. Chen, S. He, Z. Li, and S. Zhang. Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107, 2012.

[CJ10] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[CL13] A. T. Chaganty and P. Liang. Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*, 2013.

[Cla10] K. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Trans. Algorithms*, 6(4):63:1–63:30, 2010.

[CLMW11] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[Com94] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[CP11] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[CPK80] J. D. Carroll, S. Pruzansky, and J. B. Kruskal. Candelinc: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45(1):3–24, 1980.

[CPW12] V. Chandrasekaran, P. Parrilo, and A. Willsky. Latent variable graphical model selection via convex optimization. *Annals of Statistics*, 40(4):1935–1967, 2012.

[CRPW12] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[CRT06] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[CSPW11] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[DH78] J. C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432 – 444, 1978.

[DK70] C. Davis and W. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[DLDMV00] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_n)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[Don06] D. Donoho. Compressed sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, 2006.

[DR70] V.F. Demyanov and A. M. Rubinov. *Approximate methods in optimization problems*. Modern analytic and computational methods in science and mathematics. American Elsevier Pub. Co., 1970.

[DSL08] V. De Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[DSSSC08] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *ICML*, 2008.

[DWA14] F. Doshi-Velez, B. Wallace, and R. Adams. Graph-Sparse LDA: A Topic Model with Structured Sparsity. *ArXiv e-prints*, October 2014.

[EAHK13] G. Ely, S. Aeron, N. Hao, and M. E. Kilmer. 5D and 4D pre-stack seismic data completion using tensor nuclear norm (TNN). In *SEG Annual Meeting*, 2013.

[ENP12] Y. C. Eldar, D. Needell, and Y. Plan. Uniqueness conditions for low-rank matrix recovery. *Applied and Computational Harmonic Analysis*, 33(2):309–314, 2012.

[EY36] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

[Faz02] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[FG16] R. M. Freund and P. Grigas. New analysis and results for the frank–wolfe method. *Mathematical Programming*, 155(1-2):199–230, 2016.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

[GB08] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. http://stanford.edu/~boyd/graph_dcp.html.

[GB14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

[GBK01] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[GCS15] J. Gao, J. Cheng, and M. D. Sacchi. A new 5D seismic reconstruction method based on a parallel square matrix factorization algorithm. In *SEG Technical Program Expanded Abstracts 2015*, pages 3784–3788. Society of Exploration Geophysicists, 2015.

[GMWZ17] D. Goldfarb, C. Mu, J. Wright, and C. Zhou. Using negative curvature in solving nonlinear programs. *arXiv preprint arXiv:1706.00896*, 2017.

[GQ14] D. Goldfarb and Z. Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM Journal on Matrix Analysis and Applications*, 35(1):225–253, 2014.

[Gro11] D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Info. Theory*, 57(3):1548–1566, 2011.

[GRY11] S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[Han13] L. Han. An unconstrained optimization approach for finding real eigenvalues of even order symmetric tensors. *Numerical Algebra, Control and Optimization (NACO)*, 3(3):583–599, 2013.

[Har70] R. Harshman. Foundations of the PARAFAC procedure: model and conditions for an "explanatory" multi-mode factor analysis. Technical report, UCLA Working Papers in Phonetics, 1970.

[Har72] R. A. Harshman. Parafac2: Mathematical and technical notes. *UCLA working papers in phonetics*, 22(3044):122215, 1972.

[HCD14] C. Hao, C. Cui, and Y. Dai. A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors. *Numerical Linear Algebra with Applications*, 2014.

[HJLW16] J. Hu, B. Jiang, X. Liu, and Z. Wen. A note on semidefinite programming relaxations for polynomial optimization over a single sphere. *Science China Mathematics*, 59(8):1543–1560, 2016.

[HJN14] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, pages 1–38, 2014.

[HKZ11] D. Hsu, S. Kakade, and T. Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*, 57(11):7221–7234, 2011.

[HL96] R. A. Harshman and M. E. Lundy. Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/CANDECOMP. *Psychometrika*, 61(1):133–154, 1996.

[HL13] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):45:1–45:39, November 2013.

[HLL09] D. Henrion, J. B. Lasserre, and J. Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[HMGW14] B. Huang, C. Mu, D. Goldfarb, and J. Wright. Provable models for robust low-rank tensor completion. *Pacific Journal of Optimization*, 11(2):339–364, 2014.

[HQB15] L. He, Z. T. Qin, and J. Bewli. Low-rank tensor recovery for geo-demand estimation in online retailing. *Procedia Computer Science*, 53:239–247, 2015.

[HS13] T.-K. Huang and J. Schneider. Learning hidden Markov models from non-sequence data via tensor decomposition. In *Advances in Neural Information Processing Systems 26*, 2013.

[Jag13] M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013.

[JMZ14] B. Jiang, S. Ma, and S. Zhang. Tensor principal component analysis via convex optimization. *Mathematical Programming*, pages 1–35, 2014.

[JMZ15] B. Jiang, S. Ma, and S. Zhang. New ranks for even-order tensors and their applications in low-rank tensor optimization. *arXiv preprint arXiv:1501.03689*, 2015.

[JO14] P. Jain and S. Oh. Provable tensor factorization with missing data. In *NIPS*, 2014.

[JRP07] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *CVPR*, 2007.

[JS10] M. Jaggi and M. Sulovsk. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

[KABO10] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.

[KB09] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[KBK05] T.G. Kolda, B.W. Bader, and J.P. Kenny. Higher-order web link analysis using multilinear algebra. In *Fifth International Conference on Data Mining*, 2005.

[Kie00] H. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of chemometrics*, 14(3):105–122, 2000.

[KM11] T. G. Kolda and J. R. Mayo. Shifted power method for computing tensor eigenpairs. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1095–1124, 2011.

[Kol01] T. G. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.

[Kol03] T. G. Kolda. A counterexample to the possibility of an extension of the Eckart-Young low-rank approximation theorem for the orthogonal rank tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 24(3):762–767, 2003.

[Kol06] T. G. Kolda. Multilinear operators for higher-order decompositions. Technical report, Sandia National Laboratories, 2006.

[KR02] E. Kofidis and P. A. Regalia. On the best rank-1 approximation of higher-order supersymmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(3):863–884, 2002.

[Kru77] J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95–138, 1977.

[KS13] N. Kreimer and M. D. Sacchi. Nuclear norm minimization and tensor completion in exploration seismology. In *ICASSP*, 2013.

[KSV14] D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

[Lar04] R. M. Larsen. Propack-software for large and sparse svd calculations. *Available online. URL http://sun. stanford. edu/rmunk/PROPACK*, pages 2008–2009, 2004.

[Las01] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[Lau12] S. Laue. A hybrid algorithm for convex semidefinite optimization. In *ICML*, 2012.

[LC10] L.-H. Lim and P. Comon. Multiarray signal processing: Tensor decomposition meets compressed sensing. *Comptes Rendus Mecanique*, 338(6):311–320, 2010.

[LC14] L.-H. Lim and P. Comon. Blind multilinear identification. *Information Theory, IEEE Transactions on*, 60(2):1260–1280, 2014.

[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.

[LGW+09] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. In *CAMSAP*, 2009.

[LHGT04] L. Li, W. Huang, I. Y. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004.

[Lim05] L.-H. Lim. Singular values and eigenvalues of tensors: a variational approach. *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 1:129–132, 2005.

[LL10] N. Li and B. Li. Tensor completion for on-board compression of hyperspectral images. In *ICIP*, 2010.

[LML13] D. Lim, B. McFee, and G. Lanckriet. Robust structural metric learning. In *ICML*, 2013.

[LMWY09] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *ICCV*, 2009.

[LNQY09] C. Ling, J. Nie, L. Qi, and Y. Ye. Biquadratic optimization over unit spheres and semidefinite programming relaxations. *SIAM Journal on Optimization*, 20(3):1286–1310, 2009.

[LP66] E. Levitin and B. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

[LRZM12] X. Liang, X. Ren, Z. Zhang, and Y. Ma. Repairing sparse low-rank texture. In *ECCV 2012*, 2012.

[LX10] T.-K. Huang J. Schneider J. G. Carbonell L. Xiong, X. Chen. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *ICDM*, 2010.

[LYZY10] Y. Li, J. Yan, Y. Zhou, and J. Yang. Optimum subspace learning and error correction for tensors. In *ECCV*, 2010.

[McC87] P. McCullagh. *Tensor Methods in Statistics*. Chapman and Hall, 1987.

[McC13] M. McCoy. *A geometric analysis of convex demixing*. PhD thesis, California Institute of Technology, 2013.

[MGC11] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

[MHG15] C. Mu, D. Hsu, and D. Goldfarb. Successive rank-one approximations for nearly orthogonally decomposable symmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1638–1659, 2015.

[MHG17] C. Mu, D. Hsu, and D. Goldfarb. Greedy approaches to symmetric orthogonal tensor decomposition. *arXiv preprint arXiv:1706.01169*, 2017.

[MHWG13] C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square deal: Lower bounds and improved convex relaxations for tensor recovery. *arXiv preprint arXiv:1307.5870*, 2013.

[MHWG14] C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *ICML*, 2014.

[MSS06] N. Mesgarani, M. Slaney, and S.A. Shamma. Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Trans. Audio, Speech, and Language Processing*, 14(3):920–930, 2006.

[MZWG16] C. Mu, Y. Zhang, J. Wright, and D. Goldfarb. Scalable robust matrix recovery: Frank-wolfe meets proximal methods. *SIAM J. Sci. Comput.*, 38(5):3291–3317, 2016.

[MZWM10] K. Min, Z. Zhang, J. Wright, and Y. Ma. Decomposing background topics from keywords by principal component pursuit. In *CIKM*, 2010.

[Nes83] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. In *Doklady AN SSSR*, volume 269, pages 543–547, 1983.

[Nes00] Y. Nesterov. Squared functional systems and optimization problems. *High performance optimization*, 13:405–440, 2000.

[NRWY12] S. Negahban, P. Ravikumar, M. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Stat. Sci.*, 27(4):528–557, 2012.

[NS10] D. Nion and N. Sidiropoulos. Tensor algebra and multi-dimensional harmonic retrieval in signal processing for mimo radar. *IEEE Trans. on Signal Processing*, 58(11):5693–5705, 2010.

[NW14] J. Nie and L. Wang. Semidefinite relaxations for best rank-1 tensor approximations. *SIAM Journal on Matrix Analysis and Applications*, 35(3):1155–1179, 2014.

[OCS14] R. Otazo, E. Candès, and D. K. Sodickson. Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. *Magnetic Resonance in Medicine*, 2014.

[OH16] S. Oymak and B. Hassibi. Sharp MSE bounds for proximal denoising. *Foundations of Computational Mathematics*, (16):965–1029, 2016.

[OJF+12] S. Oymak, A. Jalali, M. Fazel, Y. Eldar, and B. Hassibi. Simultaneously structured models with application to sparse and low-rank matrices. *arXiv preprint arXiv:1212.3753v1*, 2012.

[OJF+15] S. Oymak, A. Jalali, M. Fazel, Y. Eldar, and B. Hassibi. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Transactions on Information Theory*, 62(5):2886–2908, 2015.

[Par00] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[Par03] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[Pat93] M. Patriksson. Partial linearization methods in nonlinear programming. *Journal of Optimization Theory and Applications*, 78(2):227–246, 1993.

[PAV+13] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. http://arxiv.org/abs/1310.4716, 2013.

[PGW+12] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2233–2246, 2012.

[RBV13] E. Richard, F. Bach, and J. Vert. Intersecting singularities for multi-structured estimation. In *ICML*, 2013.

[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.

[Roc97] R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1997.

[ROV14] E. Richard, G. R Obozinski, and J.P. Vert. Tight convex relaxations for sparse matrix factorization. In *NIPS*, 2014.

[RPABBP13] B. Romera-Paredes, H. Aung, N. Bianchi-Berthouze, and M. Pontil. Multilinear multitask learning. In *ICML*, 2013.

[SBG04] A. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley, 2004.

[SC10] A. Stegeman and P. Comon. Subtracting a best rank-1 approximation may increase tensor rank. *Linear Algebra and its Applications*, 433(7):1276–1300, 2010.

[SDS10] M. Signoretto, L. De Lathauwer, and J. Suykens. Nuclear norms for tensors and their use for convex multilinear estimation. *Submitted to Linear Algebra and Its Applications*, 43, 2010.

[SHKM14] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Transactions on Image Processing*, 23(4):1678–1693, 2014.

[Sho87] N. Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics and Systems Analysis*, 23(5):695–700, 1987.

[STDLS13] M. Signoretto, Q. Tran Dinh, L. Lathauwer, and J. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, pages 1–49, 2013.

[Ste07] A. Stegeman. Degeneracy in Candecomp/Parafac and indscal explained for several three-sliced arrays with a two-valued typical rank. *Psychometrika*, 72(4):601–619, 2007.

[Ste08] A. Stegeman. Low-rank approximation of generic p\timesq\times2 arrays and diverging components in the Candecomp/Parafac model. *SIAM Journal on Matrix Analysis and Applications*, 30(3):988–1007, 2008.

[TSHK11] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima. Statistical performance of convex tensor decomposition. In *NIPS*, 2011.

[TTT03] R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95(2):189–217, 2003.

[Tuc66] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[TWW$^+$] H. Tan, P. Wang, Y. Wu, J. Zhang, and B. Ran. High-dimension traffic data imputation based on a square norm. In *CICTP 2016*, pages 284–294.

[TY11] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[Ver07] R. Vershynin. Math 280 lectue notes. http://www-personal.umich.edu/~romanv/teaching/2006-07/280/lec6.pdf, 2007.

[Ver12] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, pages 210–268. Cambridge University Press, 2012.

[Wey12] H. Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.

[WGMM13] J. Wright, A. Ganesh, K. Min, and Y. Ma. Compressive principal component pursuit. *Information and Inference*, 2(1):32–68, 2013.

[WGS+11] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *ACCV*, 2011.

[WN99] S. J. Wright and J Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[WQ07] Y. Wang and L. Qi. On the successive supersymmetric rank-1 decomposition of higher-order supersymmetric tensors. *Numerical Linear Algebra with Applications*, 14(6):503–519, 2007.

[WS17] M. Wang and Y. Song. Tensor decompositions via two-mode higher-order svd (hosvd). In *Artificial Intelligence and Statistics*, pages 614–622, 2017.

[WYG+09] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009.

[XHYS13] Y. Xu, R. Hao, W. Yin, and Z. Su. Parallel matrix factorization for low-rank tensor completion. To appear in *Inverse Problems and Imaging*, preprint available at *arXiv:1312.1254.*, 2013.

[YY13a] J. Yang and X. Yuan. Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013.

[YY13b] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *Pacific Journal of Optimization (PJO)*, 2013.

[YYQ14] Y. Yang, Q. Yang, and L. Qi. Properties and methods for finding the best rank-one approximation to higher-order tensors. *Computational Optimization and Applications*, 58(1):105–132, 2014.

[YZ11] J. Yang and Y. Zhang. Alternating direction algorithms for ℓ_1-problems in compressive sensing. *SIAM Journal on Scientific Computing*, 33(1):250–278, 2011.

[YZ15] M Yuan and C.-H. Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, pages 1–38, 2015.

[ZG01] T. Zhang and G. Golub. Rank-one approximation to high order tensors. *SIAM Journal on Matrix Analysis and Applications*, 23(2):534–550, 2001.

[Zha03] T. Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682–691, 2003.

[ZHPA13] J. Zou, D. Hsu, D. Parkes, and R. P. Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems 26*, 2013.

[ZLQ12] X. Zhang, C. Ling, and L. Qi. The best rank-1 approximation of a symmetric tensor and related spherical optimization problems. *SIAM Journal on Matrix Analysis and Applications*, 33(3):806–821, 2012.

[ZLW+10] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. In *ISIT*, 2010.

[ZMKW13] Y. Zhang, C. Mu, H. Kuo, and J. Wright. Towards guaranteed illumination models for nonconvex objects. In *ICCV*, 2013.