4-2021

Theory and Algorithms for Reliable Multimodal Data Analysis, Machine Learning, and Signal Processing

Dimitris G. Chachlakis
dc5467@rit.edu

Theory and Algorithms for Reliable Multimodal Data Analysis, Machine Learning, and Signal Processing

by

Dimitris G. Chachlakis

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical and Computer Engineering

Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York

April, 2021

Theory and Algorithms for Reliable Multimodal Data Analysis, Machine Learning, and Signal Processing

by

Dimitris G. Chachlakis

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Electrical and Computer Engineering.

Dr. Panos P. Markopoulos Dissertation Advisor	Date
Dr. Sohail Dianat Dissertation Committee Member	Date
Dr. Andreas Savakis Dissertation Committee Member	Date
Dr. Majid Rabbani Dissertation Committee Member	Date
Dr. Nathan Cahill Dissertation Defense Chairperson	Date

Certified by:

Dr. Andres Kwasinski Ph.D. Program Director, Electrical and Computer Engineering	Date

Theory and Algorithms for Reliable Multimodal Data Analysis, Machine Learning, and Signal Processing

by

Dimitris G. Chachlakis

Submitted to the
Kate Gleason College of Engineering
Ph.D. Program in Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical and Computer Engineering
at the Rochester Institute of Technology

Abstract

Modern engineering systems collect large volumes of data measurements across diverse sensing modalities. These measurements can naturally be arranged in higher-order arrays of scalars which are commonly referred to as tensors. Tucker decomposition (TD) is a standard method for tensor analysis with applications in diverse fields of science and engineering. Despite its success, TD exhibits severe sensitivity against outliers –i.e., heavily corrupted entries that appear sporadically in modern datasets. We study L1-norm TD (L1-TD), a reformulation of TD that promotes robustness. For 3-way tensors, we show, for the first time, that L1-TD admits an exact solution via combinatorial optimization and present algorithms for its solution. We propose two novel algorithmic frameworks for approximating the exact solution to L1-TD, for general N-way tensors. We propose a novel algorithm for dynamic L1-TD –i.e., efficient and joint analysis of streaming tensors. Principal-Component Analysis (PCA) (a special case of TD) is also outlier responsive. We consider Lp-quasinorm PCA (Lp-PCA) for $p < 1$, which promotes robustness. Before this dissertation, to the best of our knowledge, the exact solution to Lp-PCA ($p < 1$) was unknown. We show, for the first time, that the problem of one principal component can be solved exactly through a combination of convex problems and we provide corresponding optimal algorithms. We propose a novel near-exact algorithm for jointly extracting multiple components. In a different, but related, research direction, we propose new theory and algorithms for robust Coprime Array (CA) processing. In Direction-of-Arrival estimation, CAs enable the identification of more sources than sensors by forming the autocorrelation matrix of a larger virtual uniform linear array which is known as coarray. We derive closed-form Mean Squared Error (MSE) expressions for the Coarray Autocorrelation Matrix (CAM) estimation error. We develop a novel approach for estimating the CAM that is designed under the Minimum-MSE optimality criterion. Finally, we propose a novel CAM estimate which, in contrast to existing estimates, satisfies the structure-properties of the nominal (true) CAM.

Acknowledgments

This doctorate dissertation would not have been possible without the guidance, support, and assistance that I have received during both my undergraduate and graduate studies.

First, I would like to express my sincerest gratitude to my doctorate dissertation advisor, Dr. Panos P. Markopoulos, whose invaluable expertise contributed significantly towards the realization of this dissertation. Thank you for your patience, unconditional support, and opportunities that you have kindly offered me. Not only have you been a fantastic advisor, but also a mentor who has always inspired me and endowed me with a remarkable work ethic.

I would also like to thank the members of my dissertation committee, Dr. Sohail Dianat, Dr. Andreas Savakis, Dr. Majid Rabbani, and Dr. Nathan Cahill, for their invaluable feedback that helped improve this dissertation, and for the time invested in meetings and reviewing my research.

Further, I want to thank my colleagues at the MILOS Lab for our collaboration and the great discussions we had throughout the years. I would like to thank my colleague, Mayur Dhanaraj, with whom I collaborated the most and developed a great friendship.

Moreover, I would like to thank Ms. Rebecca Ziebarth and Dr. Edward Hensel for the tremendous support and invaluable advise they have offered me whenever it was needed.

I would like to thank the Rochester Institute of Technology, U.S. National Science Foundation, and U.S. Air Force Research Lab, for funding my research. Also, I would like to thank the U.S. National Geospatial-Intelligence Agency for co-funding research projects I have worked on.

Notably, I would like to thank my collaborators, Dr. Ashley Prater-Bennette, Dr. Fauzia Ahmad, and Dr. Evangelos Papalexakis, for the continuous and fruitful collaboration.

I want to thank my undergraduate thesis advisor, Dr. George N. Karystinos, for introducing me to the world of research and for encouraging me to pursue a Ph.D. degree. Likewise, I would like to thank Dr. Aggelos Bletsas and Dr. Athanasios Liavas for the excellent work they have been doing at the Technical University of Crete and for being a great source of inspiration.

At last, I would like to thank my family. I want to thank my father, George and my sister, Maria-Ioanna for the unconditional support they have selflessly offered me. I also want to thank my uncle, Charalambos, my aunt, Chrysoula, and my cousins, Konstantinos and Christos, for their unrelenting support. I also want to thank my loving girlfriend, Rebecca who has been supporting me in both the good and the bad times. Thank you for the untiring support you have offered me.

In memory of my loving mother, Eleni.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this doctorate dissertation, we propose new theory and algorithms for multimodal (tensor) data analysis, machine learning, and sparse array processing.

Modern engineering systems collect large volumes of data measurements across diverse sensing modalities. These measurements can naturally be arranged in higher-order arrays of scalars which are commonly referred to as multiway arrays or *tensors*. For instance, a 1-way tensor is a standard vector, a 2-way tensor is a standard matrix, and a 3-way array is a data cube of scalars. For higher-order tensors, visualization is not a trivial task and is left to the imagination. Storing, processing, and analyzing tensor data in their natural form enables the discovery of patterns and underlying data structures that would have otherwise stayed hidden. This is often attributed to the fact that tensors naturally model higher-order correlations among the data. Tucker Decomposition (TD) and Canonical Polyadic Decomposition (CPD) are the most popular tensor analysis approaches in the literature. TD focuses more on compression and multilinear subspace analysis while CPD aims at extracting sets of non-rotatable features that promote interpretability. Both TD and CPD find applications in diverse fields of science and engineering. In this dissertation, we focus on TD. Despite its success, TD exhibits severe sensitivity against outliers –i.e., heavy magnitude peripheral entries within the processed tensor. Accordingly, applications the performance of which relies on TD may attain compromised performance. We consider outlier resistant reformulations of TD, set the theoretical foundations for these formulations, and propose new algorithms.

Principal-Component Analysis (PCA) –a special case of TD– is a standard method for data analysis with a plethora of applications among fields of science and engineering. Similar to TD, it has been well documented that PCA is outlier-responsive. Thus, applications that rely on PCA may attain compromised performance when outliers are found among the processed data. To remedy the impact of outliers, researchers have proposed an array or outlier resistant reformulations of PCA. Arguably, L1-norm PCA (L1-PCA), which derives by simple substitution of the L2-norm in the PCA formulation by the more robust L1-norm, is the most straightforward one. This change in norm promotes robustness. L1-PCA is, in fact, a special case of the general Lp-norm PCA (Lp-PCA) formulation for $p = 1$. For general values of $p \leq 1$, before this dissertation, to the best of our knowledge, the exact solution to Lp-PCA was unknown. In this dissertation, we focus on the special case that $p \leq 1$ and pursue the exact solution to Lp-quasinorm Principal-Component Analysis.

In a different, but related, research direction, we propose new theory and estimation algorithms for robust coprime array processing. Coprime Arrays (CAs) are sparse arrays which offer an increased number of Degrees-of-Freedom (DoF) when compared to equal-length Uniform Linear Arrays (ULAs). CAs have attracted significant research interest over the past years and have been successfully employed in numerous modern applications.

The main contributions of this dissertation are organized in the following Chapters.

In Chapter 2, we develop theory and algorithms for reliable tensor data analysis. Standard tensor analysis approaches in the literature are formulated based on the L2/Frobenius norm which has been shown to exhibit severe sensitivity against peripheral heavy-magnitude/tail noise points. We consider L1-norm tensor analysis formulations and set the theoretical foundations that allow us to develop, for the first time, exact and approximate algorithms that are accompanied by formal complexity and convergence analyses. Furthermore, we also consider the problem of streaming tensor data and develop a new scalable algorithm that remains robust against outliers. The merits of L1-norm tensor analysis are clearly documented in an extensive array of numerical experiments with both synthetic and real-world datasets.

Next, in Chapter 3, we study the problem of Lp-quasinorm Principal-Component Analysis (Lp-PCA) for $p \leq 1$. Before this dissertation, to the best of our knowledge, the solution to Lp-PCA was unknown. We show, for the first time, that the problem of one principal component can be solved exactly through a combination of convex problems and we provide corresponding optimal algorithms. Moreover, we propose a novel near-exact algorithm for jointly extracting multiple components. Extensive numerical studies on both synthetic and real-world medical datasets corroborate the merits of Lp-PCA compared to the standard PCA.

Finally, in Chapter 4, we steer our focus on sparse array processing. Briefly, processing at a spare array receiver can be summarized in the following steps: physical-array autocorrelation estimation, autocorrelation combining, and spatial smoothing, after which an autocorrelation matrix that corresponds to a larger virtual uniform linear array is formed. We specifically focus on the autocorrelation combining step and develop new theory and a novel autocorrelation combining method that relies on the Minimum-Mean-Squared-Error optimality criterion. In addition, we propose an algorithm for computing improved autocorrelation estimates, compared to standard counterparts, by leveraging known structure properties that derive from the received-signal model. The new theory is validated by means of numerical simulations. The performances of the new autocorrelation combining approach and the new autocorrelation matrix estimate are evaluated by means of extensive numerical experiments.

Chapter 2

L1-norm Tensor Analysis

2.1 Introduction

Data collections across diverse sensing modalities are naturally stored and processed in the form of N-way arrays, also known as *tensors*. Introduced by L. R. Tucker [1] in the mid-1960s, Tucker decomposition (TD) is a standard method for the analysis and compression of tensor data. TD finds a plethora of applications across fields of science and engineering, such as communications [2–6], data analytics [7, 8], machine learning [9–14], computer vision [15–20], biomedical signal processing [21], social-network data analysis [22, 23], and pattern recognition [24, 25], among other fields. TD is typically used for compression, denoising, and model identification, to name a few. Notably, an alternative paradigm for tensor analysis, particularly popular for data mining, is the Canonical Polyadic Decomposition (CPD), also referred to as Parallel Factor Analysis (PARAFAC) [8, 26]. In contrast to TD that focuses more on compression and multilinear subspace analysis, CPD aims at extracting sets of non-rotatable features that promote interpretability.

In many applications of interest, an N-way data tensor is formed by concatenation of $(N-1)$-way coherent (same class, or distribution) tensor samples across its, without loss of generality (w.l.o.g), N-th mode –i.e., the data tensor comprises $(N-1)$ feature modes and 1 sample mode. For such applications, TD is accordingly reformulated to Tucker2 decomposition (T2D) [27], which can be described as *joint* TD of the $(N-1)$-way tensor samples. That is, T2D strives to *jointly* decompose the collected $(N-1)$-way tensors and unveil the low-rank multilinear structure of their class, or distribution. For the special case that $N = 3$ (collection of matrix or 2-way measurements), TD/T2D take the familiar form of Principal-Component Analysis (PCA). For the same case, T2D has also been presented as Generalized Low-Rank Approximation of Matrices (GLRAM) [28, 29], or 2-D Principal Component Analysis (2DPCA) [18, 30]. For $N = 2$ (collection of vector samples), both TD and T2D boil down to standard matrix Principal-Component Analysis (PCA), computable by means of Singular-Value Decomposition (SVD) [31].

In general, conventional TD tries to minimize the L2-norm of the residual-error in the low-rank approximated tensor that derives by multi-way projection of the original tensor onto the spans of N sought-after orthonormal bases –or, equivalently, TD tries to maximize the L2-norm of this multi-way projection. Higher-Order SVD (HOSVD) and Higher-Order Orthogonal Iteration (HOOI) algorithms [32] are well-known solvers for TD and T2D. Note that both types of solvers can generally only guarantee a locally optimal solution. Furthermore, a plethora of TD variants have also been presented in the literature. Truncated HOSVD (T-HOSVD) [33, 34], Sequentially Truncated HOSVD (ST-HOSVD) [35], Hierarchical HOSVD [36], and Nonnegative Tucker [37, 38], are just a few.

TD and T2D have also been studied for applications in which the tensor measurements arrive in a streaming way. In such applications, the sought-after TD bases have to be computed incrementally. Incremental solvers are also preferred, from a computational standpoint, when there are too many collected measurements to efficiently process them as a batch. Researchers have proposed an array of algorithms for incremental TD, including Dynamic Tensor Analysis (DTA), Streaming Tensor Analysis (STA), Window-based Tensor Analysis (WTA) [39, 40], and Accelerated Online Low-Rank Tensor Learning (ALTO) [41], to name a few. Scalable, parallelized, streaming, and randomized algorithms for TD have also been proposed in [42–46].

The merits of TD have been demonstrated in a wide range of applications. However, it is well documented that TD is very sensitive against faulty measurements (outliers). Outliers appear often in modern datasets due to sensor malfunctions, errors in data storage/transfer, and even deliberate dataset contamination in adversarial environments [47–52]. The same sensitivity has also been amply documented in PCA/SVD, which is a special case of TD for 2-way tensors. For the case of matrix decomposition, researchers have shown that the impact of faulty entries can be effectively counteracted by substituting SVD with L1-norm-based PCA (L1-PCA) [53, 54]. L1-PCA is formulated similar to standard PCA as a projection maximization problem, but replaces the corruption-responsive L2-norm by the robust L1-norm. L1-PCA has exhibited solid robustness against heavily corrupted data in an array of applications [55–58]. Extending this formulation to tensor processing, one can similarly endow robustness to the TD and T2D by substituting the L2-norm in their formulations by the L1-norm (not to be confused with sparsity-inducing L1-norm regularization schemes). An approximate algorithm for L1-norm-based Tucker2 (L1-T2D) was proposed in [52] for the special case that $N = 3$ and data are processed as a batch. However, L1-T2D and L1-TD remain to date unsolved and largely unexplored.

In this Chapter, we study the theoretical foundations of L1-norm TD and T2D and develop algorithms for their solutions. Specifically, our contributions are as follows:

Contribution i. We deliver, for the first time, the exact solution to rank-1 L1-T2D decomposition by means of two novel algorithms.

Contribution ii. We present generalized L1-TD decomposition for N-way tensors and review its links to PCA, TD/T2D, and L1-PCA. We propose two new algorithmic frameworks for the solution of L1-TD/L1-T2D, namely L1-norm Higher-Order SVD (L1-HOSVD) and L1-norm Higher-Order Orthogonal Iterations (L1-HOOI), which are accompanied by complete convergence and complexity analyses.

Contribution iii. We present Dynamic L1-Tucker: a scalable method for incremental L1-TD analysis, with the ability to (1) provide quality estimates of the Tucker bases, (2) detect and reject outliers, and (3) adapt to changes of the nominal subspaces.

Contribution iv. In all the above cases, we offer numerical studies that evaluate the performance of L1-TD and compare it with state-of-the-art counterparts. Our numerical studies corroborate that L1-TD performs similar to standard TD when the processed data are nominal/clean, while it exhibits sturdy resistance against corruptions among the data.

The rest of this Chapter is organized as follows. In Section 2.2, we introduce notation and review existing tensor analysis methods. Next, in Section 2.3, we present the general formulation of L1-norm Tucker (L1-Tucker) analysis. In Section 2.4, we present, for the first time, the exact solution to rank-1 L1-norm Tucker2 Analysis. Thereafter, in Section 2.5, we offer the proposed L1-HOSVD and L1-HOOI algorithmic frameworks for the solution to L1-Tucker/L1-Tucker2. In Section 2.6, we present Dynamic L1-Tucker (D-L1-Tucker) for incremental and dynamic analysis of streaming tensor measurements.

The contributions presented in this Chapter have also been presented in [59–65].

2.2 Technical Background

Notation and Tensor Preliminaries

In this Chapter, vectors and matrices are denoted by lower- and upper-case bold letters, respectively –e.g., $\mathbf{x} \in \mathbb{R}^{D_1}$ and $\mathbf{X} \in \mathbb{R}^{D_1 \times D_2}$. N-way tensors are denoted by upper-case calligraphic bold letters –e.g., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$. An N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N}$ can also be viewed as an M-way tensor in $\mathbb{R}^{D_1 \times D_2 \times \ldots \times D_M}$, for any $M > N$, with $D_m = 1$ for $m > N$. Collections/sets of tensors are denoted by upper-case calligraphic letters –e.g., $\mathcal{X} = \{\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}\}$. The squared Frobenius/L2 norm, $\| \cdot \|_F^2$, returns the sum of squared entries of its tensor argument while the L1-norm, $\| \cdot \|_1$, returns the sum of the absolute entries of its tensor argument. $\mathbb{S}_{D \times d} = \{\mathbf{Q} \in \mathbb{R}^{D \times d} : \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_d\}$ is the Stiefel manifold of rank-d orthonormal matrices in $\mathbb{R}^D$. Each entry of $\boldsymbol{\mathcal{X}}$ is identified by N indices $\{i_n\}_{n=1}^N$ such that $i_n \leq D_n$ for every $n \in [N] = \{1, 2, \ldots, N\}$. For every $n \in [N]$, $\boldsymbol{\mathcal{X}}$ can be seen as a collection of $P_n = \prod_{m \in [N] \setminus n} D_m$ length-D_n vectors known as mode-n fibers of $\boldsymbol{\mathcal{X}}$. For instance, given a fixed set of indices $i_{m \in [N] \setminus n}$, $\boldsymbol{\mathcal{X}}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$ is a mode-n fiber of $\boldsymbol{\mathcal{X}}$. A matrix that has as columns all the mode-n fibers of $\boldsymbol{\mathcal{X}}$ is called the mode-n unfolding (or, flattening) of $\boldsymbol{\mathcal{X}}$ and will henceforth be denoted as $\mathtt{mat}(\boldsymbol{\mathcal{X}}, n) \in \mathbb{R}^{D_n \times P_n}$. The reverse procedure, known as mode-n "tensorization", rearranges the entries of matrix $\mathbf{X} \in \mathbb{R}^{D_n \times P_n}$ to form tensor $\mathtt{ten}(\mathbf{X}; n; \{D_i\}_{i \neq n}) \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N}$, so that $\mathtt{mat}(\mathtt{ten}(\mathbf{X}; n; \{D_i\}_{i \neq n}), n) = \mathbf{X}$. $\boldsymbol{\mathcal{X}} \times_n \mathbf{A}$ is the mode-n product of tensor $\boldsymbol{\mathcal{X}}$ with matrix $\mathbf{A}$ of conformable size and $\boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n^\top$ compactly denotes the multi-way product $\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}_1^\top \times_2 \mathbf{Q}_2^\top \ldots \times_n \mathbf{Q}_N^\top$. In accordance with the common convention, the order in which the mode-n fibers of $\boldsymbol{\mathcal{X}}$ appear in $\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)$ is as specified in [66]. For more details on tensor preliminaries, we refer the interested reader to [12, 66].

Tucker Decomposition

Tucker tensor decomposition factorizes $\boldsymbol{\mathcal{X}}$ into N orthonormal *bases* and a *core tensor* that constitutes a compressed version of $\boldsymbol{\mathcal{X}}$. Specifically, considering $\{d_n\}_{n \in [N]}$ with $d_n \leq D_n$ $\forall n \in [N]$, Tucker decomposition is compactly formulated as

$$\underset{\{\mathbf{U}_n \in \mathbb{S}_{D_n \times d_n}\}_{n \in [N]}}{\max.} \left\| \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{U}_n^\top \right\|_F^2 . \tag{2.1}$$

If $\{\mathbf{U}_n^{\text{tckr}}\}_{n\in[N]}$ is a solution to (2.1), then

$$\boldsymbol{\mathcal{G}}^{\text{tckr}} \triangleq \boldsymbol{\mathcal{X}} \times_{n\in[N]} {\mathbf{U}_n^{\text{tckr}}}^\top \tag{2.2}$$

is the corresponding Tucker core of $\boldsymbol{\mathcal{X}}$, and $\boldsymbol{\mathcal{X}}$ is “low-rank” approximated by

$$\hat{\boldsymbol{\mathcal{X}}} \triangleq \boldsymbol{\mathcal{G}}^{\text{tckr}} \times_{n\in[N]} \mathbf{U}_n^{\text{tckr}} = \boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^{\text{tckr}} {\mathbf{U}_n^{\text{tckr}}}^\top. \tag{2.3}$$

If $d_n = D_n \forall n$, it trivially holds that $\boldsymbol{\mathcal{X}} = \hat{\boldsymbol{\mathcal{X}}}$. The minimum values of $\{d_n\}_{n\in[N]}$ for which $\boldsymbol{\mathcal{X}} = \hat{\boldsymbol{\mathcal{X}}}$ are the respective *mode ranks* of $\boldsymbol{\mathcal{X}}$. For general values of $\{d_n\}_{n\in[N]}$, the exact solution to (2.1) remains unknown and it is commonly approximated/pursued by means of the HOSVD [17] or HOOI algorithms [32], reviewed briefly below.

In HOSVD, the N bases are optimized disjointly, setting the n-th basis, $\mathbf{U}_n^{\text{hosvd}}$, to the d_n principal components (PCs) of the mode-n unfolding $\texttt{mat}\,(\boldsymbol{\mathcal{X}}, n)$, computed by means of standard SVD.

HOOI is a converging iterative procedure that, when initialized at HOSVD, it can provably attain a higher value to the metric in (2.1) [32, 67]. For each $n \in [N]$, the n-th basis is typically (but not necessarily) initialized as $\mathbf{U}_{n,0}^{\text{hooi}} = \mathbf{U}_n^{\text{hosvd}}$. Then, HOOI updates the bases iteratively. At the t-th iteration, $t = 1, 2, \ldots$, the n-th basis $\mathbf{U}_{n,t}^{\text{hooi}}$ is updated to the d_n dominant left-singular vectors of $\texttt{mat}(\boldsymbol{\mathcal{X}} \times_{m\in[n-1]} \mathbf{U}_{m,t}^{\text{hooi}} \times_{k\in[N-n]+n} \mathbf{U}_{k,t-1}^{\text{hooi}}, n)$ –thus, in contrast to HOSVD, HOOI optimizes the N bases jointly.

Data Corruption

Large datasets often contain heavily corrupted, outlying entries due to various causes, such as errors in data storage, heavy-tail noise, intermittent variations of the sensing environment, sensor malfunctions, and even intentional contamination [68]. Regrettably, such corruptions that lie far from the sought-after subspaces, are known to significantly affect Tucker [49, 54, 59]. Accordingly, the performance of any application that relies on Tucker can be significantly compromised if the processed data are corrupted. To a high extent, this corruption sensitivity of Tucker can be attributed to its L2-norm-based formulation, which places squared emphasis on each entry of the core, thus benefiting corrupted fibers of the data tensor. To demonstrate this, we present the following numerical study. We consider tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{10\times10\times10}$ with entries independently drawn from $\mathcal{N}(0,1)$. Then, we corrupt additively the single entry $\boldsymbol{\mathcal{X}}(3,3,4)$ with a point from $\mathcal{N}(0,\mu^2)$. We apply HOSVD on $\boldsymbol{\mathcal{X}}$ to obtain the single-dimensional bases $\mathbf{u}_1 \in \mathbb{R}^{10\times1}$, $\mathbf{u}_2 \in \mathbb{R}^{10\times1}$, and

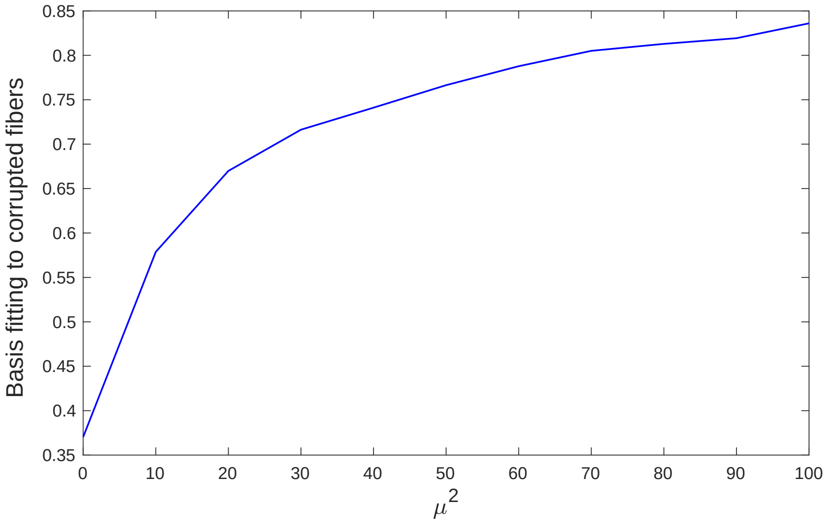


Figure 2.1. Fitting of HOSVD-derived bases to corrupted fibers, versus corruption variance μ^2.

$\mathbf{u}_3 \in \mathbb{R}^{10\times 1}$ and measure the aggregate normalized fitting of the bases to the corrupted fibers of $\boldsymbol{\mathcal{X}}$ as $f(\mu^2) = \frac{1}{3}\sum_{i=1}^{3} |\mathbf{u}_i^\top \mathbf{x}_i|^2 \|\mathbf{x}_i\|_2^{-2}$, where $\mathbf{x}_1 = \boldsymbol{\mathcal{X}}(:,3,4)$, $\mathbf{x}_2 = \boldsymbol{\mathcal{X}}(3,:,4)$, and $\mathbf{x}_3 = \boldsymbol{\mathcal{X}}(3,3,:)$. We repeat this study on 3000 distinct realizations of $\boldsymbol{\mathcal{X}}$ and plot in Fig. 2.1 the average value of $f(\mu^2)$, versus $\mu^2 = 0, 10, \ldots, 100$. We observe that, as μ increases, $\mathbf{u}_i$ tends to the corrupted fiber $\mathbf{x}_i$, for every i, and $f(\mu^2)$ increases towards 1.

To counteract the impact of corruptions, researchers have resorted in "robust" reformulations of PCA and Tucker. One popular approach seeks to approximate the processed data matrix/tensor as the summation of a sought-after low-rank component and a jointly optimized sparse component that models corruption [50, 69–71]. This approach relies on weights that regulate approximation rank, sparsity, and iteration step-size.

An alternative approach in matrix analysis replaces the corruption-responsive L2-norm in PCA by the L1-norm, resulting to L1-PCA [54]. The meaningful formulation of L1-PCA and its documented robustness in an array of applications have largely motivated the tensor-processing developments of this Chapter. Next, to set the technical background of our work, we briefly present L1-PCA.

The L1-PCA Paradigm

Given a data matrix $\mathbf{X} \in \mathbb{R}^{D_1 \times D_2}$ and $d_1 \leq \text{rank}(\mathbf{X})$, L1-PCA is defined as [54]

$$\underset{\mathbf{U} \in \mathbb{S}_{D_1 \times d_1}}{\text{max.}} \; \|\mathbf{U}^\top \mathbf{X}\|_1, \tag{2.4}$$

where the L1-norm $\|\cdot\|_1$ returns the summation of the absolute entries of its matrix argument. L1-PCA in (2.4) was solved exactly in [54], where authors presented the following Theorem 2.1.

Theorem 2.1. *Let* $\mathbf{B}_{\text{nuc}}$ *be an optimal solution to*

$$\underset{\mathbf{B} \in \{\pm 1\}^{D_2 \times d_1}}{\text{max.}} \|\mathbf{X}\mathbf{B}\|_*. \tag{2.5}$$

Then, $\mathbf{U}_{\text{L1}} = \Phi(\mathbf{X}\mathbf{B}_{\text{nuc}})$ *is an optimal solution to L1-PCA in* (2.4). *Moreover,* $\|\mathbf{X}^\top \mathbf{U}_{\text{L1}}\|_1 = \text{Tr}\left(\mathbf{U}_{\text{L1}}^\top \mathbf{X}\mathbf{B}_{\text{nuc}}\right) = \|\mathbf{X}\mathbf{B}_{\text{nuc}}\|_*$.

Nuclear norm $\|\cdot\|_*$ in (2.5) returns the sum of the singular values of its matrix argument. For any tall matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ that admits SVD $\mathbf{W}\mathbf{S}_{n \times n}\mathbf{Q}^\top$, $\Phi(\cdot)$ in Theorem 2.1 is defined as $\Phi(\mathbf{A}) \overset{\triangle}{=} \mathbf{W}\mathbf{Q}^\top$. Moreover, by the Procrustes Theorem [72], it holds that $\Phi(\mathbf{A}) = \text{argmax}_{\mathbf{U} \in \mathbb{S}_{m \times n}} \text{Tr}(\mathbf{U}^\top \mathbf{A}) = \text{argmin}_{\mathbf{U} \in \mathbb{S}_{m \times n}} \|\mathbf{U} - \mathbf{A}\|_F$.

By means of Theorem 2.1, the solution to L1-PCA is obtained by solving the Binary Nuclear-norm Maximization (BNM) in (2.5), with an additional SVD step. BNM can be solved by exhaustive search in its finite-size feasibility set, or more intelligent algorithms of lower cost, as shown in [54]. Computationally efficient, approximate solvers for (2.5) and (2.4) were presented in [55, 58, 73–76]. Incremental solvers for L1-PCA were presented in [77, 78]. Algorithms for L1-PCA of complex-valued data were recently presented in [79, 80]. To date, L1-PCA has found many applications in signal processing and machine learning, such as radar-based motion recognition and foreground-activity extraction in video sequences [56, 57].

Existing Methods for Incremental and Dynamic Tucker

Streaming and robust matrix PCA has been thoroughly studied over the past decades [77, 81–85]. However, extending matrix PCA (batch or streaming) to tensor analysis is a non-trivial task that has been attracting increasing research interest. To date, there exist multiple alternative methods for batch tensor analysis (e.g., HOSVD, HOOI, L1-HOOI) but only few for streaming/dynamic tensor analysis. For example, DTA [39, 40] efficiently approximates the HOSVD solution by processing measurements incrementally, with a fixed computational cost per update. Moreover, DTA can track multilinear changes of subspaces, weighing past measurements with a forgetting factor. STA [39, 40] is a fast alternative to DTA, particularly designed for time-critical applications. WTA is another DTA variant which, in contrast to DTA and STA, adapts to changes by considering only a sliding window of measurements. The ALTO method was presented in [41]. For each new measurement, ALTO updates the bases through a tensor regression model. In [86], authors presented another method for Low-Rank Updates to Tucker (LRUT). When a new measurement arrives, LRUT projects it on the current bases and few more randomly chosen orthogonal directions, forming an augmented core tensor. Then it updates the bases by standard Tucker (e.g., HOSVD) on this extended core. In [45], authors consider very large tensors and propose randomized algorithms for Tucker decomposition based on the TENSORSKETCH [87]. It is stated that these algorithms can also extend for processing streaming data. Randomized methods for Tucker of streaming tensor data were also proposed in [42]. These methods rely on dimension-reduction maps for sketching the Tucker decomposition and they are accompanied by probabilistic performance guarantees. More methods for incremental tensor processing were presented in [88–91], focusing on specific applications such as foreground segmentation, visual tracking, and video foreground/background separation.

Methods for incremental CPD/PARAFAC tensor analysis have also been presented. For instance, authors in [92] consider the CPD/PARAFAC factorization model and assume that N-way measurements are streaming. They propose CP-Stream, an algorithm that efficiently updates the CPD every time a new measurement arrives. CP-stream can accommodate user-defined constraints in the factorization such as non-negativity. In addition, authors in [93] consider a Bayesian probabilistic reformulation of the CPD/PARAFAC factorization, assuming that the entries of the processed tensor are streaming across all modes, and develop a posterior inference algorithm (POST). Further, the problem of robust and incremental PARAFAC has also been studied and algorithms have been presented in [94, 95]. Typically, the application spaces of CPD and TD are complementary: CPD is preferred when uniqueness and interpretability are needed; Tucker allows for the latent components to be related (dense core) and it is preferred for low-rank tensor compression and completion, among other tasks [8, 14].

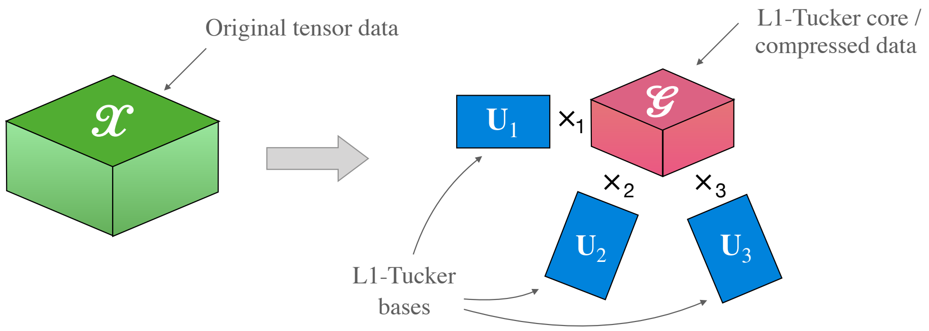


Figure 2.2. Schematic illustration of L1-Tucker decomposition for $N = 3$.

2.3 Problem Statement

Motivated by the corruption resistance of L1-PCA, we study L1-Tucker decomposition. L1-Tucker derives by simply replacing the L2-norm in (2.1) by the corruption-resistant L1-norm,[1] as

$$\underset{\{\mathbf{U}_n \in \mathbb{S}_{D_n \times d_n}\}_{n\in[N]}}{\text{max.}} \left\| \boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^\top \right\|_1 . \tag{2.6}$$

That is, L1-Tucker in (2.6) strives to maximize the sum of the absolute entries of the Tucker core $\boldsymbol{\mathcal{G}} \overset{\triangle}{=} \boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^\top$ –while standard Tucker maximizes the sum of the squared entries of the core. A schematic illustration of L1-Tucker decomposition for $N = 3$ is offered in Fig. 2.2. An interesting observation is that, for any $m \in [N]$, $\left\| \boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^\top \right\|_1 = \left\| \mathbf{U}_m^\top \mathbf{A}_m \right\|_1$, where $\mathbf{A}_m = \texttt{mat}(\boldsymbol{\mathcal{X}} \times_{n<m} \mathbf{U}_n^\top \times_{k>m} \mathbf{U}_k^\top, m)$.

In many applications, $\boldsymbol{\mathcal{X}}$ emerges as collection of D_N coherent $(N-1)$-way tensor measurements that are to be jointly decomposed. Defining $\boldsymbol{\mathcal{X}}_i \overset{\triangle}{=} \boldsymbol{\mathcal{X}}(:, \ldots, :, i) \forall i \in [D_N]$, the joint L1-Tucker analysis of $\{\boldsymbol{\mathcal{X}}_i\}_{i\in[D_N]}$ is formulated as

$$\underset{\{\mathbf{U}_n \in \mathbb{S}_{D_n \times d_n}\}_{n\in[N-1]}}{\text{max.}} \sum_{i=1}^{D_N} \left\| \boldsymbol{\mathcal{X}}_i \times_{n\in[N-1]} \mathbf{U}_n^\top \right\|_1 . \tag{2.7}$$

This formulation is henceforth referred to as L1-Tucker2, a name deriving by the special case of $N = 3$ (joint collection of 2-way matrices). Certainly, L1-Tucker2 can be expressed as L1-Tucker in (2.6), with the additional constraint $\mathbf{U}_N = \mathbf{I}_{D_N}$, since $\sum_{i=1}^{D_N} \left\| \boldsymbol{\mathcal{X}}_i \times_{n\in[N-1]} \mathbf{U}_n^\top \right\|_1 = \left\| \boldsymbol{\mathcal{X}} \times_{n\in[N-1]} \mathbf{U}_n^\top \times_N \mathbf{I}_{D_n} \right\|_1$. Conversely, L1-Tucker can be trivially written as L1-Tucker2, since

[1] The change of the projection norm from L2 in (2.1) to L1 in (2.6) should not be confused with the standard L1-norm regularization approach that is commonly employed to minimization imposed sparsity [96].

$\left\|\boldsymbol{\mathcal{X}} \times_{n\in[M-1]} \mathbf{U}_n^\top\right\|_1 = \sum_{i=1}^{D_M} \left\|\boldsymbol{\mathcal{Y}}_i \times_{n\in[M-1]} \mathbf{U}_n^\top\right\|_1$, for $M = N+1$, $D_M = 1$, and $\boldsymbol{\mathcal{Y}}_1 = \boldsymbol{\mathcal{X}}$.

2.4 Contribution 1: Exact Solution to Rank-1 L1-norm Tucker2

For the special case that $N = 3$, $\boldsymbol{\mathcal{X}}$ can, w.l.o.g., be treated as a collection of N_s real-valued matrices of equal size, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_s} \in \mathbb{R}^{D\times M}$. For any rank $d \leq \min\{D, M\}$, a *Tucker2* decomposition strives to jointly analyze $\{\mathbf{X}_i\}_{i=1}^{N_s}$, by maximizing $\sum_{i=1}^{N_s} \|\mathbf{U}^\top \mathbf{X}_i \mathbf{V}\|_F^2$ over $\mathbf{U} \in \mathbb{R}^{D\times d}$ and $\mathbf{V} \in \mathbb{R}^{M\times d}$, such that $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_d$; then, $\mathbf{X}_i$ is low-rank approximated as $\mathbf{U}\mathbf{U}^\top\mathbf{X}_i\mathbf{V}\mathbf{V}^\top$. Among other methods in the tensor-processing literature, Tucker2 coincides with Multilinear PCA [97] (for zero-centered matrices) and the Generalized Low-Rank Approximation of Matrices (GLRAM) [28]. Clearly, for $N_s = 1$, Tucker2 simplifies to the rank-d approximation of matrix $\mathbf{X}_1 \in \mathbb{R}^{D\times M}$, solved by means of the familiar singular-value decomposition (SVD) [31]; i.e., the optimal arguments $\mathbf{U}$ and $\mathbf{V}$ are built by the d left-hand and right-hand singular vectors of $\mathbf{X}_1$, respectively.

To counteract against the impact of any outliers in $\{\mathbf{X}_i\}_{i=1}^{N_s}$, in this work, we consider the L1-norm-based Tucker2 reformulation

$$\underset{\substack{\mathbf{U}\in\mathbb{R}^{D\times d};\ \mathbf{U}^\top\mathbf{U}=\mathbf{I}_d \\ \mathbf{V}\in\mathbb{R}^{M\times d};\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_d}}{\text{maximize}} \sum_{i=1}^{N_s} \|\mathbf{U}^\top\mathbf{X}_i\mathbf{V}\|_1. \tag{2.8}$$

The problem in (2.8) was studied in [52] under the title L1-Tensor Principal-Component Analysis (TPCA-L1).[2] Authors in [52] presented an approximate algorithm for its solution which they employed for image reconstruction. To date, (2.8) has not been solved exactly in the literature, even for the special case of rank-1 approximation –i.e., $d = 1$. We deliver, for the first time, the exact solution to L1-Tucker2 for $d = 1$, by means of two novel algorithms. In addition, we provide numerical studies that demonstrate the outlier-resistance of exact L1-Tucker2, and its superiority (in joint-matrix decomposition and reconstruction) over L2-norm-based (standard) Tucker2, GLRAM, TPCA-L1, PCA, and L1-PCA.

We commence our solution by showing how L1-Tucker2 ($d = 1$) can be reformulated as a combinatorial problem.

[2] In this work, we refer to the problem as L1-Tucker2, so as to highlight its connection with the Tucker2 formulation (instead of the general TUCKER formulation).

Reformulation Into Combinatorial Optimization

For rank $d = 1$, L1-Tucker2 in (2.8) takes the form

$$\underset{\mathbf{u}\in\mathbb{R}^{D\times 1};\ \mathbf{v}\in\mathbb{R}^{M\times 1};\ \|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1}{\text{maximize}} \sum_{i=1}^{N_s} |\mathbf{u}^\top \mathbf{X}_i \mathbf{v}| \tag{2.9}$$

First, we focus on the absolute value in (2.9) and notice that, for any $\mathbf{a} \in \mathbb{R}^{N_s}$, $\sum_{i=1}^{N_s} |a_i| = \sum_{i=1}^{N_s} \operatorname{sgn}(a_i)\, a_i = \operatorname{sgn}(\mathbf{a})^\top \mathbf{a} = \max_{\mathbf{b}\in\{\pm 1\}^{N_s}} \mathbf{b}^\top \mathbf{a}$, where $\operatorname{sgn}(\cdot)$ returns the $\{\pm 1\}$-sign of its (vector) argument. In view of the above, Lemma 2.1 follows.

Lemma 2.1. *For any given $\mathbf{u} \in \mathbb{R}^D$ and $\mathbf{v} \in \mathbb{R}^M$, it holds that*

$$\sum_{i=1}^{N_s} |\mathbf{u}^\top \mathbf{X}_i \mathbf{v}| = \max_{\mathbf{b}\in\{\pm 1\}^{N_s}} \mathbf{u}^\top \left(\sum_{i=1}^{N_s} b_i \mathbf{X}_i \right) \mathbf{v}. \tag{2.10}$$

The maximum in (2.10) is attained for $\mathbf{b} = [\operatorname{sgn}\left(\mathbf{u}^\top \mathbf{X}_1 \mathbf{v}\right), \operatorname{sgn}\left(\mathbf{u}^\top \mathbf{X}_2 \mathbf{v}\right), \ldots, \operatorname{sgn}\left(\mathbf{u}^\top \mathbf{X}_{N_s} \mathbf{v}\right)]^\top$.

In addition, the following well-known Lemma 2.2 derives by the matrix-approximation optimality of SVD [31].

Lemma 2.2. *For any given $\mathbf{b} \in \{\pm 1\}^{N_s}$, it holds that*

$$\max_{\substack{\mathbf{u}\in\mathbb{R}^{D\times 1};\ \|\mathbf{u}\|_2=1 \\ \mathbf{v}\in\mathbb{R}^{M\times 1};\ \|\mathbf{v}\|_2=1}} \mathbf{u}^\top \left(\sum_{i=1}^{N_s} b_i \mathbf{X}_i \right) \mathbf{v} = \sigma_{\max} \left(\sum_{i=1}^{N_s} b_i \mathbf{X}_i \right) \tag{2.11}$$

where $\sigma_{\max}(\cdot)$ returns the highest singular value of its matrix argument. The maximum in (2.11) is attained if $\mathbf{u}$ and $\mathbf{v}$ are the left-hand and right-hand dominant singular vectors of $\sum_{i=1}^{N_s} b_i \mathbf{X}_i$, respectively.

To compact our notation, we concatenate $\{\mathbf{X}_i\}_{i=1}^{N_s}$ into $\mathbf{X} \overset{\triangle}{=} [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{N_s}] \in \mathbb{R}^{D\times M N_s}$. Then, for any $\mathbf{b} \in \{\pm 1\}^{N_s}$, it holds $\sum_{i=1}^{N_s} b_i \mathbf{X}_i = \mathbf{X}(\mathbf{b} \otimes \mathbf{I}_M)$, where $\otimes$ denotes the Kronecker matrix product [98]. Then, in view of Lemma 2.1 and Lemma 2.2, we can rewrite the L1-Tucker2 in (2.9) as

$$\max_{\substack{\mathbf{u}\in\mathbb{R}^{D\times 1};\ \|\mathbf{u}\|_2=1\\ \mathbf{v}\in\mathbb{R}^{M\times 1};\ \|\mathbf{v}\|_2=1}} \sum_{i=1}^{N_s} |\mathbf{u}^\top \mathbf{X}_i \mathbf{v}| \tag{2.12}$$

$$= \max_{\substack{\mathbf{b}\in\{\pm 1\}^{N_s}\\ \mathbf{u}\in\mathbb{R}^{D\times 1};\ \|\mathbf{u}\|_2=1\\ \mathbf{v}\in\mathbb{R}^{M\times 1};\ \|\mathbf{v}\|_2=1}} \mathbf{u}^\top \left(\mathbf{X}(\mathbf{b}\otimes \mathbf{I}_M)\right)\mathbf{v} \tag{2.13}$$

$$= \max_{\mathbf{b}\in\{\pm 1\}^{N_s}} \sigma_{\max}\left(\mathbf{X}(\mathbf{b}\otimes \mathbf{I}_M)\right). \tag{2.14}$$

It is clear that (2.14) is a combinatorial problem over the size-2^{N_s} feasibility set $\{\pm 1\}^{N_s}$. The following Proposition 2.1 derives straightforwardly from Lemma 2.1, Lemma 2.2, and (2.12)-(2.14) and concludes our transformation of (2.9) into a combinatorial problem.

Proposition 2.1. *Let $\mathbf{b}_{\text{opt}}$ be a solution to the combinatorial*

$$\underset{\mathbf{b}\in\{\pm\}^{N_s}}{\text{maximize}}\sigma_{\max}(\mathbf{X}(\mathbf{b}\otimes \mathbf{I}_M)) \tag{2.15}$$

and denote by $\mathbf{u}_{\text{opt}} \in \mathbb{R}^D$ and $\mathbf{v}_{\text{opt}} \in \mathbb{R}^M$ the left- and right-hand singular vectors of $\mathbf{X}(\mathbf{b}_{\text{opt}} \otimes \mathbf{I}_M) \in \mathbb{R}^{D\times M}$, respectively. Then, $(\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}})$ is an optimal solution to (2.9). Also, $\mathbf{b}_{\text{opt}} = [\text{sgn}\left(\mathbf{u}_{\text{opt}}^\top \mathbf{X}_1 \mathbf{v}_{\text{opt}}\right), \ldots, \text{sgn}\left(\mathbf{u}_{\text{opt}}^\top \mathbf{X}_{N_s}\mathbf{v}_{\text{opt}}\right)]^\top$ and $\sum_{i=1}^{N_s} |\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}}| = \mathbf{u}_{\text{opt}}^\top(\mathbf{X}(\mathbf{b}_{\text{opt}}\otimes \mathbf{I}_M))\mathbf{v}_{\text{opt}} = \sigma_{\max}\left(\mathbf{X}(\mathbf{b}_{\text{opt}}\otimes\mathbf{I}_M)\right)$. In the special case that $\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}} = 0$, for some $i \in \{1, 2, \ldots, N_s\}$, $[\mathbf{b}_{\text{opt}}]_i$ can be set to $+1$, having no effect to the metric of (2.15).

Given $\mathbf{b}_{\text{opt}}$, $(\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}})$ are obtained by SVD of $\mathbf{X}(\mathbf{b}_{\text{opt}} \otimes \mathbf{I}_M)$. Thus, by Proposition 2.1, the solution to L1-Tucker2 for low-rank $d = 1$ is obtained by the solution of the combinatorial problem (2.15) and a D-by-M SVD.

Connection to L1-PCA and Hardness

In the sequel, we show that for $M = 1$ and $d = 1$, L1-Tucker2 in (2.9) simplifies to L1-PCA [53, 54, 58]. Specifically, for $M = 1$, matrix $\mathbf{X}_i$ is a $D \times 1$ vector, satisfying $\mathbf{X}_i = \mathbf{x}_i \overset{\triangle}{=} \text{vec}(\mathbf{X}_i)$, and (2.9) can be rewritten as

$$\max_{\mathbf{u}\in\mathbb{R}^D;\ v\in\mathbb{R};\ \|\mathbf{u}\|_2=|v|=1} \sum_{i=1}^{N_s} |\mathbf{u}^\top \mathbf{x}_i v|. \tag{2.16}$$

It is clear that for every $\mathbf{u}$, an optimal value for v is trivially $v = 1$ (or, equivalently, $v = -1$); thus, for $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_s}] \in \mathbb{R}^{D \times N_s}$, (2.16) becomes

$$\max_{\mathbf{u} \in \mathbb{R}^D;\ \|\mathbf{u}\|_2 = 1} \sum_{i=1}^{N_s} |\mathbf{u}^\top \mathbf{x}_i| = \max_{\mathbf{u} \in \mathbb{R}^D;\ \|\mathbf{u}\|_2 = 1} \|\mathbf{X}^\top \mathbf{u}\|_1, \tag{2.17}$$

which is the exact formulation of the well-studied L1-PCA problem [53, 54, 58]. We notice also that for $M = 1$ the combinatorial optimization (2.15) in Proposition 1 becomes

$$\max_{\mathbf{b} \in \{\pm 1\}^{N_s}} \sigma_{\max}(\mathbf{X}(\mathbf{b} \otimes 1)) = \max_{\mathbf{b} \in \{\pm 1\}^{N_s}} \|\mathbf{X}\mathbf{b}\|_2, \tag{2.18}$$

since the maximum singular-value of a vector coincides with its Euclidean norm, which is in accordance to the L1-PCA analysis in [54, 58]. Based of the equivalence of L1-PCA to (2.18), [54] has proven that L1-PCA of $\mathbf{X}$ is formally *NP*-hard in N_s, for jointly asymptotic N_s and rank($\mathbf{X}$). Thus, by its equivalence to L1-PCA for $d = 1$ and $M = 1$, L1-Tucker2 is also *NP*-hard in N_s, for jointly asymptotic N_s and rank($\mathbf{X}$).

Exact Algorithm 1: Exhaustive Search

Proposition 2.1 shows how the solution to (2.9) can be obtained through the solution to the combinatorial problem in (2.15). Our first exact algorithm solves (2.15) straightforwardly by an exhaustive search over its feasibility set. In fact, noticing that $\sigma_{max}(\cdot)$ is invariant to negations of its matrix argument, we obtain a solution $\mathbf{b}_{\text{opt}}$ to (2.15) by an exhaustive search in the size-2^{N_s-1} set $\mathcal{B}_{\text{ex}} = \{\mathbf{b} \in \{\pm 1\}^{N_s} : \ b_1 = 1\}$. For every value that $\mathbf{b}$ takes in $\mathcal{B}_{\text{ex}}$, we conduct SVD to $\mathbf{X}(\mathbf{b} \otimes \mathbf{I}_M)$ to calculate $\sigma_{\max}(\mathbf{X}(\mathbf{b} \otimes \mathbf{I}_M))$, with cost $\mathcal{O}(\min\{D, M\}DM)$ [31]. Since it entails 2^{N_s-1} SVD calculations, the cost of this exhaustive-search algorithm is $\mathcal{O}(2^{N_s-1}\min\{D, M\}DM)$; thus, it is exponential to the number of jointly processed matrices, N_s, and at most quadratic to the matrix sizes, D and M.

Exact Algorithm 2: Search With Cost Polynomial in N_s

In the sequel, we focus on the case where N_s is low-bounded by the constant DM and present an algorithm that solves (2.9) with polynomial cost in N_s. $DM < N_s$ emerges as a case of interest in signal processing applications when $\{\mathbf{X}_i\}_{i=1}^{N_s}$ are measurements of a $D \times M$ fixed-size sensing system (e.g., $D \times M$ images). By Proposition 2.1, for the optimal solutions $\mathbf{b}_{\text{opt}}$ and $(\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}})$ of

(2.15) and (2.9), respectively, it holds

$$\mathbf{b}_{\text{opt}} = [\text{sgn}\left(\mathbf{v}_{\text{opt}}^\top \mathbf{X}_1^\top \mathbf{u}_{\text{opt}}\right), \ldots, \text{sgn}\left(\mathbf{v}_{\text{opt}}^\top \mathbf{X}_{N_s}^\top \mathbf{u}_{\text{opt}}\right)]^\top, \tag{2.19}$$

with $\text{sgn}\left(\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}}\right) = +1$, if $\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}} = 0$. In addition, for every $i \in \{1, 2, \ldots, N_s\}$, we find that

$$\mathbf{v}_{\text{opt}}^\top \mathbf{X}_i^\top \mathbf{u}_{\text{opt}} = \text{Tr}\left(\mathbf{X}_i^\top \mathbf{u}_{\text{opt}} \mathbf{v}_{\text{opt}}^\top\right) = \mathbf{x}_i^\top (\mathbf{v}_{\text{opt}} \otimes \mathbf{u}_{\text{opt}}). \tag{2.20}$$

Therefore, defining $\mathbf{Y} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_s}] \in \mathbb{R}^{DM \times N_s}$, (2.19) can be rewritten as

$$\mathbf{b}_{\text{opt}} = \text{sgn}\left(\mathbf{Y}^\top (\mathbf{v}_{\text{opt}} \otimes \mathbf{u}_{\text{opt}})\right). \tag{2.21}$$

Consider now that $\mathbf{Y}$ is of some rank $\rho \leq \min\{DM, N_s\}$ and admits SVD $\mathbf{Y} \overset{svd}{=} \mathbf{QSW}$, where $\mathbf{Q}^\top \mathbf{Q} = \mathbf{W}\mathbf{W}^\top = \mathbf{I}_\rho$ and $\mathbf{S}$ is the $\rho \times \rho$ diagonal matrix that carries the ρ non-zero singular-values of $\mathbf{Y}$. Defining $\mathbf{p}_{\text{opt}} \overset{\triangle}{=} \mathbf{S}^\top \mathbf{Q}^\top (\mathbf{v}_{\text{opt}} \otimes \mathbf{u}_{\text{opt}})$, (2.21) can be rewritten as

$$\mathbf{b}_{\text{opt}} = \text{sgn}\left(\mathbf{W}^\top \mathbf{p}_{\text{opt}}\right). \tag{2.22}$$

In view of (2.22) and since $\text{sgn}(\cdot)$ is invariant to positive scalings of its vector argument, an optimal solution to (2.15), $\mathbf{b}_{\text{opt}}$, can be found in the binary set

$$\mathcal{B} = \{\mathbf{b} \in \{\pm 1\}^{N_s} : \ \mathbf{b} = \text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right), \ \mathbf{c} \in \mathbb{R}^\rho\}. \tag{2.23}$$

Certainly, by definition, (2.23) is a subset of $\{\pm 1\}^{N_s}$ and, thus, has finite size upper bounded by 2^{N_s}. This, in turn, implies that there exist instances of $\mathbf{c} \in \mathbb{R}^\rho$ that yield the same value in $\text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right)$. Below, we delve into this observation to build a tight superset of $\mathcal{B}$ that has polynomial size in N_s, under the following mild "general position" assumption [99].

Assumption 2.1. *For every $\mathcal{I} \subset \{1, 2, \ldots, N_s\}$ with $|\mathcal{I}| = \rho - 1$, it holds that rank$([\mathbf{W}]_{:,\mathcal{I}}) = \rho - 1$; i.e., any collection of $\rho - 1$ columns of $\mathbf{W}$ are linearly independent.*

For any $i \in \{1, 2, \ldots, N_s\}$, define $\mathbf{w}_i \overset{\triangle}{=} [\mathbf{W}]_{:,i}$ and denote by $\mathcal{N}_i$ the nullspace of $\mathbf{w}_i$. Then, for every $\mathbf{c} \in \mathcal{N}_i$, the (non-negative) angle between $\mathbf{c}$ and $\mathbf{w}_i$, $\phi(\mathbf{c}, \mathbf{w}_i)$, is equal to $\frac{\pi}{2}$ and, accordingly, $\mathbf{w}_i^\top \mathbf{c} = \|\mathbf{c}\|_2 \|\mathbf{w}_i\|_2 \cos(\phi(\mathbf{c}, \mathbf{w}_i)) = 0$. Clearly, the *hyperplane* $\mathcal{N}_i$ partitions $\mathbb{R}^\rho$ in two non-overlapping *halfspaces*, $\mathcal{H}_i^+$ and $\mathcal{H}_i^-$ [100], such that $\text{sgn}\left(\mathbf{c}^\top \mathbf{w}_i\right) = +1$ for every $\mathbf{c} \in \mathcal{H}_i^+$ and $\text{sgn}\left(\mathbf{c}^\top \mathbf{w}_i\right) = -1$ for every $\mathbf{c} \in \mathcal{H}_i^-$. In accordance with Proposition 2.1, we consider that $\mathcal{H}_i^+$ is a closed set that includes its boundary $\mathcal{N}_i$, whereas $\mathcal{H}_i^-$ is open and does not overlap with $\mathcal{N}_i$. In view of these

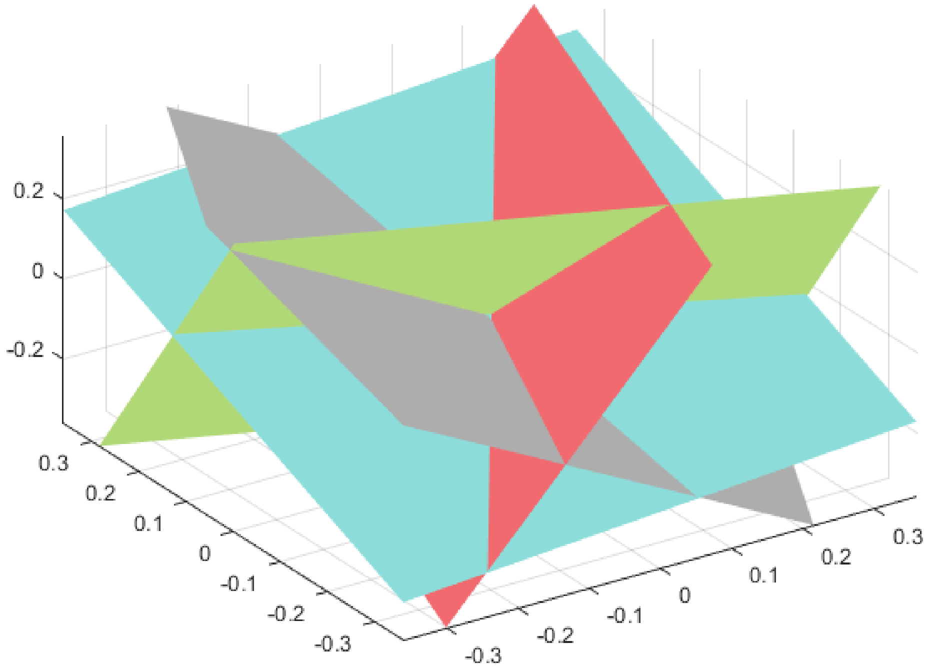


Figure 2.3. For $\rho = 3$ and $N_s = 4$, we draw $\mathbf{W} \in \mathbb{R}^{\rho \times N}$, such that $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_3$ and Assumption 1 holds true. Then, we plot the nullspaces of all 4 columns of $\mathbf{W}$ (colored planes). We observe that the planes partition $\mathbb{R}^3$ into $K = 2(\binom{3}{0} + \binom{3}{1} + \binom{3}{2}) = 2(1+3+3) = 14$ coherent cells (i.e., 7 visible cells above the cyan hyperplane and 7 cells below.)

definitions, we proceed with the following illustrative example.

Consider some $\rho > 2$ and two column indices $m < i \in \{1, 2, \ldots, N_s\}$. Then, hyperplanes $\mathcal{N}_m$ and $\mathcal{N}_i$ divide $\mathbb{R}^\rho$ in the halfspace pairs $\{\mathcal{H}_m^+, \mathcal{H}_m^-\}$ and $\{\mathcal{H}_i^+, \mathcal{H}_i^-\}$, respectively. By Assumption 2.1,[3] each one of the two halfspaces defined by $\mathcal{N}_m$ will intersect with both halfspaces defined by $\mathcal{N}_i$, forming the four halfspace-intersection "cells" $\mathcal{C}_1 = \mathcal{H}_m^+ \cap \mathcal{H}_i^+$, $\mathcal{C}_2 = \mathcal{H}_m^+ \cap \mathcal{H}_i^-$, $\mathcal{C}_3 = \mathcal{H}_m^- \cap \mathcal{H}_i^-$, $\mathcal{C}_4 = \mathcal{H}_m^- \cap \mathcal{H}_i^+$. It is now clear that, for any $k \in \{1, 2, 3, 4\}$, $[\mathrm{sgn}\left([\mathbf{W}]^\top \mathbf{c}\right)]_{m,i}$ is the same for every $\mathbf{c} \in \mathcal{C}_k$. For example, for every $\mathbf{c} \in \mathcal{C}_2$, it holds that $[\mathrm{sgn}\left([\mathbf{W}]^\top \mathbf{c}\right)]_m = +1$ and $[\mathrm{sgn}\left([\mathbf{W}]^\top \mathbf{c}\right)]_i = -1$.

Next, we go one step further and consider the arrangement of all N hyperplanes $\{\mathcal{N}_i\}_{i=1}^{N_s}$. Similar to our discussion above, these hyperplanes partition $\mathbb{R}^\rho$ in K cells $\{\mathcal{C}_k\}_{k=1}^{K}$, where K depends on ρ and N_s. Formally, for every k, the k-th halfspace-intersection set is defined as

$$\mathcal{C}_k \triangleq \bigcap_{i \in \mathcal{I}_k^+} \mathcal{H}_i^+ \bigcap_{m \in \mathcal{I}_k^{-1}} \mathcal{H}_m^-, \tag{2.24}$$

[3] If $\mathbf{w}_m$ and $\mathbf{w}_i$ are linearly independent, then $\mathcal{N}_m$ and $\mathcal{N}_i$ intersect but do not coincide.

for complementary index sets $\mathcal{I}_k^+$ and $\mathcal{I}_k^-$ that satisfy $\mathcal{I}_k^+ \cap \mathcal{I}_k^- = \emptyset$ and $\mathcal{I}_k^+ \cup \mathcal{I}_k^- = \{1, 2, \ldots, N_s\}$ [101, 102]. By the definition in (2.24), and in accordance with our example above, every $\mathbf{c} \in \mathcal{C}_k$ lies in the same intersection of halfpsaces and, thus, yields the exact same value in sgn $\left(\mathbf{W}^\top \mathbf{c}\right)$. Specifically, for every $\mathbf{c} \in \mathcal{C}_k$, it holds that

$$\left[\text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right)\right]_i = \text{sgn}\left(\mathbf{w}_i^\top \mathbf{c}\right) = \begin{cases} +1, & i \in \mathcal{I}_k^+ \\ -1, & i \in \mathcal{I}_k^- \end{cases}. \tag{2.25}$$

In view of (2.25), for every $k \in \{1, 2, \ldots, K\}$ and any $\mathbf{c} \in \mathcal{C}_k$, we define the "signature" of the k-th cell $\mathbf{b}_k \triangleq \text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right)$. Moreover, we observe that $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ for every $k \neq l$ and that $\cup_{k=1}^K \mathcal{C}_k = \mathbb{R}^\rho$. By the above observations and definitions, (2.23) can be rewritten as

$$\mathcal{B} = \bigcup_{k=1}^{K} \{\text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right) : \mathbf{c} \in \mathcal{C}_k\} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_K\}. \tag{2.26}$$

Importantly, in [101, 103], it was shown that the exact number of coherent cells formed by the nullspaces of N_s points in $\mathbb{R}^\rho$ that are in general position (under Assumption 2.1) is exactly

$$K = 2\sum_{j=0}^{\rho-1} \binom{N_s - 1}{j} \leq 2^{N_s}, \tag{2.27}$$

with equality in (2.27) if and only if $\rho = N_s$. Accordingly, per (2.27), the cardinality of $\mathcal{B}$ in (2.23) is equal to $|\mathcal{B}| = 2\sum_{j=0}^{\rho-1}\binom{N_s-1}{j}$. For clarity, in Fig. 2.3, we plot the nullspaces (colored planes) of the columns of arbitrary $\mathbf{W} \in \mathbb{R}^{3\times 4}$ that satisfies both $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_3$ and Assumption 2.1. It is interesting that exactly $K = 14 < 2^4 = 16$ coherent cells emerge by the intersection of the formed halfspaces. In the sequel, we rely on (2.26) to develop a conceptually simple method for computing a tight superset of the cell signatures in $\mathcal{B}$.

Under Assumption 2.1, for any $\mathcal{I} \subseteq \{1, 2, \ldots, N_s\}$ with $|\mathcal{I}| = \rho - 1$, the hyperplane intersection $\mathcal{V}_\mathcal{I} \triangleq \cap_{i\in\mathcal{I}}\mathcal{N}_i$ is a line (1-dimensional subspace) in $\mathbf{R}^\rho$. By its definition, this line is the verge between all cells that are jointly bounded by the $\rho - 1$ hyperplanes in $\{\mathcal{N}_i\}_{i\in\mathcal{I}}$. Consider now a vector $\mathbf{c} \in \mathbb{R}^\rho$ that crosses over the verge $\mathcal{V}_\mathcal{I}$ (at any point other than $\mathbf{0}_\rho$). By this crossing, the value of $[\text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right)]_\mathcal{I}$ will change so that sgn $\left(\mathbf{W}^\top \mathbf{c}\right)$ adjusts to the signature of the new cell to which $\mathbf{c}$ just entered. At the same time, a crossing over $\mathcal{V}_\mathcal{I}$ cannot be simultaneously over any of the hyperplanes in $\{\mathcal{N}_i\}_{i\in\mathcal{I}^c}$, for $\mathcal{I}^c \triangleq \{1, 2, \ldots, N_s\} \setminus \mathcal{I}$; this is because, under Assumption 2.1, it is only at $\mathbf{0}_\rho$ that more than $\rho - 1$ hyperplanes can intersect. Therefore, it is clear that $[\text{sgn}\left(\mathbf{W}^\top \mathbf{c}\right)]_{\mathcal{I}^c}$ will remain invariant during this crossing and, in fact, equal to $[\text{sgn}\left(\mathbf{W}^\top \mathbf{v}\right)]_{\mathcal{I}^c}$, for any $\mathbf{v} \in \mathcal{V}_\mathcal{I}$ with

```
Input: $\{\mathbf{X}_i\}_{i=1}^{N_s}$
Output: $\mathbf{b}_{\text{opt}} \leftarrow \mathbf{b}_t$, $\mathbf{u}_{\text{opt}} \leftarrow \mathbf{u}$, and $\mathbf{v}_{\text{opt}} \leftarrow \mathbf{v}$

$\mathbf{Y} \leftarrow [\text{vec}(\mathbf{X}_1), \text{vec}(\mathbf{X}_2), \ldots, \text{vec}(\mathbf{X}_{N_s})]$
$(\mathbf{Q}, \mathbf{S}_{d \times d}, \mathbf{W}) \leftarrow \text{svd}(\mathbf{Y})$, $m_t \leftarrow 0$
For every $\mathcal{I} \subseteq \{1, 2, \ldots, N_s\}$, $|\mathcal{I}| = d - 1$
    Build $\mathcal{B}_{\mathcal{I}}$ in (2.28)
    For every $\mathbf{b} \in \mathcal{B}_{\mathcal{I}}$
        $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) \leftarrow \text{svd}(\mathbf{X}(\mathbf{b} \otimes \mathbf{I}_M))$
        $m \leftarrow \max\{\text{diag}(\mathbf{\Sigma})\}$
         if $m > m_t$,
            $m_t \leftarrow m$, $\mathbf{b}_t \leftarrow \mathbf{b}$, $\mathbf{u} \leftarrow [\mathbf{U}]_{:,1}$, $\mathbf{v} \leftarrow [\mathbf{V}]_{:,1}$
```

Algorithm 2.1 Polynomial in N_s algorithm for the exact solution of rank-1 L1-Tucker2 in (2.9), with cost $\mathcal{O}(N_s^{\rho+1})$.

$\mathbf{v}^\top \mathbf{c} > 0$. In view of the above, for any $\mathbf{v} \in \mathcal{V}_{\mathcal{I}} \setminus \mathbf{0}_\rho$, the set

$$\mathcal{B}_{\mathcal{I}} \overset{\triangle}{=} \{\mathbf{b} \in \{\pm 1\}^{N_s} : [\mathbf{b}]_{\mathcal{I}^c} = [\text{sgn}\left(\mathbf{W}^\top \mathbf{v}\right)]_{\mathcal{I}^c}\} \tag{2.28}$$

contains the signatures of all sets that are bounded by the verge $\mathcal{V}_{\mathcal{I}}$. Moreover, it has been shown (see, e.g., [103]) that, for every cell, there exists at least one such verge that bounds it. Therefore, it derives that the set

$$\mathcal{B}_{\text{pol}} = \bigcup_{\mathcal{I} \subset \{1,2,\ldots,N_s\};\ |\mathcal{I}| = \rho - 1} \mathcal{B}_{\mathcal{I}} \tag{2.29}$$

includes all cell signatures and, thus, is a superset of $\mathcal{B}$. We notice that, for every $\mathcal{I}$, $\mathcal{B}_{\mathcal{I}}$ has size $2^{\rho-1}$. Since $\mathcal{I}$ can take $\binom{N_s}{\rho-1}$ distinct values, we find that $\mathcal{B}_{\text{pol}}$ is upper bounded by $2^{\rho-1}\binom{N_s}{\rho-1}$. Thus, both $|\mathcal{B}_{\text{pol}}|$ and $|\mathcal{B}|$ are polynomial, in the order of $\mathcal{O}(N_s^{\rho-1})$.

Practically, for every $\mathcal{I}$, $\mathbf{v}$ can be calculated by Gram-Schmidt orthogonalization of $[\mathbf{W}]_{:,\mathcal{I}}$ with cost $\mathcal{O}(\rho^3)$. Keeping the dominant terms, the construction of $\mathcal{B}_{\text{pol}}$ costs $\mathcal{O}(N_s^{\rho-1})$ and can be parallelized in $\binom{N_s}{\rho-1}$ processes. Then, testing every entry of $\mathcal{B}_{\text{pol}}$ for optimality in (2.15) costs an additional $\mathcal{O}(N_s)$. Thus, the overall cost of our second algorithm, taking also into account the $\mathcal{O}(N_s)$ (for constant DM) SVD cost for the formation of $\mathbf{W}$, is $\mathcal{O}(N_s^{\rho})$. The presented algorithm is summarized in Algorithm 2.1.

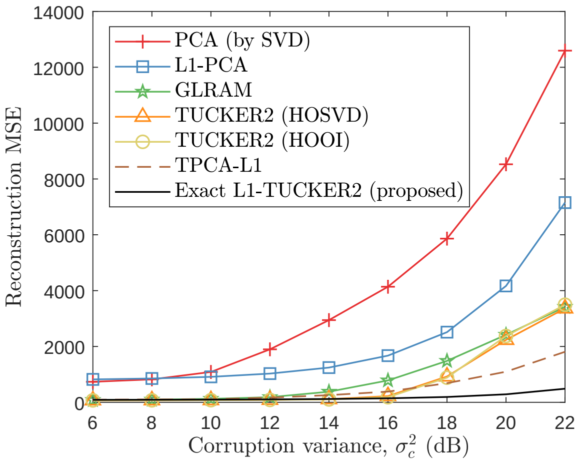


Figure 2.5. Reconstruction MSE versus corruption variance σ_c^2 (dB).

2.4.1 Numerical Studies

Consider $\{\mathbf{X}_i\}_{i=1}^{14}$ such that $\mathbf{X}_i = \mathbf{A}_i + \mathbf{N}_i \in \mathbb{R}^{20\times 20}$ where $\mathbf{A}_i = b_i \mathbf{u}\mathbf{v}^\top$ and $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$, $b_i \sim \mathcal{N}(0, 49)$, and each entry of $\mathbf{N}_i$ is additive white Gaussian noise (AWGN) from $\mathcal{N}(0, 1)$. We consider that $\mathbf{A}_i$ is the rank-1 useful data in $\mathbf{X}_i$ that we want to reconstruct, by joint analysis (Tucker2-type) of $\{\mathbf{X}_i\}_{i=1}^{14}$. By irregular corruption, 30 entries in 2 out of the 14 matrices (i.e., 60 entries out of the total 5600 entries in $\{\mathbf{X}_i\}_{i=1}^{14}$) have been further corrupted additively by noise from $\mathcal{N}(0, \sigma_c^2)$. To reconstruct $\{\mathbf{A}_i\}_{i=1}^{14}$ from $\{\mathbf{X}_i\}_{i=1}^{14}$, we follow one of the two approaches below.

In the first approach, we vectorize the matrix samples and perform standard matrix analysis. That is, we obtain the first ($d = 1$) principal component (PC) of $[\mathrm{vec}(\mathbf{X}_1), \mathrm{vec}(\mathbf{X}_2), \ldots, \mathrm{vec}(\mathbf{X}_{N_s})]$, $\mathbf{q}$. Then, for every i, we approximate $\mathbf{A}_i$ by $\hat{\mathbf{A}}_i = \mathrm{mat}(\mathbf{q}\mathbf{q}^\top \mathbf{a}_i)$, where $\mathrm{mat}(\cdot)$ reshapes its vector argument into a 20×20 matrix, in accordance with $\mathrm{vec}(\cdot)$. In the second approach, we process the samples in their natural form, as matrices, analyzing them by Tucker2. If $(\mathbf{u}, \mathbf{v})$ is the Tucker2 solution pair, then we approximate $\mathbf{A}_i$ by $\hat{\mathbf{A}}_i = \mathbf{u}\mathbf{u}^\top \mathbf{X}_i \mathbf{v}\mathbf{v}^\top$. For the first approach, we obtain $\mathbf{q}$ by PCA (i.e., SVD) and L1-PCA [54]. For the second approach, we conduct Tucker2 by HOSVD [17], HOOI [32], GLRAM [28], TPCA-L1 [52], and the proposed exact L1-Tucker2. Then, for each reconstruction method, we measure the mean of the squared error $\sum_{i=1}^{14} \|\mathbf{A}_i - \hat{\mathbf{A}}_i\|_F^2$ over 1000 independent realizations for corruption variance $\sigma_c^2 = 6, 8, \ldots, 22$dB. In Fig. 2.5, we plot the reconstruction mean squared error (MSE) for every method, versus σ_c^2. We observe that PCA and L1-PCA exhibit the highest MSE due to the vectorization operation (L1-PCA outperforms PCA clearly, across all values of σ_c^2). Then, all Tucker2-type methods perform similarly well when σ_c^2

is low. As the outlier variance σ_c^2 increases, the performance of L2-norm-based Tucker2 (HOSVD, HOOI) and GLRAM deteriorates severely. On the other hand, the L1-norm-based TPCA-L1 exhibits some robustness. The proposed exact L1-Tucker2 maintains the sturdiest resistance against the corruption, outperforming its counterparts across the board.

2.5 Contribution 2: L1-norm HOVSD and L1-norm HOOI for L1-Tucker

L1-Tucker in the general form of (2.6), for $N \geq 2$, has not been thoroughly studied to date. In this work, we present two algorithmic frameworks for the approximate solution of (2.6), which can also be modified to tackle L1-Tucker2 of N-way tensors, as defined in (2.7).

2.5.1 L1-norm HOSVD Algorithm

The first proposed algorithm, L1-HOSVD, seeks to disjointly optimize the N bases. Specifically, for every $n \in [N]$, we set the mode-n basis $\mathbf{U}_n$ to the L1-PCA solution (exact or approximate) of the mode-n matrix unfolding $\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)$. That is, L1-HOSVD approximates the jointly-optimal mode-n basis in (2.6) by

$$\mathbf{U}_n^{\text{l1-hosvd}} = \underset{\mathbf{U} \in \mathbb{S}_{D_n \times d_n}}{\operatorname{argmax}} \left\| \mathbf{U}^\top \mathtt{mat}(\boldsymbol{\mathcal{X}}, n) \right\|_1 . \tag{2.30}$$

Clearly, (2.30) is an L1-PCA problem on $\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)$ and, thus, it can be solved exactly by means of the algorithms of [54], with cost $\mathcal{O}(P_n^{D_n d_n - d_n + 1})$. Other possible L1-PCA solvers were discussed in Section 2.2. As a tensor decomposition framework, L1-HOSVD allows for the use of any solver for (2.30), allowing for different performance/cost trade-offs. For the sake of computational efficiency, in the sequel we pursue the solution to (2.30) approximately, by means of a fixed-point iteration (FPI) algorithm. According to [54], it holds

$$\max_{\mathbf{U} \in \mathbb{S}_{D_n \times d_n}} \|\mathbf{U}^\top \mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\|_1 \tag{2.31}$$

$$= \max_{\substack{\mathbf{U} \in \mathbb{S}_{D_n \times d_n} \\ \mathbf{B} \in \{\pm 1\}^{P_n \times d_n}}} \operatorname{Tr}\left(\mathbf{U}^\top \mathtt{mat}(\boldsymbol{\mathcal{X}}, n) \mathbf{B}\right) \tag{2.32}$$

$$= \max_{\mathbf{B} \in \{\pm 1\}^{P_n \times d_n}} \|\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\mathbf{B}\|_*. \tag{2.33}$$

```
Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}, \{d_n\}_{n \in [N]}$
Output: $\{\mathbf{U}_n\}_{n \in [N]}$
Initialize $\{\mathbf{U}_n\}_{n \in [N]}$ by HOSVD of $\boldsymbol{\mathcal{X}}$ for $\{d_n\}_{n \in [N]}$
For $n = 1, 2, \ldots, N$
    Until convergence/termination
        $\mathbf{U}_n \leftarrow \Phi\left(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)\text{sgn}(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)^\top \mathbf{U}_n)\right)$
```

Algorithm 2.2 L1-HOSVD algorithm for L1-Tucker ($\texttt{L1HOSVD}(\boldsymbol{\mathcal{X}}, \{d_n\}_{n \in [N]})$).

For fixed $\mathbf{B}$, (2.32) is maximized by $\mathbf{U} = \Phi(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)\mathbf{B})$. At the same time, for fixed $\mathbf{U}$, (2.32) is maximized by $\mathbf{B} = \text{sgn}(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)^\top \mathbf{U})$. Accordingly, a solution to (2.30) can be pursued in an alternating fashion, as $\mathbf{B}_t = \text{sgn}(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)^\top \mathbf{U}_{t-1}) = \text{argmax}_{\mathbf{B} \in \{\pm 1\}^{P_n \times d_n}} \text{Tr}\left(\mathbf{U}_{t-1}^\top \texttt{mat}(\boldsymbol{\mathcal{X}}, n)\mathbf{B}\right)$ and $\mathbf{U}_t = \Phi(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)\mathbf{B}_t)$=$\text{argmax}_{\mathbf{U} \in \mathbb{S}_{D_n \times d_n}} \text{Tr}\left(\mathbf{U}^\top \texttt{mat}(\boldsymbol{\mathcal{X}}, n)\mathbf{B}_t\right)$, for $t = 1, 2, \ldots$, and arbitrary initialization $\mathbf{U}_0 \in \mathbb{S}_{D_n \times d_n}$. Interestingly, the alternating optimization above, can be rewritten in the compact FPI form

$$\mathbf{U}_t = \Phi\left(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)\text{sgn}(\texttt{mat}(\boldsymbol{\mathcal{X}}, n)^\top \mathbf{U}_{t-1})\right). \tag{2.34}$$

A proof of convergence for the recursion in (2.34) is offered in the Appendix A. If T_n is the index of the converging iteration, then $\mathbf{U}_n^{\text{l1-hosvd}}$ is approximated by $\mathbf{U}_{T_n}$. Similarly, the $N-1$ first bases $\{\mathbf{U}_n^{\text{l1-hosvd}}\}_{n=1}^{N-1}$ can be used as an approximate solution to L1-Tucker2 in (2.7).

Complexity of L1-HOSVD

For any given t and n, the computational cost of (2.34) is $\mathcal{O}(d_n P)$, where $P \overset{\triangle}{=} \prod_{m \in [N]} D_m$. In practice, we have observed that, for any n, it suffices to terminate iterations at a linear multiple of D_n. Thus, for any n, $\mathbf{U}_n^{\text{l1-hosvd}}$ is approximated with cost $\mathcal{O}(D_n d_n P)$. Accordingly, the total cost of L1-HOSVD is $\mathcal{O}(\max_{n \in [N]} d_n D_n P)$. Considering, for simplicity in presentation, that $D_n = D$ and $d_n = d$ for every n, then the complexity of L1-HOSVD can be rewritten as $\mathcal{O}(dD^{N+1})$.

A pseudocode of L1-HOSVD is offered in Algorithm 2.2.

```
Input: $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}, \{d_n\}_{n \in [N]}$
Output: $\{\mathbf{U}_n\}_{n \in [N]}$
Initialize $\{\mathbf{U}_n\}_{n \in [N]}$
Until termination/convergence
    For $n = 1, 2, \ldots, N$
        $\mathbf{A} \leftarrow \texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{m<n} \mathbf{U}_m{}^\top \times_{k>n} \mathbf{U}_k{}^\top, n\right)$
        Until termination/convergence
            $\mathbf{U}_n \leftarrow \Phi(\mathbf{A}\text{sgn}(\mathbf{A}^\top \mathbf{U}_n))$
```

Algorithm 2.3 L1-HOOI algorithm for L1-Tucker ($\texttt{L1HOOI}(\boldsymbol{\mathcal{X}}, \{d_n\}_{n\in[N]})$)

2.5.2 L1-norm HOOI Algorithm

Next, we present L1-HOOI, an alternative method for jointly optimizing the L1-Tucker bases. First, L1-HOOI is initialized to N feasible bases (e.g., those returned by L1-HOSVD). Then, it conducts a sequence of iterations across which it updates all bases such that the objective value of L1-Tucker increases. Thus, when initialized to the L1-HOSVD bases, L1-HOOI is guaranteed to outperform L1-HOSVD in the L1-Tucker metric. A detailed description of L1-HOOI follows.

First, we initialize $\{\mathbf{U}_n^{(0)} \in \mathbb{S}_{D_n \times d_n}\}_{n \in [N]}$; for instance, one can set $\mathbf{U}_n^{(0)} = \mathbf{U}_n^{\text{l1-hosvd}}$. Then, at the q-th iteration, $q = 1, 2, \ldots$, all N bases are successively optimized in order of increasing mode-index $n = 1, 2, \ldots, N$. Specifically, at a given iteration q and mode index n, we fix $\mathbf{U}_m^{(q)}$ for $m < n$ and $\mathbf{U}_k^{(q-1)}$ for $k > n$ and seek the mode-n basis $\mathbf{U}_n^{(q)}$ that maximizes the L1-Tucker metric. That is, for given (q, n), we update

$$\mathbf{U}_n^{(q)} = \underset{\mathbf{U} \in \mathbb{S}_{D_n \times d_n}}{\text{argmax}} \left\| \mathbf{U}^\top \mathbf{A}_n^{(q)} \right\|_1, \tag{2.35}$$

where $\mathbf{A}_n^{(q)} \triangleq \texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{m<n} \mathbf{U}_m^{(q)\top} \times_{k>n} \mathbf{U}_k^{(q-1)\top}, n\right) \in \mathbb{R}^{D_n \times p_n}$ and $p_n \triangleq \prod_{i \in [N] \setminus n} d_i$ for every n. We notice that, in contrast to (2.30), the metric of (2.35) involves the jointly optimized bases of the other modes. Similar to L1-HOSVD, an array of L1-PCA solvers, as discussed in Section 2.2, can be used for solving (2.35), attaining different performance/cost trade-offs. For simplicity in presentation, here we employ the FPI in (2.34). That is, for any (q, n), we set $\mathbf{U}_n^{(q)}$ to the converging argument of the sequence

$$\mathbf{U}_t = \Phi\left(\mathbf{A}_n^{(q)} \text{sgn}(\mathbf{A}_n^{(q)\top} \mathbf{U}_{t-1})\right), \tag{2.36}$$

where for every n, $\mathbf{U}_0 = \mathbf{U}_n^{(q-1)}$. A pseudocode of the proposed L1-HOOI method is offered in

Algorithm 2.3. Similar to L1-HOSVD, L1-HOOI can be used for the approximate solution of L1-Tucker2 simply by fixing $\mathbf{U}_N^{(q)}$ to $\mathbf{I}_{D_N}$ for every q. A convergence analysis of the L1-HOOI iterations is presented below.

Convergence of L1-HOOI

We commence our convergence analysis with Lemma 2.3, which shows that the q-th update of the mode-n basis increases the L1-Tucker metric.

Lemma 2.3. *For given $\{\mathbf{U}_m^{(q)}\}_{m<n}$ and $\{\mathbf{U}_k^{(q-1)}\}_{k>n}$ (and, thus, given $\mathbf{A}_n^{(q)}$), it holds that*

$$\left\| {\mathbf{U}_n^{(q)}}^\top \mathbf{A}_n^{(q)} \right\|_1 \geq \left\| {\mathbf{U}_n^{(q-1)}}^\top \mathbf{A}_n^{(q)} \right\|_1 . \tag{2.37}$$

Lemma 2.3 derives straightforwardly from the convergence proof of (2.34), presented in the Appendix. Moreover, we note that Lemma 2.3 would also hold if, instead of the FPI of (2.34), we solved (2.35) by means of the bit-flipping algorithm of [58]. Also, Lemma 2.3 holds true if $\mathbf{U}_n^{(q)}$ is computed by the exact solution of (2.35) obtained by the algorithms of [54]. The following Lemma 2.4 shows that, within the same iteration, the metric increases as we successively optimize the bases.

Lemma 2.4. *For any $q > 0$ and every $n > m \in [N]$, it holds that $\left\| {\mathbf{U}_n^{(q)}}^\top \mathbf{A}_n^{(q)} \right\|_1 \geq \left\| {\mathbf{U}_m^{(q)}}^\top \mathbf{A}_m^{(q)} \right\|_1$ and $\left\| {\mathbf{U}_1^{(q)}}^\top \mathbf{A}_1^{(q)} \right\|_1 \geq \left\| {\mathbf{U}_N^{(q-1)}}^\top \mathbf{A}_N^{(q-1)} \right\|_1$.*

In view of Lemma 2.4, the following Proposition 2.2 holds true and summarizes the L1-Tucker metric increase across the L1-HOOI iterations.

Proposition 2.2. *For any $n \in [N]$ and every $q' > q$*

$$\left\| {\mathbf{U}_n^{(q')}}^\top \mathbf{A}_n^{(q')} \right\|_1 \geq \left\| {\mathbf{U}_n^{(q)}}^\top \mathbf{A}_n^{(q)} \right\|_1 . \tag{2.38}$$

Defining $p \triangleq \prod_{n \in [N]} d_n$, the following Lemma 2.5 provides an upper bound for the L1-Tucker metric.

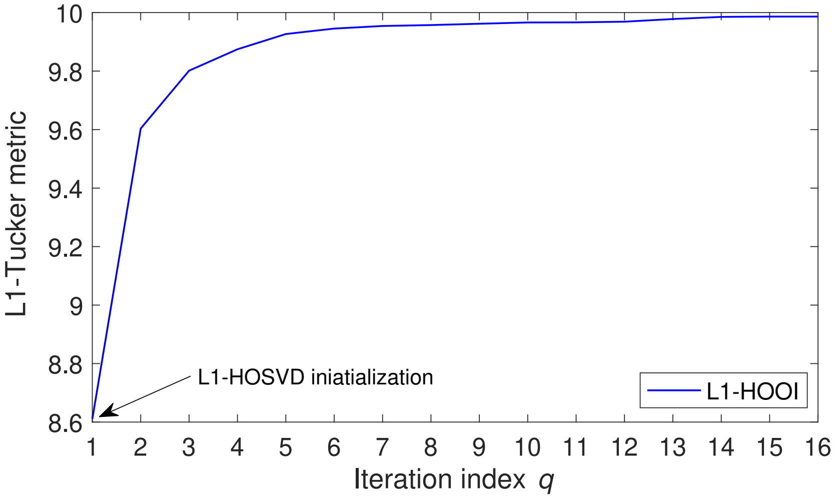


Figure 2.8. L1-Tucker metric across L1-HOOI iterations.

Lemma 2.5. *For any* $\{\mathbf{U}_n \in \mathbb{S}_{D_n \times d_n}\}_{n\in[N]}$, *it holds that*

$$\|\boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^\top\|_1 \leq \sqrt{p}\|\boldsymbol{\mathcal{X}}\|_F. \tag{2.39}$$

Lemma 2.5 shows that the L1-Tucker metric is upper bounded by $\sqrt{p}\|\boldsymbol{\mathcal{X}}\|_F$. This, in conjunction with Proposition 2.2, implies that as q increases the L1-HOOI iterations converge in the L1-Tucker metric. To visualize the convergence, we carry out the following study. We form 5-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{10\times10\times\ldots\times10}$ that draws entries independently from $\mathcal{N}(0,1)$. Then, we apply to it L1-HOOI initialized at L1-HOSVD. In Fig. 2.8, we plot the evolution of the L1-Tucker metric $\|\boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^{(q)\top}\|_1$, versus the L1-HOOI iteration index q. In accordance to our formal analysis, we observe the monotonic increase of the metric and convergence after just 16 iterations. In practice, one can terminate the L1-HOOI iterations when the metric-increase ratio $\zeta(q) \triangleq \left\|\boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^{(q)\top}\right\|_1 \left\|\boldsymbol{\mathcal{X}} \times_{n\in[N]} \mathbf{U}_n^{(q-1)\top}\right\|_1^{-1} - 1$ drops below a predetermined threshold $\tau > 0$, or when q exceeds a maximum number of permitted iterations.

Next, we discuss the computational cost of L1-HOOI.

Complexity of L1-HOOI

For simplicity in presentation, we consider again $D_i = D$ and $d_i = d$, for every $i \in [N]$. As shown above, initialization of L1-HOOI by means of L1-HOSVD costs $\mathcal{O}(dD^{N+1})$. Then, at iteration q, L1-HOOI computes matrix $\mathbf{A}_n^{(q)}$ in (2.35) and its L1-PCA, for every n. Matrix $\mathbf{A}_n^{(q)}$ can computed by a sequence of matrix-to-matrix products as follows. First, we compute the mode-k product of $\boldsymbol{\mathcal{X}}$ with $\mathbf{U}_k^{(z_k)}$, for some $k \neq n$ ($z_k = q$ if $k < n$ and $z_k = q - 1$ if $k > n$), with cost $\mathcal{O}(dD^N)$. Next, we compute the l-mode product of $\boldsymbol{\mathcal{X}} \times_k \mathbf{U}_k^{(z_k)}$ with $\mathbf{U}_l^{(z_l)}$, for some $l \notin \{n, k\}$, with cost $\mathcal{O}(d^2 D^{N-1})$. We observe that the second product (l-mode) has lower cost than the first one (k-mode), for any selection of k and l. Similarly, each subsequent mode product will have further reduced cost. Keeping as dominant term the cost of the first product, the computation of $\mathbf{A}_n^{(q)}$ costs $\mathcal{O}(dD^N)$. After $\mathbf{A}_n^{(q)} \in \mathbb{R}^{D \times d^{N-1}}$ is computed, the solution to (2.35) is approximated by fixed point iterations with cost $\mathcal{O}(D^2 d^N)$. Thus, the cost of a single iteration of L1-HOOI is $\mathcal{O}(dD^N + D^2 d^N)$. Denoting by T the maximum number of iterations permitted, the overall cost of L1-HOOI is $\mathcal{O}(T(dD^N + D^2 d^N))$, comparable to the costs of standard HOSVD and HOOI which are $\mathcal{O}(D^{N+1})$ and $\mathcal{O}(TdD^N)$, respectively. These computational costs are summarized in Table 2.1.

2.5.3 Numerical Studies

Tensor Reconstruction

We set $N = 5$, $D_1 = D_3 = D_5 = 10$, $D_2 = D_4 = 15$, $d_1 = d_2 = 6$, $d_3 = d_4 = d_5 = 4$, and generate Tucker-structured $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_{n \in [5]} \mathbf{U}_n$. The core tensor $\boldsymbol{\mathcal{G}}$ draws entries from $\mathcal{N}(0, 9)$ and, for every n, $\mathbf{U}_n$ is an arbitrary orthonormal basis. Then, we corrupt all entries of $\boldsymbol{\mathcal{X}}$ with zero-mean unit-variance additive white Gaussian noise (AWGN), disrupting its Tucker structure. Moreover, we corrupt N_o out of the 225,000 entries of $\boldsymbol{\mathcal{X}}$ by adding high-variance outliers from $\mathcal{N}(0, \sigma_o^2)$. Thus, we form $\boldsymbol{\mathcal{X}}^{\text{corr}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{O}}$, where $\boldsymbol{\mathcal{N}}$ and $\boldsymbol{\mathcal{O}}$ model AWGN and sparse outliers, respectively. Our objective is to reconstruct $\boldsymbol{\mathcal{X}}$ from the available $\boldsymbol{\mathcal{X}}^{\text{corr}}$. For that purpose, we Tucker decompose $\boldsymbol{\mathcal{X}}^{\text{corr}}$ by means of HOSVD, HOOI, L1-HOSVD, and L1-HOOI and obtain bases $\{\hat{\mathbf{U}}_n\}_{n \in [5]}$. Then, we reconstruct $\boldsymbol{\mathcal{X}}$ as $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}^{\text{corr}} \times_{n \in [5]} \hat{\mathbf{U}}_n \hat{\mathbf{U}_n}^{\top}$. The normalized squared error (NSE) is defined as $\|\boldsymbol{\mathcal{X}} - \hat{\boldsymbol{\mathcal{X}}}\|_F^2 \|\boldsymbol{\mathcal{X}}\|_F^{-2}$. In Fig. 2.9a, we set $N_o = 300$ and plot the mean NSE (MNSE), evaluated over 1000 independent noise/outlier realizations, versus outlier standard deviation $\sigma_o = 4, 8, \ldots, 28$. In the absence of outliers ($\sigma_o = 0$), all methods under comparison exhibit similarly low MNSE. As the outlier standard deviation σ_o increases the MNSE of all methods increases. We notice that the

Table 2.1: Computational costs of PCA, L1PCA-FPI, HOSVD, L1-HOSVD (proposed), HOOI, and L1-HOOI (proposed). PCA/L1PCA-FPI costs are reported for input matrix $\mathbf{X} \in \mathbb{R}^{D \times D}$ and decomposition rank d. Tucker/L1-Tucker costs are reported for N-way input tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D \times D \times \ldots \times D}$ and mode-n ranks $d_n = d \; \forall n$. T is the maximum number of iterations conducted by HOOI and L1-HOOI.

Method	**Cost**
PCA (SVD)	$\mathcal{O}\left(D^3\right)$
L1-PCA (FPI)	$\mathcal{O}\left(D^2 d\right)$
HOSVD	$\mathcal{O}\left(D^{N+1}\right)$
L1-HOSVD	$\mathcal{O}\left(dD^{N+1}\right)$
HOOI	$\mathcal{O}\left(TdD^{N}\right)$
L1-HOOI	$\mathcal{O}\left(T(dD^{N} + D^2 d^{N})\right)$

performances of HOSVD and HOOI markedly deteriorate for $\sigma_o \geq 12$ and $\sigma_o \geq 20$ respectively. On the other hand, L1-HOSVD and L1-HOOI remain robust against corruption, across the board.

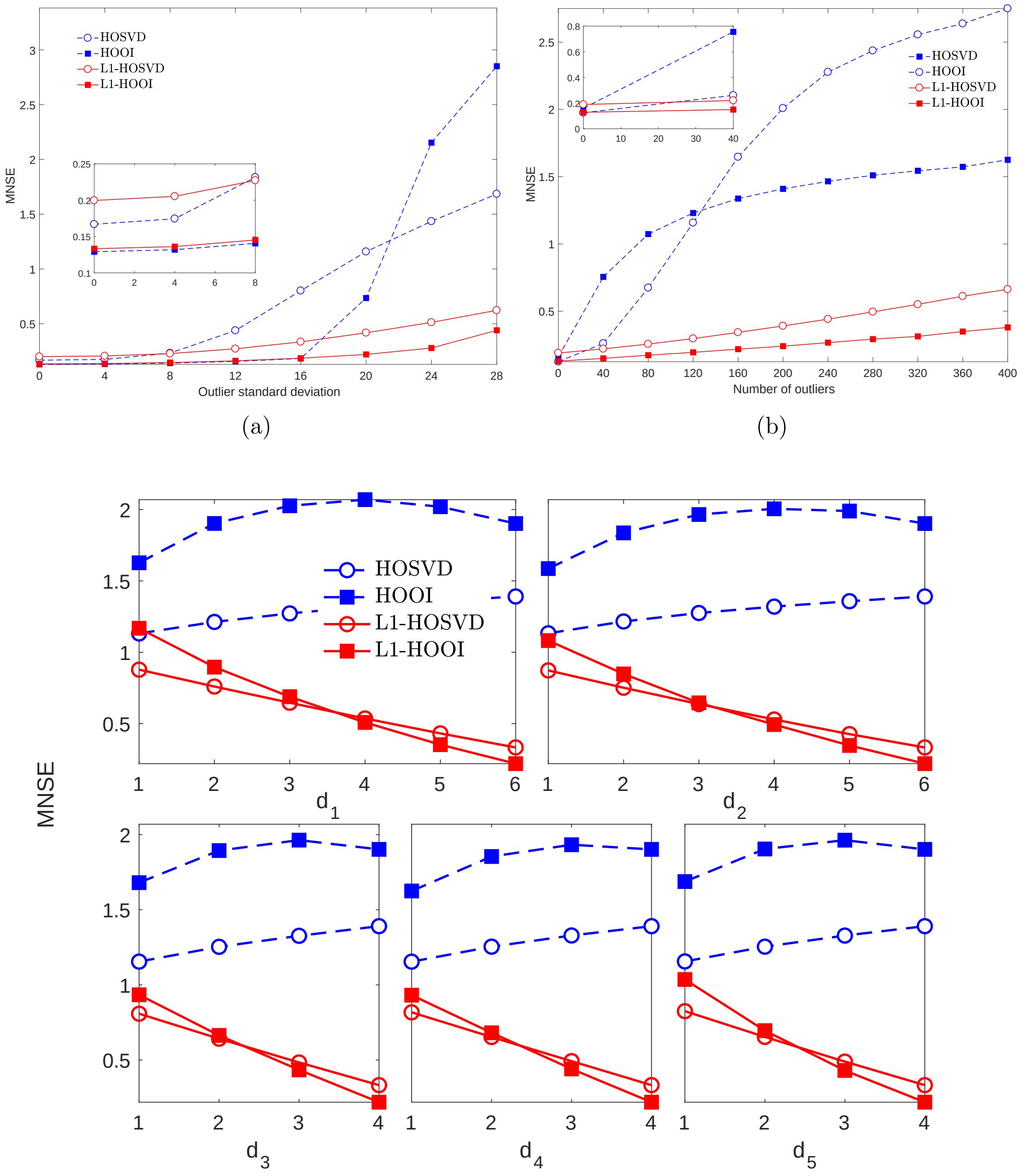
In Fig. 2.9b, we set $\sigma_o = 26$ and plot the MNSE versus number of outliers $N_o = 0, 40, \ldots, 400$. Expectedly, in the absence of outliers ($N_o = 0$), all methods exhibit low MNSE. As the number of outliers increases, HOSVD and HOOI start exhibiting high reconstruction error, while L1-HOSVD and L1-HOOI remain robust. For instance, the MNSE of L1-HOSVD for $N_o = 400$ outliers is lower than the MNSE of standard HOSVD for $N_o = 40$ (ten times fewer) outliers.

Finally, in Fig. 2.9c, we set $\sigma_o = 28$, $N_o = 150$ ($\approx 0.07\%$ of total data entries are corrupted) and plot the MNSE versus $d_n \forall n$ while d_m is set to its nominal value for every $m \in [N = 5] \setminus n$. We observe that, even for a very small fraction of outlier corrupted entries in $\boldsymbol{\mathcal{X}}^{\text{corr}}$, standard Tucker methods are clearly misled across all 5 modes. On the other hand, the proposed L1-Tucker counterparts, exhibit sturdy outlier resistance and reconstruct $\boldsymbol{\mathcal{X}}$ well, remaining almost unaffected by the outlying entries in $\boldsymbol{\mathcal{X}}^{\text{corr}}$.

A robust tensor analysis algorithm, specifically designed for counteracting sparse outliers, is the High-Order Robust PCA (HORPCA) [50]. Formally, given $\boldsymbol{\mathcal{X}}^{\text{corr}}$, HORPCA solves

$$\begin{array}{cc} \underset{\hat{\boldsymbol{\mathcal{X}}}, \hat{\boldsymbol{\mathcal{O}}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}}{\min.} & \sum_{n=1}^{N} \|\mathtt{mat}\left(\hat{\boldsymbol{\mathcal{X}}}, n\right)\|_* + \lambda \|\hat{\boldsymbol{\mathcal{O}}}\|_1 \\ \text{subject to} & \hat{\boldsymbol{\mathcal{X}}} + \hat{\boldsymbol{\mathcal{O}}} = \boldsymbol{\mathcal{X}}^{\text{corr}} \end{array}. \tag{2.40}$$

Authors in [50] presented the HoRPCA-S algorithm for the solution of (2.40) which relies on a



(a)

(b)

(c)

Figure 2.9. (a) MNSE versus standard deviation σ_o for $N_o = 300$. (b) MNSE versus number of outliers N_o for $\sigma_o = 26$. (c) MNSE versus $d_{n n\in[N=5]}$ for $N_o = 150$ and $\sigma_o = 28$; for every m, d_m is set to its nominal value and d_n variance, $n \in [5] \setminus m$.

specific sparsity penalty parameter λ, as well as a thresholding variable μ. The model in (2.40) was

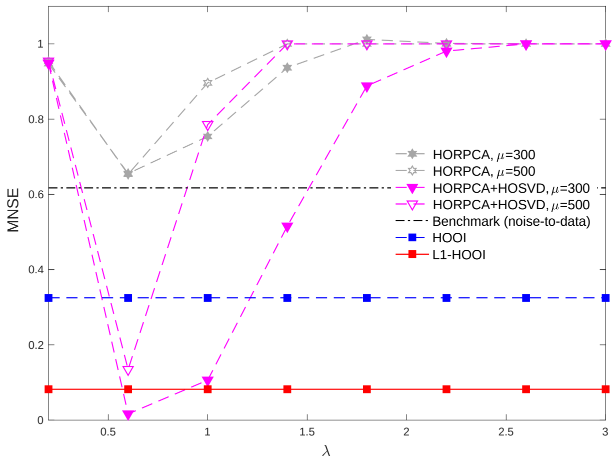


Figure 2.10. MNSE versus λ for varying μ.

introduced considering that, apart from the sparse outliers, there is no dense (full rank) corruption to $\boldsymbol{\mathcal{X}}$ (see [50], subsection 2.6). In the case of additional dense corruption, HORPCA is typically refined by HOSVD [50, 70, 71]. In the sequel, we refer to this approach as HORPCA+HOSVD.

In our next study, we set $N = 5$, $D_n = 5$, and $d_n = 2$ for every n, and build the Tucker-structured data tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_{n\in[5]} \mathbf{U}_n$, where the entries of core $\boldsymbol{\mathcal{G}}$ are independently drawn from $\mathcal{N}(0, 12^2)$. Then, we add both dense AWGN and sparse outliers, creating $\boldsymbol{\mathcal{X}}^{\text{corr}} = \boldsymbol{\mathcal{X}}+\boldsymbol{\mathcal{N}}+\boldsymbol{\mathcal{O}}$, where the entries of noise $\boldsymbol{\mathcal{N}}$ are drawn independently from $\mathcal{N}(0, 1)$ and the 15 non-zero entries of $\boldsymbol{\mathcal{O}}$ (in arbitrary locations) are drawn from $\mathcal{N}(0, 20^2)$. Then, we attempt to reconstruct $\boldsymbol{\mathcal{X}}$ from the available $\boldsymbol{\mathcal{X}}^{\text{corr}}$ using HOOI, HORPCA (for $\lambda = 0.2, 0.6, \ldots, 3$ and $\mu = 300, 500$), HORPCA+HOSVD (same λ and μ combinations as HORPCA), and the proposed L1-HOOI.

In Fig. 2.10, we plot MNSE computed over 50 data/noise/corruption realizations, versus λ for the four methods. In addition, we plot the average noise-to-data benchmark $\|\boldsymbol{\mathcal{N}}\|_F^2\|\boldsymbol{\mathcal{X}}\|_F^{-2}$. In accordance with our previous studies, we observe that L1-HOOI offers markedly lower MNSE than standard HOOI. In addition, we notice that for specific selection of μ and λ ($\mu = 300$ and $\lambda = 0.6$) HORPCA+HOSVD attains MNSE slightly lower than L1-HOOI. However, for any different selection of λ L1-HOOI attains markedly better reconstruction. In addition, we plot the performance of HORPCA when it is not refined by HOSVD. We notice that, expectedly, for specific selections of μ and λ the method is capable of removing the outliers, but not the dense noise component –thus, the MNSE approaches the average noise-to-data benchmark. This study corroborates the corruption-resistance of L1-HOOI, while, similar to HOOI, it does not depend on any tunable parameters, other than $\{d_n\}_{n\in[N]}$.

Classification

Tucker decomposition is commonly employed for classification of multi-way data samples. Below, we consider the Tucker-based classification framework originally presented in [104]. That is, we consider C classes of order-N tensor objects of size $D_1 \times D_2 \times \ldots \times D_N$ and M_c labeled samples available from the c-th class, $c \in [C]$, that can be used for training a classifier. The training data from class c are organized in tensor $\boldsymbol{\mathcal{X}}_c \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N \times M_c}$ and the total of $M = \sum_{c=1}^{C} M_c$ training data are organized in tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times \ldots \times D_N \times M}$, constructed by concatenation of $\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_C$ across mode $(N+1)$.

In the first processing step, $\boldsymbol{\mathcal{X}}$ is Tucker decomposed, obtaining feature bases $\{\mathbf{U}_n \in \mathbb{S}_{D_n \times d_n}\}_{n \in [N]}$ for the first N modes (feature modes) and the sample basis $\mathbf{Q} \in \mathbb{S}_{M \times M}$ for the $(N+1)$-th mode (sample mode). The obtained feature bases are then used to compress the training data, as

$$\boldsymbol{\mathcal{G}}_c = \boldsymbol{\mathcal{X}}_c \times_{n \in [N]} \mathbf{U}_n^\top \in \mathbb{R}^{d_1 \times \ldots \times d_N \times M_c} \tag{2.41}$$

for every $c \in [C]$. Then, the M_c compressed tensor objects from the c-th class are vectorized (equivalent to mode-$(N+1)$ flattening) and stored in the data matrix

$$\mathbf{G}_c = \mathtt{mat}\left(\boldsymbol{\mathcal{G}}_c, N+1\right)^\top \in \mathbb{R}^{p \times M_c}, \tag{2.42}$$

where $p = \prod_{n \in [N]} d_n$. Finally, the labeled columns of $\{\mathbf{G}_c\}_{c \in [C]}$ are used to train any standard vector-based classifier, such as support vector machines (SVM), or k-nearest-neighbors (k-NN). When an unlabeled testing point $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$ is received, it is first compressed using the Tucker-trained bases as $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{Y}} \times_{n \in [N]} \mathbf{U}_n^\top$. Then, $\boldsymbol{\mathcal{Z}}$ is vectorized as $\mathbf{z} = \text{vec}(\boldsymbol{\mathcal{Z}}) = \mathtt{mat}\left(\boldsymbol{\mathcal{Z}}, N+1\right)^\top$. Finally, vector $\mathbf{z}$ is classified based on the standard vector classifier trained above.

In this study, we focus on the classification of order-2 data ($N = 2$) from the MNIST image dataset of handwritten digits [105]. Specifically, we consider $C = 5$ digit classes (digits $0, 1, \ldots, 4$) and $M_1 = \ldots = M_5 = 10$ image samples of size $(D = D_1 = 28) \times (D = D_2)$ available from each class. To make the classification task more challenging, we consider that each training image is corrupted by heavy-tail noise with probability α. Then, each pixel of a corrupted image is additively corrupted by a noise component $w \sim \text{unif}(0, v)$, with probability β. Denoting the average pixel energy by $E = \frac{1}{D^2 M}\|\boldsymbol{\mathcal{X}}\|_F^2$, we choose v so that $\sqrt{\frac{E}{\mathbb{E}\{w^2\}}} = 10$. We conduct Tucker-based classification as described above, for $d = d_1 = d_2$, using a nearest-neighbor (NN) classifier (i.e., 1-NN), by which

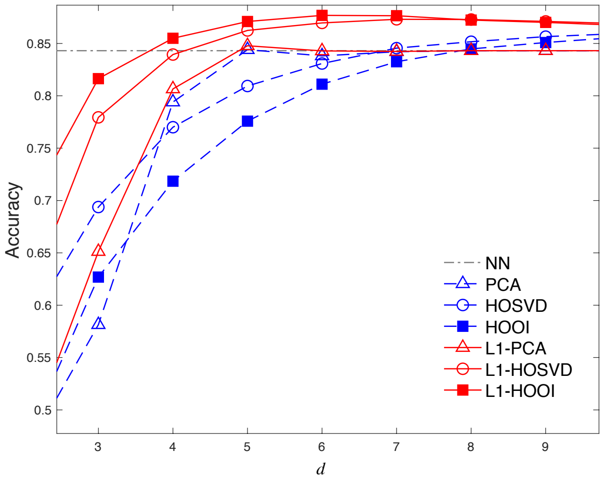


Figure 2.11. Classification accuracy versus d, for $\alpha = 0.2$ and $\beta = 0.5$.

testing sample $\mathbf{z}$ is assigned to class[4]

$$c^* = \underset{c \in [C]}{\operatorname{argmin}} \left\{ \min_{j \in [M_c]} \|\mathbf{z} - [\mathbf{G}_c]_{:,j}\|_2^2 \right\}. \tag{2.43}$$

For a given training dataset, we classify 500 testing points from each class. Then, we repeat the training/classification procedure on 300 distinct realizations of training data, testing data, and corruptions. In Fig. 2.11, we plot the average classification accuracy versus d for $\alpha = 0.2$ and $\beta = 0.5$, for HOSVD, HOOI, L1-HOSVD, L1-HOOI, as well as PCA, L1-PCA,[5] and plain NN classifier that returns the label of the nearest column of $\mathtt{mat}\left(\boldsymbol{\mathcal{X}}, N+1\right)^\top \in \mathbb{R}^{P \times M}$ to the vectorized testing sample $\text{vec}(\boldsymbol{\mathcal{Y}})$. We observe that, in general, the compression-based methods can attain superior performance than plain NN. Moreover, we notice that $d > 7$ implies $p > M$ and, thus, the PCA/L1-PCA methods attain constant performance, equal to plain NN. Moreover, we notice that L1-PCA outperforms PCA, for every value of $d \leq 7$. For $4 \leq d \leq 7$, PCA/L1-PCA outperform the Tucker methods. Finally, the proposed L1-Tucker methods outperform standard Tucker and PCA/L1-PCA, for every d, and attain the highest classification accuracy of about 89% for $d = 6$ (5% higher than plain NN).

Next, we fix $d = 5$ and $\beta = 0.8$ and plot in Fig. 2.12 the average classification accuracy, versus α. This figure reveals the sensitivity of standard HOSVD and HOOI as the training data corruption

[4] We consider a simple classifier, so that the study focuses to the impact of each compression method.

[5] Denoting by $\mathbf{U}$ the $\min\{p, M\}$ PCs/L1-PCs of $\mathtt{mat}\left(\boldsymbol{\mathcal{X}}, N+1\right)^\top \in \mathbb{R}^{P \times M}$, we train any classifier on the labeled columns of $\mathbf{U}^\top \mathtt{mat}\left(\boldsymbol{\mathcal{X}}, N+1\right)$ and classify the vectorized and projected testing sample $\mathbf{U}^\top \text{vec}(\boldsymbol{\mathcal{Y}})$.

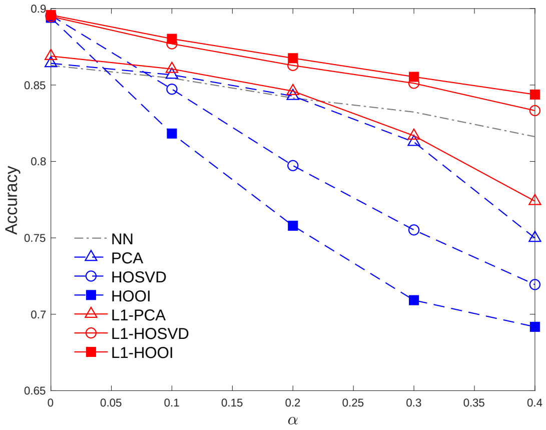


Figure 2.12. Classification accuracy versus α, for $d = 5$ and $\beta = 0.8$.

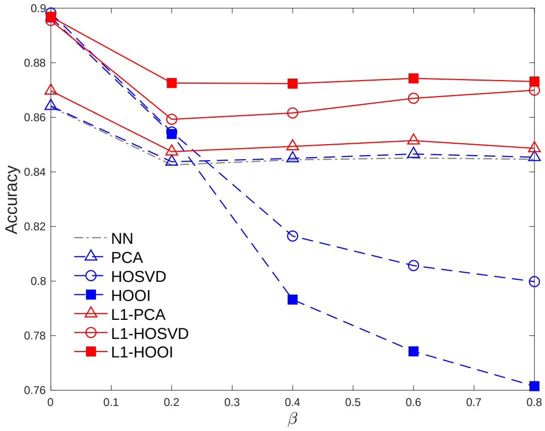


Figure 2.13. Classification accuracy versus β, for $\alpha = 0.2$ and $d = 5$.

probability increases. At the same time, the proposed L1-Tucker methods exhibit robustness against the corruption, maintaining the highest average accuracy for every value of α. For instance, for image-corruption probability $\alpha = 0.3$, L1-HOSVD and L1-HOOI attain about 87% accuracy, while HOSVD and HOOI attain accuracy 75% and 71%, respectively.

Last, in Fig. 2.13, we plot the average classification accuracy, versus the pixel corruption probability β, fixing again $\alpha = 0.2$ and $d = 5$. We observe that, for any value of β, the performance of the L1-HOSVD and L1-HOOI does not drop below 86% and 87.5%, respectively. On the other hand, as β increases, NN and PCA-based methods perform close to 85%. The performance of standard Tucker methods decreases markedly, even as low as 76%, for intense corruption with $\beta = 0.8$. The above studies highlight the benefit of L1-Tucker compared to standard Tucker and PCA counterparts.

Compression

Standard Tucker decomposition is often employed for compression of tensors. In this study, we consider that a small fraction of the entries of the processed tensor has been outlier corrupted by high magnitude/peripheral entries. We explore the capacity of standard Tucker solvers, the proposed L1-Tucker solvers, and other popular tensor decomposition approaches in the literature, in compressing the processed tensor in the presence of outliers. In order to evaluate the success of each method, we reverse the compression operation by reprojecting the compressed tensor to a tensor estimate with size equal to the the size of the processed tensor. Then, we measure the normalized reconstruction error attained by the reconstructed tensor in estimating the nominal tensor.

We work on a dataset from the Formidable Repository of Open Sparse Tensors and Tools (FROSTT) [106]. Specifically, we consider the "Uber Pickups" tensor which is a $(N = 4)$-way array of size $(D_1 = 183$ days$) \times (D_2 = 24$ hours$) \times (D_3 = 1140$ latitudes$) \times (D_4 = 1717$ longitutes$)$. Each entry of this tensor models number of Uber pickups in New York City over a period of time in a specific area of the city. The Uber Pickups tensor can be treated as a collection of 183 $(N = 3)$-way tensors each of which is obtained by fixing the day index (mode-1 index). We fix the day index to 1 and retain a $(N = 3)$-way tensor of size $24 \times 1140 \times 1717$. Then, for a fixed hour index we split each horizontal slab of size 1140×1717 in 20-by-20 blocks and carry out undersampling of the resolution in the latitude and longitude modalities by summing all entries comprising each block. We repeat this procedure for each of the 24 horizontal slabs. After the last operation we obtain tensor $\boldsymbol{\mathcal{X}}_{\text{uber}} \in \mathbb{R}^{(D1=24)\times(D2=57)\times(D3=86)}$ which we will henceforth treat as the ground truth tensor. Visual illustrations of the 1-st, 7-th, 13-th, and 20-th horizontal slabs of $\boldsymbol{\mathcal{X}}_{\text{uber}}$ are offered in Fig. 2.14.

In this study, we employ the following methods for compression of the processed tensor: (i) Standard Tucker implemented by means of the HOSVD and HOOI algorithms; (ii) L1-Tucker implemented by means of the proposed L1-HOSVD and L1-HOOI algorithms; (iii) HORPCA followed by HOSVD as described previously in subsection 2.5.3; and (iv) Robust Tucker Tensor Decomposition (RTTD) [107] the performance of which, similar to HORPCA, depends on an ad-hoc parameter μ in accordance with [107].

By a visual inspection at the horizontal slab samples of $\boldsymbol{\mathcal{X}}_{\text{uber}}$ in Fig. 2.14, we observe that the

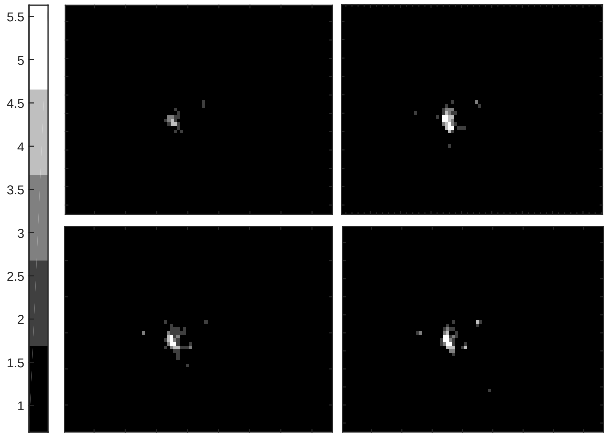


Figure 2.14. Visual illustration (in logarithmic scale) of the 1-st, 7-th, 13-th, and 20-th horizontal slabs of $\boldsymbol{\mathcal{X}}_{\text{uber}}$.

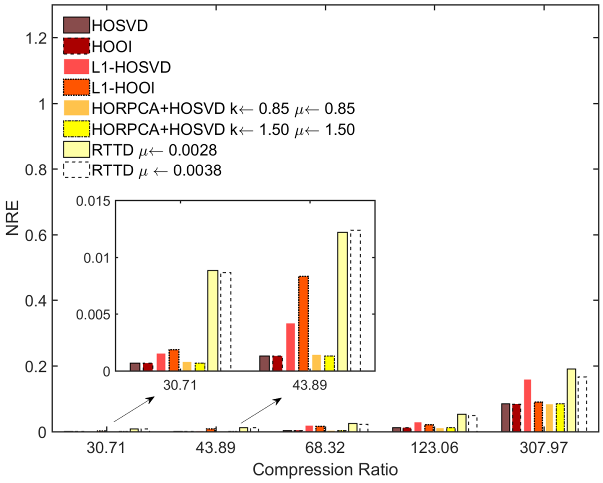


Figure 2.15. Compression error versus compression ratio on the nominal tensor $\boldsymbol{\mathcal{X}}_{\text{uber}}$.

slabs have sparse structure and that the positive counts of each slab are concentrated in a small portion of the slab close to its center. These observations in turn imply that each slab has low-rank structure. In support of this low-rank structure intuition, we first compress the nominal/clean ground-truth tensor and evaluate the performance of each method. That is, we consider processed tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{\text{uber}}$, fix $\mathbf{U}_1 = \mathbf{I}_{24}$, $d_2 = d_3 = d$, and carry out tensor decomposition with HOSVD, HOOI, L1-HOSVD, L1-HOOI, HORPCA+HOSVD, and RTTD. Each method returns bases $\mathbf{U}_2 \in \mathbb{R}^{D_2 \times d}, \mathbf{U}_3 \in \mathbb{R}^{D_3 \times d}$, and core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{24 \times d \times d}$. Accordingly, for each compressed tensor the total number of stored variables is $24d^2 + d(D_2 + D_3)$ while the total number of entries in

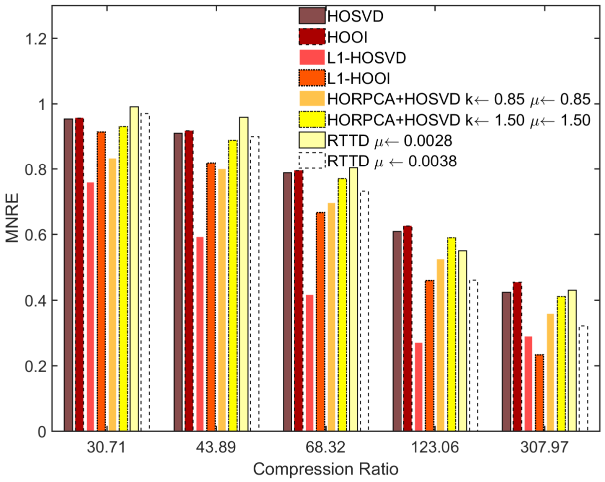


Figure 2.16. Normalized reconstruction error versus compression ratio in the presence of $N_o = 12$ outliers.

the ground truth/processed tensor is $D_1 D_2 D_3$. In view of the above, we define the compression ratio $\mathrm{CR}(d) = \frac{D_1 D_2 D_3}{24d^2 + d(D_2 + D_3)}$.To measure the success of each method, we obtain low-rank tensor estimate $\widehat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{G}} \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$ and measure the normalized reconstruction error (NRE) $\|\boldsymbol{\mathcal{X}}_{\mathrm{uber}} - \widehat{\boldsymbol{\mathcal{X}}}\|_F^2 \|\boldsymbol{\mathcal{X}}_{\mathrm{uber}}\|_F^{-2}$.

In Fig. 2.15, we report the NRE versus compression ratio when the compression ratio varies, as d varies in $\{10, 8, 6, 4, 2\}$. We observe that for compression ratio less that 69, all methods reconstruct the nominal tensor well attaining similar performance. For compression ratio greater than 124 the reconstruction error of all methods increases but the reconstruction performance remains high. In this case where the data are nominal/clean, standard Tucker solvers attain slightly higher performance than the other counterparts.

Next, we consider that N_o entries of a single arbitrarily chosen horizontal slab of $\boldsymbol{\mathcal{X}}_{\mathrm{uber}}$ with index s_{cor} are additively corrupted by pseudorandom scalar integers between 1 and 500. That is, we consider processed tensor $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_{\mathrm{uber}} + \boldsymbol{\mathcal{X}}_{\mathrm{cor}}$, where $[\boldsymbol{\mathcal{X}}_{\mathrm{cor}}]_{s_{\mathrm{cor}},:,:}$ has N_o non-zero entries between 1 and 500 and for any index $s \in \{1, 2 \ldots, 24\} \setminus s_{\mathrm{cor}}$ it holds $[\boldsymbol{\mathcal{X}}_{\mathrm{cor}}]_{s,:,:} = \mathbf{0}_{57 \times 86}$. As before, we compute a set of bases and core tensor with each method. Then, we compute the reconstructed tensor $\widehat{\boldsymbol{\mathcal{X}}}$ and measure the NRE in approximating the nominal tensor $\boldsymbol{\mathcal{X}}_{\mathrm{uber}}$. In Fig. 2.16, we fix $N_o = 12$ and report the mean NRE (MNRE) computed based on 1000 distinct realizations of corruption ($\boldsymbol{\mathcal{X}}_{\mathrm{cor}}$ and s_{cor}). We observe that for low compression ratio 30.71 all methods exhibit large reconstruction error with L1-HOSVD attaining the lowest. As the compression ratio increases (d decreases) the performance of all methods improves. For compression ratio up to 123.06 L1-HOSVD outperforms

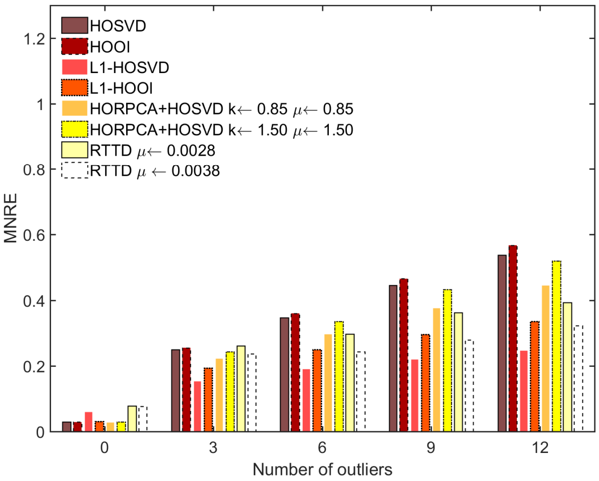


Figure 2.17. Normalized reconstruction error versus number of outliers. Compression ratio is set to 182.40 ($d = 3$).

all counterparts while for compression ratio 307.97 L1-HOOI attains the highest reconstruction performance among all compared methods. Moreover, we observe that different ad-hoc parameter selections for HORPCA+HOSVD and RTTD yield different reconstruction performances. Favorable ad-hoc parameter selections may exist such that HORPCA+HOSVD and RTTD attain high reconstruction performance but these parameters require fine tuning. To the best of our knowledge, there is no prescribed way of setting these parameters in favor of high performance, the selection of which depends on the decomposition rank, the data distribution of the processed tensor, the number of outliers, the magnitude of outliers, the sparsity of the processed tensor, etc. Finally, we fix the compression ratio to 182.40 ($d = 3$), let the number of outliers N_o vary in $\{0, 3, 6, 9, 12\}$, and for each value N_o compute the MNRE based on 1000 independent and distinct realizations of corruption. In Fig. 2.17, we illustrate the computed MNRE. Expectedly, in the absence of outliers ($N_o = 0$) all methods attain high, similar performance. As N_o increases the reconstruction error of all methods increases along. L1-HOSVD attains the highest reconstruction performance for any $N_o > 0$. L1-HOOI and RTTD ($\mu = 0.0038$) follow with almost identical performance. Standard Tucker solvers HOSVD and HOOI attain low reconstruction performance for any $N_o > 0$. Regarding HORPCA+HOSVD and RTTD, once more we observe that different reconstruction performances are attained for different ad-hoc parameters selections.

2.6 Contribution 3: Dynamic L1-norm Tucker

Focusing on outlier-resistant tensor processing, we wish to estimate the L1-Tucker bases of a tensor-data model, as formulated in (2.7). We assume, however, that the measurements $\{\boldsymbol{\mathcal{X}}_t\}_{t=1}^{T}$ are originally unavailable and collected in a streaming fashion, one at a time.

To set our algorithmic guidelines, we start by considering two simplistic antipodal approaches. On the one hand, an *instantaneous* approach would L1-Tucker-decompose each new measurement to return new bases, independently of any previously seen data. While this approach is memoryless and computationally simple, its bases estimation performance is bound to be limited, especially in low Signal-to-Noise Ratio (SNR). On the other hand, an *increasing-batch* approach would append the new measurement to the already collected ones and re-solve the L1-Tucker problem from scratch. As the data collection increases, this method could attain superior bases estimation performance at the expense of increasingly high computational and storage overhead.

Both these extreme approaches exhibit an unfavorable performance/cost trade-off. In contrast, a preferred method would leverage each new measurement, together with previous ones, to efficiently update the existing bases. The development of such a method is the main contribution of this Section, as presented in detail in the following Section 2.6.1.

In the algorithmic developments of this Section, we implement L1-Tucker (batch processing) with L1-norm Bit-Flipping (L1-BF) algorithm [58] as the underlying L1-PCA solver. For the sake of completeness, a brief description of L1-BF follows.

Consider matrix $\mathbf{X} \in \mathbb{R}^{Z \times Q}$, for $Q \geq Z$, and the L1-PCA

$$\underset{\mathbf{Q} \in \mathbb{S}_{Z \times z}}{\text{max.}} \left\| \mathbf{X}^\top \mathbf{Q} \right\|_1 . \tag{2.44}$$

L1-BF is based on the following Theorem, presented in [54].

Theorem 2.2. *[54] Let $\mathbf{B}_{opt} \in \{\pm 1\}^{Q \times z}$ be a solution to* $\text{max.}_{\mathbf{B} \in \{\pm 1\}^{Q \times z}} \|\mathbf{X}\mathbf{B}\|_*$. $\texttt{Proc}(\mathbf{X}\mathbf{B}_{opt})$ *is an exact solution to L1-PCA in* (2.44).

The nuclear norm $\| \cdot \|_*$ returns the sum of the singular values of its argument and, for any tall matrix $\mathbf{A} \in \mathbb{R}^{Z \times z}$ that admits SVD $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}_{z \times z}\mathbf{V}^\top$, $\texttt{Proc}(\mathbf{A}) = \mathbf{U}\mathbf{V}^\top$.

In view of Theorem 2.2, [58] proposed to initialize at arbitrary $\mathbf{B}_0 \in \{\pm 1\}^{Q \times z}$ and iteratively conduct optimal single-bit flips (negations). Let $\mathbf{e}_{q,Q}$ denote the q-th column of the size-Q identity

```
Input: $\boldsymbol{\mathcal{X}}$, $\{\mathbf{Q}_n\}_{n\in[N]}$
Output: $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_N$

Until convergence/termination
    For $n \in [N]$
        $\mathbf{A} \leftarrow \texttt{mat}(\boldsymbol{\mathcal{X}} \times_{m\in[n-1]} \mathbf{Q}_m^\top \times_{k\in[N-n]+n} \mathbf{Q}_k^\top, n)$
        $\mathbf{B} \leftarrow \mathrm{sgn}(\mathbf{A}^\top \mathbf{Q}_n)$
        $\mathbf{Q}_n \leftarrow \texttt{L1-BF}(\mathbf{A}, \mathbf{B})$

Function: $\texttt{L1-BF}(\mathbf{A}, \mathbf{B})$,   % $\mathbf{B} \in \{\pm 1\}^{Q\times z}$
Output: $\mathbf{Q}$

Until convergence/termination
    $(\bar{k}, \bar{l}) \leftarrow \underset{(k,l)\in[Q]\times[z]}{\mathrm{argmax}} \left\| \mathbf{A}\left(\mathbf{B} - 2\mathbf{e}_{k,Q}\mathbf{e}_{l,z}^\top[\mathbf{B}]_{k,l}\right)\right\|_*$
    $\mathbf{B} \leftarrow \mathbf{B} - 2\mathbf{e}_{\bar{k},Q}\mathbf{e}_{\bar{l},z}[\mathbf{B}]_{\bar{k},\bar{l}}$
$\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top \leftarrow \texttt{svd}(\mathbf{A}\mathbf{B})$
$\mathbf{Q} \leftarrow \mathbf{U}\mathbf{V}^\top$
```

Algorithm 2.4 L1-norm Tucker Decomposition algorithm for batch-processing.

matrix $\mathbf{I}_Q$. Then, at iteration $i \geq 1$, L1-BF solves

$$(k', l') = \underset{(k,l)\in[Q]\times[z]}{\mathrm{argmax}} \left\| \mathbf{X}\left(\mathbf{B}_{i-1} - 2\mathbf{e}_{k,Q}\mathbf{e}_{l,z}^\top[\mathbf{B}_{i-1}]_{k,l}\right)\right\|_* \tag{2.45}$$

and updates $\mathbf{B}_i = \mathbf{B}_{i-1} - 2\mathbf{e}_{k',Q}\mathbf{e}_{l',z}[\mathbf{B}_{i-1}]_{k',l'}$. Among all possible single bit-flips, negation of the (k', l')-th entry of $\mathbf{B}_{i-1}$ offers the maximum possible value in $\|\mathbf{X}\mathbf{B}_i\|_*$. Importantly, L1-BF is guaranteed to monotonically increase the metric and converge in finite (in practice, few) iterations.

A pseudocode of L1-Tucker (batch processing), implemented by means of L1-HOOI relying on L1-BF is offered in Algorithm 2.4.

2.6.1 Dynamic L1-Tucker Algorithm

The proposed *Dynamic L1-Tucker Decomposition (D-L1-Tucker)* is a method for incremental estimation of the L1-Tucker bases. D-L1-Tucker is designed to (i) attain high bases estimation performance, (ii) suppress outliers, and (iii) adapt to changes of the nominal subspaces. In this Section, we present D-L1-Tucker in detail, addressing bases initialization, bases updates, parameter tuning, and modifications for long-term efficiency.

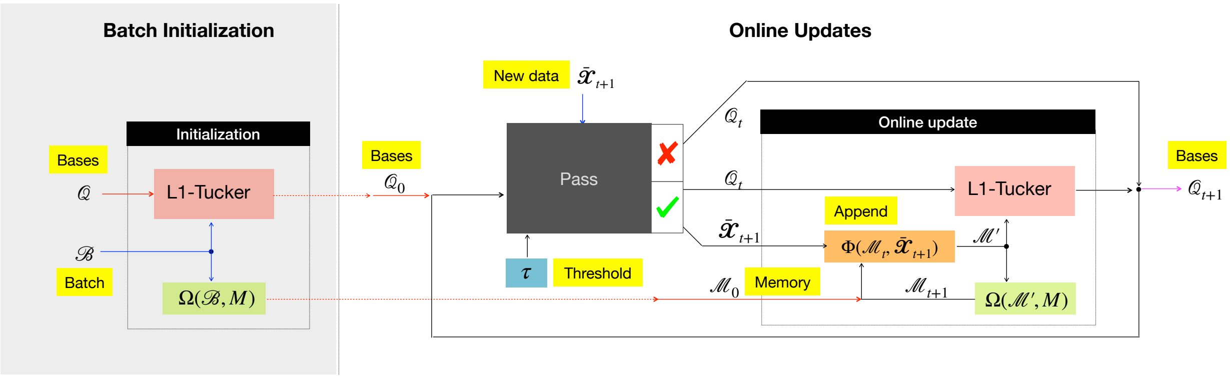


Figure 2.19. A schematic illustration of the proposed algorithm for streaming L1-norm Tucker decomposition.

Batch Initialization

Considering the availability of an initial batch of $B \ll T$ measurements, $\mathcal{B} = \{\boldsymbol{\mathcal{X}}_1, \ldots, \boldsymbol{\mathcal{X}}_B\}$, we run on it L1-HOSVD or L1-HOOI to obtain an initial set of L1-Tucker estimates $\mathcal{Q}_0 = \{\mathbf{Q}_1^{(0)}, \ldots, \mathbf{Q}_N^{(0)}\}$.

Apart from $\mathcal{Q}_0$, we also initialize a *memory set* $\mathcal{M}_0 = \Omega(\mathcal{B}, M)$, for some maximum memory size $M \geq 0$. For any ordered set $\mathcal{I}$ and integer $Z \geq 0$, we define

$$\Omega(\mathcal{I}, Z) = \begin{cases} \mathcal{I}, & \text{if} \quad |\mathcal{I}| \leq Z, \\ [\mathcal{I}]_{|\mathcal{I}|-Z+1:|\mathcal{I}|}, & \text{if} \quad |\mathcal{I}| > Z, \end{cases} \tag{2.46}$$

where $|\cdot|$ denotes the cardinality (number of elements in a set) of its input argument. That is, $\Omega(\mathcal{B}, M)$ returns the last $\min\{M, B\}$ elements in $\mathcal{B}$.

If an initialization batch $\mathcal{B}$ is not available, the bases in $\mathcal{Q}_0$ are chosen arbitrarily and the initial memory $\mathcal{M}_0$ is empty. In this case, D-L1-Tucker becomes purely streaming.

Streaming Updates

When a new measurement $\bar{\boldsymbol{\mathcal{X}}}_t \neq \mathbf{0}$, $t \geq 1$, is collected,[6] we first perform a check on it to assess its reliability based on the most recently updated set of bases $\mathcal{Q}_{t-1}$. Motivated by [55, 77, 108, 109],

[6] A bar over a tensor denotes that it is streaming.

we define the reliability as

$$r_t = \left\| \bar{\boldsymbol{\mathcal{X}}}_t \times_{n \in [N]} \mathbf{Q}_n^{(t-1)^\top} \right\|_F^2 \left\| \bar{\boldsymbol{\mathcal{X}}}_t \right\|_F^{-2} \quad \in [0,1]. \tag{2.47}$$

After some algebraic manipulations, (2.47) can be rewritten as

$$r_t = \cos^2(\phi(\mathrm{vec}(\bar{\boldsymbol{\mathcal{X}}}_t \times_{n \in [N]} \mathbf{Q}_n^{(t-1)} \mathbf{Q}_n^{(t-1)^\top}), \mathrm{vec}(\bar{\boldsymbol{\mathcal{X}}}_t))), \tag{2.48}$$

where $\phi(\cdot,\cdot)$ returns the angle between its two vector arguments. Intuitively, r_t quantifies how much measurement $\bar{\boldsymbol{\mathcal{X}}}_t$ conforms to the multi-way subspace spanned by $\{\mathbf{Q}_n^{(t-1)}\}_{n \in [N]}$, or, the angular proximity of $\mathrm{vec}(\bar{\boldsymbol{\mathcal{X}}}_t \times_{n \in [N]} \mathbf{Q}_n^{(t-1)} \mathbf{Q}_n^{(t-1)^\top})$ to $\mathrm{vec}(\bar{\boldsymbol{\mathcal{X}}}_t)$. This check of reliability/conformity inherits its robustness from the L1-Tucker-derived bases upon which it is defined. Moreover, if an outlier happens to pass the reliability check, L1-Tucker will try to suppress it, providing again robust bases. By definition, the value of r_t will be between 0 and 1. If $r_t = 1$, then the bases in $\mathcal{Q}_{t-1}$ perfectly describe $\bar{\boldsymbol{\mathcal{X}}}_t$. In contrast, if $r_t = 0$, then the set $\mathcal{Q}_{t-1}$ does not capture any component of $\bar{\boldsymbol{\mathcal{X}}}_t$. Then, we introduce a user-defined parameter τ and consider that $\bar{\boldsymbol{\mathcal{X}}}_t$ is reliable for processing if $r_t \geq \tau$. Otherwise, $\bar{\boldsymbol{\mathcal{X}}}_t$ is considered to be an outlier and it is rejected.

If $\bar{\boldsymbol{\mathcal{X}}}_t$ passes the reliability check, we use it to update the bases and memory as follows. First, we append the new measurement to the most recent memory set $\mathcal{M}_{t-1}$ by computing the extended memory $\mathcal{M}' = \Phi(\mathcal{M}_{t-1}, \bar{\boldsymbol{\mathcal{X}}}_t) = \mathcal{M}_{t-1} \cup \bar{\boldsymbol{\mathcal{X}}}_t$. Then, we update the set of bases to $\mathcal{Q}_t$ by running L1-HOOI on $\mathcal{M}'$, initialized to the bases in $\mathcal{Q}_{t-1}$. Finally, we update the memory by discarding the oldest measurement, as

$$\mathcal{M}_t = \Omega(\mathcal{M}', M). \tag{2.49}$$

In view of the above, the cost of the L1-HOOI algorithm remains low across updates because, at any given instance, the extended memory $\mathcal{M}'$ will comprise at most $M+1$ measurements.

If $\bar{\boldsymbol{\mathcal{X}}}_t$ fails the reliability check, we discard it and update the bases and memory by setting $\mathcal{Q}_t = \mathcal{Q}_{t-1}$ and $\mathcal{M}_t = \mathcal{M}_{t-1}$, respectively. A schematic representation of the proposed algorithm is offered in Fig. 2.19. Here, it is worth noting that the proposed approach focuses on temporal coherence of streaming measurements. That is, temporally sporadic points from a second nominal source of measurements could be perceived as outliers.

Zero Centering

In some applications –most notable in image processing– we are interested in subspaces of zero-centered data. To this end, we can modify the proposed algorithm so that, at every update instance $(t-1)$, it computes and maintains the mean $\boldsymbol{\mathcal{C}}_{t-1} = (1/M)\sum_{m=1}^{M}[\mathcal{M}_{t-1}]_m$. Then, when $\bar{\boldsymbol{\mathcal{X}}}_t$ is collected, it will first be zero-centered as $\bar{\boldsymbol{\mathcal{X}}}_t^c = \bar{\boldsymbol{\mathcal{X}}}_t - \boldsymbol{\mathcal{C}}_{t-1}$. If $\bar{\boldsymbol{\mathcal{X}}}_t^c$ passes the reliability check, then it will be used to update the bases, as described above.

Adaptation to Subspace Changes

In many applications of interest, the underlying data subspaces change across time. In such cases, an ambiguity naturally rises on whether a rejected measurement was actually an outlier or the nominal data subspaces have changed and need to be tracked. To resolve this ambiguity and allow D-L1-Tucker to adapt, we work as follows.

First, we make the minor assumption that outlying measurements appear sporadically. Then, we introduce a buffer of ambiguous measurements, $\mathcal{W}$, with capacity $W > 0$. When a streaming measurement fails the reliability check, we insert it to $\mathcal{W}$. If a measurement passes the reliability check, then we empty $\mathcal{W}$. If at any update instance $|\mathcal{W}|$ reaches W –i.e., W consecutive streaming measurements were rejected as outliers– then we detect a change of the nominal subspaces.

In order to adapt to these changes, we empty the memory, set $\mathcal{B} = \mathcal{W}$, and re-initialize (reset) the bases and memory, as described in Section 2.6.1. Next, the updates proceed as described in Section 2.6.1. A pseudocode of the proposed D-L1-Tucker algorithm is presented in Algorithm 2.5.

Long-Run Efficiency

As measurements are streaming, D-L1-Tucker keeps refining the bases estimates. Naturally, after a sufficiently large number of measurements have been processed, the enhancement rate of the bases estimates can be so low that does not justify the computational effort expended for the update.

In view of this observation, we can enhance the long-run efficiency of D-L1-Tucker by introducing an exponentially decreasing probability ρ_t to determine whether or not the t-th measurement will be processed. Intuitively, when a large number of reliable measurements have been processed, ρ_t should be low enough to limit the number of updates performed. For example, let us denote by α_{t-1}

```
Input: $\{\bar{\boldsymbol{\mathcal{X}}}_t\}_{t \in [T]}$, $B$, $M$, $W$, $\tau$, $\mathcal{Q} \leftarrow \cup_{n \in [N]} \mathbf{Q}_n$
Output: $\mathcal{Q} \rightarrow \{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_N\}$

$\mathcal{B} \leftarrow \bigcup_{t \in [B]} \bar{\boldsymbol{\mathcal{X}}}_t$
$(\mathcal{Q}, \mathcal{M}, \mathcal{W}) \leftarrow$ batch-init$(\mathcal{B}, \mathcal{Q}, M)$
For $t > B$
      $(\mathcal{Q}, \mathcal{M}, \mathcal{W}) \leftarrow$ online-updates$(\bar{\boldsymbol{\mathcal{X}}}_t, \mathcal{M}, \mathcal{Q}, \mathcal{W}, \tau)$

Function: batch-init$(\mathcal{B}, \mathcal{Q}, M)$
Output: $\mathcal{Q}, \mathcal{M}, \mathcal{W}$

$\mathcal{Q} \leftarrow$ L1-Tucker$(\mathcal{B}, \mathcal{Q})$
$\mathcal{M} \leftarrow \Omega(\mathcal{B}, M)$
$\mathcal{W} \leftarrow \varnothing$

Function: online-updates$(\bar{\boldsymbol{\mathcal{X}}}_t, \mathcal{M}, \mathcal{Q}, \mathcal{W}, \tau)$
Output: $\mathcal{Q}, \mathcal{M}, \mathcal{W}$

$r_t \leftarrow \|\bar{\boldsymbol{\mathcal{X}}}_t \times_{n \in [N]} \mathbf{Q}_n^\top\|_F^2 \|\bar{\boldsymbol{\mathcal{X}}}_t\|_F^{-2}$
If $r_t > \tau$
   $\mathcal{M}' \leftarrow \Phi(\mathcal{M}, \bar{\boldsymbol{\mathcal{X}}}_t)$
   $\mathcal{Q} \leftarrow$ L1-Tucker$(\mathcal{M}', \mathcal{Q})$
   $\mathcal{M} \leftarrow \Omega(\mathcal{M}', M)$
   $\mathcal{W} \leftarrow \varnothing$
Else
   $\mathcal{W} \leftarrow \mathcal{W} \cup \bar{\boldsymbol{\mathcal{X}}}_t$
   If $|\mathcal{W}| = W$
      $(\mathcal{Q}, \mathcal{M}, \mathcal{W}) \leftarrow$ batch-init$(\mathcal{W}, \mathcal{Q}, M)$
```

Algorithm 2.5 Proposed Dynamic L1-Tucker Decomposition algorithm.

the number of consecutive measurements that have passed the reliability check at update instance $t-1$. Then, if $\bar{\boldsymbol{\mathcal{X}}}_t$ passes the reliability check, it will be processed with probability $\rho_t = \rho^{\alpha_{t-1}+1}$, for some initial probability $\rho > 0$, close to 1. If $\bar{\boldsymbol{\mathcal{X}}}_t$ fails the reliability check, then it is rejected and α_t is reset to 0.

Parameter Configuration

The performance of D-L1-Tucker largely depends on three parameters: the initialization batch size B, the memory size M, and the reliability threshold τ. Here, we discuss how to select these parameters.

Batch size B: B determines the quality of the initial set of bases. That is, higher values of B will generally offer better set of bases. Naturally, a very large B would contradict the streaming nature of the method.

Memory size M: M determines how many measurements L1-Tucker will process at each time instance. Similar to B, higher values of M can enable superior estimation performance. At the same time, high values of M increase the overhead of storage and computation (cost of L1-Tucker updates). Thus, a rule of thumb is to set M as high as the storage/computation limitations of the application permit.

Reliability threshold τ: For $\tau = 0$, all measurements will be processed (including outliers); for $\tau = 1$, all measurements will fail the reliability check and no bases updates will take place. Appropriate tuning of τ between 0 and 1 may ask for some prior knowledge on the SNR quality of the nominal data. Alternatively, in the sequel we present a data-driven method for setting τ.

We start with the reasonable assumption that the initialization batch $\mathcal{B}$ is outlier-free. Then, we conduct on $\mathcal{B}$ a leave-one-out cross-validation to tune τ. For every $i \in [B]$, we first form $\mathcal{B}_i = \mathcal{B} \backslash \boldsymbol{\mathcal{X}}_i$. Then, we obtain the set of bases $\mathcal{Q}_i$ by running L1-HOOI on $\mathcal{B}_i$. Next, we capture in r_i the reliability of $\boldsymbol{\mathcal{X}}_i$ evaluated on $\mathcal{Q}_i$ (notice that $\boldsymbol{\mathcal{X}}_i$ did not participate in the computation of $\mathcal{Q}_i$). Finally, we set τ to the minimum, median, or maximum value of the cross-validated reliabilities $\{r_1, \ldots, r_B\}$, depending on the noise-tolerance/outlier-robustness level that we want to enforce.

2.6.2 Experimental Studies

Testing Parameter Configurations

We first study the performance of the proposed D-L1-Tucker algorithm across varying parameter configurations. We consider T N-way measurements $\bar{\boldsymbol{\mathcal{X}}}_1, \ldots, \bar{\boldsymbol{\mathcal{X}}}_T$, where

$$\bar{\boldsymbol{\mathcal{X}}}_t = \boldsymbol{\mathcal{G}}_t \times_{n\in[N]} \mathbf{Q}_n^{\text{nom}} + \boldsymbol{\mathcal{N}}_t + \boldsymbol{\mathcal{O}}_t \in \mathbb{R}^{D\times D\times \ldots \times D}, \tag{2.50}$$

for a nominal set of bases $\mathcal{Q}^{\text{nom}} = \{\mathbf{Q}_n^{\text{nom}} \in \mathbb{S}_{D\times d}\}_{n\in[N]}$. The core tensor $\boldsymbol{\mathcal{G}}_t \in \mathbb{R}^{d\times d\times \ldots \times d}$ draws entries independently from $\mathcal{N}(0, \sigma_s^2)$. $\boldsymbol{\mathcal{N}}_t$ models Additive White Gaussian Noise (AWGN) and draws entries from $\mathcal{N}(0, \sigma_n^2)$. $\boldsymbol{\mathcal{O}}_t$ models sporadic heavy outlier corruption and is non-zero with probability p_o. When non-zero, $\boldsymbol{\mathcal{O}}_t$ draws entries from $\mathcal{N}(0, \sigma_o^2)$. In order to measure data quality, we define the SNR as

$$\text{SNR} = \frac{\mathbb{E}\{\|\boldsymbol{\mathcal{X}}_t\|_F^2\}}{\mathbb{E}\{\|\boldsymbol{\mathcal{N}}_t\|_F^2\}} = \frac{\sigma_s^2}{\sigma_n^2}\left(\frac{d}{D}\right)^N \tag{2.51}$$

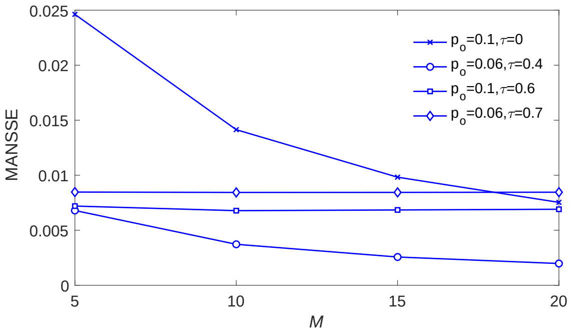


Figure 2.21. MANSSE vs memory size. $N = 3$, $D = 10$, $d = 5$, $B = 5$, $T = 30$, SNR = 0dB, ONR = 14dB, 3000 realizations.

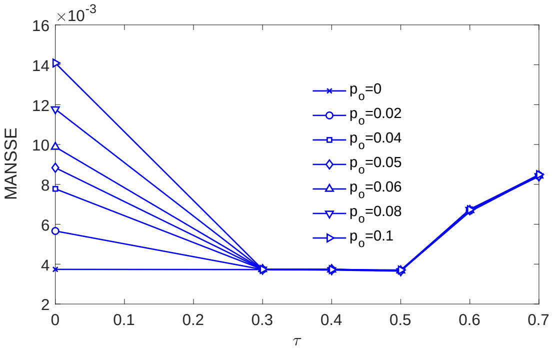


Figure 2.22. MANSSE vs reliability threshold. $N = 3$, $D = 10$, $d = 5$, $B = 5$, $M = 10$, $T = 30$, SNR = 0dB, ONR = 14dB, 3000 realizations.

and the Outlier-to-Noise Ratio (ONR)

$$\text{ONR} = \frac{\mathbb{E}\{\|\boldsymbol{\mathcal{O}}_t\|_F^2\}}{\mathbb{E}\{\|\boldsymbol{\mathcal{N}}_t\|_F^2\}} = \frac{\sigma_o^2}{\sigma_n^2}. \tag{2.52}$$

Our objective is to recover $\mathcal{Q}^{\text{nom}}$ by processing the measurements $\{\bar{\boldsymbol{\mathcal{X}}}_t\}_{t\in[T]}$ in a streaming way. Denoting by $\hat{\mathbf{Q}}_n$ the estimate of $\mathbf{Q}_n^{\text{nom}}$, we quantify performance by means of the Mean Aggregate Normalized Subspace Squared Error (MANSSE)

$$\text{MANSSE} = \frac{1}{2Nd} \sum_{n\in[N]} \left\| \mathbf{Q}_n^{\text{nom}} \mathbf{Q}_n^{\text{nom}\top} - \hat{\mathbf{Q}}_n \hat{\mathbf{Q}}_n^\top \right\|_F^2. \tag{2.53}$$

First, we set $N = 3$, $D = 10$, $d = 5$, $B = 5$, and $T = 30$. Moreover, we set σ_s^2, σ_n^2, and σ_o^2 such that SNR = 0dB and ONR = 14dB. In Fig. 2.21, we plot the MANSSE metric versus varying $M \in \{5, 10, 15, 20\}$ and fixed $(p_0, \tau) \in \{(0.1, 0), (0.06, 0.4), (0.1, 0.6), (0.06, 0.7)\}$. We observe that

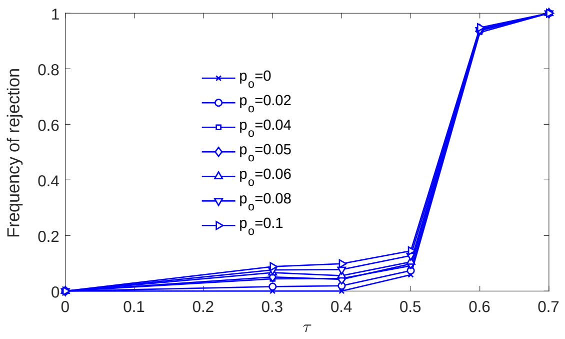


Figure 2.23. Frequency of rejection vs reliability threshold. $N = 3$, $D = 10$, $d = 5$, $B = 5$, $M = 10$, $T = 30$, SNR $=$ 0dB, ONR $=$ 14dB, 3000 realizations.

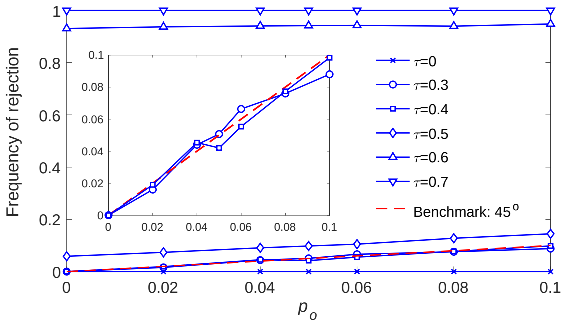


Figure 2.24. Frequency of rejection vs outlier probability. $N = 3$, $D = 10$, $d = 5$, $B = 5$, $M = 10$, $T = 30$, SNR $=$ 0dB, ONR $=$ 14dB, 3000 realizations.

the curves corresponding to $\tau \geq 0.6$ are almost horizontal. This implies that these values of τ are too strict, rejecting almost all measurements. For $\tau = 0$, all measurements are processed (outliers and nominal ones); therefore, we see that the estimation performance improves as M increases, however, the estimation error is somewhat high because of the processed outliers. The curve corresponding to $\tau = 0.4$ exhibits the best performance across the board.

Next, motivated by the above observations, we fix SNR $=$ 0dB and let τ vary in $[0.0, 0.7]$ with 0.1 increments. In Fig. 2.22, we plot the MANSSE versus τ for different values of outlier probability p_o. We notice that for any $\tau \in [0.3, 0.5]$, D-L1-Tucker exhibits high, almost identical MANSSE performance independently of p_o. This, in turn, suggests that the SNR plays an important role in determining the optimal value of τ, for which nominal measurements will be processed and outliers will be rejected with high probability. For the same study, we present the frequency of rejection versus τ in Fig. 2.23. Again, we notice that for very low values of τ, most measurements are accepted for processing. In contrast, for very high values of τ, most measurements are rejected.

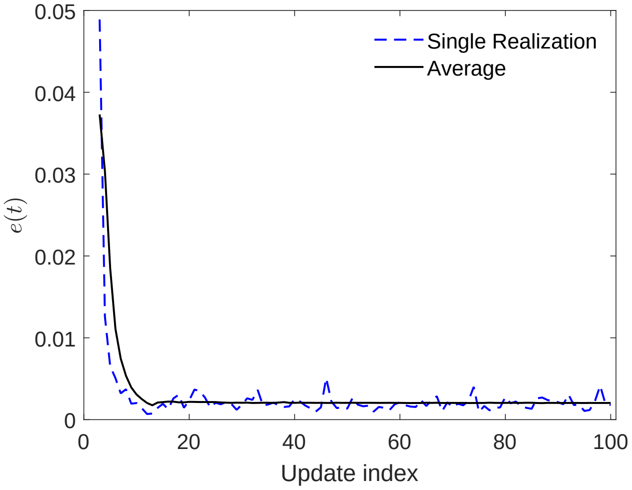


Figure 2.25. Empirical convergence. $N = 3$, $D = 10$, $d = 3$, $T = 100$, $B = 2$, $M = 12$, $W = 0$, SNR $= -6$dB, data-driven τ, 20000 realizations.

Interestingly, this figure suggests that for any given parameter configuration there will be an optimal value of τ for which the frequency of rejection will approach the probability of outliers p_o –which, in turn, implies that in general outliers will rejected and nominal data will be processed.

Finally, we let p_o vary in $\{0, 0.02, 0.06, 0.08, 0.1\}$ and, in Fig. 2.24, we plot the frequency of rejection versus p_o. In accordance with previous observations, we see that high values of τ result in high rejection frequency, independently of the value of p_o. Interestingly, we see that values of τ within $[0.3, 0.4]$ appear to be near-optimal for this particular SNR and parameter configuration, as their performance almost coincides with the $45°$ slope, at all points of which the frequency of rejection equals the outlier probability p_o.

Convergence

At any fixed update index, D-L1-Tucker is guaranteed to converge. That is, when a measurement is deemed reliable, the proposed algorithm processes the measurements in memory after appending the new measurement by means of L1-HOOI the convergence of which has been formally proven [63]. Further, due to the sturdiness of L1-Tucker it is expected that, after many updates, the replacement of a single nominal measurement in the memory set will not cause much of a shift to the bases. To illustrate this, we conduct the following study. We process measurements $\bar{\boldsymbol{\mathcal{X}}}_1, \ldots, \bar{\boldsymbol{\mathcal{X}}}_T$ in the form of (2.50). D-L1-Tucker returns $\hat{\mathcal{Q}}_t = \{\hat{\mathbf{Q}}_{n,t} \in \mathbb{S}_{D \times d}\}_{n \in [N]}$. In order to evaluate convergence across updates, we measure $e(t) = \frac{1}{2Nd} \sum_{n \in [N]} \|\hat{\mathbf{Q}}_{n,t}\hat{\mathbf{Q}}_{n,t}^\top - \hat{\mathbf{Q}}_{n,t-1}\hat{\mathbf{Q}}_{n,t-1}^\top\|_F^2$. In Fig. 2.25, we plot

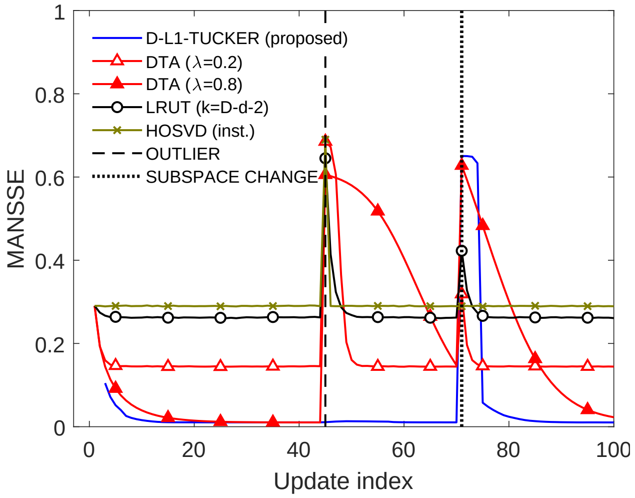


Figure 2.26. MANSSE vs update index. $N = 3$, $D = 10$, $d = 3$, $T_1 = 70$, $T_2 = 30$, $B = 2$, $M = 12$, $W = 4$, SNR $= -6$dB, ONR $= 18$dB, data-driven τ, 20000 realizations.

$e(t)$ versus update index t for a single realization of measurements. Moreover, we plot $e(t)$ when it is sample-average computed over 20000 statistically independent realizations of measurements. As expected, we see that after enough measurements have been processed, $e(t)$ remains low and very close to the average expected performance. We conclude that, upon nominal operation and large enough M, bases changes will be minuscule in the long run. This will be even more emphatic for high SNR. Finally, the long-run efficiency feature of D-L1-Tucker, introduced in Section 2.6.1, can also enforce convergence/termination.

Dynamic Subspace Adaptation

We consider a total of $T = T_1 + T_2$ streaming measurements, in the form of (2.50). The first T_1 measurements are generated by nominal bases $\mathcal{Q}^{\text{nom},1}$. For $t > T_1$ and on, the measurements are generated by bases $\mathcal{Q}^{\text{nom},2}$. The angular proximity of $\mathbf{Q}_n^{\text{nom},1}$ to $\mathbf{Q}_n^{\text{nom},2}$, defined as $1 - \|\mathbf{Q}_n^{\text{nom},1}\mathbf{Q}_n^{\text{nom},1\top} - \mathbf{Q}_n^{\text{nom},2}\mathbf{Q}_n^{\text{nom},2\top}\|_F^2(2d)^{-1}$, is set between 30% and 40% for every $n \in [N]$. Moreover, we consider that the outlier is only active at instance $t = t_o = 45$. We set $N = 3$, $D = 10$, $d = 3$, $T_1 = 70$, and $T_2 = 30$. The SNR and ONR are set to -6dB and 18dB, respectively. We process all measurements by the proposed D-L1-Tucker algorithm for $B = 2$, $M = 12$, $W = 4$, and data-driven τ (median of cross-validated batch reliabilities). We also process the streaming measurements with DTA ($\lambda = 0.2, 0.8$), LRUT (additional core dimensions $k = D - d - 2$), and

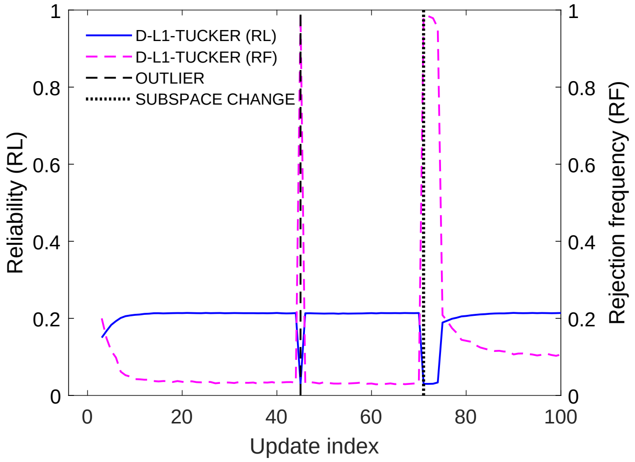


Figure 2.27. Reliability and rejection frequency vs update index. $N = 3$, $D = 10$, $d = 3$, $T_1 = 70$, $T_2 = 30$, $B = 2$, $M = 12$, $W = 4$, SNR $= -6$dB, ONR $= 18$dB, data-driven τ, 20000 realizations.

instantaneous HOSVD[7] counterparts.

In Fig. 2.26, we plot the MANSSE versus update index t. All methods, except for the instantaneous HOSVD, start from a higher MANSSE value and refine their bases by processing streaming measurements until they reach a low plateau. At $t = 45$, when the outlier appears, we observe that all competing methods suffer a significant performance loss. In contrast, the proposed D-L1-Tucker algorithm discards the outlier and its performance remains unaffected. When subsequent measurements are streaming, the competing methods start recovering until they reach again a low plateau, which is largely determined by the SNR and the parameter configuration of each method. Interestingly, the instantaneous HOSVD recovers rapidly, after just one measurement, because it is memoryless. DTA ($\lambda = 0.2$) recovers faster than DTA ($\lambda = 0.8$) but its MANSSE plateau is higher. LRUT also recovers and reaches its plateau performance after it has seen about 10 measurements after the outlier. At time instance 71 the nominal data subspaces shift, affecting all methods expect for the memoryless/instantaneous HOSVD. D-L1-Tucker attains a high value of MANSSE for about W time instances while its ambiguity window is being filled. Right after, it rapidly recovers to a low MANSSE value and keeps refining as more measurements are streaming. DTA and LRUT are also adapting to the new underlying structure after processing a few measurements. Another interesting observation is that the low plateau level for each method appears to be the same in the two distinct coherence windows.

In Fig. 2.27, we plot the reliability of the streaming measurements across updates in accordance

[7] At update instance t, instantaneous HOSVD returns the HOSVD solution of $\bar{\boldsymbol{\mathcal{X}}}_t$, independently of any previous measurements.

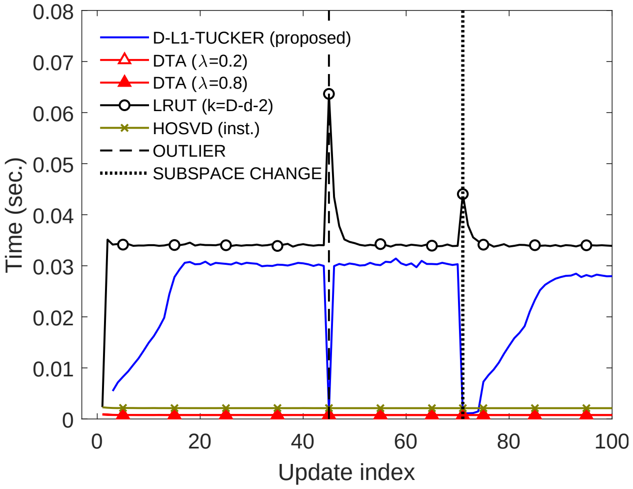


Figure 2.28. Time (sec.) vs update index. $N = 3$, $D = 10$, $d = 3$, $T_1 = 70$, $T_2 = 30$, $B = 2$, $M = 12$, $W = 4$, SNR $= -6$dB, ONR $= 18$dB, data-driven τ, 20000 realizations.

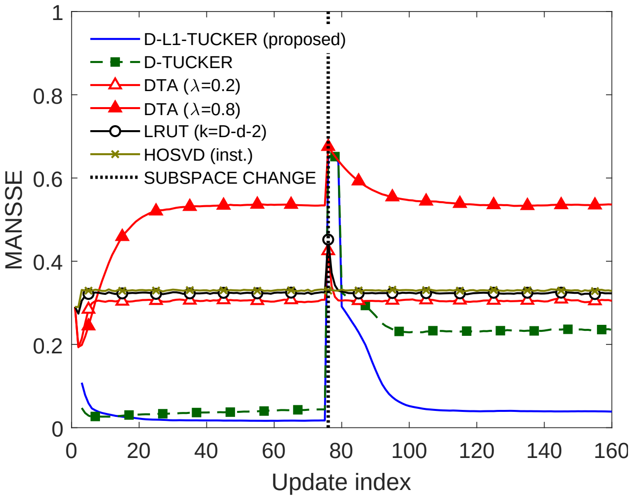


Figure 2.29. MANSSE vs update index. $N = 3$, $D = 10$, $d = 3$, $T_1 = 75$, $T_2 = 85$, $B = 2$, $M = 12$, $W = 4$, SNR $= -6$dB, ONR $= 18$dB, data-driven τ, 20000 realizations.

with (2.47). At the same figure, we illustrate the frequency of rejection; that is, the frequency by which measurements fail the reliability check. We notice that the outlier at $t = 45$ and the W measurements following the change of subspaces are rejected with probability close to 1. In addition, we observe the instantaneous reliability drop when the outlier appears and when nominal subspaces change. For this value of SNR $= -6$dB, the reliability level for nominal measurements is about 0.2 and our data-driven τ is accordingly low.

We conclude this study by comparing the run time of each method across updates. In Fig. 2.28,

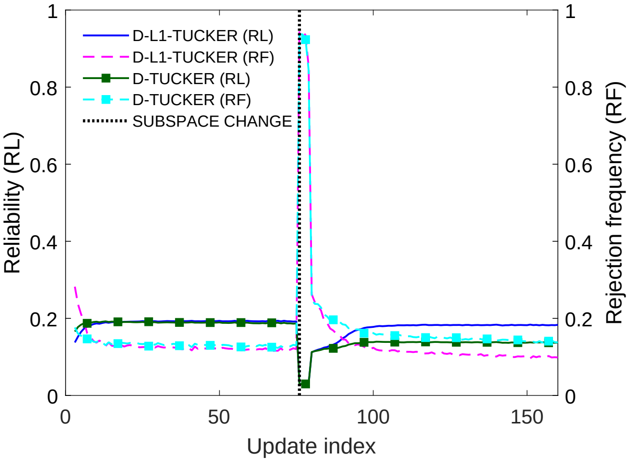


Figure 2.30. Reliability and rejection frequency vs update index. $N = 3$, $D = 10$, $d = 3$, $T_1 = 75$, $T_2 = 85$, $B = 2$, $M = 12$, $W = 4$, SNR $= -6$dB, ONR $= 18$dB, data-driven τ, 20000 realizations.

we plot the instantaneous run times. We observe that the instantaneous HOSVD and DTA exhibit constant run time across updates independently of outliers or changes of subspaces. D-L1-Tucker also exhibits about constant runtime after its memory has been filled. Moreover, we notice an instantaneous drop in the runtime at index $t = 45$ which is because D-L1-Tucker discarded the outlier and did not process it. In contrast, when the outlier appears and when the subspaces change, LRUT attains an increase in runtime, as it tries to adapt.

Next, we repeat the above study. This time, instead of having a fixed outlier at an index, every measurement with index $t > B$ is outlier corrupted with probability $p_o = 0.1$. Moreover, $T_1 = 75$ and $T_2 = 85$. This time, we include a curve which corresponds to a method we label D-Tucker. D-Tucker is identical to D-L1-Tucker with the exception that standard Tucker by means of HOOI is employed instead of L1-Tucker.

In Fig. 2.29, we plot the MANSSE versus update index. We observe that the estimation performance of the DTA curves degrades due to the outliers until a plateau is reached. Their estimation error increases at the subspaces change index and returns to its plateau performance after a few measurements. Expectedly, the instantaneous HOSVD appears to exhibit constant performance for any index $t > B$. A similar observation is made for the LRUT curve with the exception of update indices 75 to 80 where it adjusts to the new underlying data structure. D-Tucker starts from a low MANSSE value and improves for a while, however, its performance slowly drops as measurements are streaming. This is because outliers are passing the reliability check of the L2-norm derived bases which, in turn, affects the performance of the memory batch processing. In contrast, we see that D-L1-Tucker keeps improving its performance up to update index 75 where the underlying

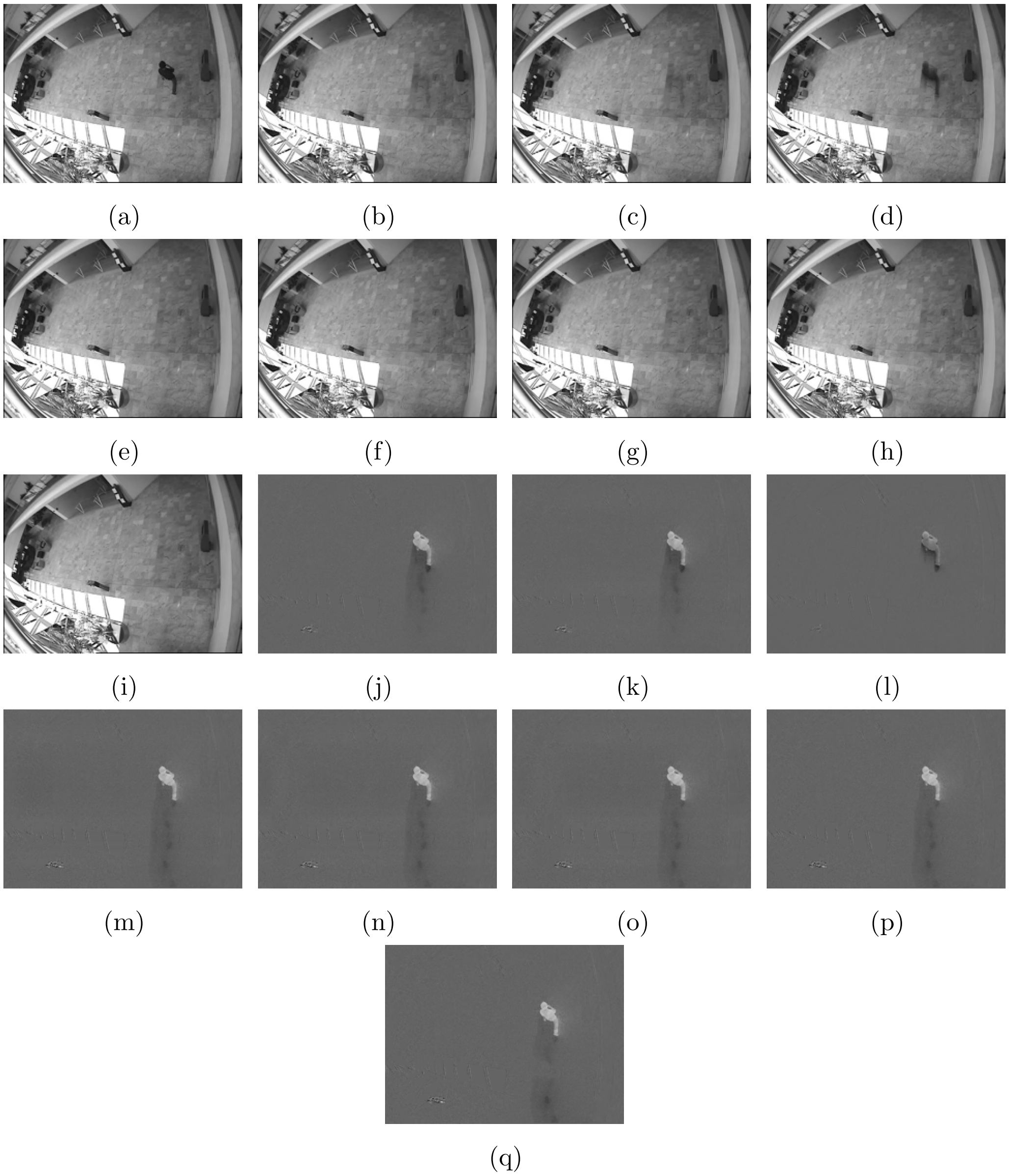
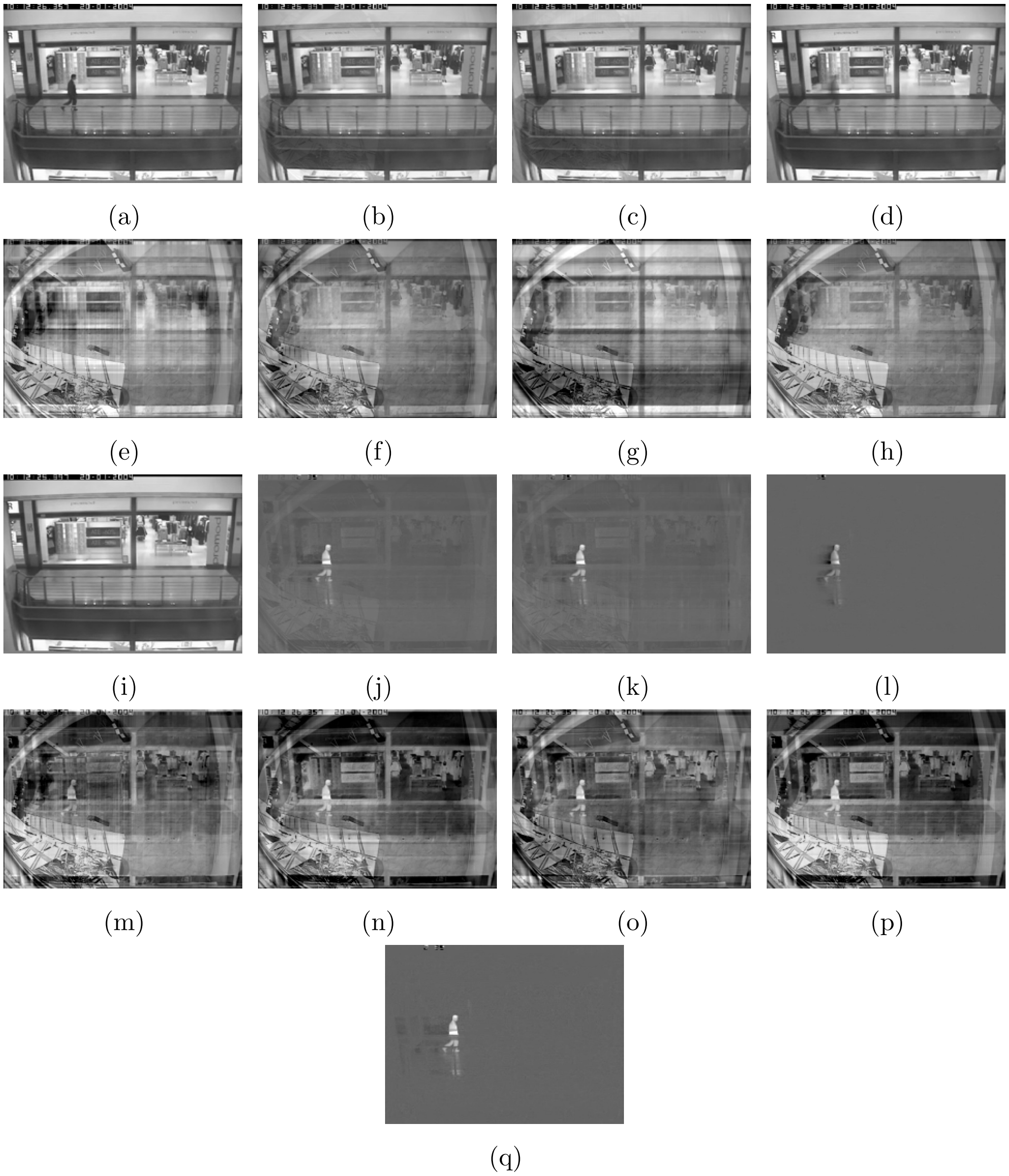


Figure 2.31. Dynamic video foreground/background separation experiment. (a) Original 75-th frame (scene 1). Background extracted by (b) Adaptive Mean ($\lambda = 0.95$), (c) DTA ($\lambda = 0.95$), (d) DTA ($\lambda = 0.7$), (e) LRUT, (f) OSTD, (g) HOOI (increasing memory), (h) L1-HOOI (increasing memory), and (i) D-L1-TUCKER (proposed). Foreground extracted by (j) Adaptive Mean ($\lambda = 0.95$), (k) DTA ($\lambda = 0.95$), (l) DTA ($\lambda = 0.7$), (m) LRUT, (n) OSTD, (o) HOOI (increasing memory), (p) L1-HOOI (increasing memory), and (q) D-L1-TUCKER (proposed).

data structure changes. Then, after the ambiguity batch windows of D-Tucker and D-L1-Tucker are filled, they both reset based on the measurements in their window –each measurement of which



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q)

Figure 2.32. Dynamic video foreground/background separation experiment. (a) Original 150-th frame (scene 2). Background extracted by (b) Adaptive Mean ($\lambda = 0.95$), (c) DTA ($\lambda = 0.95$), (d) DTA ($\lambda = 0.7$), (e) LRUT, (f) OSTD, (g) HOOI (increasing memory), (h) L1-HOOI (increasing memory), and (i) D-L1-TUCKER (proposed). Foreground extracted by (j) Adaptive Mean ($\lambda = 0.95$), (k) DTA ($\lambda = 0.95$), (l) DTA ($\lambda = 0.7$), (m) LRUT, (n) OSTD, (o) HOOI (increasing memory), (p) L1-HOOI (increasing memory), and (q) D-L1-TUCKER (proposed).

is outlier corrupted with probability p_o. Due to its inherent L1-norm robustness, D-L1-Tucker is able to recover and learn the underlying data structure of the new coherence window. In contrast, we see that D-Tucker fails to recover due to the outlying measurements in its ambiguity window.

In Fig. 2.30, we illustrate the reliability values and rejection frequencies as they were computed by the proposed D-L1-Tucker and its counterpart D-Tucker. We observe that during the first coherence window ($t \leq T_1$), both methods exhibit almost identical reliabilities and rejection frequencies. However, in view of Fig. 2.29, we infer that D-L1-Tucker is more successful at rejecting outliers compared to D-Tucker. A few measurements after the change of subspaces at index $t = 75$, we see that D-L1-Tucker converges to reliability and rejection frequency values similar to those of the first coherence window which implies that it has adapted nicely to the new coherence window. On the other hand, the reliability and rejection frequency values to which D-Tucker converges in the second coherence window, both diverge from their corresponding values in the first coherence window which, in turn, implies that D-Tucker was not able to adapt well. This is also depicted in Fig. 2.29 where we see that in the second coherence window D-Tucker converged to a high MANSSE value.

Dynamic Video Foreground/Background Separation

Foreground-background separation is a common task in video processing applications, including moving object tracking, security surveillance, and traffic monitoring. The omnipresent background in a static camera scene determines a nominal subspace, while any foreground movement –e.g., by vehicles or people– represent intermittent outliers. In this experiment, we use D-L1-Tucker to estimate the background and foreground of incoming video frames and compare its performance with that of state-of-the-art alternatives.

For this experiment, we use videos from the CAVIAR database [110]. Specifically, we use two videos, each containing 100 frames of size $(D_1 = 173) \times (D_2 = 231)$ and capturing a different scene. Each video is viewed as collection of frames –i.e., 2-way tensor measurements. We collate the two videos, one behind the other to form the data stream $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times (T=200)}$. Below, we denote by $\bar{\boldsymbol{\mathcal{X}}}_t$ the t-th frontal slab of $\boldsymbol{\mathcal{X}}$ (i.e., the t-th video frame).

We apply the proposed algorithm on $\boldsymbol{\mathcal{X}}$ by setting $d_n = d = 3 \ \forall n \in [2]$, $B = 5$, $M = 10$, $W = 20$, and τ by the proposed batch reliability cross-validation. For every $t \in [T - B] + B$, we obtain bases $\mathbf{Q}_1^{(t)}$ and $\mathbf{Q}_2^{(t)}$ and the mean frame $\boldsymbol{\mathcal{C}}_t$. Accordingly, we estimate the background as $\boldsymbol{\mathcal{X}}_t^{\mathrm{BG}} = \mathbf{Q}_1^{(t)} \mathbf{Q}_2^{(t)}{}^{\top} \left(\bar{\boldsymbol{\mathcal{X}}}_t - \boldsymbol{\mathcal{C}}_t\right) \mathbf{Q}_2^{(t)} \mathbf{Q}_2^{(t)}{}^{\top} + \boldsymbol{\mathcal{C}}_t$ and the foreground as $\boldsymbol{\mathcal{X}}_t^{\mathrm{FG}} = \boldsymbol{\mathcal{X}}_t - \boldsymbol{\mathcal{X}}_t^{\mathrm{BG}}$.

We compare the performance of the proposed algorithm with that of DTA, LRUT, OSTD,[8] HOOI (increasing batch), and L1-HOOI (increasing batch). For the last two benchmark approaches, at any frame index t, we run HOOI/L1-HOOI on the entire batch $\{\bar{\boldsymbol{\mathcal{X}}}_j\}_{j \in [t]}$, starting from arbitrary initialization. We notice that DTA is capable of tracking scene changes using a forgetting factor λ. Since the background estimation involves mean subtraction, for a fair comparison with the proposed method, we enable mean tracking for DTA by computing $\boldsymbol{\mathcal{C}}_t^{\mathrm{DTA}} = \lambda \boldsymbol{\mathcal{C}}_{t-1}^{\mathrm{DTA}} + (1-\lambda)\bar{\boldsymbol{\mathcal{X}}}_t$, for $\boldsymbol{\mathcal{C}}_1^{\mathrm{DTA}} = \bar{\boldsymbol{\mathcal{X}}}_1$. For all other methods, we compute the mean incrementally at any t as $\boldsymbol{\mathcal{C}}_t = \left((t-1)\boldsymbol{\mathcal{C}}_{t-1} + \bar{\boldsymbol{\mathcal{X}}}_t\right)/t$. For DTA, we use two values of forgetting factor, $\lambda = 0.95, 0.7$ and for LRUT we set the number of additional core dimensions to $k_n = D_n - d - 3 \ \forall n \in [2]$.

In Fig. 2.31 and Fig. 2.32, we present the backgrounds and foregrounds obtained by the proposed method and existing counterparts at the 75-th frame (scene 1) and the 150-th frame (scene 2), respectively. We observe from Fig. 2.31 that HOOI (increasing batch), LRUT, and OSTD perform similarly leaving a trail of a ghostly appearance behind the person in their respective foreground frames. We notice that OSTD and L1-HOOI (increasing batch) perform better with a smoother trail behind the person in their foreground frames. DTA with $\lambda = 0.7$ captures the person in its background, leading to an undesirably smudged foreground estimate. DTA with $\lambda = 0.95$ demonstrates a cleaner foreground estimate, similar to that of the adaptive mean (background estimated by the same adaptive mean that we use for DTA), however their backgrounds contain a ghostly appearance of the person. The proposed method extracts a cleaner background and foreground owing to its outlier rejection capability.

In Fig. 2.32, we demonstrate the performance after the scene changes at $t = 100$ by presenting the estimated backgrounds and foregrounds at frame index $t = 150$. We observe that HOOI, L1-HOOI, OSTD, and LRUT perform poorly because they are not designed to track changes in the scene. DTA with $\lambda = 0.95$ demonstrates better performance compared to that of $\lambda = 0.75$ at frame 75, however, at frame 150, we observe that DTA with $\lambda = 0.9$ captures some of the background from scene 1, while DTA with $\lambda = 0.7$ obtains a clean background and hence a smooth foreground, wherein the person appears slightly blurry. The proposed method is capable of tracking scene changes and we observe that it obtains a good estimate of the background and a clear foreground. To quantify the background/foreground estimation performance, we compute, for every frame, the Peak Signal-to-Noise Ratio (PSNR) defined as PSNR $= 10\mathtt{log}_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)$, where MSE is the mean squared error of the estimated background and the ground truth (clean) background. In Fig. 2.33, we plot the PSNR versus frame index and observe that all methods begin with high PSNR and, as

[8] OSTD –i.e., Online Stochastic Tensor Decomposition– was specifically designed for background/foreground separation in multispectral video sequences [91].

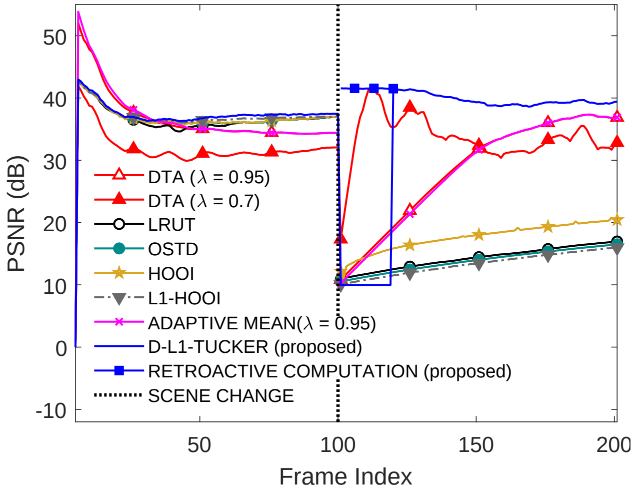


Figure 2.33. Dynamic video foreground/background separation experiment. PSNR (dB) versus frame index.

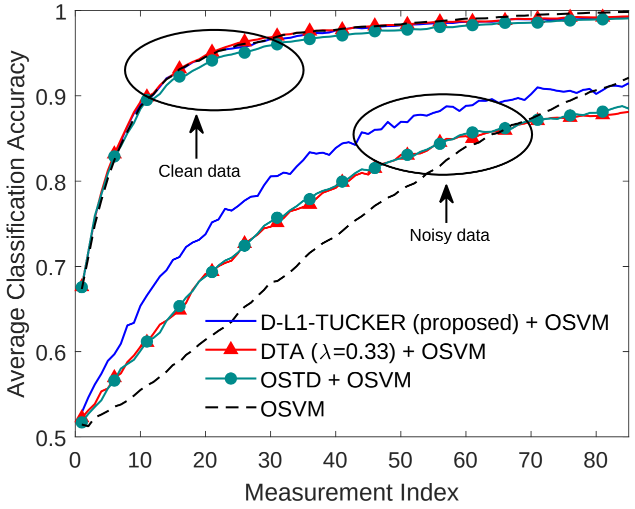


Figure 2.34. Online tensor compression and classification experiment. Average classification accuracy versus update index.

they process frames with foreground movement, the PSNR drops. We observe that the PSNR of the proposed method is the highest after approximately frame 25. When the scene changes, the PSNR of all methods drops instantaneously. The PSNR values of HOOI, L1-HOOI, LRUT, and OSTD increase at a low rate as they process frames from the new scene. Adaptive mean and DTA with $\lambda = 0.95$ demonstrate better performance with faster PSNR increase. DTA with $\lambda = 0.5$ adapts to the new scene very quickly, but it is affected by foreground movement (depicted by oscillations in its PSNR values). The proposed method adapts to the new scene after it processes $W = 20$

measurements and attains the highest PSNR values across all frame indices thereafter. Certainly, after adaptation, the proposed method is straightforwardly capable of accurately extracting the background and foreground of all ambiguous frames in $\mathcal{W}$ in a retroactive fashion, as shown in Fig. 2.33.

Online Tucker and Classification

In this experiment, we perform joint Tucker feature extraction and online classification. We use the Extended Yale Face Database B [99], consisting of face images of many individuals, captured at varying illumination. For this experiment, we use the face images of subject 02 and subject 23 to perform binary classification. Each class has 64 images in total, out of which, at every realization, we choose 45 for training and the remaining 19 for testing. The size of each image is originally 192×168 and we down-sample it to 96×84. Therefore, at every realization, we have a tensor with training data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{96 \times 84 \times 90}$ containing 90 measurements in total (45 from each class) and a tensor with testing data $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{96 \times 84 \times 38}$ containing 38 measurements in total (19 from each class). At every realization, we arbitrarily shuffle the training data and follow a parallel online feature extraction and classification approach as follows.

At update index t, we process the t-th training sample in $\boldsymbol{\mathcal{X}}$, $\bar{\boldsymbol{\mathcal{X}}}_t$, and update $\mathbf{Q}_1^{(t)}$ and $\mathbf{Q}_2^{(t)}$ using the proposed method ($B = 5$, $M = 10$, $d_1 = 15$, $d_2 = 6$, and cross-validated τ). Next, we use the updated bases to compress all previously seen training data $\{\bar{\boldsymbol{\mathcal{X}}}_i\}_{i \in [t]}$ as $\boldsymbol{\mathcal{Z}}_i = \mathbf{Q}_1^{(t)\top} \bar{\boldsymbol{\mathcal{X}}}_i \mathbf{Q}_2^{(t)}$. We vectorize the compressed training measurements and give them as input to the Online Support Vector Machine (OSVM) classifier of [111].[9] We test the performance of the classifier on testing data, compressed using the same bases, and record the classification accuracy for every update index t. We repeat the experiment 300 times and plot the average classification accuracy versus update index in Fig. 2.34. Along with the proposed algorithm, we also plot the performance of the plain OSVM classifier, i.e., OSVM classifier run on vectorized (uncompressed) data, DTA with $\lambda = 0.33$, and OSTD. In Fig. 2.34, we observe that all compared methods attain almost identical performance. The classification accuracy starts low and as the update index increases it tends to 1.

Next, we repeat the experiment with the same setting, by corrupting each training measurement outside the initial memory batch $\mathcal{B}$ with noise from $\mathcal{N}(2, 5)$, cropping pixel intensities outside $[0, 255]$. We compute the average classification accuracy over 300 realizations and plot it versus

[9] Matlab code available at https://www.cpdiehl.org/code.html.

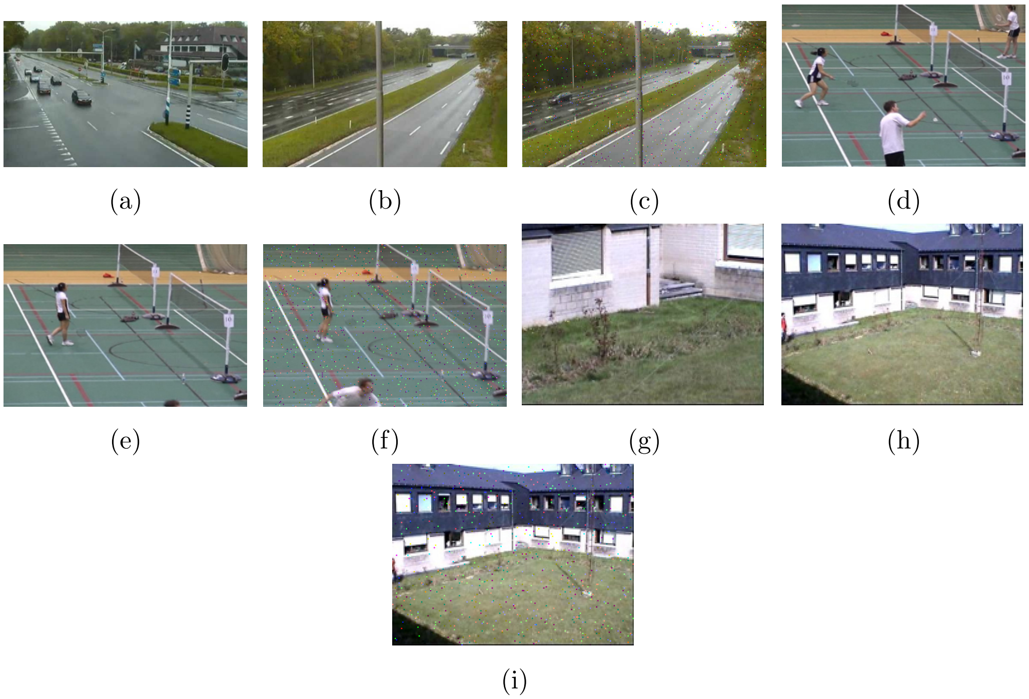

(a) (b) (c) (d)
(e) (f) (g) (h)
(i)

Figure 2.35. Frame instances per scene for the three videos. Video 1: (a) scene 1, (b) scene 2, and (c) noisy frame. Video 2: (d) scene 1, (e) scene 2, and (f) noisy frame. Video 3: (g) scene 1, (h) scene 2, and (i) noisy frame. Probability of noise corruption per pixel is 10% for all noisy frames.

update index in Fig. 2.34. In this case, we notice that plain OSVM is significantly affected by the noise. DTA, OSTD, and D-L1-Tucker demonstrate resistance to noise corruption, especially for earlier update indices.

Online Video Scene Change Detection

In this experiment, we demonstrate the efficacy of the proposed method in online video scene change detection. We operate on Red-Green-Blue (RGB) videos from the benchmark video scene change detection dataset in [112]. Specifically, we operate on three video frame-sequences from the dataset. Video 1 (`twoPositionPTZCam`) captures a street with a single scene change. Video 2 (`badminton`) captures badminton players in action with camera jittering. Video 3 (`zoomInzoomOut`) captures the backyard of a house with a single scene change. Each video can be seen as a collection of 3-way tensors $\bar{\boldsymbol{\mathcal{X}}}_t \in \mathbb{R}^{D_1 \times D_2 \times D_3}$ which are the video frames streaming across time $t = 1, 2, \ldots$. Videos 1, 2, and 3 are cropped to consist of $T = 202$, 220, and 172 frames, respectively, and each frame is of size $85 \times 143 \times 3$, $94 \times 144 \times 3$, and $60 \times 80 \times 3$, respectively. For videos 1 and 3, scene change

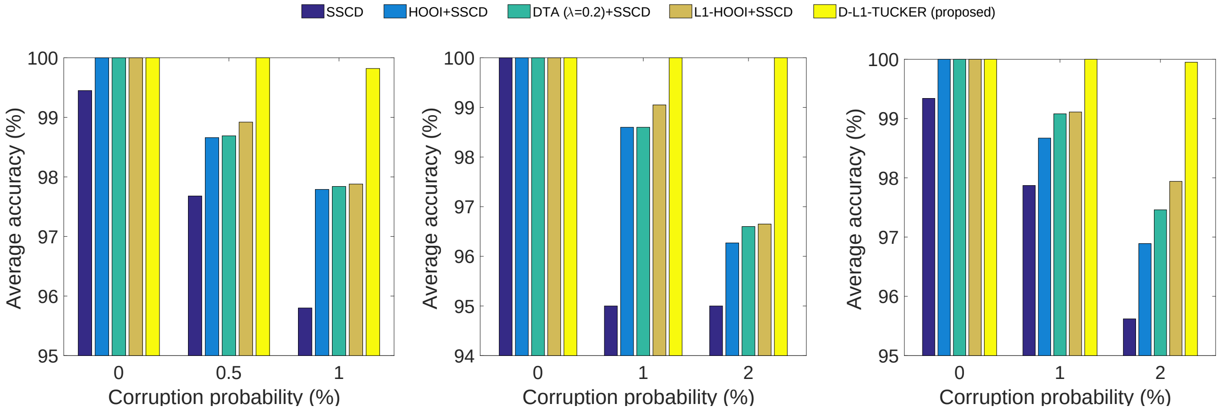


Figure 2.36. Average online video scene change detection accuracy versus probability of noise corruption per pixel. For videos 1 and 3, the probability of frame corruption p_f is set to 0.1 while for video 2 it is set to 0.25. Video 1 (left), video 2 (middle), and video 3 (right).

occurs at frames 102 and 92, respectively.

We run the proposed algorithm on all three videos with $B = 20$, $M = 10$, $W = 5$, and cross-validated threshold τ set to 88% of the median reliability of the initialization batch. For all videos we set $d_3 = 1$. For videos 1 and 2 we set $d_1 = d_2 = 2$ while for video 3 we set $d_1 = d_2 = 5$.

A scene change is detected when the ambiguity batch overflows. The frame index of the first frame to enter the ambiguity batch is returned as the index of scene change. We quantify the performance of the proposed algorithm by means of the standard accuracy metric

$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}, \tag{2.54}$$

where TP is the number of correct scene change detections, TN is the number of frames correctly identified as non-scene-changes, FP is the number of falsely identified scene changes, and FN is the number of missed detections. We compare the performance of the proposed algorithm with that of the state-of-the-art subspace-based scene change detection (SSCD) method in [113]. In addition, we extend SSCD to handle multi-way/tensor subspaces and replace the matrix products in Algorithm 1 of [113] by tensor products. Then, we apply SSCD on the tensor bases obtained by means of HOOI (batch), L1-HOOI (batch), and DTA ($\lambda = 0.2$). The values of d_1, d_2, and d_3 are the same as those used with the proposed algorithm. Other hyper-parameters of SSCD include positive constants b and c which are optimally tuned, individually for each method. To evaluate the robustness of each method against corruptions, we corrupt each frame of all videos with probability p_f. To each pixel of a corrupted frame, we add salt-and-pepper noise with probability p_n. In Fig.

2.35, we illustrate a frame instance per scene for each video along with a noisy frame. For every value of p_n, we repeat the experiment 250 times.

For video 1 and $p_f = 0.1$, we illustrate the average detection accuracy of each method in Fig. 2.36 (left). We observe that under nominal conditions (no noise), the proposed algorithm and the tensor-based SSCD methods (HOOI + SSCD, L1-HOOI + SSCD, and DTA + SSCD), with $c = 10^7$ and $b = 1$ attain perfect performance by correctly identifying the scene change, without any false positives. In contrast, the plain SSCD method with $c = 10^{7.4}$ and $b = 1$ that relies on the $K = 2$ orthonormal bases obtained by SVD on the video frames demonstrates slightly degraded performance. This can be attributed to the loss of spatial context when the video frames are vectorized. As p_n increases in steps of 0.005, we notice that plain SSCD is affected the most, followed by HOOI + SSCD and DTA + SSCD. L1-HOOI + SSCD exhibits some robustness to noise, comparatively. The proposed method maintains the best performance across the board.

For video 2 and $p_f = 0.25$, we report the average detection performance of each method in Fig. 2.36 (middle). The tensor-based SSCD methods use the same parameters as before while plain SSCD uses the same K value, $c = 10^{7.5}$, and $b = 1$. Because this video is jittery, SSCD methods exhibit some robustness to noise for small p_n. As p_n increases in steps of 1%, SSCD methods perceive camera jitter as scene change. In contrast, D-L1-Tucker remains robust to noise and is not misled.

Finally, for Video 3 and $p_f = 0.1$, we plot the detection performance of all methods in Fig. 2.36 (right). p_n varies in steps of 0.01 from 0 to 0.02. We use the same parameters for the tensor-based SSCD methods while for the plain SSCD method we use the same K, $c = 10^{7.25}$, and $b = 1$. We observe similar results to those of Fig. 2.36 (left) and Fig. 2.36 (middle). Although p_n is small, the number of corrupted pixels per frame is significant –e.g., consider video 1 where each frame is of size $85 \times 143 \times 3$ and, on average, a probability of pixel corruption $p_n = 0.01$ results in approximately 365 noisy pixels per corrupted frame.

Online Anomaly Detection

We consider the "Uber Pickups" tensor of the Formidable Repository of Open Sparse Tensors and Tools (FROSTT) [106] which is a $(N = 4)$-way tensor of size D_1-by-D_2-by-D_3-by-D_4 where $D_1 = 1140$ latitudes, $D_2 = 1717$ longitudes, $D_3 = 24$ hours, and $D_4 = 183$ days. Each entry of the tensor models number of Uber pickups in New York City over a period of about 6 months.

Pre-processing: To make the tensor more manageable in size, we first take a summation across the day mode and obtain a size D_1-by-D_2-by-D_3 tensor where $D_3 = 183$ days –i.e., a collection of 183 size 1140-by-1717 matrix measurements. We further reduce the size of the matrix measurements by retaining a 250-by-250 area centered at Manhattan wherein most of the activity –in terms of Uber Pickups– occurs. We consider the resulting tensor $\boldsymbol{\mathcal{X}}_{\text{uber}} \in \mathbb{R}^{250\times 250\times 183}$ to be a collection of 183 streaming measurements, one for each day.

Streaming processing: $\boldsymbol{\mathcal{X}}_{\text{uber}}$ can be seen as a data stream of matrix measurements each of which corresponds to a day. Accordingly, 7 successive measurements across the day index must correspond to a week which, in turn, is separated into weekdays and Saturdays. We assume that traffic during the weekdays is not the same as traffic on Saturdays and conjecture that weekdays belong to a coherent class/distribution while Saturdays belong to another.

We assume that we are given $B = 5$ measurements that correspond to weekdays and use those measurements to initialize D-L1-Tucker with memory size $M = 5$ and $d = 10$ components per mode. Moreover, we leverage these B measurements to tune τ using the leave-one-out cross-validation approach that we presented above. We set τ to the median value of the B collected reliability values in $\mathbf{r}$. Then, we update the decomposition of D-L1-Tucker by processing the rest of the measurements one by one. In Fig. 2.37, we plot the reliability of streaming measurements versus day (update) index. Moreover, we add the data-driven value of τ as a horizontal line. Each measurement with reliability value above this curve is deemed reliable for processing, while each measurement with reliability value below that curve is considered to be an anomaly (outlier). For better understanding, we include vertical dotted lines on the day indices which correspond to Saturdays. The reported curves are averaged over 300 random initializations of bases and runs of D-L1-Tucker on $\boldsymbol{\mathcal{X}}_{\text{uber}}$. Quite consistently, days that correspond to Saturdays exhibit reliability values that are clearly below the τ threshold and are considered anomalies. In contrast, almost all weekdays exhibit reliability values above the τ threshold. A different selection of the threshold τ may slightly improve the results, however, even with the data-driven tuning of τ, the reliability check feature of D-L1-Tucker offers high accuracy in identifying anomalies/outliers.

2.7 Conclusions

In this Chapter, we studied L1-Tucker and L1-Tucker2 decompositions for outlier-resistant analysis of tensors, derived new theory, and proposed new algorithms. Our contributions are summarized as follows: (i) We showed that L1-Tucker2 decomposition can be cast to combinatorial optimization and offered the first algorithms for its solution. Numerical studies corroborate the outlier-resistance

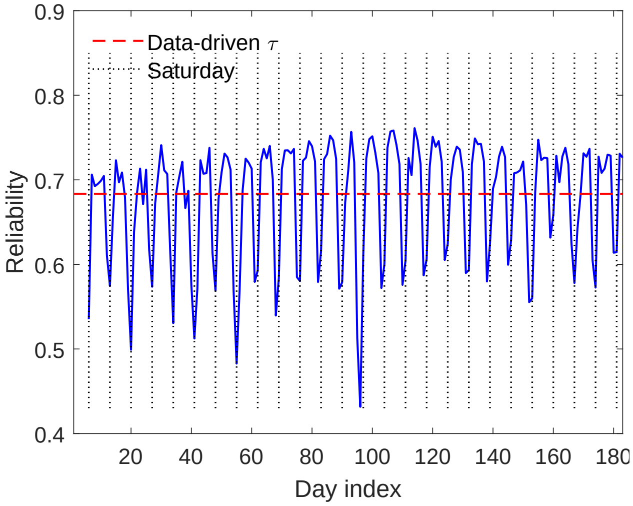


Figure 2.37. Online anomaly detection. $B = 5$, $M = 5$, $d = 10$, 300 realizations, data-driven τ.

of L1-Tucker2 against standard counterparts. (ii) We presented L1-Tucker and L1-Tucker2 decomposition of general-order tensors and proposed two new algorithmic frameworks for their solution, L1-HOSVD and L1-HOOI. The presented methods are accompanied by formal convergence and complexity analyses. Our numerical studies on data restoration and classification demonstrate that, in contrast to standard Tucker, L1-Tucker decomposition exhibits sturdy resistance against corruptions among the processed data. (iii) We presented D-L1-Tucker: an algorithm for dynamic and outlier-resistant Tucker analysis of tensor data. Our experimental studies on real and synthetic datasets corroborate that the proposed method (1) attains high bases estimation performance, (2) suppresses outliers, and (3) adapts to changes of the nominal subspaces.

Chapter 3

Lp-quasinorm Principal-Component Analysis

3.1 Introduction

Principal-Component Analysis (PCA) is a fundamental method for data analysis, machine learning, and pattern recognition [114]. PCA finds applications in virtually every field of science and engineering. Computer vision, robotics, neuroscience, signal processing, and medicine are just a few applications in which PCA has successfully been employed. PCA is typically used for dimensionality reduction, denoising, classification, clustering, and feature extraction, among others.

PCA is typically formulated as a L2-norm error minimization, or, equivalently, a L2-norm projection maximization problem. Mathematically, for given matrix $\mathbf{X}$ of size D-by-N, PCA seeks a collection of K D-dimensional unit-norm vectors arranged as columns of a subspace basis matrix $\mathbf{Q}$ that minimizes $\|\mathbf{X} - \mathbf{Q}\mathbf{Q}^\top\mathbf{X}\|_F^2$, or equivalently, maximizes $\|\mathbf{X}^\top\mathbf{Q}\|_F^2$, where the squared Frobenius/L2 norm $\|\cdot\|_F^2$ returns the sum of the squared entries of its argument. The solution to PCA is typically computed by Singular-Value Decomposition (SVD) of $\mathbf{X}$ [115]. Considering that $\mathbf{X} \in \mathbb{R}^{D\times N}$ has rank $\rho \leq \{D, N\}$ and admits SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}_{\rho\times\rho}\mathbf{V}^\top$ such that $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}_\rho$, $\mathbf{\Sigma} = \text{diag}(\boldsymbol{\sigma})$, and $[\boldsymbol{\sigma}]_1 \geq [\boldsymbol{\sigma}]_2 \geq \ldots \geq [\boldsymbol{\sigma}]_\rho > 0$, then the solution to PCA is given by the K left-hand signular-valued singular vectors of $\mathbf{X}$ –i.e., $\mathbf{Q} = [\mathbf{U}]_{:,1:K}$ solves PCA of $\mathbf{X}$.

Despite its documented success, PCA is known to exhibit severe sensitivity against outliers within the processed data [47, 48]. Outliers are high-magnitude/peripheral data points that lie far away from the nominal data subspace and commonly appear in modern datasets –e.g., due to data storage/transfer errors, faulty sensors, or deliberate data contamination in adversarial environments [116]. The outlier-sensitivity of PCA is often attributed to its L2-norm-based formulation which places squared emphasis to each data point, thus promoting the impact of outliers. Accordingly, applications that rely on PCA are severely affected when outliers exist in the processed data [117].

To counteract the impact of outliers, researchers have proposed an array of "robust" PCA formulations [117, 118]. For instance, one approach considers a L1-norm based residual-error minimization formulation the (approximate) solution to which is computed by means of alternating-optimization [119]. Similarly, the authors of [120] study low-rank approximation of a matrix by a residual error minimization formulation based on the Lp-norm for $p \geq 1$. The author of [121] considers the same problem and presents practical algorithms specifically for $p = 1$ or $p = \infty$.

A more straightforward approach considers the projection maximization PCA formulation and replaces the outlier-sensitive L2-norm by the more robust L1-norm (special case of Lp-PCA for $p = 1$). This results in the popular L1-norm PCA (L1-PCA) formulation. L1-PCA was solved exactly in [53, 54]. Efficient [58, 73], adaptive [77], incremental [122], stochastic [123], and complex-valued data [124, 125] solvers have also been proposed. L1-PCA has been successfully employed in an array of applications [55–57, 126, 127] where it has been documented that it attains similar performance to PCA when the processed data are nominal/clean while it exhibits strong resistance against outliers. L1-norm formulations have recently been proposed for robust tensor analysis –e.g., L1-Tucker [59, 63] and L1-Rescal [128]). Similar to L1-PCA, L1-Tucker and L1-Rescal exhibited significant robustness against data corruptions.

Interestingly, Lp-norm formulations and algorithms have been studied for popular problems other than PCA. For example, the authors of [129] considered the popular Independent Component Analysis (ICA) problem and present a variant formulated based on the Lp-norm. Authors in [130, 131] have studied Linear Discriminant Analysis (LDA) formulations based on the Lp-norm. Block Principal-Component Analysis has also been studied from the scope of Lp-norm formulations [132]. Finally, Lp-norm based constraints have been considered for infrared small target detection [133].

In this work, we study the problem of projection maximization Lp-quasinorm ($p \leq 1$) Principal-Component Analysis, the theoretical foundations of which, remain largely unexplored to date. This problem was originally studied in [134] where the author considered any general value of $p > 0$ and presented algorithms for the solution to Lp-PCA. However, the presented algorithms are only convergent for the special case that $p \geq 1$. Moreover, the same problem was also studied in [135] where a Grassmann manifold optimization algorithm was proposed for the solution to Lp-PCA. This algorithm is guaranteed to converge to a locally optimal solution. For $p \leq 1$, to the best of our knowledge, to date there exists no algorithm for the exact solution to Lp-PCA. With this work, we strive to fill this void.

In this Chapter, we study the theoretical foundations of Lp-quasinorm PCA and propose algorithms for its solution. Specifically, our contributions are as follows:

Contribution i. We show that, for $K = 1$, Lp-PCA can always be solved exactly by means of combinatorial optimization and offer the first algorithms for its solution.

Contribution ii. For the special case of non-negative data, we show that a combinatorial search can be evaded and finding the dominant Lp-PC of a matrix simplifies to a convex problem.

Contribution iii. We show that, for general $K \geq 1$, the solution to Lp-PCA can also be pursued by combinatorial optimization and present a novel near-exact algorithm.

Contribution iv. We propose a refining strategy, that, in some cases, can further enhance a near-exact solution.

The rest of this Chapter is organized as follows. In Section 3.2, we introduce the problem statement. In Section 3.3, we present exact algorithms for Lp-PCA ($K = 1$). In Section 3.4, we show that, for non-negative data, computing the dominant Lp-norm component simplifies to a convex problem. In Section 3.5, we present a near-exact algorithm for jointly computing multiple Lp-norm PCs. In Section 3.6, we propose a solution-refinement strategy that, in some cases, can further enhance a near exact solution. Section 3.7 holds numerical experimental studies on synthetic and real-world medical data. Concluding remarks are drawn in Section 3.8.

The contributions presented in this Chapter have also been presented in [136–138].

3.2 Problem Statement

Before we present the problem statement, we introduce notation that will facilitate the presentation of this Chapter.

Vectors and matrices are denoted by lower-case and upper-case bold letters –i.e., $\mathbf{a}$ and $\mathbf{A}$, respectively. $\mathbb{S}_{D\times K} = \{\mathbf{Q} \in \mathbb{R}^{D\times K} : \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_K\}$ is the Stiefel manifold of all rank-K orthonormal matrices in $\mathbb{R}^D$. The set $\mathbb{B}_{D\times K} := \{\mathbf{Q} \in \mathbb{R}^{D\times K} : \|\mathbf{Q}\|_2 \leq 1\}$ is the convex hull of $\mathbb{S}_{D\times K}$. For the special case $K = 1$, $\mathbb{B}_{D\times K}$ boils down to the unit-radius hyperball. For any $\mathbf{A} \in \mathbb{R}^{N\times K}$ and $p > 0$, $\|\mathbf{A}\|_p = \|\mathrm{vec}(\mathbf{A})\|_p = (\sum_{n=1}^{N}\sum_{k=1}^{K} |[\mathbf{A}]_{n,k}|^p)^{\frac{1}{p}}$. For $p \geq 1$, $\|\cdot\|_p$ is a *proper norm* function that, for any $\mathbf{x} \in \mathbb{R}^D$, $\mathbf{y} \in \mathbb{R}^D$, and $\alpha \in \mathbb{R}$, satisfies (i) $\|\alpha\mathbf{x}\|_p = |a|\|\mathbf{x}\|_p$, (ii) $\|\mathbf{x}\|_p = 0$ implies that $\mathbf{x} = \mathbf{0}_D$, and (iii) $\|\mathbf{x}+\mathbf{y}\|_p \leq \|\mathbf{x}\|_p + \|\mathbf{y}\|_p$. In contrast, for $p \leq 1$, $\|\cdot\|_p$ is a *quasinorm* function and satisfies (i) and (ii) while (iii) is substituted by $\|\mathbf{x}+\mathbf{y}\|_p \leq C(\|\mathbf{x}\|_p + \|\mathbf{y}\|_p)$, where it holds that the constant $C = 2^{\frac{1-p}{p}} > 0$ [139]. $\mathrm{sgn}\,(\cdot)$ returns the signs of its input argument –i.e, $\mathrm{sgn}\,(\alpha) = +1$ if $\alpha > 0$ and $\mathrm{sgn}\,(\alpha) = -1$ if $\alpha < 0$. Without loss of generality, we set $\mathrm{sgn}\,(\alpha) = +1$ if $\alpha = 0$. For any positive integer N, $[N] = \{1, 2, \ldots, N\}$. "$\odot$" denotes the element-wise multiplication operation.

We consider a collection of N vector measurements arranged as columns of $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$. The problem of interest is *Lp-quasinorm Principal-Component Analysis* (Lp-PCA) of $\mathbf{X}$, for any $p \leq 1$, –i.e., we seek a column-wise orthonormal matrix corresponding to a lower dimensional signal subspace such that the Lp-quasinorm derived projection variance is maximized. Mathematically, for any $K < \mathrm{rank}(\mathbf{X}) \leq \min\{D, N\}$, Lp-PCA is formulated as

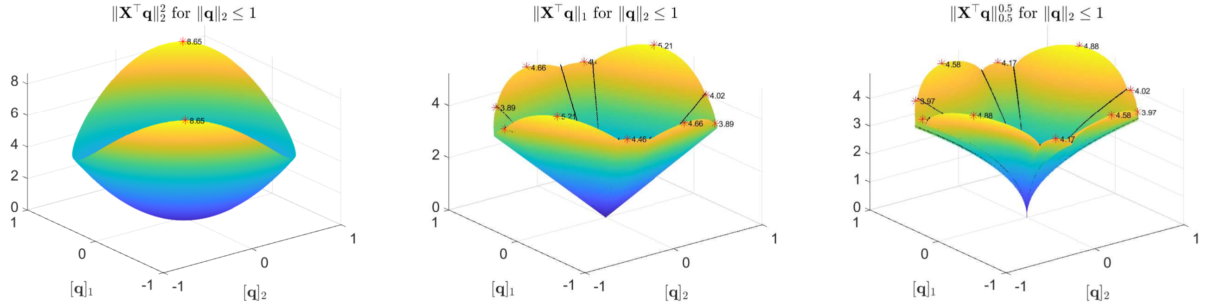
$$\underset{\mathbf{Q}\in\mathbb{S}_{D\times K}}{\max.} \left\|\mathbf{X}^\top\mathbf{Q}\right\|_p^p. \tag{3.1}$$

A solution to Lp-PCA exists only for the very special case that $p = 1$. For the area of interest –general values of $p \leq 1$– the solution to Lp-PCA remains unknown to date.

The problem of finding the exact solution to Lp-PCA in (3.1) (or, nearly-exact in a few cases), is the main focus of this work. Before we commence our developments, we present Lemma 3.1 which states that, without loss of generality (w.l.o.g.), it suffices to solve (3.1) for a full row rank matrix.

Lemma 3.1. *$\mathbf{X}$ has rank-ρ and admits SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{U} \in \mathbb{S}_{D\times\rho}$, $\mathbf{V} \in \mathbb{S}_{N\times\rho}$, and $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma}) \in \mathbb{R}^{\rho\times\rho}$. Let $\mathbf{Y} = \mathbf{\Sigma}\mathbf{V}^\top$. If $\mathbf{A}_{\mathrm{opt}}$ solves $\max._{\mathbf{A}\in\mathbb{S}_{\rho\times K}} \|\mathbf{Y}^\top\mathbf{A}\|_p^p$, then $\mathbf{Q}_{\mathrm{opt}} = \mathbf{U}\mathbf{A}_{\mathrm{opt}}$ solves* (3.1).

By Lemma 3.1, we henceforth consider that $\mathbf{X}$ has full row rank –i.e., $\rho = D \leq N$.



(a) $p = 2$ (b) $p = 1$ (c) $p = 0.5$

Figure 3.1. Metric surface of $\|\mathbf{X}^\top\mathbf{q}\|_p^p$ for arbitrary $\mathbf{X} \in \mathbb{R}^{2\times 5}$ and $p \in \{2, 1, 0.5\}$.

3.3 Contribution 1: The Lp-quasinorm Principal-Component

First, we steer our focus on the special case $K = 1$ in which the problem of interest simplifies to

$$\underset{\mathbf{q}\in\mathbb{S}_{D\times 1}}{\text{max.}} \left\|\mathbf{X}^\top\mathbf{q}\right\|_p^p. \tag{3.2}$$

For context, we illustrate the cost function of (3.2) for arbitrary $\mathbf{X} \in \mathbb{R}^{2\times 5}$ over the unit-radius hyperball and for $p = 2$, $p = 1$, and $p = 0.5$, in Fig. 3.1a, Fig. 3.1b, and Fig. 3.1c, respectively. For $K = 1$, the Stiefel manifold simplifies to the D-dimensional unit-radius hypersphere which comprises all D-dimensional unit-norm vectors.

Next, we present Lemma 3.2.

Lemma 3.2. *For any* $\mathbf{X} \in \mathbb{R}^{D\times N}$*, solving* (3.2) *is equivalent to solving* $\max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p$.

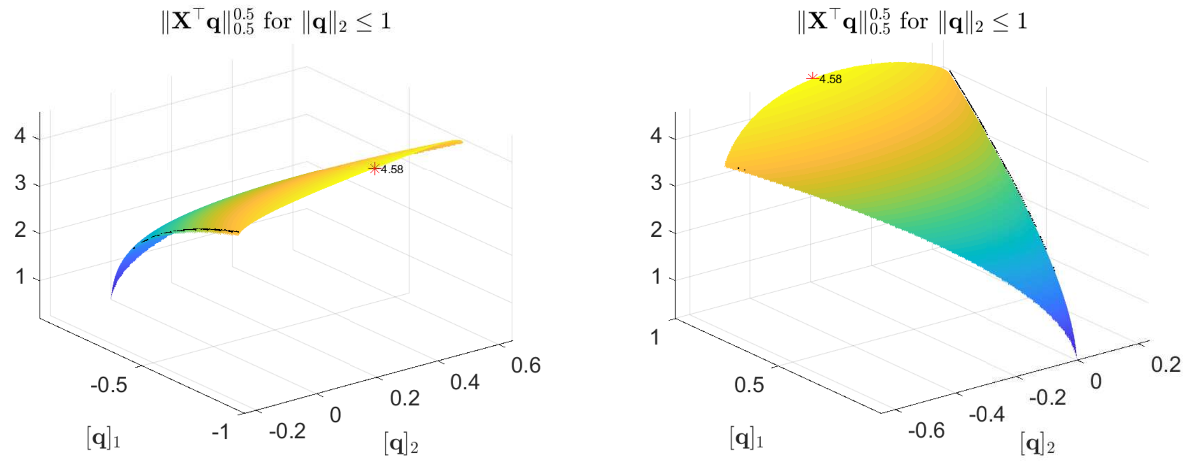
Lemma 3.2 states that the solution to Lp-PCA can equivalently be pursued over the closed unit-radius D-dimensional hyperball. Next, we consider $\mathbf{b} = \{\pm 1\}^N$ and define the set

$$\mathcal{C}(\mathbf{b}) = \{\mathbf{q} \in \mathbb{B}_{D\times 1} : \text{sgn}\left(\mathbf{X}^\top\mathbf{q}\right) = \mathbf{b}\}. \tag{3.3}$$

In practice, $\mathcal{C}(\mathbf{b})$ is a subset of the unit-radius hyperball. Some instances of $\mathbf{b}$ will be such that $\mathcal{C}(\mathbf{b}) = \emptyset$. Further, every distinct $\mathbf{b}$ uniquely identifies a subset of the hyperball in accordance with the following Lemma 3.3.

Lemma 3.3. *For every* $\mathbf{b}, \mathbf{b}' \in \{\pm 1\}^N$ *such that* $\mathbf{b} \neq \mathbf{b}'$*, it holds that* $\mathcal{C}(\mathbf{b}) \cap \mathcal{C}(\mathbf{b}') = \emptyset$*.*

All sets $\mathcal{C}(\mathbf{b})$ formed by $\mathbf{b}$ when it scans the finite-size set $\{\pm 1\}^N$ are, in practice, a partition of



(a) $\mathbf{b} = [-1, 1, -1, 1, -1]^\top$ (b) $\mathbf{b} = [1, -1, 1, -1, 1]^\top$

Figure 3.2. Metric surface of $\|\mathbf{X}^\top\mathbf{q}\|_p^p$ for $p = 0.5$ and arbitrary $\mathbf{X} \in \mathbb{R}^{2\times 5}$ when $\mathbf{q}$ scans $\mathcal{C}(\mathbf{b})$ for fixed $\mathbf{b}$.

Input: $\mathbf{X} \in \mathbb{R}^{D\times N}$, $\mathbf{b} \in \{\pm 1\}^N$, $p \leq 1$
Output: $\mathbf{q} \in \mathbb{R}^D$

```
Y = X * diag(b);
cvx_begin
variable q(D)
maximize(sum(pow_p(Y' * q, p))
q' * q <= 1
Y' * q >= 0
cvx_end
```

Figure 3.3. CVX code (Matlab) for the solution to (3.4).

the hyperball which corresponds to the data matrix $\mathbf{X}$ as shown in the following Lemma 3.4.

Lemma 3.4. *It holds that* $\mathbb{B}_{D\times 1} = \bigcup_{\mathbf{b}\in\{\pm 1\}^N} \mathcal{C}(\mathbf{b})$.

By Lemma 3.2 and Lemma 3.4, $\max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \left\|\mathbf{X}^\top\mathbf{q}\right\|_p^p = \max_{\mathbf{b}\in\{\pm 1\}^N} \max_{\mathbf{q}\in\mathcal{C}(\mathbf{b})} \left\|\mathbf{X}^\top\mathbf{q}\right\|_p^p$. Accoridngly, the solution to Lp-PCA can be pursued by combinatorial search over $\{\pm 1\}^N$ –i.e., by separately solving

$$\max_{\mathbf{q}\in\mathcal{C}(\mathbf{b})} \left\|\mathbf{X}^\top\mathbf{q}\right\|_p^p \tag{3.4}$$

for every $\mathbf{b} \in \{\pm 1\}^N$.

Next, we show that the solution to (3.4) exists and can be found exactly, in accordance with the

```
Input: $\mathbf{X} \in \mathbb{R}^{D \times N}$, $\mathbf{b} \in \{\pm 1\}^N$, $p \leq 1$
Output: $\mathbf{q} \in \mathbb{R}^D$
$v \leftarrow 0$
For every $\mathbf{b} \in \{\pm 1\}^N$
    $\mathbf{q} \leftarrow \text{argmin}_{\mathbf{q} \in \mathcal{C}(\mathbf{b})} - \sum_{n \in [N]} ([\mathbf{b}]_n \mathbf{q}^\top \mathbf{x}_n)^p$
    If $\|\mathbf{X}^\top \mathbf{q}\|_p^p > v$, $v \leftarrow \|\mathbf{X}^\top \mathbf{q}\|_p^p$, $\mathbf{q}_{\text{Lp}} \leftarrow \mathbf{q}$
```

Algorithm 3.1 Exhaustive search for the exact solution to Lp-PCA.

following Lemma 3.5 and Lemma 3.6.

Lemma 3.5. *For any $\mathbf{b} \in \{\pm 1\}^N$, $\mathcal{C}(\mathbf{b})$ is a convex set.*

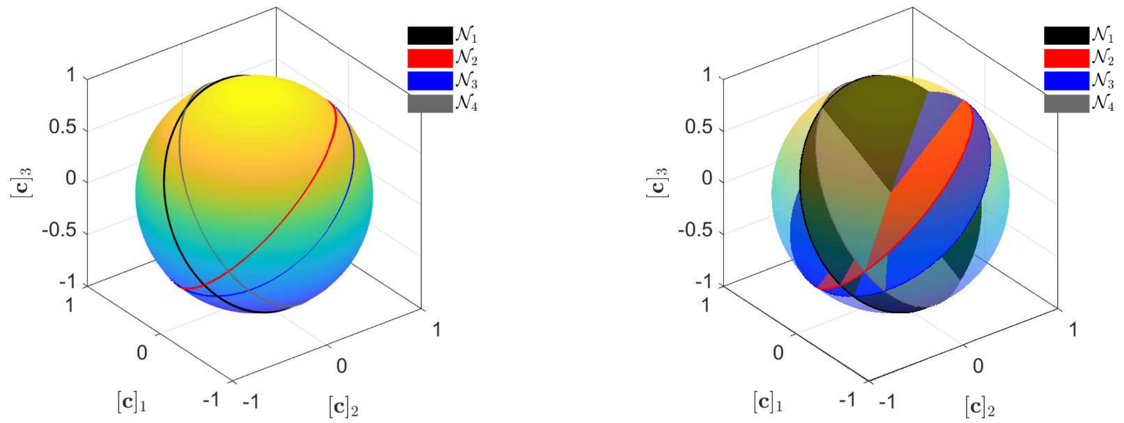
Lemma 3.6. *Let $p \leq 1$ and $\mathbf{b} \in \{\pm 1\}^N$ such that $\mathcal{C}(\mathbf{b}) \neq \emptyset$. For any $\mathbf{q} \in \mathcal{C}(\mathbf{b})$, $[\mathbf{b}]_n \mathbf{q}^\top \mathbf{x}_n \geq 0 \forall n \in [N]$. In turn, $\|\mathbf{X}^\top \mathbf{q}\|_p^p = \sum_{n \in [N]} |\mathbf{x}_n^\top \mathbf{q}|^p = \sum_{n \in [N]} ([\mathbf{b}]_n \mathbf{q}^\top \mathbf{x}_n)^p$ is concave with respect to $\mathbf{q} \in \mathcal{C}(\mathbf{b})$.*

In Fig. 3.2, we plot $\|\mathbf{X}^\top \mathbf{q}\|_p^p$ for $p = 0.5$ when $\mathbf{q}$ scans $\mathcal{C}(\mathbf{b})$ for fixed $\mathbf{b}$ where Lemma 3.6 is illustrated –i.e., $\|\mathbf{X}^\top \mathbf{q}\|_p^p$ is piece-wise concave for $p \leq 1$. By Lemma 3.5 and Lemma 3.6, Proposition 2.25 follows.

Proposition 3.1. *The optimization problem in (3.4) is a convex problem and can be equivalently rewritten as $\min_{\mathbf{q} \in \mathcal{C}(\mathbf{b})}. - \sum_{n \in [N]} ([\mathbf{b}]_n \mathbf{q}^\top \mathbf{x}_n)^p$.*

3.3.1 Exact Algorithm 1: Exhaustive Search

By Proposition 3.1, for any $\mathbf{b} \in \{\pm 1\}^N$, (3.4) is a convex problem and can be solved exactly –e.g., by interior point methods. For instance, considering a standard primal-dual interior-point solver based on Newton's method [140, 141], (3.4) can be solved with about cubic cost in D, N. More sophisticated solvers with lower computational cost can also be derived, but this is beyond the scope of this Chapter. For simplicity in presentation, in this work we solve (3.4) using CVX [142, 143]. In Fig. 3.3, we offer a CVX code snippet for Matlab for the solution to (3.4). The proposed algorithm for exact Lp-PCA via exhaustive search over $\{\pm 1\}^N$ is summarized in Algorithm 1.



(a) Partition of hypersphere.

(b) Partition of hyperball.

Figure 3.5. Partition of the hypersphere/hyperball to a set of halfspaces-intersection "cells" for an arbitrary $\mathbf{X} \in \mathbb{R}^{(D=3)\times(N=4)}$.

3.3.2 Exact Algorithm 2: Search Over Set With Size Polynomial in N

We steer our focus on deriving a more intelligent algorithm for computing the dominant Lp-PC of $\mathbf{X}$. Specifically, we design an algorithm with asymptotic cost polynomial in N for the special case of interest that $\mathbf{X}$ has rank-ρ and $\rho \ll N$. We commence our developments with the following Lemma 3.7.

Lemma 3.7. *Let $\mathbf{q}_{\text{opt}}$ denote the solution to* (3.1)*. It holds that $\mathbf{q}_{\text{opt}} \in \mathcal{C}(\mathbf{b}_{\text{opt}})$ where $\mathbf{b}_{\text{opt}} = \text{sgn}(\mathbf{X}^\top \mathbf{q}_{\text{opt}})$. Let $\mathbf{s}_{\text{opt}} = \mathbf{\Sigma}\mathbf{U}^\top \mathbf{q}_{\text{opt}}$. Then, $\mathbf{b}_{\text{opt}} = \text{sgn}(\mathbf{V}\mathbf{s}_{\text{opt}})$.*

By Lemma 3.7, the optimal binary vector $\mathbf{b}_{\text{opt}}$, and thus $\mathcal{C}(\mathbf{b}_{\text{opt}})$, can be found in

$$\mathcal{B} = \left\{\mathbf{b} \in \{\pm 1\}^N : \mathbf{b} = \text{sgn}(\mathbf{V}\mathbf{c}),\ \mathbf{c} \in \mathbb{R}^\rho\right\}. \tag{3.5}$$

Of course, $\mathcal{B}$ is a subset of $\{\pm 1\}^N$ and has finite size which is upper-bounded by $|\{\pm 1\}^N| = 2^N$ –i.e., $|\mathcal{B}| \leq 2^N$. We understand that there exist instances of $\mathbf{c} \in \mathbb{R}^\rho$ for which $\text{sgn}(\mathbf{V}\mathbf{c})$ remains invariant. In view of this observation, we work as follows to build a tight superset of $\mathcal{B}$ with size polynomial in N. We make the following mild assumption [144]:

Assumption 3.1. *Any collection of ρ rows of $\mathbf{V}$ are linearly independent. That is, for every index-set $\mathcal{I} \subseteq [N]$ with $|\mathcal{I}| = \rho$, it holds $\text{rank}(\mathbf{V}_{\mathcal{I},:}) = \rho$.*

Next, we define $\mathbf{v}_n = [\mathbf{V}]_{n,:}{}^\top$ for every $n \in [N]$ and denote by $\mathcal{N}(n)$ the nullspace of $\mathbf{v}_n$ –i.e.,

```
**Input:** $\mathbf{X} \in \mathbb{R}^{D \times N}$, $p \leq 1$
**Output:** $\mathbf{q}_{\text{opt}} \in \mathbb{R}^{D}$
$v \leftarrow 0$
$(\mathbf{U}, \mathbf{\Sigma}_{\rho \times \rho}, \mathbf{V}^{\top}) \leftarrow \text{svd}(\mathbf{X})$
For every $\mathcal{J} \subset [N],\ |\mathcal{J}| = \rho - 1$
    Build $\mathcal{B}_{\mathcal{J}}$ in (3.16)
    For every $\mathbf{b} \in \mathcal{B}_{\mathcal{J}}$
        $\mathbf{q} \leftarrow \arg\min_{\mathbf{q} \in \mathcal{C}(\mathbf{b})} - \sum_{n \in [N]} ([\mathbf{b}]_n \mathbf{x}_n^{\top} \mathbf{q})^p$
        If $\|\mathbf{X}^{\top}\mathbf{q}\|_p^p > v, v \leftarrow \|\mathbf{X}^{\top}\mathbf{q}\|_p^p, \mathbf{q}_{\text{opt}} \leftarrow \mathbf{q}$
```

Algorithm 3.2 Exact Lp-PCA via search over polynomial (in N) size set.

$\mathcal{N}(n) = \text{span}(\mathbf{I}_{\rho} - \mathbf{v}_n \mathbf{v}_n^{\top})$. We observe that, for every $\mathbf{c} \in \mathcal{N}(n)$, the non-negative angle between $\mathbf{c}$ and $\mathbf{v}_n$, $\phi(\mathbf{c}, \mathbf{v}_n) = \frac{\pi}{2}$ and, accordingly, $\mathbf{c}^{\top}\mathbf{v}_n = \|\mathbf{c}\|_2 \|\mathbf{v}_n\|_2 \cos(\phi(\mathbf{c}, \mathbf{v}_n)) = 0$. This implies that $\mathcal{N}(n)$ divides $\mathbb{R}^{\rho}$ into two non-overlapping hyperplanes (halfspaces) $\mathcal{H}(n)^{+}$ and $\mathcal{H}(n)^{-}$ such that [100]

$$\mathcal{H}(n)^{+} = \left\{ \mathbf{c} \in \mathbb{R}^{\rho} : \text{sgn}(\mathbf{c}^{\top}\mathbf{v}_n) = +1 \right\}, \tag{3.6}$$

$$\mathcal{H}(n)^{-} = \left\{ \mathbf{c} \in \mathbb{R}^{\rho} : \text{sgn}(\mathbf{c}^{\top}\mathbf{v}_n) = -1 \right\}. \tag{3.7}$$

By the definition of sgn $(\cdot)$, $\mathcal{H}(n)^{+}$ is an open set including the boundary $\mathcal{N}(n)$ while $\mathcal{H}(n)^{-}$ is an open set that does not include the boundary $\mathcal{N}(n)$. Accordingly, it holds that $\mathcal{H}(n)^{+} \cap \mathcal{H}(n)^{-} = \emptyset$. For example, consider some $\rho \geq 3$ and indices $m < s \in [N]$. $\mathcal{N}(m)$ and $\mathcal{N}(s)$ divide $\mathbb{R}^{\rho}$ in the halfspaces $\{\mathcal{H}(m)^{+}, \mathcal{H}(m)^{-}\}$ and $\{\mathcal{H}(s)^{+}, \mathcal{H}(s)^{-}\}$, respectively. By Assumption 3.1, each of the two halfspaces defined by $\mathcal{N}(i)$ for some $i \in \{m, s\}$ will intersect with each of two halfspaces defined by $\mathcal{N}(j)$, $j \in \{m, s\} \setminus i$. These intersections result in the four halfspaces-intersection "cells"

$$\mathcal{C}_1 = \mathcal{H}(m)^{-} \cap \mathcal{H}(s)^{-}, \tag{3.8}$$

$$\mathcal{C}_2 = \mathcal{H}(m)^{-} \cap \mathcal{H}(s)^{+}, \tag{3.9}$$

$$\mathcal{C}_3 = \mathcal{H}(m)^{+} \cap \mathcal{H}(s)^{-}, \tag{3.10}$$

$$\mathcal{C}_4 = \mathcal{H}(m)^{+} \cap \mathcal{H}(s)^{+}. \tag{3.11}$$

Importantly, for any $j \in [4]$ and $i \in \{m, s\}$, $[\text{sgn}(\mathbf{Vc})]_i$ remains the same across $\mathcal{C}_j$ –i.e., for any $\mathbf{c} \in \mathcal{C}_3$, $[\text{sgn}(\mathbf{Vc})]_m = +1$ and $[\text{sgn}(\mathbf{Vc})]_s = -1$. To better understand how the halfspaces-intersection "cells" are formed, in Fig. 3.5, we offer a visual illustration for arbitrary matrix $\mathbf{X} \in \mathbb{R}^{(D=3) \times (N=4)}$.

Next, we generalize the above discussion and consider the arrangement of all N hyperplanes $\mathcal{N}(n)\forall n \in [N]$. These hyperplanes partition $\mathbb{R}^\rho$ in L non-overlapping cells for some finite L which depends on ρ and N. Formally, for every $l \in [L]$, the l-th halfspaces-intersection cell is defined as

$$\mathcal{C}_l = \bigcap_{i\in\mathcal{I}^+} \mathcal{H}(i)^+ \bigcap_{j\in\mathcal{I}^-} \mathcal{H}(j)^-, \tag{3.12}$$

where $\mathcal{I}^+ \subset [N]$, $\mathcal{I}^- \subset [N]$, $\mathcal{I}^+ \cap \mathcal{I}^- = \emptyset$, and $\mathcal{I}^+ \cup \mathcal{I}^- = [N]$ [145, 146]. By the above discussions, every $\mathbf{c} \in \mathcal{C}_l$ lies in the same intersection of halfspaces and yields the same $\operatorname{sgn}(\mathbf{V}\mathbf{c})$. It holds

$$[\operatorname{sgn}(\mathbf{V}\mathbf{c})]_n = \operatorname{sgn}(\mathbf{v}_n^\top \mathbf{c}) = \begin{cases} +1, & n \in \mathcal{I}^+, \\ -1, & n \in \mathcal{I}^-. \end{cases} \tag{3.13}$$

Thereafter, we define a signature for each cell, $\mathbf{b}_l = \operatorname{sgn}(\mathbf{V}\mathbf{c})$ for some $\mathbf{c} \in \mathcal{C}_l$. We notice that, for every $(k,l) \in [N] \times [N]$ such that $k \neq l$, $\mathcal{C}_k \cap \mathcal{C}_l = \emptyset$ and that $\bigcup_{l\in[L]} \mathcal{C}_l = \mathbb{R}^\rho$. By the above analysis, the set $\mathcal{B}$ in (3.5) takes the equivalent form

$$\mathcal{B} = \bigcup_{l\in[L]} \{\operatorname{sgn}(\mathbf{V}\mathbf{c}) \ : \ \mathbf{c} \in \mathcal{C}_l\} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_L\}. \tag{3.14}$$

It has been shown [103, 145] that $L = 2\sum_{k=0}^{\rho-1}\binom{N-1}{k} \leq 2^N$ with equality attained if and only if $\rho = N$. It follows that the cardinality of the signature set $\mathcal{B}$ is also L. In the sequel, we rely on the above discussions and construct a tight superset of $\mathcal{B}$. By Assumption 3.1, for any index-set $\mathcal{J} \subseteq [N]$ with $|\mathcal{J}| = \rho - 1$ and complementary set $\mathcal{J}^c = [N] \setminus \mathcal{J}$, the halfspaces-intersection

$$\mathcal{S}_\mathcal{J} = \bigcap_{n\in[\mathcal{J}]} \mathcal{N}(n) \tag{3.15}$$

is a line (one-dimensional subspace) in $\mathbb{R}^\rho$. By definition, this line is the verge between all cells that are jointly bounded by the $\rho - 1$ hyperlanes. Then, we consider a vector $\mathbf{c} \in \mathbb{R}^\rho$ that crosses the verge $\mathcal{S}_\mathcal{J}$ (at any point other than $\mathbf{0}_\rho$). By this crossing, the value of $[\operatorname{sgn}(\mathbf{V}\mathbf{c})]_\mathcal{J}$ will change so that it adjusts to the cell it just entered while the value of $[\operatorname{sgn}(\mathbf{V}\mathbf{c})]_{\mathcal{J}^c}$ will remain invariant. This follows straightforwardly by the fact that a crossing over $\mathcal{S}_\mathcal{J}$ can not be simultaneous over a crossing of any of the hyperplanes corresponding to $\mathcal{J}^c$. Accordingly, for any $\mathbf{c} \in \mathcal{S}_\mathcal{J} \setminus \mathbf{0}_\rho$, the set

$$\mathcal{B}_\mathcal{J} = \left\{\mathbf{b} \in \{\pm 1\}^N \ : \ [\mathbf{b}]_{\mathcal{J}^c} = [\operatorname{sgn}(\mathbf{V}\mathbf{c})]_{\mathcal{J}^c}\right\} \tag{3.16}$$

contains all the signatures of all the cells are that are bounded by the verge $\mathcal{S}_\mathcal{J}$ –i.e., all non-zero points $\mathbf{c}$ asymptotically close to $\mathcal{S}_{\mathcal{J}}$ yield sgn($\mathbf{Vc}$) that lies in $\mathcal{B}_{\mathcal{J}}$. Moreover, there exists at least one such verge that bounds it. Therefore, the set

$$\mathcal{B}^{\star} = \bigcup_{\mathcal{J}\subseteq[N]\ :\ |\mathcal{J}|=\rho-1} \mathcal{B}_{\mathcal{J}} \tag{3.17}$$

includes all cell signatures and, thus, is a superset of $\mathcal{B}$.

Next, we study the size of $\mathcal{B}^{\star}$ and the computational effort required for forming it. We notice that for every instance of $\mathcal{J}$, it holds that $|\mathcal{B}_{\mathcal{J}}| = 2^{\rho-1}$, which derives by the sign ambiguity by the nullspace of each row of $\mathbf{V}$. Moreover, there exist $\binom{N}{\rho-1}$ distinct instances of $\mathcal{J}$ which implies that $|\mathcal{B}^{\star}|$ is upper bounded by $2^{\rho-1}\binom{N}{\rho-1}$. Both $\mathcal{B}^{\star}$ and $\mathcal{B}$ have polynomial size in N. In practice, for every instance of $\mathcal{J}$, $\mathbf{c}$ can be computed by Gram-Schmidt orthogonalization of $[\mathbf{V}]_{\mathcal{J},:}{}^{\top}$ with cost $\mathcal{O}(\rho^3)$. Retaining the dominant terms, $\mathcal{B}^{\star}$ can be computed with serial cost $\mathcal{O}(N^{\rho-1})$ or by $\binom{N}{\rho-1}$ parallel processes. The exact Lp-PC of $\mathbf{X}$ can be found by a search over $\mathcal{B}^{\star}$. The proposed algorithm for exact Lp-PCA via a search over $\mathcal{B}^{\star}$ is summarized in Algorithm 2.

In view of the above, the Lp-PC of a matrix can be computed exactly by a search over $\{\pm 1\}^N$ or (more intelligently) by a search over $\mathcal{B}^{\star}$. In both cases, multiple instances of (3.4) must be solved. Accordingly, a discussion on the computational effort required for solving (3.4) is in order. First, we note that numerous distinct solvers can be developed for solving (3.4) enabling different performance/complexity trade-offs. For simplicity in presentation and to remain within the scope of this chapter, we presented a solution to (3.4) via CVX's [142, 143] standardized algorithms. For a better understanding with respect to the computational effort required for solving (3.4), we developed a modified primal-dual Newton's algorithm the presentation of which is omitted as it extends well beyond the scope of this chapter: to study the underlying mechanics of Lp-PCA and develop new theory that enables us to compute the exact solution. The per-iteration cost of this algorithm is dominated by the inversion of a Jacobian matrix with cost $\mathcal{O}(N^3 + D^3)$ –i.e., cubic in both the number of features and number of points in $\mathbf{X}$. Letting T denote the number of iterations required for convergence, the overall complexity becomes $\mathcal{O}(T(N^3 + D^3))$. In practice, we have observed that $T \ll \min\{D, N\}$. Certainly, more sophisticated algorithms can be designed further reducing the computational effort required for solving Lp-PCA exactly.

3.4 Contribution 2: The Dominant Lp-PC of a Non-Negative Matrix

We consider the special case of interest that $\mathbf{X}$ has non-negative entries –i.e., for every $(i, j) \in [D] \times [N]$, $[\mathbf{X}]_{i.j} \geq 0$. The following Proposition 3.2 states that the Lp-PC of $\mathbf{X}$ can be found exactly without a search.

Proposition 3.2. *If $\mathbf{X}$ has non-negative entries, then*

$$\max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p = \max_{\mathbf{q}\in\mathcal{C}(\mathbf{1}_N)} \|\mathbf{X}^\top\mathbf{q}\|_p^p. \tag{3.18}$$

By Proposition 3.2, forming $\mathcal{B}^\star$ is not necessary when $\mathbf{X}$ has non-negative entries. Solving (3.4) for $\mathbf{b} = \mathbf{1}_N$ suffices for computing the Lp-PC of $\mathbf{X}$.

3.5 Contribution 3: Joint Extraction of Multiple Lp-quasinorm Principal-Components

We now consider the problem of jointly computing K Lp-PCs of $\mathbf{X}$, for general $K > 1$. For readability, we present again the formulation of Lp-PCA

$$\underset{\mathbf{Q}\in\mathbb{S}_{D\times K}}{\text{max.}} \left\|\mathbf{X}^\top\mathbf{Q}\right\|_p^p. \tag{3.19}$$

First, similar to the presented theory for $K = 1$, we notice that the Stiefel manifold can be partitioned into a finite number of non-overlapping sets. That is, for every $\mathbf{B} \in \{\pm 1\}^{N\times K}$, we define

$$\mathcal{C}(\mathbf{B}) = \{\mathbf{Q} \in \mathbb{S}_{D\times K} : \text{sgn}\left(\mathbf{X}^\top\mathbf{Q}\right) = \mathbf{B}\}. \tag{3.20}$$

Accordingly, w.l.o.g., Lp-PCA in (3.1) can equivalently be written as

$$\underset{\mathbf{Q}\in\bigcup_{\mathbf{B}\in\{\pm 1\}^{N\times K}}\mathcal{C}(\mathbf{B})}{\text{max.}} \left\|\mathbf{X}^\top\mathbf{Q}\right\|_p^p. \tag{3.21}$$

Input: $\mathbf{X} \in \mathbb{R}^{D \times N}$, $\mathbf{B} \in \{\pm 1\}^{N \times K}$, $p \leq 1$
Output: $\mathbf{Q} \in \mathbb{R}^{D \times K}$

```
cvx_begin
variable Q(D,K)
cost = 0;
for k = 1:K
        Yk = X * diag(B(:, k));
        cost = cost + sum(pow_p(Yk' * Q(:,k), p);
        Yk' * Q(:, k) >= 0
end
maximize(cost)
norm(Q, 2) <= 1
cvx_end
```

Figure 3.7. CVX code (Matlab) for the solution to (3.24).

In view of (3.21), we consider fixed $\mathbf{B} \in \{\pm 1\}^{N \times K}$ and steer our focus on solving

$$\max_{\mathbf{Q} \in \mathcal{C}(\mathbf{B})} \left\| \mathbf{X}^\top \mathbf{Q} \right\|_p^p . \tag{3.22}$$

By Lemma 3.6 and the fact that $\|\mathbf{X}^\top \mathbf{Q}\|_p^p = \sum_{k \in [K]} \|\mathbf{X}^\top \mathbf{q}_k\|_p^p$ for any $\mathbf{Q} \in \mathbb{R}^{D \times K}$, we present the following Lemma 3.8.

Lemma 3.8. *The cost function* $\|\mathbf{X}^\top \mathbf{Q}\|_p^p = \sum_{k \in [K]} \|\mathbf{X}^\top \mathbf{q}_k\|_p^p = \sum_{n \in [N]} \sum_{k \in [K]} (\mathbf{x}_n^\top \mathbf{q}_k [\mathbf{B}]_{n,k})$ *is concave with respect to* $\mathbf{Q} \in \mathcal{C}(\mathbf{B})$.

Lemma 3.8 derives straightforwardly by applying Lemma 3.6 K times. Even though the cost function of (3.22) is concave with respect to the maximizing argument, the optimization problem itself is not convex nor concave because of the Stiefel manifold constraint (orthogonality constraints) and, accordingly, solving (3.22) is not a trivial task. Similar to standard practice in the literature [147, 148], we consider a convex relaxation formulation of (3.22) by substituting the Stiefel manifold constraint –i.e., $\mathbf{Q} \in \mathbb{S}_{D \times K}$– by $\|\mathbf{Q}\|_2 = \sigma_{\max}(\mathbf{Q}) \leq 1$.[1] The set $\{\mathbf{Q} \in \mathbb{R}^{D \times K} : \|\mathbf{Q}\|_2 \leq 1\}$ is the convex hull of $\mathbb{S}_{D \times K}$ –i.e., the smallest convex set which includes it [149].

Formally, we define the set

$$\bar{\mathcal{C}}(\mathbf{B}) = \{\mathbf{Q} \in \mathbb{R}^{D \times K} : \|\mathbf{Q}\|_2 \leq 1, \operatorname{sgn}\left(\mathbf{X}^\top \mathbf{Q}\right) = \mathbf{B}\} \tag{3.23}$$

[1] For any matrix $\mathbf{A}$, $\sigma_{\max}(\mathbf{A})$ denotes the maximum singular value of $\mathbf{A}$.

and consider a convex relaxation formulation of (3.22) of the form

$$\max_{\mathbf{Q}\in\bar{\mathcal{C}}(\mathbf{B})} \left\|\mathbf{X}^\top\mathbf{Q}\right\|_p^p. \tag{3.24}$$

We have already established that the cost function of (3.22) (and, thus, (3.24)) is concave with respect to the maximizing argument. The following Lemma 3.9 states that the set $\bar{\mathcal{C}}(\mathbf{B})$ is a convex set.

Lemma 3.9. *For any $\mathbf{B}\in\{\pm1\}^{N\times K}$, $\bar{\mathcal{C}}(\mathbf{B})$ is a convex set.*

By Lemma 3.8 and Lemma 3.9, the following Proposition 3.3 naturally follows.

Proposition 3.3. *The optimization problem in (3.24) is a convex problem and can be equivalently rewritten as $\min_{\mathbf{Q}\in\bar{\mathcal{C}}(\mathbf{B})} -\sum_{n\in[N]}\sum_{k\in[K]}([\mathbf{B}]_{n,k}\mathbf{x}_n^\top\mathbf{q}_k)^p$.*

For simplicity, in this work we solve (3.24) with CVX [142, 143]. In Fig. 3.7, we offer a CVX code snippet for Matlab for the solution to (3.24). In the sequel, we discuss how solving (3.24), for a fixed $\mathbf{B}$, can be leveraged for deriving a solution to the original problem of interest (3.1) (or, (3.21)).

A straightforward yet naive approach would be to solve (3.24) for every $\mathbf{B}\in\{\pm1\}^{N\times K}$ (exhaustive search) and retain the solution that attains the highest metric. While this approach computes the exact solution to

$$\max_{\mathbf{Q}\in\mathbb{R}^{D\times K}:\|\mathbf{Q}\|_2\leq1} \left\|\mathbf{X}^\top\mathbf{Q}\right\|_p^p, \tag{3.25}$$

an exhaustive search is not necessary in accordance with the following Lemma 3.10

Lemma 3.10. *Let $\bar{\mathbf{Q}}_{\text{opt}}$ be the exact solution to $\max_{\mathbf{Q}\in\bar{\mathcal{C}}(\mathbf{B})}\max_{\mathbf{B}\in\{\pm1\}^{N\times K}}\|\mathbf{X}^\top\mathbf{Q}\|_p^p$ (equivalently (3.25)). $\bar{\mathbf{B}}_{\text{opt}}=\text{sgn}(\mathbf{X}^\top\bar{\mathbf{Q}}_{\text{opt}})$ is such that, for every $k\in[K]$, $[\bar{\mathbf{B}}_{\text{opt}}]_{:,k}\in\mathbf{B}^\star$.*

By Lemma 3.10, the exact solution to (3.25) can be computed by a search over the set

$$\mathcal{B}^{\star,K}=\{\mathbf{B}\in\{\pm1\}^{N\times K}:[\mathbf{B}]_{:,k}\in\mathbf{B}^\star\forall k\in[K]\}. \tag{3.26}$$

In practice, $\mathcal{B}^{\star,K}$ is the K-fold Cartesian product of $\mathcal{B}^\star$. Moreover, it holds that $\mathcal{B}^{\star,K}\subseteq\{\pm1\}^{N\times K}$ and that $|\mathcal{B}^{\star,K}|=|\mathcal{B}^\star|^K\leq|\{\pm1\}^{N\times K}|$. In view of this discussion, we propose to solve (3.24) for every $\mathbf{B}\in\mathcal{B}^{\star,K}$. The proposed algorithm the solving (3.25) is summarized in Algorithm 3.

```
**Input:** $\mathbf{X} \in \mathbb{R}^{D \times N}$, $K$, $p \leq 1$
**Output:** $\bar{\mathbf{Q}}_{\text{opt}} \in \mathbb{R}^{D}$
$v \leftarrow 0$
For every $\mathbf{B} \in \mathcal{B}^{\star,K}$
    $\mathbf{Q} \leftarrow \arg\min_{\mathbf{Q}\in\bar{\mathcal{C}}(\mathbf{B})} -\sum_{n\in[N]}\sum_{k\in[K]}([\mathbf{B}]_{n,k}\mathbf{x}_n^{\top}\mathbf{q}_k)^p$
    If $\|\mathbf{X}^{\top}\mathbf{Q}\|_p^p > v, v \leftarrow \|\mathbf{X}^{\top}\mathbf{Q}\|_p^p, \bar{\mathbf{Q}}_{\text{opt}} \leftarrow \mathbf{Q}$
```

Algorithm 3.3 Multiple Lp-PCs (convex relaxation) via search over polynomial (in N) size set.

Formally, by Algorithm 3 we obtain the exact solution to (3.25) by solving

$$\max_{\mathbf{Q}\in\bar{\mathcal{C}}(\mathbf{B})} \max_{\mathbf{B}\in\mathcal{B}^{\star,K}} \|\mathbf{X}^{\top}\mathbf{Q}\|_p^p. \tag{3.27}$$

Let $\bar{\mathbf{Q}}_{\text{opt}}$ denote the exact solution to (3.27). We distinguish two cases for $\bar{\mathbf{Q}}_{\text{opt}}$: (i) $\bar{\mathbf{Q}}_{\text{opt}} \in \mathbb{S}_{D\times K}$ and (ii) $\bar{\mathbf{Q}}_{\text{opt}} \notin \mathbb{S}_{D\times K}$. Then, we present the following Proposition 3.4

Proposition 3.4. *If $\bar{\mathbf{Q}}_{opt} \in \mathbb{S}_{D\times K}$, then it solves Lp-PCA in* (3.1) *exactly.*

Proposition 3.4 states that when the solution of the convex relaxation formulation is in the Stiefel manifold, then it is also a solution to the original Lp-PCA formulation. Our numerical studies have shown that this is often the case. In the sequel, we propose an algorithm for computing a solution in the Stiefel manifold when $\bar{\mathbf{Q}}_{\text{opt}} \notin \mathbb{S}_{D\times K}$.

3.6 Contribution 4: Stiefel Manifold Solution Refinement

In this Section, we propose a method for computing a solution to the original problem in (3.1) for the special case that $\bar{\mathbf{Q}}_{\text{opt}} \notin \mathbb{S}_{D\times K}$.

In this case, the most straightforward approach would be to refine $\bar{\mathbf{Q}}_{\text{opt}}$ by solving

$$\underset{\mathbf{Q}\in\mathbb{S}_{D\times K}}{\arg\min} \|\mathbf{Q} - \bar{\mathbf{Q}}_{\text{opt}}\|_F^2. \tag{3.28}$$

This problem admits an exact solution by means of the Procrustes Theorem. Our numerical studies have shown that, although simple, this approach is inferior with respect to the Lp-PCA metric, when compared with the more sophisticated approach that we propose next as follows.

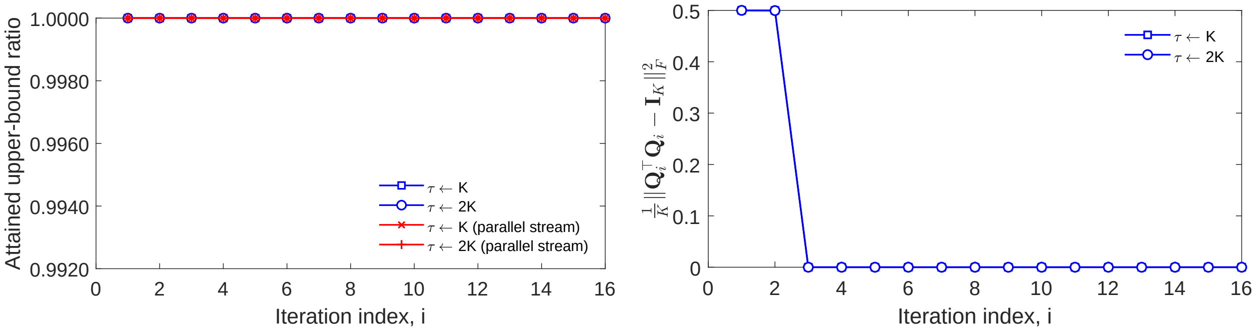
Even if $\bar{\mathbf{Q}}_{\text{opt}} \notin \mathbb{S}_{D\times K}$, it provides valuable information about the solution to (3.1). First, we understand that $\bar{v} = \|\mathbf{X}^\top \bar{\mathbf{Q}}_{\text{opt}}\|_p^p$ serves as an upper-bound of the maximum attainable metric of the cost function of (3.1). That is, if we denote by $\mathbf{Q}_{\text{opt}}$ the exact solution to (3.1), then it always holds that $\|\mathbf{X}^\top \mathbf{Q}_{\text{opt}}\|_p^p \leq \|\mathbf{X}^\top \bar{\mathbf{Q}}_{\text{opt}}\|_p^p$. Moreover, it is reasonable to expect that $\mathbf{Q}_{\text{opt}}$ can exhibit low Euclidean distance from $\bar{\mathbf{Q}}_{\text{opt}}$ when we consider that the metric surface of the cost function is concave over $\bar{\mathcal{C}}(\mathbf{B}_{\text{opt}})$, where $\mathbf{B}_{\text{opt}} = \text{sgn}(\mathbf{X}^\top \bar{\mathbf{Q}}_{\text{opt}})$. Accordingly, we propose to use $\bar{\mathbf{Q}}_{\text{opt}}$ as an "anchor" and search over an area near it while at the same time trying to maximize the metric of interest. To this end, we propose an iterative algorithm for computing a solution in the Stiefel manifold by leveraging $\bar{\mathbf{Q}}_{\text{opt}}$ as follows. First, we initialize $\mathbf{Q}_1 = \bar{\mathbf{Q}}_{\text{opt}}$. Then, at every iteration $i \geq 1$, we update

$$\mathbf{Q}_{i+1} = \underset{\mathbf{Q} \in \bar{\mathcal{C}}(\mathbf{B}_{\text{opt}})}{\arg\max} \frac{1}{\bar{v}} \sum_{n \in [N]} \sum_{k \in [K]} ([\mathbf{B}]_{n,k} \mathbf{x}_n^\top \mathbf{q}_k)^p + \tau \text{Tr}(\mathbf{Q}^\top \mathbf{Q}_i), \tag{3.29}$$

where $\tau > 0$ is a regularization coefficient that promotes orthogonality among the columns of $\mathbf{Q}$. For any fixed iteration index i, the above problem is, of course, a convex problem. Moreover, at every iteration the compound metric in (3.29) is increasing. At the same time, the compound metric is upper bounded. Accordingly, convergence is guaranteed. Importantly, for an appropriate choice of τ, the term $\tau \text{Tr}(\mathbf{Q}^\top \mathbf{Q}_i)$ will enforce the Stiefel manifold constraint. Interestingly, the proposed iteration in (3.29) is similar, in formulation, with the general form of proximal algorithms [150], a special class of optimization algorithms. Furthermore, in view of (3.29), we consider a parallel stream of matrices each of which are guaranteed to be in the Stiefel manifold, independently of the choice of τ. That is, at every iteration index $i \geq 1$, we compute $\mathbf{W}_i = \arg\min_{\mathbf{W} \in \mathbb{S}_{D \times K}} \|\mathbf{W} - \mathbf{Q}_i\|_F^2$.

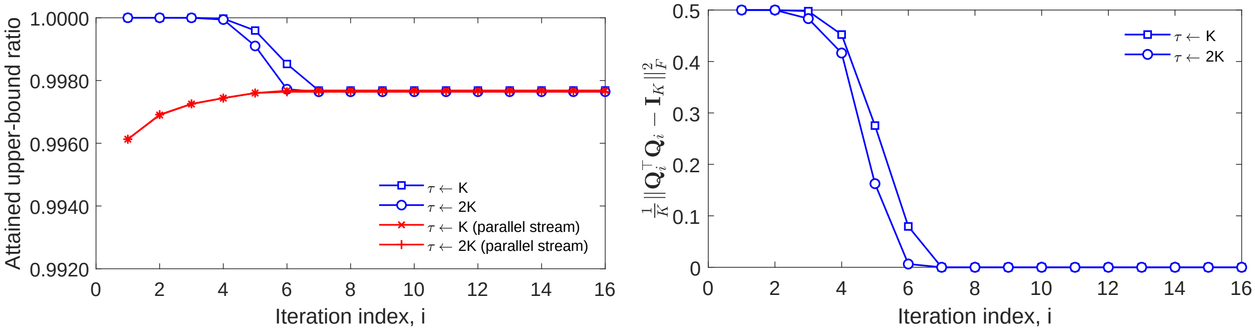
Next, we evaluate the performance of the proposed iteration in(3.29) by evaluating the Lp-PCA metric attained by both $\mathbf{Q}_i$ and $\mathbf{W}_i$, for every $i \geq 1$. As measures of success, we consider two metrics. First, at any fixed iteration index i, we measure the ratio of the Lp-PCA metric attained by $\mathbf{Q}_i$ and $\mathbf{W}_i$ with respect to the known upper-bound $\bar{v}$ by computing $v_i = \frac{1}{\bar{v}} \|\mathbf{X}^\top \mathbf{Q}_i\|_p^p$ and $\tilde{v}_i = \frac{1}{\bar{v}} \|\mathbf{X}^\top \mathbf{W}_i\|_p^p$, respectively. Certainly, it holds that $v_i \leq 1$ and that $\tilde{v}_i \leq 1$. Second, we measure the proximity of $\mathbf{Q}_i$ to the Stiefel manifold by measuring $e(i) = \frac{1}{K} \|\mathbf{Q}_i^\top \mathbf{Q}_i - \mathbf{I}_K\|_F^2$. If $e(i) = 0$ then $\mathbf{Q}_i$ is in the Stiefel manifold. Let I denote the converging iteration of (3.29). If $v_I = 1$ and $e(I) = 0$ at the same time, then the proposed iteration in (3.29) has returned the exact solution to (3.1).

We consider data matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ drawing entries from $\mathcal{N}(0, 1)$. We run Algorithm 3 on $\mathbf{X}$ and compute $\bar{\mathbf{Q}}_{\text{opt}}$. If $\bar{\mathbf{Q}}_{\text{opt}} \in \mathbb{S}_{D \times K}$, then we discard it. Otherwise, we give it as input to the iteration in (3.29) for refining its solution to a solution in the Stiefel manifold. We repeat this process until we collect 30 realizations of $\mathbf{X}$ such that $\bar{\mathbf{Q}}_{\text{opt}} \notin \mathbb{S}_{D \times K}$.



(a) Attained upper-bound ratio vs iteration index. (b) Proximity to Stiefel manifold vs iteration index.

Figure 3.9. Evaluation of the proposed iteration in (3.29) for $p = 1$, $D = 3$, $N = 5$, and $K = 2$. Reported curve-values are averages over 30 realizations of data.

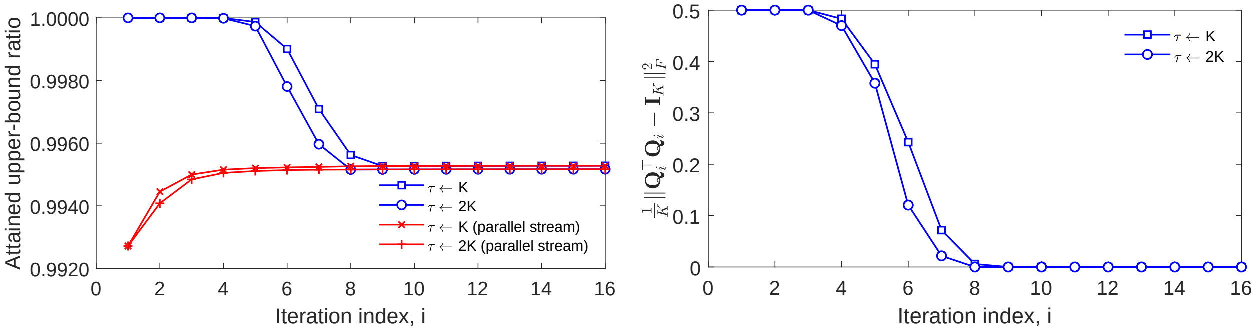


(a) Attained upper-bound ratio vs iteration index. (b) Proximity to Stiefel manifold vs iteration index.

Figure 3.10. Evaluation of the proposed iteration in (3.29) for $p = 0.90$, $D = 3$, $N = 5$, and $K = 2$. Reported curve-values are averages over 30 realizations of data.

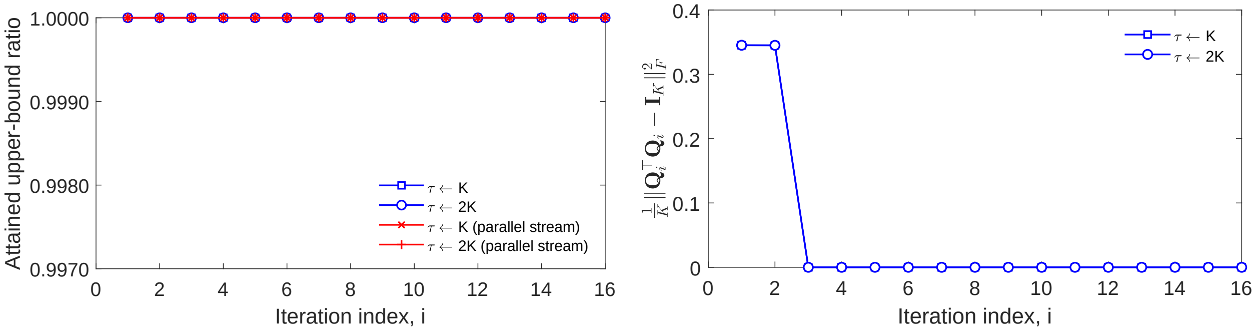
First, we set $D = 3$, $N = 5$, $K = 2$, and $p = 1$. In Fig. 3.9a and Fig. 3.9b, we plot $(v_i, \tilde{v}_i)$ and $e(i)$, respectively, for $\tau = K$ and $\tau = 2K$. The reported performances are averages over 30 realizations. We observe that both streams of matrices attain the maximum attainable metric across the board and, at the converging iteration I, both $\mathbf{Q}_I$ and $\mathbf{W}_I$ are in the Stiefel manifold –i.e., the solution is exact. Next, we repeat the above study, this time for $p = 0.9$. We present the performance curves in Fig. 3.10a and Fig. 3.10b, computed again over 30 realizations. We notice that $v_i \geq \tilde{v}_i$ for every iteration index. Specifically, in early iterations the matrices of the original stream offer slightly higher metric compared to the matrices of the parallel stream. However, upon convergence, the matrices of the two streams coincide and converge to the same metric. We repeat the above experiment, this time for $p = 0.75$. We illustrate the average performances in Fig. 3.11a and Fig. 3.11b. The observations we make are the same as above.

Thereafter, we adjust the parameter configuration as follows. We set $D = 4$, $N = 5$, $K = 3$, and



(a) Attained upper-bound ratio vs iteration index. (b) Proximity to Stiefel manifold vs iteration index.

Figure 3.11. Evaluation of the proposed iteration in (3.29) for $p = 0.75$, $D = 3$, $N = 5$, and $K = 2$. Reported curve-values are averages over 30 realizations of data.

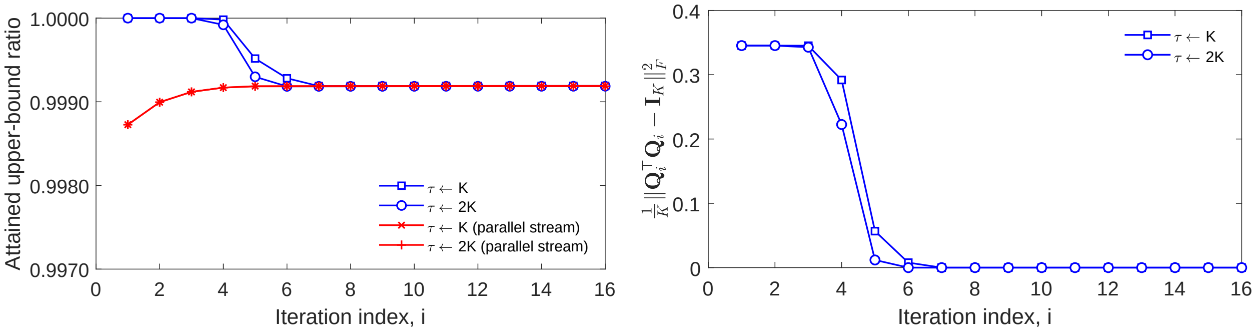


(a) Attained upper-bound ratio vs iteration index. (b) Proximity to Stiefel manifold vs iteration index.

Figure 3.12. Evaluation of the proposed iteration in (3.29) for $p = 1$, $D = 4$, $N = 5$, and $K = 3$. Reported curve-values are averages over 30 realizations of data.

repeat the above experiments for $p = 1$, $p = 0.9$, and $p = 0.75$. The average performances are reported in Fig. 3.12, Fig. 3.13, and Fig. 3.14, respectively. The observations we make are inline with the observation made the smaller matrix size previously studied.

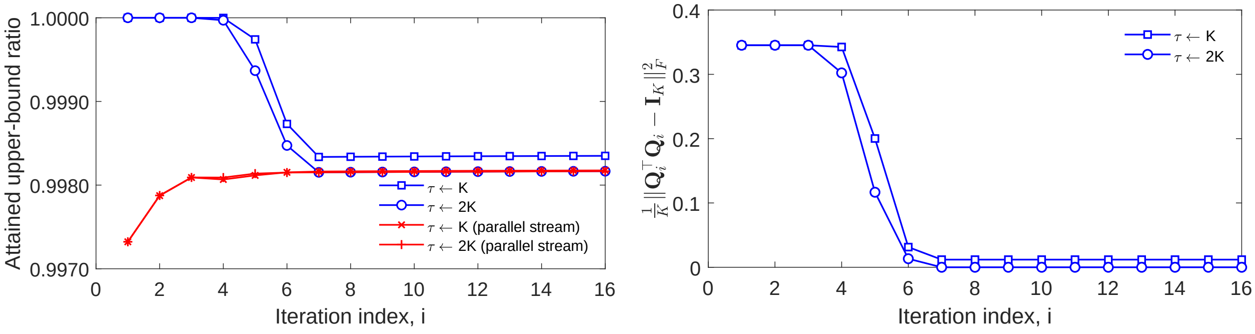
In view of the above, we infer that, for any matrix $\mathbf{X}$, if we left $\mathbf{Q}_{\text{opt}}$ denote the exact solution to Lp-PCA, then the following holds: $\|\mathbf{Q}_{\text{opt}}^\top\mathbf{X}\|_p^p - \|\mathbf{W}_I^\top\mathbf{X}\|_p^p \leq \alpha = \|\bar{\mathbf{Q}}_{\text{opt}}^\top\mathbf{X}\|_p^p - \|\mathbf{W}_I^\top\mathbf{X}\|_p^p$. That is, the difference of the Lp-PCA metric attained by the solution over the spectral ball and the converging argument of the parallel stream, serves as a lower bound α of the performance attained by $\mathbf{W}_I$, over the Stiefel manifold, with respect to the Lp-PCA metric. Our numerical experiments have shown that that the lower-bound α is, in general, very small.



(a) Attained upper-bound ratio vs iteration index.

(b) Proximity to Stiefel manifold vs iteration index.

Figure 3.13. Evaluation of the proposed iteration in (3.29) for $p = 0.90$, $D = 4$, $N = 5$, and $K = 3$. Reported curve-values are averages over 30 realizations of data.



(a) Attained upper-bound ratio vs iteration index.

(b) Proximity to Stiefel manifold vs iteration index.

Figure 3.14. Evaluation of the proposed iteration in (3.29) for $p = 0.75$, $D = 4$, $N = 5$, and $K = 3$. Reported curve-values are averages over 30 realizations of data.

3.7 Numerical Experiments

In this chapter, we set the theoretical foundations for Lp-quasinorm Principal-Component analysis and proposed algorithms for its solution. The proposed algorithms are of theoretical value –i.e., they reveal the underlying properties and mechanics of the problem which, in turn, set the foundations for developing efficient and scalable algorithms. Nonetheless, the development of such algorithms is beyond the scope of this Chapter. In the sequel, we conduct numerical experiments on rather small-sized matrices. The purpose of these studies is to demonstrate the merits of Lp-PCA processing.

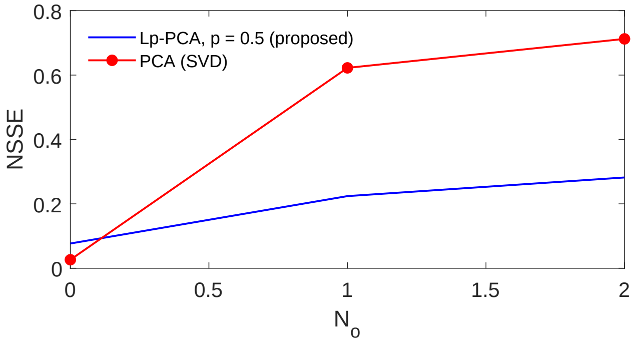


Figure 3.15. Subspace recovery: NSSE vs number of outliers N_o. $D = 7$, $N = 7$, $K = 1$, $p = 0.5$, SNR = 8dB, ONR = 18dB.

3.7.1 Subspace Recovery

We consider availability of matrix

$$\mathbf{X} = \mathbf{X}_s + \mathbf{N} + \mathbf{O} \in \mathbb{R}^{D \times N}. \tag{3.30}$$

Matrix $\mathbf{X}_s = \mathbf{Q}_{\text{nom}}\mathbf{U}^\top$ is the signal-of-interest carrying matrix such that $\mathbf{Q}_{\text{nom}} \in \mathbb{S}_{D\times K}$ is the signal-of-interest subspace basis and $\mathbf{U} \in \mathbb{R}^{N\times K}$ draws entries from $\mathcal{N}(0,1)$ –i.e., $\mathbb{E}\{\|\mathbf{U}\|_F^2\} = KN$. Matrix $\mathbf{N}$ models Additive White Gaussian Noise (AWGN) and draws entries from $\mathcal{N}(0,\sigma_n^2)$. Matrix $\mathbf{O}$ models sparse outliers –i.e., it has N_o non-zero entries drawing values from $\mathcal{N}(0,\sigma_o^2)$. The objective is to recover $\mathbf{Q}_{\text{nom}}$ from the available for processing matrix $\mathbf{X}$. To measure data quality, we define the Signal-to-Noise-Ratio (SNR)

$$\text{SNR} = \mathbb{E}\left\{\frac{\|\mathbf{X}_s\|_F^2}{\|\mathbf{N}\|_F^2}\right\} \tag{3.31}$$

and the Outlier-to-Noise-Ratio (ONR)

$$\text{ONR} = \mathbb{E}\left\{\frac{\|\mathbf{O}\|_F^2}{\|\mathbf{N}\|_F^2}\right\}. \tag{3.32}$$

As a performance evaluation metric, for any estimate of $\mathbf{Q}_{\text{nom}}$, $\widehat{\mathbf{Q}} \in \mathbb{S}_{D\times K}$, we define the Normalized Subspace Squared Error (NSSE)

$$\text{NSSE} = \frac{1}{2K}\left\|\mathbf{Q}_{\text{nom}}\mathbf{Q}_{\text{nom}}^\top - \widehat{\mathbf{Q}}\widehat{\mathbf{Q}}^\top\right\|_F^2 \in [0,1]. \tag{3.33}$$

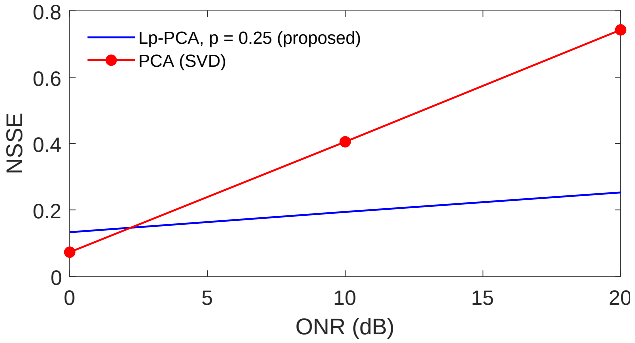


Figure 3.16. Subspace recovery: NSSE vs ONR (dB). $D = 7$, $N = 7$, $K = 1$, $p = 0.25$, SNR = 8dB, $N_o = 2$.

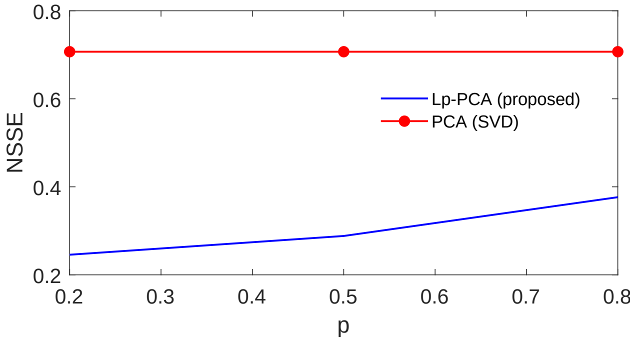


Figure 3.17. Subspace recovery: NSSE vs p. $D = 7$, $N = 7$, $K = 1$, ONR = 18dB , SNR = 8dB, $N_o = 2$.

In Fig. 3.15, we set $D = 7$, $N = 7$, $K = 1$, SNR = 8dB and ONR = 18dB. We let N_o vary in $\{0, 1, 2\}$ and measure the average NSSE (computed over 1000 statistically independent realizations of noise and corruption) attained by the proposed Lp-PCA algorithm ($p = 0.5$) and compare against standard PCA implemented by means of SVD. We observe that, under nominal operation ($N_o = 0$), both methods attain high basis estimation performance. However, for $N_o > 0$, Lp-PCA ($p = 0.5$) clearly outperforms standard PCA in subspace basis estimation performance.

In Fig. 3.16, we consider the same parameter configuration as above with the exception that, this time, we fix $N_o = 2$ and let ONR vary in $\{0, 10, 20\}$dB. We measure the average NSSE (computed over 1000 statistically independent realizations of noise and corruption) attained by the proposed Lp-PCA algorithm ($p = 0.25$) and compare against standard PCA. For benign corruption (ONR = 0dB), both approaches exhibit high estimation performance. In contrast, the robustness of Lp-PCA ($p = 0.25$) and its superiority against standard PCA are clearly documented when ONR is greater or equal than 10dB.

Finally, in Fig. 3.17, we consider again the same parameter configuration. This time, we fix ONR = 18dB and let p vary in $\{0.2, 0.5, 0.8\}$. We illustrate the average NSSE performance vs p (computed over 1000 statistically independent realizations of noise and corruption). As a benchmark, we include the performance of standard PCA (horizontal line). Once more, the merit of Lp-PCA processing and its superiority over standard PCA are clearly documented.

3.7.2 Classification of Biomedical Data

We consider length-($D = 30$) vector samples computed from a digitized image of a fine needle aspirate (FNA) of a breast mass which are available in the Breast Cancer Wisconsin (Diagnostic) dataset [151, 152]. There are 569 data samples in total, each of which is labeled as malignant or benign tissue. There are 212 malignant and 357 benign tissue-samples available. We arrange all the samples in data matrix $\mathbf{X} \in \mathbb{R}^{(D=30)\times(N=569)}$ and define label-vector $\mathbf{z} \in \{0, 1\}^{569}$, where $[\mathbf{z}]_i = 1$ if $[\mathbf{X}]_{:,i}$ corresponds to a malignant tissue and $[\mathbf{z}]_i = 0$ if $[\mathbf{X}]_{:,i}$ corresponds to a benign tissue. We consider availability of N_{train} points from each class for training and N_{test} for testing. We let $\mathbf{X}_b = [\mathbf{X}]_{:,\mathcal{I}_b} \in \mathbb{R}^{D\times N_{\text{train}}}$ and $\mathbf{X}_m = [\mathbf{X}]_{:,\mathcal{I}_m} \in \mathbb{R}^{D\times N_{\text{train}}}$ denote the benign and malignant training data samples, respectively, where $\mathcal{I}_b \subset [569]$, $|\mathcal{I}_b| = N_{\text{train}}$, $\mathcal{I}_m \subset [569]$, and $|\mathcal{I}_m| = N_{\text{train}}$. Similarly, $\mathbf{Y}_b = [\mathbf{X}]_{:,\mathcal{J}_b} \in \mathbb{R}^{D\times N_{\text{test}}}$ and $\mathbf{Y}_m = [\mathbf{X}]_{:,\mathcal{J}_m} \in \mathbb{R}^{D\times N_{\text{test}}}$ denote the benign and malignant data samples, respectively, for testing, where $\mathcal{J}_b \subset [569]$, $|\mathcal{J}_b| = N_{\text{test}}$, $\mathcal{J}_m \subset [569]$, and $|\mathcal{J}_m| = N_{\text{test}}$. There is no overlap between the training and testing data of each class –i.e., $\mathcal{I}_m \cap \mathcal{J}_m = \emptyset$ and $\mathcal{I}_b \cap \mathcal{J}_b = \emptyset$. During training, we compute $\mathbf{q}_b = \arg\max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}_b^\top \mathbf{q}\|_p^p$ and $\mathbf{q}_m = \arg\max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}_m^\top \mathbf{q}\|_p^p$ by means of the proposed algorithm. Given a testing sample $\mathbf{y}$ from $\mathbf{Y}_m$ or $\mathbf{Y}_b$, we classify it according to

$$(\mathbf{q}_m^\top \mathbf{y})^2 \|\mathbf{y}\|^{-2} \underset{\text{malignant}}{\overset{\text{benign}}{\lessgtr}} (\mathbf{q}_b^\top \mathbf{y})^2 \|\mathbf{y}\|^{-2}. \tag{3.34}$$

In Fig. 3.18, we set $N_{\text{train}} = 16$ and $N_{\text{test}} = 100$. We plot the average classification accuracy (computed over 1000 distinct realizations of $\mathcal{I}_m, \mathcal{I}_b, \mathcal{J}_m, \mathcal{J}_b$) when p varies in

$$\{0.01, 0.05, 0.2, 0.35, 0.50, 0.65, 0.80, 0.95, 1\}. \tag{3.35}$$

Moreover, we include as benchmarks the classification accuracies of the k-nearest neighbor classifier for $k = 1$ (NN) and the stanard Principal-Component Analysis implemented by means of SVD. We observe that all methods exhibit high performance. NN exhibits the best performance, slightly higher than that of Lp-PCA for $p = 0.01$. Standard PCA, implemented by means of SVD, exhibits

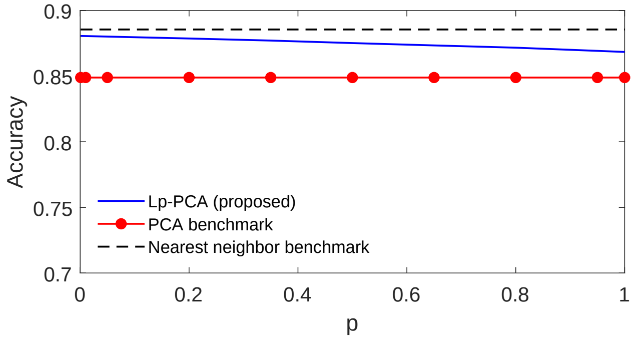


Figure 3.18. Breast Cancer Wisconsin (Diagnostic) Dataset: Classification accuracy vs p. $N_{\text{train}} = 20$, $N_{\text{test}} = 100$, $m = 0$.

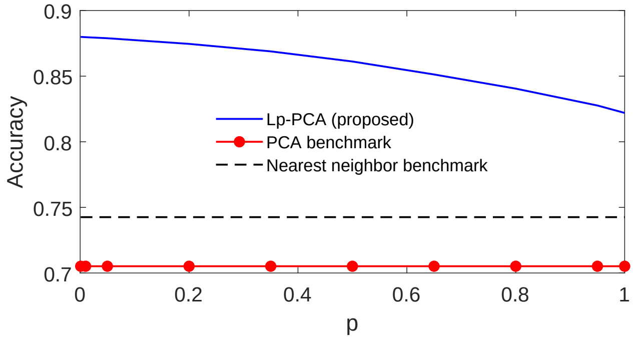


Figure 3.19. Breast Cancer Wisconsin (Diagnostic) Dataset: Classification accuracy vs p. $N_{\text{train}} = 20$, $N_{\text{test}} = 100$, $m = 3$.

the lowest performance.

In Fig. 3.19, we repeat the above experiment but this time we consider that $m = 3$ malignant samples have mistakenly been labeled as benign samples and $m = 3$ benign samples have been mislabeled as malignant. We notice that the performances of NN and SVD are significantly compromised by the mislabelings –i.e., the performance of NN dropped to 0.74 from 0.89 and the performance of PCA dropped to 0.71 from 0.85. On the other hand, Lp-PCA attained classification performance as high as 0.88 ($p = 0.1$) and no lower than 0.825 ($p = 1$).

Finally, in Fig. 3.20, we set $N_{\text{train}} = 20$, $N_{\text{test}} = 100$, and $p = 0.1$. We let the number of mislabelings, m, vary in $\{0, 1, 2, 3, 4\}$.[2] Then, we plot the classification accuracy attained by each method (computed over 1000 distinct realizations of data and corruptions). Expectedly, when $m = 0$ the performances of all methods are similar to their performances in Fig. 3.18. However, as

[2] That is, m benign samples have wrongly been labeled as malignant and m malignant samples have been labeled as benign samples.

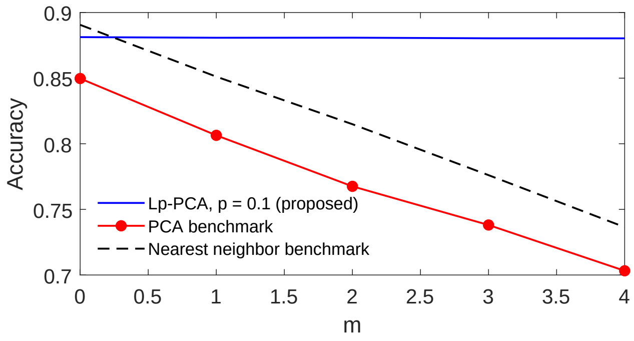


Figure 3.20. Breast Cancer Wisconsin (Diagnostic) Dataset: Classification accuracy vs number of mislabelings. $N_{\text{train}} = 20$, $N_{\text{test}} = 100$, $p = 0.1$.

m increases, the performances of NN and PCA are compromised. Interestingly, the performance of Lp-PCA ($p = 0.1$) appears to be affected very little by the mislabelings for all values of m.

3.8 Conclusions

In this Chapter, we proposed new theory and algorithms for Lp-quasinorm Principal-Component Analysis. Specifically: (i) We showed, for the first time, that rank-1 Lp-PCA admits an exact solution by means of combinatorial optimization and presented the first algorithms for its solution. (ii) We proved that, for non-negative data, the dominant Lp-PC can be computed by solving a convex problem. (iii) We proposed a novel near-exact algorithm for jointly extracting multiple components. (iv) We proposed a solution-refinement strategy which, in some cases, further enhances a near-exact solution to Lp-PCA.

Chapter 4

Coprime Array Processing

4.1 Introduction

Coprime Arrays (CAs) are non-uniform linear arrays the element locations of which are derived by coprime number theory [153]. CAs are a special class of sparse arrays [153, 154] which are often preferred due to their desirable bearing properties –i.e., enhanced Degrees-of-Freedom (DoF) and closed-form expressions for element-locations. CAs have attracted significant research interest over the past years and have been successfully employed in applications such as Direction-of-Arrival (DoA) estimation [155–166], beamforming [167–169], interference localization and mitigation in satellite systems [170], and space-time adaptive processing [171, 172], to name a few.

Interpolation methods for CAs which further enhance DoF [173–180] have also been studied. Furthermore, scholars have considered CAs for underwater localization [181, 182], channel estimation in MIMO communications via tensor decomposition [183], receivers capable of two-dimensional DoA estimation [184], and receivers on moving platforms which also promote increased number of DoF [185–187]. Other non-uniform arrays with increased DoF and closed-form expressions are the nested and MISC arrays [154, 188].

In standard DoA estimation with CAs [153], the receiver conducts a series of intelligent processing steps and assembles an autocorrelation matrix which corresponds to a larger virtual Uniform Linear Array (ULA) which is commonly referred to as *coarray*. Accordingly, CAs enable the identification of more sources than sensors compared to equal-length ULAs [189–195]. Processing at a coprime array receiver commences with the estimation of the nominal (true) physical-array autocorrelations. The receiver processes the estimated autocorrelations so that each coarray element is represented by one autocorrelation estimate. Next, the processed autocorrelations undergo spatial smoothing, forming an autocorrelation matrix estimate which corresponds to the coarray. By the received-signal model, the nominal autocorrelation matrix of the coarray has a specific structure: it is (i) Positive Definite (PD), (ii) Hermitian, (iii) Toeplitz, and (iv) its noise-subspace eigenvalues are equal. Finally, a DoA estimation approach such as the MUltiple SIgnal Classification (MUSIC) algorithm, for instance, can be applied to the resulting autocorrelation matrix estimate for identifying the source directions.

In practice, the autocorrelations of the physical-array elements are estimated by processing a collection of received-signal snapshots and diverge from the nominal ones. Accordingly, existing approaches offer autocorrelation-matrix estimates which diverge from the nominal one, while at the same time, violate at least one of the above structure-properties. At the autocorrelation processing step, the estimated autocorrelations are commonly processed by *selection combining* [153], retaining only one autocorrelation sample for each coarray element. Alternatively, an autocorrelation estimate for each coarray element is obtained by *averaging combining* [162] all available sample-estimates corresponding to a particular coarray element. The two methods coincide in Mean-Squared-Error (MSE) estimation performance when applied to the nominal physical-array autocorrelations –which the receiver could only estimate with asymptotically large number of received-signal snapshots. Due to a finite number of received-signal snapshots available at the receiver and the fact that these methods have been designed under no optimality criterion, the estimated autocorrelations diverge from the nominal ones and attain arbitrary MSE performance. In this case, the two methods no longer coincide in MSE estimation performance.

In this Chapter, we focus on the autocorrelation combining step of coprime array processing and make the following contributions:

Contribution i. For the first time, we present in closed-form the MSE of both selection and averaging autocorrelation combining and show that, for any number of sample support, averaging combining offers superior MSE estimation performance compared to selection combining. The theoretical results are validated with numerical simulations.

Contribution ii. Motivated by prior works which treat source angles as statistical random variables [196, 197], we make the mild assumption that the source directions are independently and identically distributed random variables. Under this assumption, we design a novel coprime array receiver equipped with a linear autocorrelation combiner designed under the Minimum-MSE (MMSE) optimality criterion. The proposed MMSE combiner minimizes in the mean –i.e., for any configuration of DoAs– the error in estimating the physical-array autocorrelations with respect to the MSE metric. We conduct extensive numerical studies and compare the performance of the proposed MMSE combiner to existing counterparts, with respect to autocorrelation estimation error and DoA estimation.

Contribution iii. We propose an optimization framework for computing an improved coarray autocorrelation matrix estimate that satisfies the structure properties (i)-(iv). In practice, we iteratively solve a sequence of distinct structure-optimization problems, obtaining upon convergence, an improved estimate that satisfies properties (i)-(iv). The proposed framework is accompanied by formal convergence analysis. Numerical studies illustrate that the proposed method outperforms standard counterparts, both in estimation error and DoA estimation.

The rest of this Chapter is organized as follows. In Section 4.2, we present the signal model and state the problem of interest. In Section 4.3, we review the existing selection and averaging autocorrelation combining methods for coprime arrays and existing coarray autocorrelation matrix estimates. In Section 4.4, we present the closed-form expressions of selection and averaging autocorrelation combining approaches. Then, in Section 4.5, we present the proposed MMSE autocorrelation combining. After, in Section 4.6, we present the proposed structured coarray autocorrelation matrix estimate. Conclusions are drawn in Section 4.7.

The contributions presented in this Chapter have also been presented in [155, 156, 198, 199].

4.2 Signal Model and Problem Statement

Consider coprime naturals (M, N) such that $M < N$. A coprime array equipped with $L = 2M+N-1$ elements is formed by overlapping a ULA with N antenna elements at positions $p_{M,i} = (i-1)M\Delta$, $i = 1, 2, \ldots, N$, and a ULA equipped with $2M-1$ antenna elements at positions $p_{N,i} = iN\Delta$, $i = 1, 2, \ldots, 2M-1$. The reference unit-spacing Δ is typically set to one-half wavelength at the operating frequency. The positions of the L elements are the entries of the element-location vector $\mathbf{p} \overset{\triangle}{=} \text{sort}([p_{M,1}, \ldots, p_{M,N}, p_{N,1}, \ldots, p_{N,2M-1}]^\top)$, where sort$(\cdot)$ sorts the entries of its vector argument in ascending order and the superscript '$\top$' denotes matrix transpose. We assume that narrowband signals impinge on the array from $K < MN + M$ sources with propagation speed c and carrier frequency f_c. Assuming far-field conditions, a signal from source $k \in \{1, 2, \ldots, K\}$ impinges on the array from direction $\theta_k \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ with respect to the broadside. The array response vector for source k is $\mathbf{s}(\theta_k) \triangleq [v(\theta_k)^{[\mathbf{p}]_1}, \ldots, v(\theta_k)^{[\mathbf{p}]_L}]^\top \in \mathbb{C}^{L\times 1}$, with $v(\theta) \triangleq \exp\left(\frac{-j2\pi f_c}{c}\sin(\theta)\right)$ for every $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$. Accordingly, the qth collected received-signal snapshot is of the form

$$\mathbf{y}_q = \sum_{k=1}^{K} \mathbf{s}(\theta_k)\xi_{q,k} + \mathbf{n}_q \in \mathbb{C}^{L\times 1}, \tag{4.1}$$

where $\xi_{q,k} \sim \mathcal{CN}(0, d_k)$ is the qth symbol for source k (power-scaled and flat-fading-channel processed) and $\mathbf{n}_q \sim \mathcal{CN}(\mathbf{0}_L, \sigma^2\mathbf{I}_L)$ models Additive White Gaussian Noise (AWGN). We make the common assumptions that the random variables are statistically independent across different snapshots and that symbols from different sources are independent of each other and of every entry of $\mathbf{n}_q$. The received-signal autocorrelation matrix is given by

$$\mathbf{R}_y \triangleq \mathbb{E}\{\mathbf{y}_q\mathbf{y}_q^H\} = \mathbf{S}\ \mathrm{diag}(\mathbf{d})\ \mathbf{S}^H + \sigma^2\mathbf{I}_L, \tag{4.2}$$

where $\mathbf{d} \triangleq [d_1, \ldots, d_K]^\top \in \mathbb{R}_+^{K\times 1}$ is the source-power vector and $\mathbf{S} \triangleq [\mathbf{s}(\theta_1), \ldots, \mathbf{s}(\theta_K)] \in \mathbb{C}^{L\times K}$ is the array-response matrix. We define

$$\mathbf{r} \triangleq \mathrm{vec}(\mathbf{R}_y) = \sum_{i=1}^{K} \mathbf{a}(\theta_i)d_i + \sigma^2\mathbf{i}_L \in \mathbb{C}^{L^2\times 1}, \tag{4.3}$$

where $\mathrm{vec}(\cdot)$ returns the column-wise vectorization of its matrix argument, $\mathbf{a}(\theta_i) \triangleq \mathbf{s}(\theta_i)^* \otimes \mathbf{s}(\theta_i)$, $\mathbf{i}_L \triangleq \mathrm{vec}(\mathbf{I}_L)$, the superscript '$*$' denotes complex conjugate, and '$\otimes$' is the Kronecker product of matrices [200]. By coprime number theory [153], for every $n \in \{-L'+1, -L'+2, \ldots, L'-1\}$ with $L' \triangleq MN + M$, there exists a well-defined set of indices $\mathcal{J}_n \subset \{1, 2, \ldots, L^2\}$, such that

$$[\mathbf{a}(\theta)]_j = v(\theta)^n \quad \forall j \in \mathcal{J}_n, \tag{4.4}$$

for every $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$. We henceforth consider that $\mathcal{J}_n$ contains all j indices that satisfy (4.4). In view of (4.4), a coprime array receiver assembles a linear combining matrix $\mathbf{E} \in \mathbb{R}^{L^2\times(2L'-1)}$ and forms a length-$(2L'-1)$ autocorrelation-vector $\mathbf{r}_{\mathrm{co}}$, each element of which corresponds to a single set $\mathcal{J}_n$, for every $n \in \{1-L', 2-L', \ldots, L'-1\}$, by conducting linear *processing*[1] (e.g., $\mathbf{E}^\top\mathbf{r}$) to

[1] Existing coprime array autocorrelation processing methods in the literature are reviewed in Section 4.3.

the autocorrelations in $\mathbf{r}$. That is, there exists linear combiner $\mathbf{E}$ such that

$$\mathbf{r}_{\text{co}} = \mathbf{E}^\top \mathbf{r} = \sum_{k=1}^{K} \mathbf{a}_{\text{co}}(\theta_k) d_k + \sigma^2 \mathbf{e}_{L',2L'-1}, \tag{4.5}$$

where $\mathbf{a}_{\text{co}}(\theta) \triangleq [v(\theta)^{1-L'}, v(\theta)^{2-L'}, \ldots, v(\theta)^{L'-1}]$ for any $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, and, for any $p \leq P \in \mathbb{N}_+$, $\mathbf{e}_{p,P}$ is the pth column of $\mathbf{I}_P$. Thereafter, the receiver applies spatial-smoothing to organize the sampled autocorrelations as the matrix

$$\mathbf{Z} \triangleq \Phi\left(\mathbf{r}_{\text{co}}\right) \triangleq \mathbf{F}(\mathbf{I}_{L'} \otimes \mathbf{r}_{\text{co}}) \in \mathbb{C}^{L' \times L'}, \tag{4.6}$$

where $\mathbf{F} \triangleq [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{L'}]$ and $\mathbf{F}_m \triangleq [\mathbf{0}_{L' \times (L'-m)}, \mathbf{I}_{L'}, \mathbf{0}_{L' \times (m-1)}]$ $\forall m \in \{1, 2, \ldots, L'\}$. This approach is also known as the *Augmented Matrix* in the coprime array processing literature [162]. Importantly, under nominal statistics $\mathbf{Z}$ becomes the autocorrelation matrix of a length-L' ULA with antenna elements at locations $\{0, 1, \ldots, L'-1\}\Delta$. That is,

$$\mathbf{Z} = \mathbf{S}_{\text{co}} \text{diag}(\mathbf{d}) \mathbf{S}_{\text{co}}^H + \sigma^2 \mathbf{I}_{L'}, \tag{4.7}$$

where it holds that $[\mathbf{S}_{\text{co}}]_{m,k} = v(\theta_k)^{m-1}$ for every $m \in \{1, 2, \ldots, L'\}$ and $k \in \{1, 2, \ldots, K\}$. Standard MUSIC DoA estimation can be applied to $\mathbf{Z}$. Let the columns of $\mathbf{U} \in \mathbb{C}^{L' \times K}$ be the dominant left-hand singular vectors of $\mathbf{Z}$, corresponding to its K highest singular values, acquired by means of Singular Value Decomposition (SVD). Defining $\mathbf{v}(\theta) = [1, v(\theta), \ldots, v(\theta)^{L'-1}]^\top$, we can accurately decide that $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$ belongs in $\Theta \triangleq \{\theta_1, \theta_2, \ldots, \theta_K\}$ if $(\mathbf{I}_{L'} - \mathbf{U}\mathbf{U}^H)\mathbf{v}(\theta) = \mathbf{0}_{L'}$ is satisfied for some θ. Equivalently, we can resolve the angles in Θ by the K (smallest) local minima of the standard MUSIC spectrum [201]

$$P_{\text{MUSIC}}(\theta) = \left\| \left(\mathbf{I}_{L'} - \mathbf{U}\mathbf{U}^H \right) \mathbf{v}(\theta) \right\|_2^2. \tag{4.8}$$

In practice, $\mathbf{R}_y$ in (4.2) is unknown to the receiver and sample-average estimated by a collection of Q received-signal snapshots in $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_Q]$ by

$$\hat{\mathbf{R}}_y = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{y}_q \mathbf{y}_q^H. \tag{4.9}$$

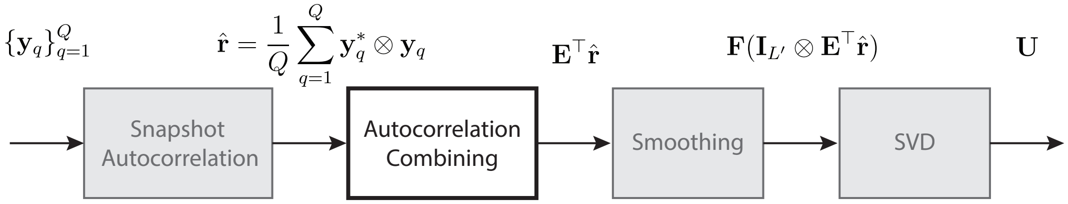

$\{\mathbf{y}_q\}_{q=1}^{Q}$
$\hat{\mathbf{r}} = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{y}_q^* \otimes \mathbf{y}_q$
$\mathbf{E}^\top\hat{\mathbf{r}}$
$\mathbf{F}(\mathbf{I}_{L'} \otimes \mathbf{E}^\top\hat{\mathbf{r}})$
$\mathbf{U}$
Snapshot Autocorrelation
Autocorrelation Combining
Smoothing
SVD

Figure 4.1. Coprime processing steps: from a collection of samples $\{\mathbf{y}_q\}_{q=1}^{Q}$ to the estimated coarray signal-subspace basis $\mathbf{U}$.

Accordingly, the physical-array autocorrelation-vector $\mathbf{r}$ in (4.3) is estimated by

$$\hat{\mathbf{r}} \overset{\triangle}{=} \operatorname{vec}(\hat{\mathbf{R}}_y) = \frac{1}{Q}\sum_{q=1}^{Q}\mathbf{y}_q^* \otimes \mathbf{y}_q. \tag{4.10}$$

The receiver then conducts linear combining to the estimated autocorrelation vector $\hat{\mathbf{r}}$ to obtain an estimate of $\mathbf{r}_{\text{co}}$, $\hat{\mathbf{r}}_{\text{co}} = \mathbf{E}^\top\hat{\mathbf{r}}$. The estimation error $\|\mathbf{r}_{\text{co}} - \hat{\mathbf{r}}_{\text{co}}\|_2$ depends on how well the linear combiner $\mathbf{E}$ estimates the nominal physical-array autocorrelations. Accordingly, $\mathbf{Z}$ in (4.7) is estimated by $\hat{\mathbf{Z}} \overset{\triangle}{=} \Phi\left(\hat{\mathbf{r}}_{\text{co}}\right)$.

Finally, MUSIC DoA estimation can be applied using the K dominant left-hand singular vectors of $\hat{\mathbf{Z}}$ instead of those of $\mathbf{Z}$. Of course, in practice, there is an inherent DoA estimation error due to the mismatch between $\mathbf{Z}$ and $\hat{\mathbf{Z}}$. A schematic illustration of the coprime array processing steps presented above is offered in Fig. 4.1.

4.3 Technical Background

4.3.1 Selection Combining

The most commonly considered autocorrelation combining method is *selection combining* [153] based on which the receiver selects any single index $j_n \in \mathcal{J}_n$, for $n \in \{-L'+1, \ldots, L'-1\}$, and forms the *selection combining* matrix

$$\mathbf{E}_{\text{sel}} \overset{\triangle}{=} \left[\mathbf{e}_{j_{1-L'},L^2}, \mathbf{e}_{j_{2-L'},L^2}, \ldots, \mathbf{e}_{j_{L'-1},L^2}\right] \in \mathbb{R}^{L^2\times(2L'-1)}, \tag{4.11}$$

by which it processes the autocorrelations in $\mathbf{r}$, discarding by selection all duplicates –i.e., every entry with index in $\mathcal{J}_n \setminus j_n$, for every n– to form autocorrelation vector

$$\mathbf{r}_{\text{sel}} \triangleq \mathbf{E}_{\text{sel}}^{\top}\mathbf{r} \in \mathbb{R}^{2L'-1}. \tag{4.12}$$

Importantly, when the nominal entries of $\mathbf{r}$ are known to the receiver, $\mathbf{r}_{\text{sel}}$ coincides with $\mathbf{r}_{\text{co}}$ in (4.5), thus, applying spatial smoothing on $\mathbf{r}_{\text{sel}}$ yields the exact coarray autocorrelation matrix $\mathbf{Z} = \Phi\left(\mathbf{r}_{\text{sel}}\right)$. In contrast, when $\mathbf{r}$ is unknown to the receiver and estimated by $\hat{\mathbf{r}}$ in (4.10), $\mathbf{r}_{\text{sel}}$ in (4.12) is estimated by

$$\hat{\mathbf{r}}_{\text{sel}} = \mathbf{E}_{\text{sel}}^{\top}\hat{\mathbf{r}} \in \mathbb{R}^{2L'-1}. \tag{4.13}$$

Accordingly, the coarray autocorrelation matrix is estimated by $\hat{\mathbf{Z}}_{\text{sel}} \triangleq \Phi\left(\hat{\mathbf{r}}_{\text{sel}}\right)$.

4.3.2 Averaging Combining

Instead of selecting a single index in $\mathcal{J}_n$ by discarding duplicates, averaging combining [162] conducts averaging on all autocorrelation estimates in $\mathcal{J}_n$, for every $n \in \{1-L', 2-L', \ldots, L'-1\}$. That is, the receiver forms the *averaging combining* matrix

$$\mathbf{E}_{\text{avg}} \triangleq \left[\frac{1}{|\mathcal{J}_{1-L'}|}\sum_{j\in\mathcal{J}_{1-L'}}\mathbf{e}_{j,L^2}, \ldots, \frac{1}{|\mathcal{J}_{L'-1}|}\sum_{j\in\mathcal{J}_{L'-1}}\mathbf{e}_{j,L^2}\right]. \tag{4.14}$$

$|\cdot|$ denotes the cardinality (number of elements in a set) of its argument. Then, it processes the autocorrelation vector $\mathbf{r}$ to obtain

$$\mathbf{r}_{\text{avg}} \triangleq \mathbf{E}_{\text{avg}}^{\top}\mathbf{r} \in \mathbb{R}^{2L'-1}. \tag{4.15}$$

By (4.4) and the fact that $[\mathbf{i}_L]_j$ equals 1, if $j \in \mathcal{J}_0$ and 0 otherwise, it holds that, for any $n \in \{-L'+1, \ldots, L'-1\}$, $[\mathbf{r}]_j = \mathbf{e}_{j,L^2}^{\top}\mathbf{r}$ takes a constant value for every $j \in \mathcal{J}_n$. Thus, $\mathbf{r}_{\text{avg}}$ coincides with $\mathbf{r}_{\text{sel}}$ and $\mathbf{r}_{\text{co}}$. Therefore, similar to the selection combining approach, when $\mathbf{r}$ is known to the receiver, applying spatial smoothing on $\mathbf{r}_{\text{avg}}$ yields $\mathbf{Z} = \Phi\left(\mathbf{r}_{\text{avg}}\right)$. In practice, when $\mathbf{R}_y$ in (4.2) is estimated by $\hat{\mathbf{R}}_y$ in (4.9), $\mathbf{r}_{\text{avg}}$ is estimated by

$$\hat{\mathbf{r}}_{\text{avg}} \triangleq {\mathbf{E}_{\text{avg}}}^{\top}\hat{\mathbf{r}} \in \mathbb{R}^{2L'-1}. \tag{4.16}$$

Accordingly, $\mathbf{Z}$ is estimated by $\hat{\mathbf{Z}}_{\text{avg}} \stackrel{\triangle}{=} \Phi\left(\hat{\mathbf{r}}_{\text{avg}}\right)$.

4.3.3 Remarks on Existing Coarray Autocorrelation Matrix Estimates

The true coarray autocorrelation matrix $\mathbf{Z}$ has full-rank, is PD, Hermitian, and Toeplitz. If we consider $\mathbf{Z}_{\text{ss}} \stackrel{\triangle}{=} \frac{1}{L'}\mathbf{Z}\mathbf{Z}^H$, then an autocorrelation matrix estimate can be extracted from $\mathbf{Z}_{\text{ss}}$ as a scaled version of its Principal Squared Root (PSR) [153]

$$\mathbf{Z}_{\text{psr}} \stackrel{\triangle}{=} \sqrt{L'}\mathbf{Z}_{\text{ss}}^{\frac{1}{2}}. \tag{4.17}$$

We notice that $\mathbf{Z}\mathbf{Z}^H = \mathbf{Z}^2 = L'\mathbf{Z}_{\text{ss}}$. Moreover, $\mathbf{Z}_{\text{ss}}$ admits SVD $\mathbf{Z}_{\text{ss}} \stackrel{\text{svd}}{=} \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^H$ which implies that $\mathbf{Z}_{\text{psr}} = \mathbf{U}(\sqrt{L'}\boldsymbol{\Lambda}^{\frac{1}{2}})\mathbf{V}^H = \mathbf{Z}$. That is, $\mathbf{Z}_{\text{psr}}$ and $\mathbf{Z}$ coincide. Accordingly, in the (ideal) case of known statistics to the receiver, all estimates that we have discussed above coincide with the nominal autocorrelation matrix of the coarray and satisfy (i)-(iv).

However, in the practical case of unknown statistics (case of interest) to the receiver, the estimates above diverge from $\mathbf{Z}$ and satisfy only a subset of (i)-(iv): That is, $\hat{\mathbf{Z}}_{\text{avg}}$ and $\hat{\mathbf{Z}}_{\text{sel}}$ are guaranteed to be Hermitian and Toeplitz, however, they are not guaranteed to be PD –i.e., $\hat{\mathbf{Z}}_{\text{avg}}$ and $\hat{\mathbf{Z}}_{\text{sel}}$ can be indefinite estimates of $\mathbf{Z}$ [162]. Similarly, by adopting the superior averaging autocorrelation combining approach, $\mathbf{Z}_{\text{ss}}$ and $\mathbf{Z}_{\text{psr}}$ can be estimated by $\widehat{\mathbf{Z}}_{\text{ss}} \stackrel{\triangle}{=} \frac{1}{L'}\hat{\mathbf{Z}}_{\text{avg}}\hat{\mathbf{Z}}_{\text{avg}}^H$ and $\widehat{\mathbf{Z}}_{\text{psr}} \stackrel{\triangle}{=} \sqrt{L'}\widehat{\mathbf{Z}}_{\text{ss}}^{\frac{1}{2}}$, respectively. In view of the above, $\widehat{\mathbf{Z}}_{\text{psr}}$ is by construction a PD and Hermitian matrix estimate of the coarray autocorrelation matrix, however, it violates the Toeplitz structure-property of $\mathbf{Z}$. It follows that $\widehat{\mathbf{Z}}_{\text{psr}}$ and $\hat{\mathbf{Z}}_{\text{avg}}$ no longer coincide, however, their left-hand singular-valued singular vectors span the same signal subspace.

4.4 Contribution 1: Closed-form MSE Expressions for Selection and Averaging Combining

In general, estimates $\hat{\mathbf{r}}_{\text{sel}}$ and $\hat{\mathbf{r}}_{\text{avg}}$ diverge from $\mathbf{r}_{\text{co}}$ and attain MSE $\text{err}_r(\hat{\mathbf{r}}_{\text{sel}}) \stackrel{\triangle}{=} \mathbb{E}\{\|\mathbf{r}_{\text{co}} - \hat{\mathbf{r}}_{\text{sel}}\|_2^2\}$ and $\text{err}_r(\hat{\mathbf{r}}_{\text{avg}})$, respectively. $\hat{\mathbf{Z}}_{\text{sel}}$ and $\hat{\mathbf{Z}}_{\text{avg}}$ diverge from the true $\mathbf{Z}$ and attain MSE $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}}) \stackrel{\triangle}{=} \mathbb{E}\{\|\mathbf{Z} - \hat{\mathbf{Z}}_{\text{sel}}\|_F^2\}$ and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$, respectively. In this Section, we present for the first time closed-form MSE expressions for the estimation errors attained by selection and averaging combining.

For any sample support Q, the following Lemma 4.1 and Lemma 4.2 express in closed-form the

MSE attained by $\hat{\mathbf{r}}_{\text{sel}}$.

Lemma 4.1. *For any $n \in \{-L'+1, -L'+2, \ldots, L'-1\}$ and $j \in \mathcal{J}_n$, it holds*

$$e = \mathbb{E}\{|[\mathbf{r}]_j - [\hat{\mathbf{r}}]_j|^2\} = \frac{(\mathbf{1}_K^\top \mathbf{d} + \sigma^2)^2}{Q}. \tag{4.18}$$

Lemma 4.2. $\hat{\mathbf{r}}_{\text{sel}}$ *attains MSE* $\text{err}_r(\hat{\mathbf{r}}_{\text{sel}}) = (2L'-1)\, e$.

In view of Lemma 4.2, the following Proposition naturally follows.

Proposition 4.1. $\hat{\mathbf{Z}}_{\text{sel}}$ *attains MSE* $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}}) = {L'}^2 e$.

Expectedly, as the sample-support Q grows asymptotically, e, $\text{err}_r(\hat{\mathbf{r}}_{\text{sel}})$, and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}})$ converge to zero.

For any sample support Q, the following Lemma 4.3 and Lemma 4.4 express in closed-form the MSE attained by $\hat{\mathbf{r}}_{\text{avg}}$, where $\dot{\mathbf{p}} \triangleq \mathbf{p} \otimes \mathbf{1}_L$, $\dot{\omega}_{i,j} \triangleq [\dot{\mathbf{p}}]_i - [\dot{\mathbf{p}}]_j$, and $\mathbf{z}_{i,j} \triangleq [v(\theta_1)^{\dot{\omega}_{i,j}}\ v(\theta_2)^{\dot{\omega}_{i,j}} \ldots\ v(\theta_K)^{\dot{\omega}_{i,j}}]^H$.

Lemma 4.3. *For any $n \in \{-L'+1, \ldots, L'-1\}$ and $j_n \in \mathcal{J}_n$, it holds that*

$$e_n = \mathbb{E}\left\{\left|[\mathbf{r}]_{j_n} - \frac{1}{|\mathcal{J}_n|}\sum_{j \in \mathcal{J}_n} [\hat{\mathbf{r}}]_j\right|^2\right\} = \frac{1}{Q}\left(\frac{2\sigma^2 \mathbf{1}_K^\top \mathbf{d} + \sigma^4}{|\mathcal{J}_n|} + \sum_{i \in \mathcal{J}_n}\sum_{j \in \mathcal{J}_n} \frac{|\mathbf{z}_{i,j}^H \mathbf{d}|^2}{|\mathcal{J}_n|^2}\right). \tag{4.19}$$

Lemma 4.4. $\hat{\mathbf{r}}_{\text{avg}}$ *attains MSE* $\text{err}_r(\hat{\mathbf{r}}_{\text{avg}}) = \sum_{n=1-L'}^{L'-1} e_n$.

By Lemma 4.4, the following Proposition naturally follows.

Proposition 4.2. $\hat{\mathbf{Z}}_{\text{avg}}$ *attains MSE* $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}}) = \sum\limits_{m=1}^{L'} \sum\limits_{n=1-m}^{L'-m} e_n$.

Similar to selection combining, as the sample-support Q grows asymptotically, e_n, $\text{err}_r(\hat{\mathbf{r}}_{\text{avg}})$, and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$ converge to zero. Complete proofs for the above Lemmas and Propositions are offered for the first time in the Appendix.

Remarks on Selection and Averaging Combining

In view of (4.19), $e = e_n$, when $|\mathcal{J}_n| = 1$. Also, (4.18) and (4.19) show that, as expected, e and e_n tend to zero as Q increases asymptotically. Using (4.18) and (4.19), we prove below the inequality $e_n \leq e$, which documents the MSE superiority of average-sampling over selection-sampling. We observe that, for any i, j, $|\mathbf{z}_{i,j}^H \mathbf{d}|^2 = |\sum_{k=1}^{K} [\mathbf{z}_{i,j}]_k^* d_k|^2 \leq |\sum_{k=1}^{K} |[\mathbf{z}_{i,j}]_k^* d_k||^2 = (\mathbf{1}_K^\top \mathbf{d})^2$ and, therefore,

$$e - e_n \geq \beta_n \triangleq \frac{|\mathcal{J}_n| - 1}{|\mathcal{J}_n| Q} (2\sigma^2 \mathbf{1}_K^\top \mathbf{d} + \sigma^4) \geq 0. \tag{4.20}$$

4.4.1 Numerical Studies

We consider an $(M = 2, N = 3)$ extended coprime array of $L = 6$ elements and $K = 7$ sources, transmitting from angles $\Theta = \{\theta_1 = -60^\circ, \theta_2 = -28^\circ, \theta_3 = -4^\circ, \theta_4 = 5^\circ, \theta_5 = 18^\circ, \theta_6 = 34^\circ, \theta_7 = 86^\circ\}$. We set $\sigma^2 = 1$ and signal-to-noise ratio (SNR) to 10dB, for all K sources; i.e., $d_k = 10\ \forall k$.

First, we let the number of snapshots Q vary from 20 to 140 in increments of 10. For each value of Q, we simulate estimation of $\mathbf{Z}$ by both selection-sampling ($\hat{\mathbf{Z}}_{\text{sel}}$) and averaging-sampling ($\hat{\mathbf{Z}}_{\text{avg}}$); then, we measure the corresponding squared-estimation errors $\|\mathbf{Z} - \hat{\mathbf{Z}}_{\text{sel}}\|_F^2$ and $\|\mathbf{Z} - \hat{\mathbf{Z}}_{\text{avg}}\|_F^2$. We repeat this procedure for $10,000$ independent symbol/noise realizations and plot the averaged squared-error in Fig. 4.2. In the same figure, we plot the theoretical MSEs $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}})$ and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$, as presented in Proposition 4.1 and Proposition 4.2, respectively. We observe that the numerically and theoretically calculated values for the MSEs coincide. All MSE curves start from a high value for $Q = 20$ and decrease monotonically as Q increases, with a trend for asymptotic convergence to 0, as $Q \to \infty$. We observe that, in accordance to Proposition 4.1 and Proposition 4.2, $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$ is significantly lower than $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}})$, for every value of Q.

Next, we set $Q = 30$ and calculate the MSE of selection and averaging in the estimation of $\mathbf{Z}$ for different values of K. Specifically, in Fig. 4.3, we plot the theoretical and numerical MSEs attained by the two estimators versus the number of transmitting sources, $K = 1, 2, \ldots, 7$, i.e., for $K = 1$, we consider signal transmission only from angle $\theta_1 = -60^\circ$ in Θ, while for $K = 3$, we consider transmission from angles $\theta_1 = -60^\circ, \theta_2 = -28^\circ$, and $\theta_3 = -4^\circ$. We observe that the performance gap between selection and averaging increases monotonically with K. This is in accordance with the observation that the lower-bound of the gap, β_n in (4.20), increases monotonically as K increases, for every n such that $|\mathcal{J}_n| > 1$ (when $|\mathcal{J}_n| = 1$, $\beta_n = 0$).

In the next two examples, we examine the association between the MSE in the estimation of $\mathbf{Z}$

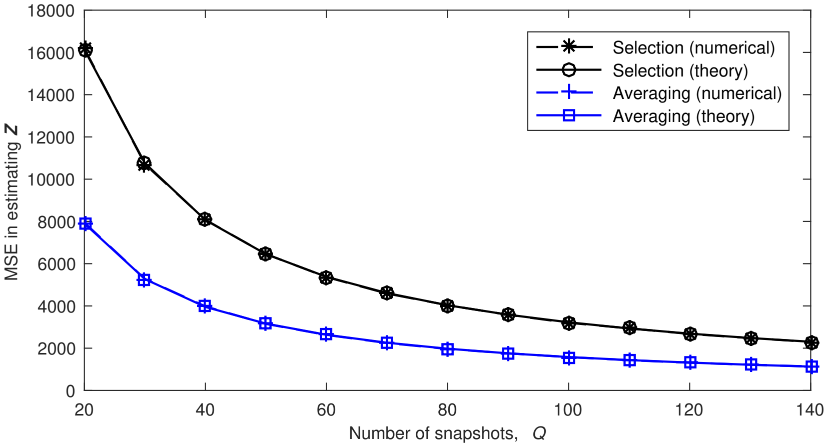


Figure 4.2. Theoretically and numerically calculated MSEs $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}})$ and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$, versus sample-support Q.

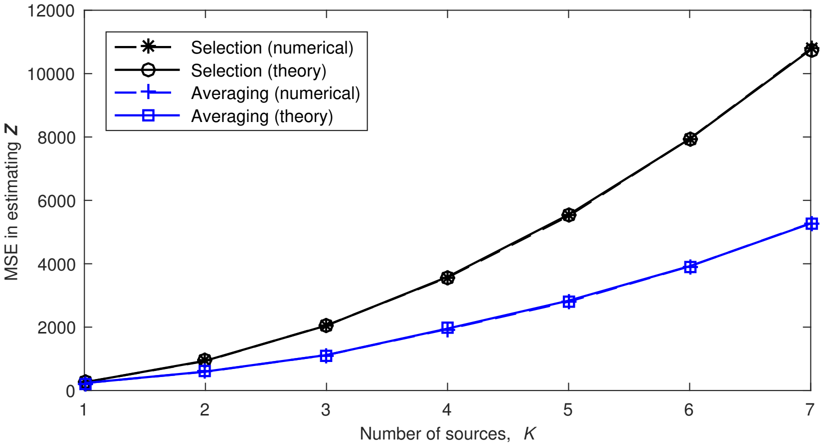


Figure 4.3. Theoretically and numerically calculated MSEs $\text{err}_Z(\hat{\mathbf{Z}}_{\text{sel}})$ and $\text{err}_Z(\hat{\mathbf{Z}}_{\text{avg}})$, versus the number of transmitting sources K.

and the MUSIC DoA-estimation performance. We start with measuring the error in the estimation of the signal-subspace span($\mathbf{S}_{co}$) by the dominant singular vectors of $\hat{\mathbf{Z}}_{\text{sel}}$ and $\hat{\mathbf{Z}}_{\text{avg}}$. Specifically, we set $K = 7$ and let Q vary in $\{20, 30, \ldots, 140\}$. Then, we compute over $10,000$ realizations the average values of the (normalized) squared subspace errors (SSE) $\frac{1}{2K}\|\mathbf{U}\mathbf{U}^H - \hat{\mathbf{Q}}_{sel}\hat{\mathbf{Q}}_{sel}^H\|_F^2$ and $\frac{1}{2K}\|\mathbf{U}\mathbf{U}^H - \hat{\mathbf{Q}}_{avg}\hat{\mathbf{Q}}_{avg}^H\|_F^2$, where $\mathbf{U}$ is a matrix with columns the K dominant left-hand singular vectors of $\mathbf{Z}$, and $\hat{\mathbf{Q}}_{sel}$ and $\hat{\mathbf{Q}}_{avg}$ are the K dominant left-hand singular vectors of $\hat{\mathbf{Z}}_{\text{sel}}$ and $\hat{\mathbf{Z}}_{\text{avg}}$, respectively, obtained through SVD. We plot the calculated subspace-MSEs in Fig. 4.4, versus Q. We observe that, as expected, the estimation errors in $\mathbf{Z}$ translate accurately to subspace-

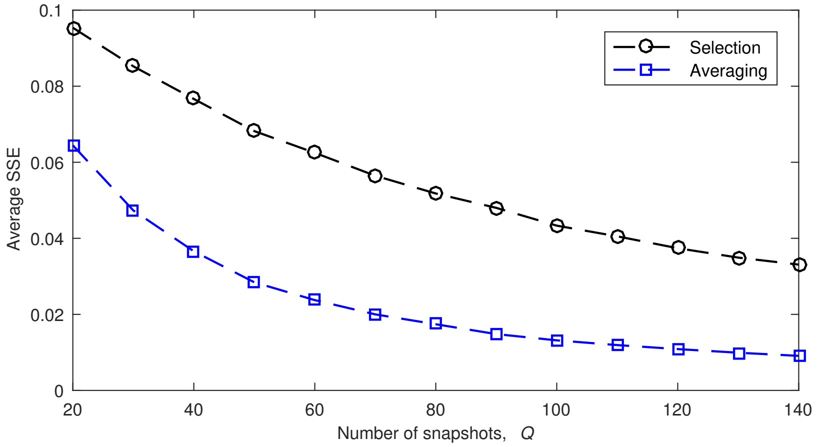


Figure 4.4. Average squared subspace error attained by selection and averaging sampling, versus sample-support Q.

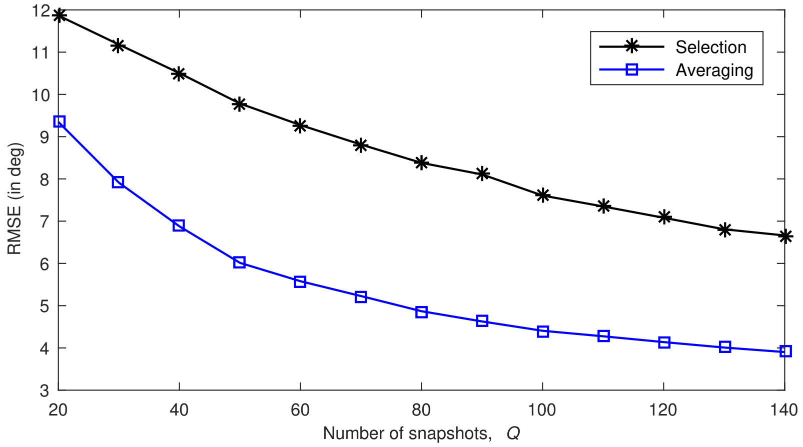


Figure 4.5. RMSE of coprime MUSIC DoA estimation attained by selection and averaging sampling, versus sample-support Q.

estimation errors. Thus, averaging autocorrelation sampling attains lower subspace estimation error than selection sampling, for every value of Q, with the maximum difference documented at the minimum tested value of Q.

Finally, for the same setup ($K = 7$, $Q = 20, 30, \ldots, 140$), we conduct MUSIC DoA estimation in the form of (4.8), employing, instead of the ideal signal-subspace descriptor $\mathbf{U}\mathbf{U}^H$, the estimated projection matrices $\hat{\mathbf{Q}}_{sel}\hat{\mathbf{Q}}_{sel}^H$ (selection) and $\hat{\mathbf{Q}}_{avg}\hat{\mathbf{Q}}_{avg}^H$ (sampling), calculated upon the estimates $\hat{\mathbf{Z}}_{\text{sel}}$ and $\hat{\mathbf{Z}}_{\text{avg}}$, respectively. After $10,000$ independent realizations, we compute and plot in Fig. 4.5 the root-mean-squared-error (RMSE) for the two estimators, RMSE $= \sqrt{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{10000}\sum_{\text{r}=1}^{10000}(\theta_k - \hat{\theta}_{k,r})^2}$, versus the sample support Q. In the RMSE expression, r is the realization index and $\hat{\theta}_{k,r}$ is the kth angle, as estimated at the rth realization. We observe that, in accordance with the theoretical MSE results, averaging autocorrelation sampling allows for superior DoA estimation, compared to selection sampling, for every sample support Q.

4.5 Contribution 2: Minimum-MSE Autocorrelation Combining

We focus on the autocorrelation combining step of coprime-array processing (see Fig. 4.1) where the receiver applies linear combining matrix $\mathbf{E}$ to the estimated autocorrelations of the physical array. Arguably, a preferred receiver will attain consistently –i.e., for any possible configuration of DoAs, Θ– low squared-estimation error $\|\mathbf{r}_{\text{co}} - \mathbf{E}^\top \hat{\mathbf{r}}\|_2^2$. If this error is exactly equal to zero, then at the fourth step of coprime-array processing, MUSIC will identify the exact DoAs in Θ.

In this work, we treat the DoAs in Θ as independently and identically distributed (i.i.d.) random variables and design a coprime array receiver equipped with linear combiner $\mathbf{E}$ such that $\|\mathbf{r}_{\text{co}} - \mathbf{E}^\top \hat{\mathbf{r}}\|_2^2$ is minimized in the mean. We assume that, $\forall k, \theta_k \in \Theta$ is a random variable with probability distribution $\mathcal{D}(a,b)$ –e.g., uniform, truncated normal, or, other– where a and b denote the limits of the support of $\mathcal{D}$. Under this assumption, we seek the Minimum-MSE combining matrix $\mathbf{E}$ that minimizes $\mathbb{E}\{\|\mathbf{r}_{\text{co}} - \mathbf{E}^\top \hat{\mathbf{r}}\|_2^2\}$.

Probability distributions of angular variables is not a new concept. In fact, angular variables have been modeled by the *von Mises* Probability Density Function (PDF) which can include (or, nearly approximate) standard distributions such as uniform, Gaussian, and wrapped Gaussian, to name a few, by tuning a parameter in the PDF expression [196, 197]. Angular distributions have also been considered for Bayesian-based beamforming [202, 203].

In the most general case and in lieu of any pertinent prior information at the receiver, the DoAs in Θ can, for instance, be assumed to be the uniformly distributed in $(-\frac{\pi}{2}, \frac{\pi}{2}]$ –i.e., $\mathcal{D}(a,b) \equiv \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$. In the sequel, we derive the Minimum-MSE combining matrix *for any* continuous probability distribution $\mathcal{D}(a,b)$, $-\frac{\pi}{2} < a < b \leq \frac{\pi}{2}$.

First, we introduce new notation to the problem statement and formulate the MSE-Minimization problem. We let

$$\mathbf{A} \overset{\triangle}{=} \Big[\mathbf{S}\ \text{diag}([\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_K}]), \sigma \mathbf{I}_L\Big] \in \mathbb{C}^{L \times (K+L)}. \tag{4.21}$$

The autocorrelation matrix $\mathbf{R}_y$ in (4.2) is factorized as $\mathbf{R}_y = \mathbf{A}\mathbf{A}^H$. Then, we define

$$\mathbf{V} \overset{\triangle}{=} \mathbf{A}^* \otimes \mathbf{A} \in \mathbb{C}^{L^2 \times (K+L)^2} \tag{4.22}$$

and $\mathbf{i} \overset{\triangle}{=} \text{vec}(\mathbf{I}_{K+L}) \in \mathbb{R}^{(K+L)^2}$, where $\mathbf{I}_{K+L}$ is the $(K+L)$-size identity matrix. It follows that $\mathbf{r}$ takes the form

$$\mathbf{r} = \mathbf{V}\mathbf{i}. \tag{4.23}$$

In view of the above, $\mathbf{r}_{\text{co}}$ (or, $\mathbf{r}_{\text{sel}}$ and $\mathbf{r}_{\text{avg}}$ in (4.12) and (4.15), respectively) can be expressed as $\mathbf{E}_{\text{sel}}^\top \mathbf{V}\mathbf{i} = \mathbf{E}_{\text{avg}}^\top \mathbf{V}\mathbf{i}$. We observe that $\forall q \in \{1, 2, \ldots, Q\}$, there exists vector $\mathbf{x}_q \sim \mathcal{CN}(\mathbf{0}_{K+L}, \mathbf{I}_{K+L})$, pertinent to $\mathbf{y}_q$, such that $\mathbf{y}_q = \mathbf{A}\mathbf{x}_q$. By the signal model, $\mathbf{x}_q$ is statistically independent from $\mathbf{x}_p$ for any $(p,q) \in \{1, 2, \ldots, Q\}^2$ such that $p \neq q$. The physical-array autocorrelation matrix estimate $\hat{\mathbf{R}}_y$ in (4.9) is expressed as $\hat{\mathbf{R}}_y = \mathbf{A}\mathbf{W}\mathbf{A}^H$, where $\mathbf{W} \overset{\triangle}{=} \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{x}_q \mathbf{x}_q^H$. Moreover, by defining $\mathbf{w} \overset{\triangle}{=} \text{vec}(\mathbf{W}) = \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{x}_q^* \otimes \mathbf{x}_q$, the estimate $\hat{\mathbf{r}}$ in (4.10) takes the form

$$\hat{\mathbf{r}} = \mathbf{V}\mathbf{w}. \tag{4.24}$$

By (4.23) and (4.24), we propose to design linear combiner $\mathbf{E}_{\text{MMSE}}$ by formulating and solving the MSE-Minimization

$$\underset{\mathbf{E} \in \mathbb{C}^{L^2 \times (2L'-1)}}{\arg\min} \; \underset{\Theta, \mathbf{w}}{\mathbb{E}} \left\{ \left\| \mathbf{E}^H \mathbf{V}\mathbf{w} - \mathbf{E}_{\text{sel}}^\top \mathbf{V}\mathbf{i} \right\|_2^2 \right\}. \tag{4.25}$$

Of course, if we replace $\mathbf{E}_{\text{sel}}$ by $\mathbf{E}_{\text{avg}}$ in (4.25), the resulting problem will be equivalent. In the sequel, we show that a closed-form solution to (4.25) exists for any finite value of sample support Q and present it step-by-step.

We commence our solution by defining $\mathbf{G} \overset{\triangle}{=} \mathbf{V}\mathbf{w}\mathbf{w}^H\mathbf{V}^H \in \mathbb{C}^{L^2 \times L^2}$ and $\mathbf{H} \overset{\triangle}{=} \mathbf{V}\mathbf{w}\mathbf{i}^H\mathbf{V}^H \in \mathbb{C}^{L^2 \times L^2}$. Then, the problem in (4.25) simplifies to

$$\underset{\mathbf{E} \in \mathbb{C}^{L^2 \times (2L'-1)}}{\arg\min} \; \underset{\Theta, \mathbf{w}}{\mathbb{E}} \left\{ \text{Tr}\left(\mathbf{E}^H \mathbf{G}\mathbf{E}\right) - 2\Re\left\{ \text{Tr}\left(\mathbf{E}^H \mathbf{H}\mathbf{E}_{\text{sel}}\right)\right\}\right\}, \tag{4.26}$$

where $\Re\{\cdot\}$ extracts the real part of its argument and $\text{Tr}(\cdot)$ returns the sum of the diagonal entries of its argument. Furthermore, we define $\mathbf{G}_{\mathbb{E}} \overset{\triangle}{=} \mathbb{E}_{\Theta,\mathbf{w}}\{\mathbf{G}\}$ and $\mathbf{H}_{\mathbb{E}} \overset{\triangle}{=} \mathbb{E}_{\Theta,\mathbf{w}}\{\mathbf{H}\}$. Then, (4.26) takes

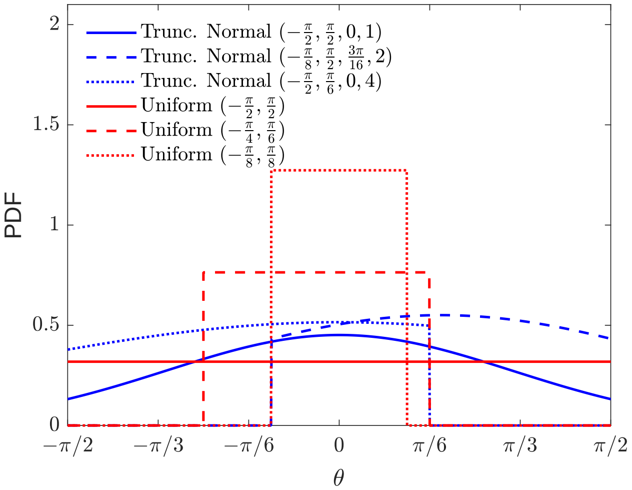


Figure 4.6. Probability density function $f(\theta)$ for different distributions and support sets.

the equivalent form

$$\underset{\mathbf{E} \in \mathbb{C}^{L^2 \times (2L'-1)}}{\arg\min} \left\{ \operatorname{Tr}\left(\mathbf{E}^H \mathbf{G}_{\mathbb{E}} \mathbf{E}\right) - 2\Re\left\{ \operatorname{Tr}\left(\mathbf{E}^H \mathbf{H}_{\mathbb{E}} \mathbf{E}_{\text{sel}}\right)\right\}\right\}. \tag{4.27}$$

Next, we focus on deriving closed-form expressions for $\mathbf{G}_{\mathbb{E}}$ and $\mathbf{H}_{\mathbb{E}}$ that will allow us to solve (4.27) and obtain the Minimum-MSE linear combiner $\mathbf{E}_{\text{MMSE}}$. At the core of our developments lies the observation that, $\forall \theta \sim \mathcal{D}(a, b)$ with PDF $f(\theta)$ and scalar $x \in \mathbb{R}$, it holds

$$\mathcal{I}(x) \overset{\triangle}{=} \underset{\theta}{\mathbb{E}}\{v(\theta)^x\} = \int_a^b f(\theta) \exp\left(-jx \frac{2\pi f_c}{c} \sin\theta\right) d\theta. \tag{4.28}$$

The integral $\mathcal{I}(x)$ can be approximated within some numerical error tolerance with numerically efficient vectorized methods [204–206]. In Fig. 4.6, we offer visual illustration examples of $f(\theta)$ when θ follows a uniform distribution in (a, b) –i.e., $\theta \sim \mathcal{U}(a, b)$– or, a truncated normal distribution in (a, b) with mean μ and variance σ^2 –i.e., $\theta \sim \mathcal{TN}(a, b, \mu, \sigma^2)$.

That is,

$$f(\theta) = \begin{cases} \dfrac{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\theta-\mu}{\sigma}\right)^2\right)}{\frac{1}{2}\left(\operatorname{erf}\left(\frac{b-\mu}{\sigma\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a-\mu}{\sigma\sqrt{2}}\right)\right)}, & \theta \sim \mathcal{TN}(a, b, \mu, \sigma^2), \\ \frac{1}{b-a}, & \theta \sim \mathcal{U}(a, b), \end{cases} \tag{4.29}$$

where $\operatorname{erf}(\cdot)$ denotes the Error Function [207]. In the special case that $\mathcal{D}(a, b) \equiv \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$, $\mathcal{I}(x)$ coincides with $J_0(x\frac{2\pi f_c}{c})$: the 0th order Bessel function of the first kind [208] for which there exist

Table 4.1: Entry-wise closed-form expression for matrix $\mathbf{V}$ defined in (4.22).

$[\mathbf{V}]_{i,j}$	Condition on (i,j)
$\sqrt{[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}}v(\theta_{[\ddot{\mathbf{u}}]_j})^{[\ddot{\mathbf{p}}]_i}v(\theta_{[\dot{\mathbf{u}}]_j})^{-[\dot{\mathbf{p}}]_i}$	$[\ddot{\mathbf{u}}]_j, [\dot{\mathbf{u}}]_j \leq K$
$\sigma\sqrt{[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}}v(\theta_{[\dot{\mathbf{u}}]_j})^{-[\dot{\mathbf{p}}]_i}$	$[\dot{\mathbf{u}}]_j \leq K$ and $[\ddot{\mathbf{u}}]_j = [\ddot{\mathbf{v}}]_i + K$
$\sigma\sqrt{[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}}v(\theta_{[\ddot{\mathbf{u}}]_j})^{[\ddot{\mathbf{p}}]_i}$	$[\ddot{\mathbf{u}}]_j \leq K$ and $[\dot{\mathbf{u}}]_j = [\dot{\mathbf{v}}]_i + K$
σ^2	$[\dot{\mathbf{u}}]_j = [\dot{\mathbf{v}}]_i + K$ and $[\ddot{\mathbf{u}}]_j = [\ddot{\mathbf{v}}]_i + K$
0	otherwise
Auxiliary variables used in the above conditions/expressions: $\mathbf{s}_x = [1,2,\ldots,x]^\top$, $\dot{\mathbf{u}} = \mathbf{1}_{K+L} \otimes \mathbf{s}_{K+L}$, $\ddot{\mathbf{u}} = \mathbf{s}_{K+L} \otimes \mathbf{1}_{K+L}$, $\dot{\mathbf{v}} = \mathbf{1}_L \otimes \mathbf{s}_L$, $\ddot{\mathbf{v}} = \mathbf{s}_L \otimes \mathbf{1}_L$.	

look-up tables. Next, we define the indicator function $\delta(x)$ which equals 1 if $x = 0$ and assumes a value of zero otherwise. We provide in the following Lemma the statistics of the random variable $\mathbf{w}$ which appear in the closed-form expressions of $\mathbf{G}_{\mathbb{E}}$ and $\mathbf{H}_{\mathbb{E}}$.

Lemma 4.5. *The first-order and second-order statistics of the random variable* $\mathbf{w}$ *are given by* $\mathbb{E}_{\mathbf{w}}\{\mathbf{w}\} = \mathbf{i} \in \mathbb{R}^{(K+L)^2}$ *and* $\mathbb{E}_{\mathbf{w}}\{\mathbf{w}\mathbf{w}^H\} = \mathbf{i}\mathbf{i}^\top + \frac{1}{Q}\mathbf{I}_{(K+L)^2} \in \mathbb{R}^{(K+L)^2 \times (K+L)^2}$*, respectively.* □

A proof for Lemma 4.5 is offered in the Appendix. Next, we define $\ddot{\mathbf{p}} \overset{\triangle}{=} \mathbf{1}_L \otimes \mathbf{p}$ and $\omega_i \overset{\triangle}{=} [\dot{\mathbf{p}}]_i - [\ddot{\mathbf{p}}]_i$. In view of Lemma 4.5, we present an entry-wise closed-form expression for $\mathbf{H}_{\mathbb{E}}$ in the following Lemma.

Lemma 4.6. *For any* $(i,m) \in \{1,2,\ldots,L^2\}^2$,

$$[\mathbf{H}_{\mathbb{E}}]_{i,m} = \|\mathbf{d}\|_2^2 \mathcal{I}\left(\omega_i - \omega_m\right) + \sigma^4\delta(\omega_i)\delta(\omega_m) + \mathcal{I}(\omega_i)\mathcal{I}(-\omega_m)\left((\mathbf{1}_K^\top\mathbf{d})^2 - \|\mathbf{d}\|_2^2\right) \tag{4.30}$$

$$+ \sigma^2\left(\mathbf{1}_K^\top\mathbf{d}\right)\left(\delta(\omega_i)\mathcal{I}(-\omega_m) + \mathcal{I}(\omega_i)\delta(-\omega_m)\right). \tag{4.31}$$

□

A complete proof for Lemma 4.6 is also provided in the Appendix. Hereafter, we focus on deriving a closed-form expression for $\mathbf{G}_{\mathbb{E}}$. First, we define $\tilde{\mathbf{V}} \overset{\triangle}{=} \mathbf{V}\mathbf{V}^H$ the expectation of which, $\tilde{\mathbf{V}}_{\mathbb{E}} \overset{\triangle}{=} \underset{\Theta}{\mathbb{E}}\{\tilde{\mathbf{V}}\}$, appears in $\mathbf{G}_{\mathbb{E}}$. We observe that each entry of $\tilde{\mathbf{V}}$ can be expressed as a linear combination of the entries of $\mathbf{V}$. That is, for any $(i,m) \in \{1,2,\ldots,L^2\}^2$ and $j \in \{1,2,\ldots,(K+L)^2\}$, it holds

$$[\tilde{\mathbf{V}}]_{i,m} = \sum_{j=1}^{(K+L)^2} [\mathbf{V}]_{i,j}[\mathbf{V}^*]_{m,j}. \tag{4.32}$$

Table 4.2: Closed-form expression for $\gamma_j^{(i,m)}$ defined in (4.33) .

$\gamma_j^{(i,m)}$	Condition on (i,m,j)
$[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}v(\theta_{[\ddot{\mathbf{u}}]_{j]}})^{[\ddot{\mathbf{p}}]_i-[\ddot{\mathbf{p}}]_m}v(\theta_{[\dot{\mathbf{u}}]_j})^{[\dot{\mathbf{p}}]_m-[\dot{\mathbf{p}}]_i}$	$[\dot{\mathbf{u}}]_j{=}[\ddot{\mathbf{u}}]_j \leq K,$
$[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}v(\theta_{[\ddot{\mathbf{u}}]_j})^{[\ddot{\mathbf{p}}]_i-[\ddot{\mathbf{p}}]_m}v(\theta_{[\dot{\mathbf{u}}]_j})^{[\dot{\mathbf{p}}]_m-[\dot{\mathbf{p}}]_i}$	$[\dot{\mathbf{u}}]_j, [\ddot{\mathbf{u}}]_j \leq K;\ [\ddot{\mathbf{u}}]_j \neq [\dot{\mathbf{u}}]_j$
$\sigma^2[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}v(\theta_{[\dot{\mathbf{u}}]_j})^{[\dot{\mathbf{p}}]_m-[\dot{\mathbf{p}}]_i}$	$[\dot{\mathbf{u}}]_j \leq K;\ [\ddot{\mathbf{u}}]_j{-}K{=}[\ddot{\mathbf{v}}]_i{=}[\ddot{\mathbf{v}}]_m$
$\sigma^2[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}v(\theta_{[\ddot{\mathbf{u}}]_j})^{[\ddot{\mathbf{p}}]_i-[\ddot{\mathbf{p}}]_m}$	$[\ddot{\mathbf{u}}]_j \leq K;\ [\dot{\mathbf{u}}]_j{-}K{=}[\dot{\mathbf{v}}]_i{=}[\dot{\mathbf{v}}]_m$
σ^4	$[\dot{\mathbf{u}}]_j{-}K{=}[\dot{\mathbf{v}}]_i{=}[\dot{\mathbf{v}}]_m;\ [\ddot{\mathbf{u}}]_j{-}K{=}[\ddot{\mathbf{v}}]_i = [\ddot{\mathbf{v}}]_m$
0	otherwise

Auxiliary variables used in the above conditions/expressions:
$\dot{\mathbf{p}} = \mathbf{p} \otimes \mathbf{1}_L$, $\ddot{\mathbf{p}} = \mathbf{1}_L \otimes \mathbf{p}$, $\mathbf{s}_x = [1, 2, \ldots, x]^\top$, $\dot{\mathbf{u}} = \mathbf{1}_{K+L} \otimes \mathbf{s}_{K+L}$, $\ddot{\mathbf{u}} = \mathbf{s}_{K+L} \otimes \mathbf{1}_{K+L}$, $\dot{\mathbf{v}} = \mathbf{1}_L \otimes \mathbf{s}_L$, $\ddot{\mathbf{v}} = \mathbf{s}_L \otimes \mathbf{1}_L$.

Table 4.3: Closed-form expression for $\mathbb{E}_\Theta\{\gamma_j^{(i,m)}\}$.

$\mathbb{E}_\theta\{\gamma_j^{(i,m)}\}$	Condition on (i,m,j)
$[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}\mathcal{I}(\ddot{\omega}_{m,i} + \dot{\omega}_{i,m})$	$[\dot{\mathbf{u}}]_j{=}[\ddot{\mathbf{u}}]_j \leq K,$
$[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}\mathcal{I}(\ddot{\omega}_{m,i})\mathcal{I}(\dot{\omega}_{i,m})$	$[\dot{\mathbf{u}}]_j, [\ddot{\mathbf{u}}]_j \leq K;\ [\ddot{\mathbf{u}}]_j \neq [\dot{\mathbf{u}}]_j$
$\sigma^2[\mathbf{d}]_{[\dot{\mathbf{u}}]_j}\mathcal{I}(\ddot{\omega}_{m,i})$	$[\dot{\mathbf{u}}]_j \leq K;\ [\ddot{\mathbf{u}}]_j{-}K{=}[\ddot{\mathbf{v}}]_i{=}[\ddot{\mathbf{v}}]_m$
$\sigma^2[\mathbf{d}]_{[\ddot{\mathbf{u}}]_j}\mathcal{I}(\dot{\omega}_{i,m})$	$[\ddot{\mathbf{u}}]_j \leq K;\ [\dot{\mathbf{u}}]_j{-}K{=}[\dot{\mathbf{v}}]_i{=}[\dot{\mathbf{v}}]_m$
σ^4	$[\dot{\mathbf{u}}]_j{-}K{=}[\dot{\mathbf{v}}]_i{=}[\dot{\mathbf{v}}]_m;\ [\ddot{\mathbf{u}}]_j{-}K{=}[\ddot{\mathbf{v}}]_i = [\ddot{\mathbf{v}}]_m$
0	otherwise

Auxiliary variables used in the above conditions/expressions:
$\dot{\mathbf{p}} = \mathbf{p} \otimes \mathbf{1}_L$, $\ddot{\mathbf{p}} = \mathbf{1}_L \otimes \mathbf{p}$, $\dot{\omega}_{i,j} = [\dot{\mathbf{p}}]_i - [\dot{\mathbf{p}}]_j$, $\ddot{\omega}_{i,j} = [\ddot{\mathbf{p}}]_i - [\ddot{\mathbf{p}}]_j$, $\mathbf{s}_x = [1, 2, \ldots, x]^\top$, $\dot{\mathbf{u}} = \mathbf{1}_{K+L} \otimes \mathbf{s}_{K+L}$, $\ddot{\mathbf{u}} = \mathbf{s}_{K+L} \otimes \mathbf{1}_{K+L}$, $\dot{\mathbf{v}} = \mathbf{1}_L \otimes \mathbf{s}_L$, $\ddot{\mathbf{v}} = \mathbf{s}_L \otimes \mathbf{1}_L$.

An entry-wise closed-form expression for $\mathbf{V}$, in terms of Θ, is offered in Table 4.1. Then, for any triplet (i, m, j) such that $(i, m) \in \{1, 2, \ldots, L^2\}^2$ and $j \in \{1, 2, \ldots, (K + L)^2\}$, we derive a closed-form expression for

$$\gamma_j^{(i,m)} \triangleq [\mathbf{V}]_{i,j}[\mathbf{V}^*]_{m,j}, \tag{4.33}$$

in Table 4.2. Accordingly, $\mathbb{E}_\Theta\{\gamma_j^{(i,m)}\}$ is offered in Table 4.3, based on which, the (i, m)th entry of $\tilde{\mathbf{V}}_\mathbb{E}$ is computed as

$$[\tilde{\mathbf{V}}_\mathbb{E}]_{i,m} = \sum_{j=1}^{(K+L)^2} \mathbb{E}_\Theta\{\gamma_j^{(i,m)}\}. \tag{4.34}$$

In view of the above, we offer a closed-form expression for matrix $\mathbf{G}_\mathbb{E}$ in the following Lemma.

Lemma 4.7. *Matrix $\mathbf{G}_{\mathbb{E}}$ is given by $\mathbf{G}_{\mathbb{E}} = \mathbf{H}_{\mathbb{E}} + \frac{1}{Q}\tilde{\mathbf{V}}_{\mathbb{E}}$.* □

A proof for Lemma 4.7 is offered in the Appendix. Next, we differentiate (4.27) with respect to $\mathbf{E}$, set its derivative to zero, and obtain

$$\left(\mathbf{H}_{\mathbb{E}} + \frac{1}{Q}\tilde{\mathbf{V}}_{\mathbb{E}}\right)\mathbf{E}_{\text{MMSE}} = \mathbf{H}_{\mathbb{E}}\mathbf{E}_{\text{sel}}. \tag{4.35}$$

We observe that (4.35) is, in practice, a collection of $(2L'-1)$ systems of linear equations. Let $\mathbf{E}_{\text{MMSE}} = [\mathbf{e}_1, \ldots, \mathbf{e}_{2L'-1}]$ and $\mathbf{c}_i = [\mathbf{H}_{\mathbb{E}}\mathbf{E}_{\text{sel}}]_{:,i} \forall i \in \{1, 2, \ldots, 2L'-1\}$. Solving (4.35) is equivalent to solving, for every i,

$$\mathbf{G}_{\mathbb{E}}\mathbf{e}_i = \mathbf{c}_i. \tag{4.36}$$

For any i such that $\mathbf{c}_i \in \text{span}(\mathbf{G}_{\mathbb{E}})$, (4.36) has at least one exact solution $\mathbf{e}_i = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{c}_i + \mathbf{b}_i$, where $\mathbf{G}_{\mathbb{E}}$ admits SVD $\mathbf{U}_{L^2\times\rho}\mathbf{\Sigma}_{\rho\times\rho}\mathbf{V}^H_{\rho\times L^2}$, $\rho = \text{rank}(\mathbf{G}_{\mathbb{E}})$, and $\mathbf{b}_i$ is an arbitrary vector in the nullspace of $\mathbf{G}_{\mathbb{E}}$ which is denoted by $\mathcal{N}(\mathbf{G}_{\mathbb{E}})$. In the special case that $\rho = L^2$, that is, $\mathbf{G}_{\mathbb{E}}$ has full-rank, then $\mathcal{N}(\mathbf{G}_{\mathbb{E}}) = \mathbf{0}_{L^2}$ and there exists a unique solution $\mathbf{e}_i = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{c}_i$. If, on the other hand, $\exists i$ such that $\mathbf{c}_i \notin \text{span}(\mathbf{G}_{\mathbb{E}})$, then (4.36) does not have an exact solution and a Least Squares (LS) approach can be followed by solving $\min_{\mathbf{e}_i} \|\mathbf{G}_{\mathbb{E}}\mathbf{e}_i - \mathbf{c}_i\|_2^2$. Interestingly, it is easy to show that the LS solution is the same as before –i.e., $\mathbf{e}_i = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{c}_i + \mathbf{b}_i$, where $\mathbf{b}_i \in \mathcal{N}(\mathbf{G}_{\mathbb{E}})$. In every case, each column of $\mathbf{E}_{\text{MMSE}}$ can be computed in closed-form as

$$\mathbf{e}_i = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^H\mathbf{c}_i + \mathbf{b}_i, \quad \mathbf{b}_i \in \mathcal{N}(\mathbf{G}_{\mathbb{E}}). \tag{4.37}$$

In view of the above, we propose to process the autocorrelations in $\hat{\mathbf{r}}$ by the linear combiner $\mathbf{E}_{\text{MMSE}}$ to obtain the MMSE estimate of $\mathbf{r}_{\text{co}}$,

$$\hat{\mathbf{r}}_{\text{MMSE}} \stackrel{\triangle}{=} \mathbf{E}^{\top}_{\text{MMSE}}\hat{\mathbf{r}}. \tag{4.38}$$

In turn, we propose to Minimum-MSE estimate $\mathbf{Z}$ in (4.6) by $\hat{\mathbf{Z}}_{\text{MMSE}} = \Phi\left(\hat{\mathbf{r}}_{\text{MMSE}}\right)$.

Next, we discuss the computational complexity of forming $\mathbf{E}_{\text{MMSE}}$. First, we consider the cost of numerically approximating the integral $\mathcal{I}(x)$ in (4.28) for any DoA probability distribution $f(\theta)$, support set (a, b), and scalar x. There is rich literature on theory, methods, and complexity/accuracy trade-offs in numerical integration –a topic that extends well beyond the scope of this article. For the purpose of our complexity analysis, we consider that $\mathcal{I}(x)$ can be approximated within numerical error tolerance $\epsilon > 0$ by, e.g., trapezoidal, midpoint, and Simpson's interpolatory quadrature rules [204–206], with cost $\mathcal{O}(C)$, where $C = \frac{1}{\sqrt{\epsilon}}$. Next, we note that evaluating $\mathcal{I}(x)$ for every $x \in \mathcal{V} \overset{\triangle}{=} \{\{\omega_i\}_{i=1}^{L^2} \cup \{\omega_i - \omega_m\}_{i,m=1}^{L^2} \cup \{\dot{\omega}_i\}_{i=1}^{L^2} \cup \{\ddot{\omega}_i\}_{i=1}^{L^2} \cup \{\ddot{\omega}_{m,i} + \dot{\omega}_{i,m}\}_{i,m=1}^{L^2}\}$ costs at most $\mathcal{O}(MNC)$. Having computed $\mathcal{I}(x)$ for every $x \in \mathcal{V}$, we can form matrices $\mathbf{H}_{\mathbb{E}}$ and $\tilde{\mathbf{V}}_{\mathbb{E}}$ with costs $\mathcal{O}(L^4)$ and $\mathcal{O}(L^6 + KL^5)$, respectively. Thereafter, the SVD of $\mathbf{G}_{\mathbb{E}} = \mathbf{H}_{\mathbb{E}} + \frac{1}{Q}\tilde{\mathbf{V}}_{\mathbb{E}}$ for solving (4.36) requires $\mathcal{O}(L^6)$ computations. Observing that $L = 2M + N - 1$ and $K \leq MN + M$ and keeping only the dominant terms, the computational complexity of forming $\mathbf{E}_{\text{MMSE}}$ is $\mathcal{O}(L^6 + MNC)$. In addition, we note that the computation of $\mathcal{I}(x)$ for different values of x can be performed in parallel, tremendously reducing the computation time. In this work, we conduct numerical integration by means of the vectorized adaptive quadrature approach of [204] with absolute error tolerance $\epsilon = 1e - 10$. As a simple numerical example, for $(M, N) = (2, 3)$, $K = 7$, $\epsilon = 1e - 10$, and serial computation of the numerical integrals, in MATLAB 2019a (generic software) running on an Intel(R) core(TM) i7-8700 processor at 3.2 GHz and 32GB RAM (generic hardware), we computed $\mathbf{E}_{\text{MMSE}}$ in a very small fraction of a second.

Finally, a remark is in order on the source powers and noise variance. The proposed linear combiner $\mathbf{E}_{\text{MMSE}}$ depends on $\mathbf{H}_{\mathbb{E}}$ and $\tilde{\mathbf{V}}_{\mathbb{E}}$ which, in turn, depend on the powers $d_1, d_2, \ldots, d_K, \sigma^2$ associated to the source DoAs $\theta_1, \theta_2, \ldots, \theta_K$ and noise, respectively. In this Section, we assume the signal-source powers and noise variance to be known. Joint power/DoA estimation is beyond the scope of this Section. For the problem of the power estimation, we refer the interested reader to, e.g., the works in [209–214].

4.5.1 Numerical Studies

We consider coprime naturals (M, N) with $M < N$ and form a length-$(L = 2M + N - 1)$ coprime array which corresponds to a length-$(L' = MN + M)$ virtual ULA –i.e., the coarray. Signals from K sources impinge on the array with equal transmit power $d_1 = d_2 = ... = d_K = \alpha^2 = 10$dB. The noise variance is set to $\sigma^2 = 0$dB. Accordingly, the Signal-to-Noise-Ratio (SNR) is set to 10dB for every DoA signal source.

We commence our studies by computing the empirical Cumulative Distribution Function (CDF) of the Normalized-MSE (NMSE) in estimating $\mathbf{Z}$ for a given DoA collection $\Theta = \{\theta_1, \ldots, \theta_K\}$ such that the DoAs in Θ are i.i.d. –i.e., $\theta_k \sim \mathcal{D}(a, b) \forall k$. More specifically, we consider fixed sample-support $Q = 10$ and for each estimate $\hat{\mathbf{Z}} \in \{\hat{\mathbf{Z}}_{\text{sel}}, \hat{\mathbf{Z}}_{\text{avg}}, \hat{\mathbf{Z}}_{\text{MMSE}}\}$, we compute the estimation

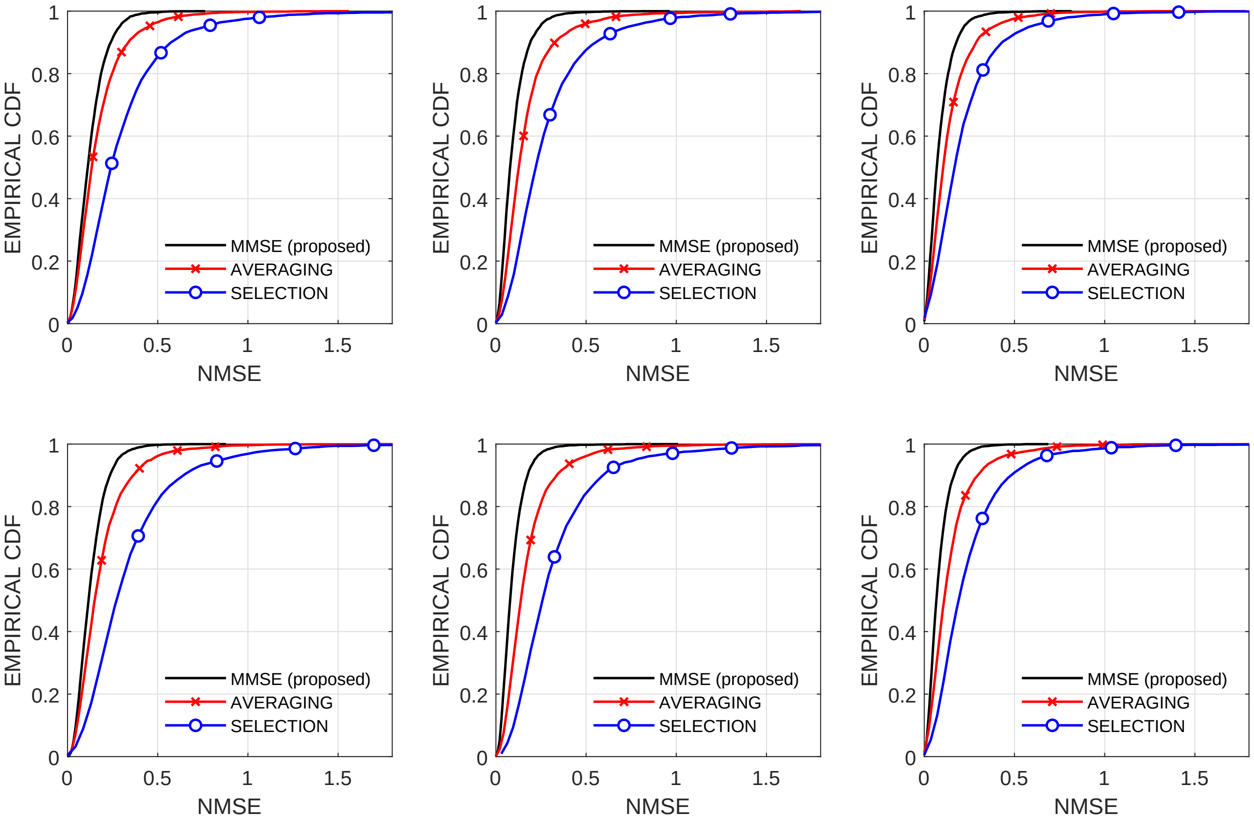


Figure 4.7. Empirical CDF of the MSE in estimating $\mathbf{Z}$ for $(M, N) = (2, 3)$, SNR= 10dB, $Q = 10$, $K = 5$ (top), and $K = 7$ (bottom). $\forall k, \theta_k \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ (left), $\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$ (center), $\mathcal{TN}(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)$ (right).

error

$$\text{NMSE} = \left\|\mathbf{Z} - \hat{\mathbf{Z}}\right\|_F^2 \|\mathbf{Z}\|_F^{-2}. \tag{4.39}$$

We repeat this process over 4000 statistically independent realizations of Θ and AWGN. We collect 4000 NMSE measurements based on which we plot, in Fig. 4.7, the empirical CDF of the NMSE in estimating $\mathbf{Z}$ for fixed $(M, N) = (2, 3)$, $K \in \{5, 7\}$, and

$$\mathcal{D}(a, b) \in \{\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2}), \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6}), \mathcal{T}N(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)\}. \tag{4.40}$$

We observe that the proposed MMSE combining approach attains superior MSE in estimating $\mathbf{Z}$ for any distribution and support set considered for the DoAs in Θ. Moreover, we notice that the averaging combining approach consistently outperforms the selection combining approach in accordance with our theoretical finding that averaging combining attains superior performance

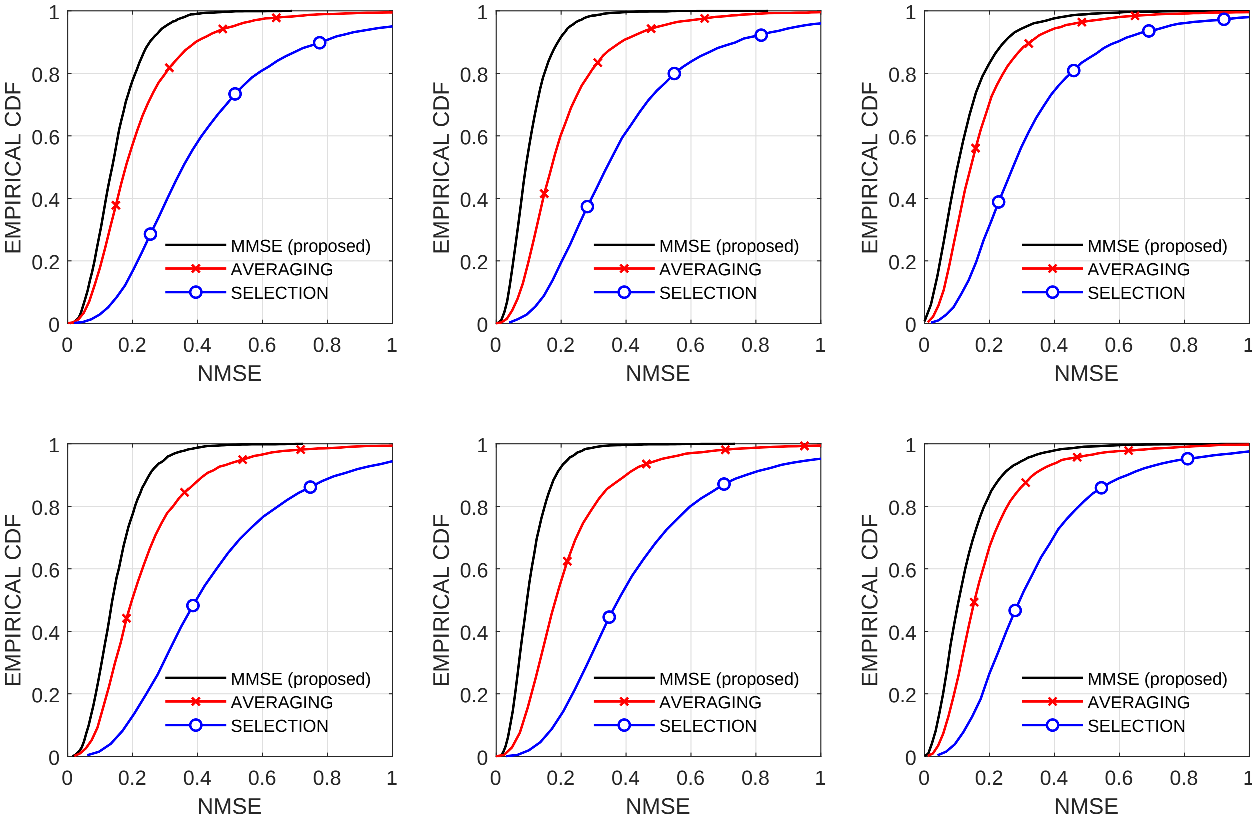


Figure 4.8. Empirical CDF of the NMSE in estimating $\mathbf{Z}$ for $(M, N) = (2, 5)$, SNR= 10dB, $Q = 10$, $K = 7$ (top), and $K = 9$ (bottom). $\forall k, \theta_k \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ (left), $\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$ (center), and $\mathcal{TN}(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)$ (right).

compared to selection combining. Furthermore, we observe that the performance advantage of the proposed MMSE combining approach over its standard counterparts is more emphatic when $\theta_k \sim \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$. Last, we notice that for $\theta_k \sim \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$, the performance gap between the proposed MMSE combining and the standard averaging approach is greater for $K = 7$. We repeat the last study for $(M, N) = (2, 5)$, $K \in \{7, 9\}$, and $\mathcal{D}(a, b) \in \{\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2}), \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6}), \mathcal{TN}(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)\}$. We illustrate the new CDFs in Fig. 4.8. Similar observations as in Fig. 4.7 are made. The proposed MMSE combining approach clearly outperforms its standard counterparts for any distribution assumption and support set for $\theta_k \forall k$.

For fixed $K = 7$ and every other parameter same as above, we plot the NMSE (averaged over 4000 realizations) versus sample support $Q \in \{1, 10, 100, 1000, 10000\}$, in Fig. 4.9. Consistent with the observations above, we notice that selection attains the highest NMSE while the proposed MMSE combiner attains, expectedly, the lowest NMSE in estimating $\mathbf{Z}$ across the board. The performance gap between the proposed MMSE combining estimate and the estimates based on

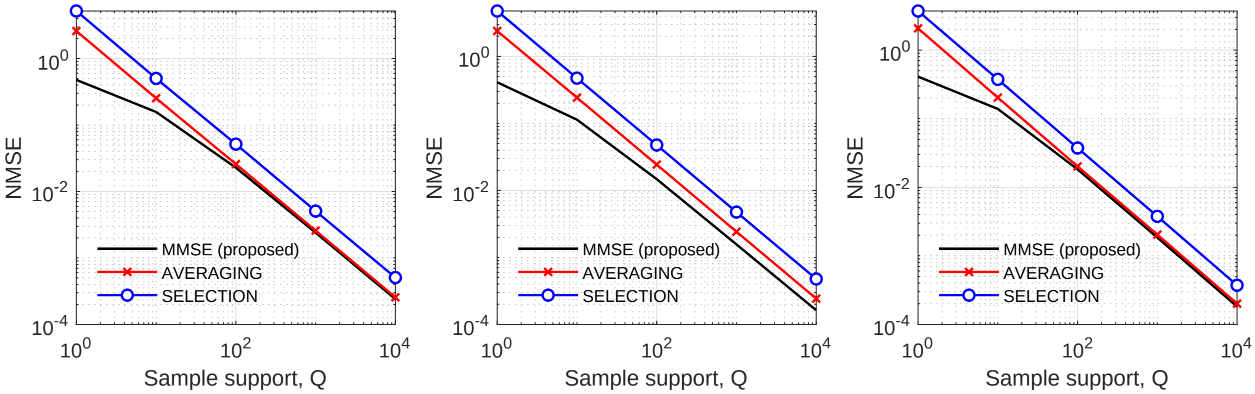


Figure 4.9. NMSE in estimating $\mathbf{Z}$, versus sample support, Q. $(M, N) = (2, 5)$, SNR= 10dB, and $K = 7$. $\forall k, \theta_k \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ (left), $\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$ (center), $\mathcal{TN}(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)$ (right).

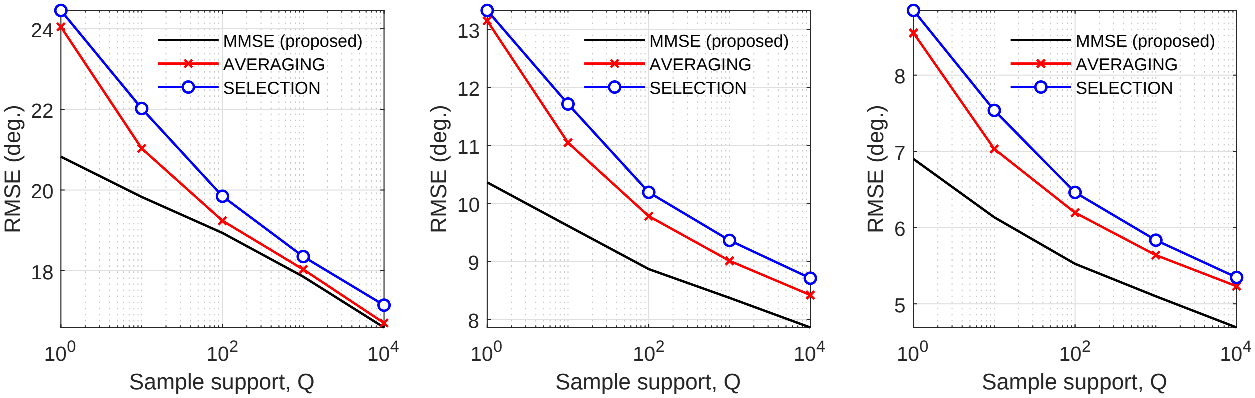


Figure 4.10. RMSE (degrees) in estimating the DoA set Θ, versus sample support, Q. $(M, N) = (2, 5)$, SNR= 10dB, and $K = 7$. $\forall k, \theta_k \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ (left), $\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$ (center), $\mathcal{TN}(-\frac{\pi}{8}, \frac{\pi}{8}, 0, 1)$ (right).

existing combining approaches decreases as the sample support Q increases. Nonetheless, it remains superior, in many cases, even for high values of Q –e.g., $Q = 10^4$. Moreover, in the first two subplots (uniform DoA distribution), we notice that the performance gap between the MMSE combining approach and the averaging approach is wider when the range of the support set (a, b) is narrower.

Next, we evaluate the performance of the proposed MMSE combining approach and its counterparts by measuring the Root-MSE (RMSE) in estimating the DoAs in Θ versus sample support Q. First, we evaluate the MUSIC spectrum for each estimate $\hat{\mathbf{Z}} \in \{\hat{\mathbf{Z}}_{\text{sel}}, \hat{\mathbf{Z}}_{\text{avg}}, \hat{\mathbf{Z}}_{\text{MMSE}}\}$ over the probability distribution support set (a, b) and obtain DoA estimates in $\hat{\Theta} = \{\hat{\theta}_1, \ldots, \hat{\theta}_K\}$. Then, we compute

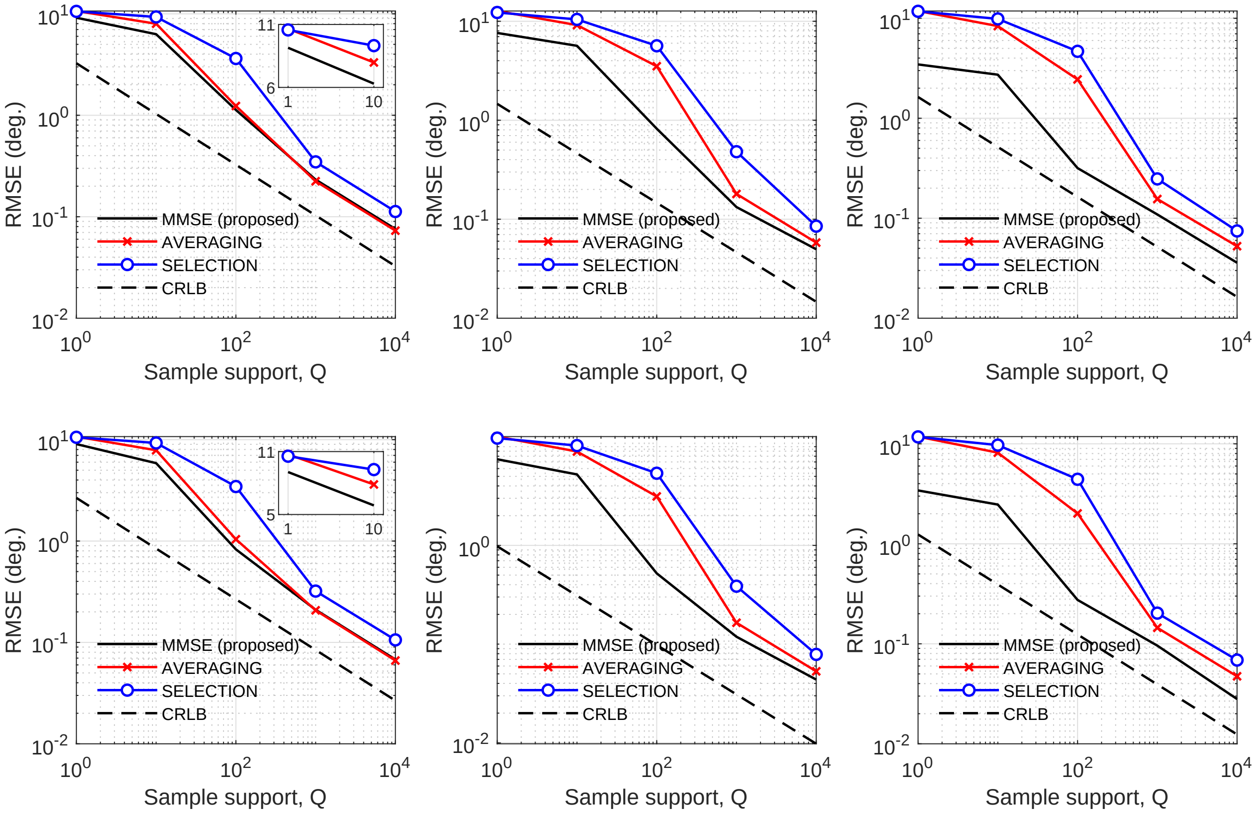


Figure 4.11. RMSE (degrees) in estimating the DoA set Θ, versus sample support, Q. $(M, N) = (3, 5)$ and $K = 11$. $(\Theta, \text{SNR}, \mathcal{D}(a, b)) = (\Theta_1, 0\text{dB}, \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2}))$ –top left, $(\Theta_1, 8\text{dB}, \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2}))$ –bottom left, $(\Theta_2, 0\text{dB}, \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{4}))$ –top center, $(\Theta_2, 8\text{dB}, \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{4}))$ –bottom center, $(\Theta_3, 0\text{dB}, \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6}))$ –top right, $(\Theta_3, 8\text{dB}, \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6}))$ –bottom right.

the estimation error

$$\text{MSE}(\hat{\Theta}) = \frac{1}{K} \sum_{k=1}^{K} (\theta_k - \hat{\theta}_k)^2. \tag{4.41}$$

We compute the RMSE by taking the square root of the $\text{MSE}(\hat{\Theta})$ computed over 4000 statistically independent realizations of Θ and AWGN. The resulting RMSE curves are depicted in Fig. 4.10. For every DoA distribution (even the most general case of uniform distribution in $(-\frac{\pi}{2}, \frac{\pi}{2})$) and every sample support (even as high as 10^4), the proposed method attains the lowest RMSE.

We conclude our studies by evaluating the DoA estimation performance of the proposed MMSE combiner and its counterparts for fixed DoAs. This time, we consider coprime naturals $(M, N) = (3, 5)$ forming an array with $L = 10$ sensors, fixed number of sources $K = 11$, and three distinct realizations of DoA set $\Theta = \{\theta_1, \ldots, \theta_K\}$:

$$\Theta_1 = \{-79, -55, -43, -25, -15, -6, 7, 21, 34, 54, 63\}^\circ, \tag{4.42}$$

$$\Theta_2 = \{-44, -38, -27, -12, -3, 6, 17, 24, 31, 37, 44\}^\circ, \tag{4.43}$$

$$\Theta_3 = \{-44, -37, -31, -23, -15, -8, -2, 4, 17, 22, 29\}^\circ. \tag{4.44}$$

For every fixed Θ_i, $i \in \{1, 2, 3\}$, we proceed as follows in order to compute the RMSE curves. First, we consider 4000 statistically independent realizations of AWGN for SNR $\in \{0, 8\}$dB. For every realization, we compute DoA estimate $\hat{\Theta}_i$ as previously described and evaluate the MUSIC spectrum over a probability distribution support set. We then measure MSE$(\hat{\Theta}_i)$ as described above and compute the RMSE by taking the square root of the sample-average of the 4000 MSE$(\hat{\Theta}_i)$ measurements. As a benchmark, we also compute the Cramér Rao Lower Bound (CRLB) curves computed by means of the closed-form expression that follows, first derived in [165], as[2]

$$\texttt{CRLB}_\Theta = \frac{1}{Q} \left(\mathbf{M}_\Theta^H \mathbf{\Pi} \mathbf{M}_\Theta \right)^{-1}, \tag{4.45}$$

where

$$\mathbf{M}_\Theta = (\mathbf{R}_y^\top \otimes \mathbf{R}_y)^{-\frac{1}{2}} \dot{\mathbf{S}}_d \mathrm{diag}(\mathbf{d}), \tag{4.46}$$

$$\mathbf{\Pi} = \mathbf{I}_{L^2} - \mathbf{M}_s (\mathbf{M}_s^H \mathbf{M}_s)^{-1} \mathbf{M}_s^H, \tag{4.47}$$

$$\mathbf{M}_s = (\mathbf{R}_y^\top \otimes \mathbf{R}_y)^{-\frac{1}{2}} [\mathbf{S}_d \ \mathrm{vec}(\mathbf{I}_L)], \tag{4.48}$$

$$\mathbf{S}_d = \mathbf{S}^* \odot \mathbf{S}, \tag{4.49}$$

$$\dot{\mathbf{S}}_d = \dot{\mathbf{S}}^* \odot \mathbf{S} + \mathbf{S}^* \odot \dot{\mathbf{S}}, \tag{4.50}$$

$$\dot{\mathbf{S}} = [\frac{\partial \mathbf{s}(\theta_1)}{\partial \theta_1}, \ldots, \frac{\partial \mathbf{s}(\theta_K)}{\partial \theta_K}]. \tag{4.51}$$

In Fig. 4.11 (left), we plot the RMSE and CRLB curves versus sample support for $\Theta = \Theta_1$, and SNR= 0dB (top-left) and SNR= 8dB (bottom-left), respectively. We form the MMSE combiner by considering $\mathcal{D}(a, b) = \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$. We observe that for SNR $\in \{0, 8\}$dB and low sample support, $Q \leq 100$, the proposed MMSE combiner clearly outperforms its counterparts in DoA estimation accuracy. For $Q \geq 1000$, the proposed MMSE and the averaging combiners attain almost identical performance, superior to that of the selection combiner.

In Fig. 4.11 (center), we plot the RMSE and CRLB curves versus sample support for $\Theta = \Theta_2$, and SNR= 0dB (top-center) and SNR= 8dB (bottom-center), respectively. We consider $\mathcal{D}(a, b) =$

[2] Operator '$\odot$' denotes the column-wise Kronecker product of matrices, also known as Khatri-Rao product.

$\mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{4})$. In this case, we observe that the proposed MMSE combiner outperforms its counterparts across the board with respect to both sample support and SNR.

Finally, we consider $\mathcal{D}(a, b) = \mathcal{U}(-\frac{\pi}{4}, \frac{\pi}{6})$ and in Fig. 4.11 (right), we plot the RMSE and CRLB curves versus sample support for $\Theta = \Theta_3$, and SNR= 0dB (top-right) and SNR= 8dB (bottom-right), respectively. In accordance with earlier observations, now that the probability distribution support set is even narrower, the proposed MMSE combiner attains an even more superior DoA estimation performance compared to its counterparts.

4.6 Contribution 3: Structured Coarray Autocorrelation Matrix Estimation

We propose an algorithm which iteratively solves a sequence of optimization problems returning, upon convergence, an improved coarray autocorrelation matrix estimate. Motivated by [155], where it was formally proven that averaging autocorrelation sampling attains superior autocorrelation estimates compared to selection sampling with respect to the Mean-Squared-Error (MSE) metric, we propose to initialize the proposed algorithm to $\mathbf{P}_0 = \sqrt{\hat{\mathbf{Z}}_{\text{avg}} \hat{\mathbf{Z}}_{\text{avg}}^H}$. At iteration $i \geq 0$, the proposed algorithm computes

$$\mathbf{Q}_i = \Upsilon(\mathbf{P}_i), \tag{4.52}$$

$$\mathbf{R}_i = \Psi(\mathbf{Q}_i), \tag{4.53}$$

$$\mathbf{P}_{i+1} = \Omega(\mathbf{R}_i), \tag{4.54}$$

where for any $\mathbf{X} = \mathbf{X}^H \in \mathbb{C}^{D \times D}$ with Eigenvalue Decomposition (EVD) $\mathbf{X} \overset{\text{evd}}{=} \mathbf{U}\text{diag}(\boldsymbol{\lambda}_{\boldsymbol{X}})\mathbf{U}^H$ the following hold.

Definition 4.1. $\Upsilon(\mathbf{X})$ *returns the nearest Toeplitz matrix, in the Euclidean norm sense,*[3] *to* $\mathbf{X}$*:* $\Upsilon(\mathbf{X}) \overset{\triangle}{=} \underset{\mathbf{T}_0 \in \mathcal{T}^D}{\text{argmin}} \|\mathbf{X} - \mathbf{T}_0\|_F^2$, *where* $\mathcal{T}^D \overset{\triangle}{=} \{\mathbf{T} \in \mathbb{C}^{D \times D} \mid \mathbf{T} \text{ is Toeplitz}\}$.

Definition 4.2. $\Psi(\mathbf{X})$ *returns the nearest Positive Semidefinite (PSD) matrix to* $\mathbf{X}$*:* $\Psi(\mathbf{X}) \overset{\triangle}{=} \underset{\mathbf{X}_0 \in \mathbb{S}_+^D}{\text{argmin}} \|\mathbf{X} - \mathbf{X}_0\|_F^2$, *where* $\mathbb{S}_+^D \overset{\triangle}{=} \{\mathbf{A} \in \mathbb{C}^{D \times D} \mid \mathbf{A} = \mathbf{A}^H \succeq \mathbf{0}\}$.

Definition 4.3. $\Omega(\mathbf{X})$ *performs an eigenvalue-correction operation to the* $D - \rho$ *smallest eigen-*

[3] Otherwise known as Frobenius norm: $\|\cdot\|_F^2$ returns the sum of the squared entries of its argument.

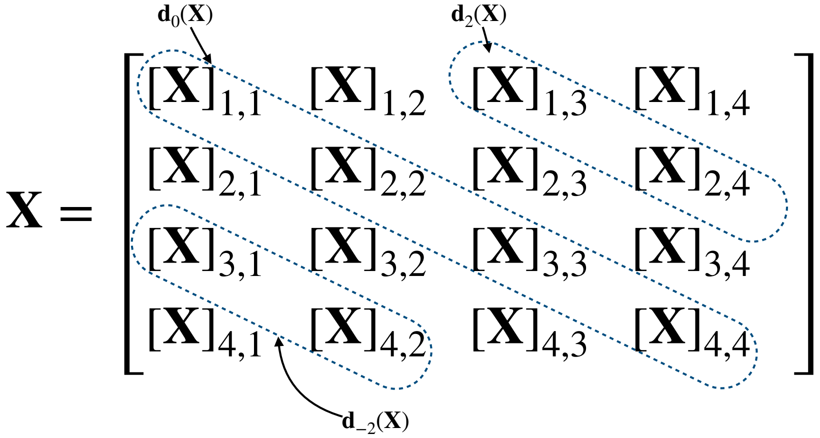

$\mathbf{d}_0(\mathbf{X})$
$\mathbf{d}_2(\mathbf{X})$
$\mathbf{X} =$
$[\mathbf{X}]_{1,1}$ $[\mathbf{X}]_{1,2}$ $[\mathbf{X}]_{1,3}$ $[\mathbf{X}]_{1,4}$
$[\mathbf{X}]_{2,1}$ $[\mathbf{X}]_{2,2}$ $[\mathbf{X}]_{2,3}$ $[\mathbf{X}]_{2,4}$
$[\mathbf{X}]_{3,1}$ $[\mathbf{X}]_{3,2}$ $[\mathbf{X}]_{3,3}$ $[\mathbf{X}]_{3,4}$
$[\mathbf{X}]_{4,1}$ $[\mathbf{X}]_{4,2}$ $[\mathbf{X}]_{4,3}$ $[\mathbf{X}]_{4,4}$
$\mathbf{d}_{-2}(\mathbf{X})$

Figure 4.12. Illustration of the ith diagonal of $\mathbf{X} \in \mathbb{C}^{4\times 4}$, $\mathbf{d}_i(\mathbf{X})$, $i \in \{0, \pm 2\}$.

values of $\mathbf{X}$. *For some general* $\rho \in \{1, \ldots, D-1\}$, $\Omega(\mathbf{X}) \overset{\triangle}{=} \mathbf{U}\text{diag}(\overline{\boldsymbol{\lambda}}_X)\mathbf{U}^H$, *where*

$$[\overline{\boldsymbol{\lambda}}_X]_i = \begin{cases} [\boldsymbol{\lambda}_{\boldsymbol{X}}]_i, & i \leq D-\rho+1, \\ \frac{1}{D-\rho}\sum\limits_{j=\rho+1}^{D} [\boldsymbol{\lambda}_{\boldsymbol{X}}]_j, & i > D-\rho+1. \end{cases} \tag{4.55}$$

In view of the above, the proposed algorithm seeks to optimize the $D-\rho$ smallest eigenvalues of the autocorrelation matrix estimate at which it is initialized while preserving the PSD, Hermitian, and Toeplitz structure. Next, we conduct formal convergence analysis of the proposed algorithm. Consider arbitrary $\mathbf{X} = \mathbf{X}^H \in \mathbb{C}^{D\times D}$ and let $\mathbf{d}_i(\mathbf{X})$ denote a diagonal of $\mathbf{X}$ (see Fig. 4.12) such that

$$[\mathbf{d}_i(\mathbf{X})]_j = \begin{cases} [\mathbf{X}]_{j-i,j}, & i \leq 0, \\ [\mathbf{X}]_{j,j+i}, & i > 0, \end{cases} \tag{4.56}$$

for any $j \in \{1, 2, \ldots, D - |i|\}$. The following remarks hold.

Remark 4.1. *It holds that* $\mathbf{d}_i(\mathbf{X}) = \mathbf{d}^*_{i'}(\mathbf{X})$, *if* $|i| = |i'|$.

Remark 4.2. *Let* $\mathbf{T} \overset{\triangle}{=} \Upsilon(\mathbf{X}) = \min_{\mathbf{T}_0 \in \mathcal{T}^D} \|\mathbf{X} - \mathbf{T}_0\|_F^2$. *For any* $i \in \{1-D, \ldots, D-1\}$, *it holds that* $\mathbf{d}_i(\mathbf{T}) = \frac{1}{D-|i|}\mathbf{1}_{D-|i|}^\top \mathbf{d}_i(\mathbf{X})\mathbf{1}_{D-|i|}$ *where, for any integer* $x \geq 1$, $\mathbf{1}_x = [1, 1, \ldots, 1]^\top \in \mathbb{R}^x$.

Remark 4.2 illustrates that the nearest Toeplitz matrix to $\mathbf{X}$ can be found in closed-form.

Remark 4.3. *Let* $\mathbf{T} \overset{\triangle}{=} \Upsilon(\mathbf{X})$. *It holds that* $\|\mathbf{T}\|_F^2 \leq \|\mathbf{X}\|_F^2$.

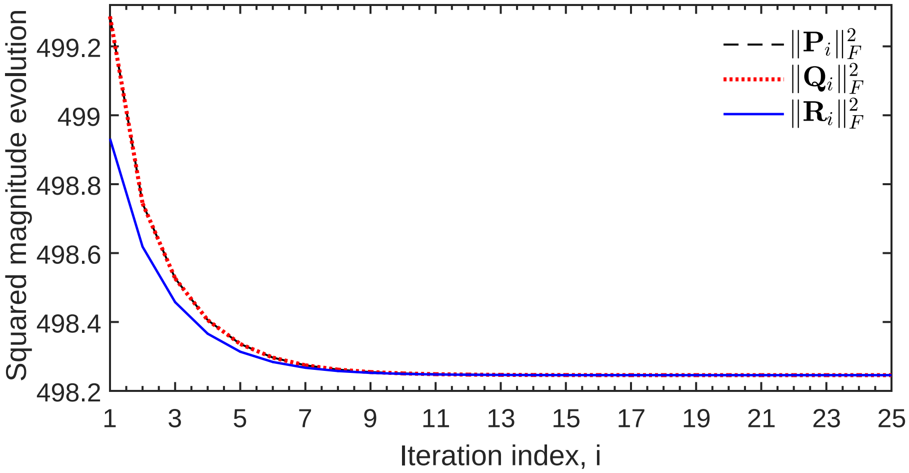


Figure 4.13. Illustration of Proposition 4.3. $(M, N) = (2, 3)$, $\Theta = \{-43°, -21°, -10°, 17°, 29°, 54°\}$, $d_k = 0\text{dB}\ \forall k$, $\sigma^2 = 1$, $Q = 50$.

By Remark 4.3, $\mathbf{T}$ is the nearest Toeplitz matrix to $\mathbf{X}$ and exhibits squared Frobenius norm lower or equal than that of $\mathbf{X}$.

Remark 4.4. *Let $\mathbf{T} \overset{\triangle}{=} \Upsilon(\mathbf{X})$ admit EVD*[4] $\mathbf{T} = \mathbf{U}\text{diag}(\boldsymbol{\lambda}_T)\mathbf{U}^H$.[5] *It holds that*

$$\|\mathbf{T}\|_F^2 = \text{Tr}(\mathbf{U}\text{diag}(\boldsymbol{\lambda}_T)\text{diag}(\boldsymbol{\lambda}_T)\mathbf{U}^H) = \|\boldsymbol{\lambda}_T\|_2^2. \tag{4.57}$$

Remark 4.5. *Let $\mathbf{T} \overset{\triangle}{=} \Upsilon(\mathbf{X})$ admit EVD $\mathbf{T} = \mathbf{U}\text{diag}(\boldsymbol{\lambda}_T)\mathbf{U}^H$ and define $\mathbf{P}$ such that $\mathbf{P} = \mathbf{U}\text{diag}(\boldsymbol{\lambda}_P)\mathbf{U}^H$, where*[6] $\boldsymbol{\lambda}_P = \boldsymbol{\lambda}_T^+$*–i.e., $\forall i \in [D]$, $[\boldsymbol{\lambda}_P]_i = \max\{[\boldsymbol{\lambda}_T]_i, 0\}$. It holds that $\mathbf{P}$ is the solution to $\min_{\mathbf{P}_0 \in \mathbb{S}_+^D} \|\mathbf{T} - \mathbf{P}_0\|_F^2$. A proof for Remark 4.5 was first offered for real matrices in [215]. For completeness purposes, we offer an analogous proof for complex-valued matrices in the Appendix.*

By Remark 4.5, the nearest PSD matrix to $\mathbf{T}$ can be found in closed-form.

[4] A Hermitian matrix $\mathbf{A}$ can be expressed as $\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H$, where $\boldsymbol{\Lambda}$ is an upper diagonal with the eigenvalues of $\mathbf{A}$ in its main diagonal. If $\mathbf{A}$ is normal (i.e., $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}$), then $\boldsymbol{\Lambda}$ is diagonal. Every Hermitian matrix is normal.

[5] For any normal matrix $\mathbf{W}$, we denote its eigenvalues by vector $\boldsymbol{\lambda}_W$. For instance, $\boldsymbol{\lambda}_T$ denotes the eigenvalues of $\mathbf{T}$ while $\boldsymbol{\lambda}_P$ denotes the eigenvalues of $\mathbf{P}$.

[6] $\boldsymbol{\lambda}_T^+$ returns a vector with length equal to that of $\boldsymbol{\lambda}_T$ with the positive entries of $\boldsymbol{\lambda}_T$ intact and its negative entries set to zero.

```
Input: Coarray autocorrelation matrix estimate $\widehat{\mathbf{R}}$
Output: $\mathbf{R} \leftarrow \mathbf{P}_i$
$\mathbf{P}_0 \leftarrow \widehat{\mathbf{R}}$          % Initialization
Until convergence/termination
    $\mathbf{Q}_i \leftarrow \Upsilon(\mathbf{P}_i)$    % Nearest Toeplitz to $\mathbf{P}_i$
    $\mathbf{R}_i \leftarrow \Psi(\mathbf{Q}_i)$    % Nearest PSD to $\mathbf{Q}_i$
    $\mathbf{P}_{i+1} \leftarrow \Omega(\mathbf{R}_i)$  % Eigenvalue-correction
```

Algorithm 4.1 Structured coarray autocorrelation matrix estimation

Remark 4.6. *For $\mathbf{P} = \Psi(\mathbf{T})$, it holds that $\|\mathbf{P}\|_F^2 \leq \|\mathbf{T}\|_F^2$. Formally,*

$$\|\mathbf{P}\|_F^2{=}\|\boldsymbol{\lambda}_P\|_2^2 = \sum_{i=1}^{D}[\boldsymbol{\lambda}_P]_i^2 \leq \sum_{i=1}^{D}[\boldsymbol{\lambda}_T]_i^2 = \|\mathbf{T}\|_F^2 \tag{4.58}$$

By Remark 4.6, $\mathbf{P}$ is the nearest PSD matrix to $\mathbf{T}$ and attains squared Frobenius norm lower or equal than that of $\mathbf{T}$.

Remark 4.7. *Let $\mathbf{A} = \Omega(\mathbf{P}) = \mathbf{U}\text{diag}(\boldsymbol{\lambda}_A)\mathbf{U}^H$, where*

$$[\boldsymbol{\lambda}_A]_i = \begin{cases} [\boldsymbol{\lambda}_P]_i, & i \leq D - \rho + 1, \\ \frac{1}{D-\rho}\sum\limits_{j=\rho+1}^{D}[\boldsymbol{\lambda}_P]_j, & i > D - \rho + 1, \end{cases} \tag{4.59}$$

for $\rho \in \{1, \ldots, D-1\}$. It holds that $\|\mathbf{A}\|_F^2 \leq \|\mathbf{P}\|_F^2$.

By Remark 4.7, the eigenvalue correction operation on a normal matrix returns a matrix with squared-Frobenius norm lower or equal than that of its input argument. In view of Remarks 4.1-4.7, the following Proposition derives.

Proposition 4.3. *For $\mathbf{Q}_i, \mathbf{R}_i$, and $\mathbf{P}_{i+1}$ in* (4.52)-(4.54)*, it holds that*

$$\|\mathbf{Q}_i\|_F^2 \geq \|\mathbf{R}_i\|_F^2 \geq \|\mathbf{P}_{i+1}\|_F^2 \geq \|\mathbf{Q}_{i+1}\|_F^2 \geq \ldots \geq 0 \forall i \geq 0. \tag{4.60}$$

Proposition 4.3 states that the proposed algorithm is guaranteed to converge. In practice, one can terminate the iterations when $\|\mathbf{P}_{i+1}{-}\mathbf{P}_i\| \leq \epsilon$, for some $\epsilon \geq 0$. For sufficiently small ϵ, Proposition 1

Matrix estimate	Autocorrelation sampling approach	Positive Definite	Toeplitz	Hermitian	Equal noise-subspace eigenvalues
$\hat{\mathbf{Z}}_{\text{sel}}$	Selection	✗	✓	✓	✗
$\hat{\mathbf{Z}}_{\text{avg}}$	Averaging	✗	✓	✓	✗
$\widehat{\mathbf{Z}}_{\text{psr}}$	Averaging	✓	✗	✓	✗
Structured (proposed)	Averaging	✓	✓	✓	✓

Table 4.4: Comparison of coarray autocorrelation matrix estimates: autocorrelation sampling approach and structure properties.

implies that, at convergence, $\mathbf{P}_{i+1} = \mathbf{R}_i = \mathbf{Q}_i$ which, in turn, implies that the algorithm converged to a PD, Hermitian, and Toeplitz matrix the noise-subspace eigenvalues of which are equal. A visual illustration of Proposition 4.3 and a pseudocode of the proposed algorithm are offered in Fig. 4.13 and Algorithm 4.1, respectively. Importantly, $\forall i \geq 0$, the Algorithm of Fig. 4.14 computes $\mathbf{Q}_i, \mathbf{R}_i, \mathbf{P}_{i+1}$ by closed-form expressions with cost at most the cost of EVD–i.e., $\mathcal{O}(D^3)$. Overall, the cost of the proposed algorithm is $\mathcal{O}(TD^3)$ where T is the number of iterations required for convergence.

We summarize the above estimates in Table 4.4, where for each estimate we mention the employed autocorrelation sampling approach. Moreover, for each structure property guaranteed to be satisfied, we place a '✓', otherwise, we place a '✗'. Given a coarray autocorrelation matrix estimate $\widehat{\mathbf{R}} \in \{\hat{\mathbf{Z}}_{\text{sel}}, \hat{\mathbf{Z}}_{\text{avg}}, \hat{\mathbf{Z}}_{\text{psr}}\}$, a standard DoA estimation approach –e.g., MUltiple SIgnal Classification (MUSIC) –is applied for identifying the DoAs in Θ.

4.6.1 Numerical Studies

We consider coprime naturals $(M, N) = (3, 5)$ and form coprime array with $L = 10$ elements. Source-signals impinge on the array from $K = 13$ DoAs $\{\theta_k\}_{k=1}^{13}$, $\theta_k = (-75 + (k-1)12)^\circ$. The noise variance is set to $\sigma^2 = 0$dB. All sources emit signals with equal power $d_k = \alpha^2$dB. Accordingly, the Signal-to-Noise Ratio SNR $= \alpha^2$. The receiver collects $Q \in \{150, 300, 450, 600\}$ received-signal snapshots. For every Q, we consider $\Gamma = 3000$ statistically independent realizations of noise; i.e., $\{\mathbf{y}_{r,1}, \ldots, \mathbf{y}_{r,Q}\}_{r=1}^{\Gamma}$. At every realization r, we compute coarray autocorrelation matrix estimates corresponding to the augmented matrix approach (AM), principal square root of the spatial smoothed matrix (PSR), nearest Hermitian, PSD, and Toeplitz (H-PSD-T) approach of [180][7], and the proposed structured estimate. We take a moment and discuss similarities and

[7] H-PSD-T seeks a Hermitian-PSD-Toeplitz matrix which fills the gaps of the coarray. When the uniform segment of the coarray is considered, H-PSD-T returns $\underset{\mathbf{R} \in \mathbb{S}_{+}^{L'}}{\operatorname{argmin}} \|\mathbf{R} - \hat{\mathbf{Z}}_{\text{avg}}\|_F^2 + \mu \|\mathbf{R}\|_*$, where $\mu\|\mathbf{R}\|_*$ is a regularization term that moderates overfitting.

differences between H-PSD-T and the proposed framework: (1) H-PSD-T is guaranteed to satisfy the Toeplitz, Hermitian, and Positive Semidefinite structure properties while the proposed estimate is guaranteed to satisfy all four, (2) H-PSD-T requires tuning of an ad-hoc parameter that moderates overfitting while the proposed framework is ad-hoc parameter free, and (3) H-PSD-T computes an estimate by iterative solvers for convex optimization problems while the proposed approach enjoys closed-form solutions for each individual optimization problem in its sequence. For every method and estimate $\widehat{\mathbf{R}}_r$ at realization r, we compute the Normalized Squared Error

$$\mathrm{NSE}(\widehat{\mathbf{R}}_r) = \|\widehat{\mathbf{R}}_r - \mathbf{Z}\|_F^2 \|\mathbf{Z}\|_F^{-2}. \tag{4.61}$$

Then, we compute the Root Mean Normalized Squared Error

$$\mathrm{RMNSE} = \sqrt{\frac{1}{\Gamma}\sum_{r=1}^{\Gamma} \mathrm{NSE}(\widehat{\mathbf{R}}_r)}. \tag{4.62}$$

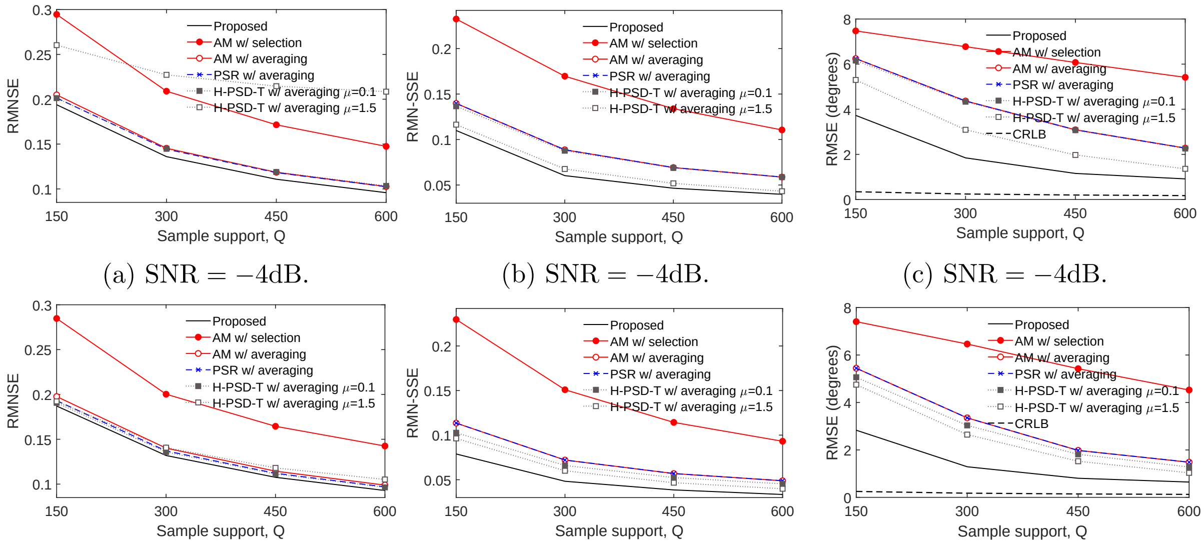
In Fig. 4.15a and Fig. 4.15d, we plot the RMNSE versus sample support, Q, for SNR $= -4$dB and SNR $= 2$dB, respectively. Expectedly, we observe that all methods employing averaging-sampling perform similarly well. The proposed estimate attains superior estimation performance across the board. Moreover, we notice the sensitivity of H-PSD-T with respect to the ad-hoc parameter μ; e.g., for SNR $= -4$dB, H-PSD-T with $\mu = 1.5$ exhibits low performance while for SNR $= 2$dB it exhibits high estimation performance. Thereafter, we consider that the nominal coarray autocorrelation matrix admits SVD $\mathbf{Z} = \mathbf{Q}_{\mathrm{co}}\mathbf{\Sigma}_{\mathrm{co}}\mathbf{V}_{\mathrm{co}}^H + \sigma^2\bar{\mathbf{Q}}_{\mathrm{co}}\bar{\mathbf{Q}}_{\mathrm{co}}^H$, where $\mathbf{Q}_{\mathrm{co}}$ and $\bar{\mathbf{Q}}_{\mathrm{co}}$ correspond to the signal and noise subspace bases, respectively. Similarly, every coarray autocorrelation matrix estimate $\widehat{\mathbf{R}}_r$ admits SVD $\widehat{\mathbf{R}}_r = \mathbf{Q}_r\mathbf{\Sigma}_r\mathbf{V}_r^H + \bar{\mathbf{Q}}_r\bar{\mathbf{\Sigma}}_r\bar{\mathbf{V}}_r^H$, where $\mathbf{Q}_r$ denotes the signal-subspace-basis of the K dominant left-hand singular valued singular vectors of $\widehat{\mathbf{R}}_r$. At each realization and for every value of Q, we compute the Normalized Squared Subspace Error

$$\mathrm{NSSE}(\widehat{\mathbf{Q}}_r) = \|\widehat{\mathbf{Q}}_r\widehat{\mathbf{Q}}_r^H - \mathbf{Q}_{\mathrm{co}}\mathbf{Q}_{\mathrm{co}}^H\|_F^2 (2K)^{-1}. \tag{4.63}$$

Then, we compute the Root Mean Normalized Squared Subspace Error

$$\mathrm{RMN\text{-}SSE} = \sqrt{\frac{1}{\Gamma}\sum_{r=1}^{\Gamma} \mathrm{NSSE}(\widehat{\mathbf{Q}}_r)}. \tag{4.64}$$

In Fig. 4.15b and Fig. 4.15e, we plot the RMN-SSE versus sample support for SNR $= -4$dB and 2dB, respectively. We notice the influence of the ad-hoc parameter μ with respect to H-PSD-T and observe that the proposed structured estimate clearly outperforms all counterparts across the



(a) SNR $= -4$dB.

(b) SNR $= -4$dB.

(c) SNR $= -4$dB.

(d) SNR $= 2$dB.

(e) SNR $= 2$dB.

(f) SNR $= 2$dB.

Figure 4.15. Root Mean Normalized Squared Error (RMNSE) with respect to (w.r.t.) $\mathbf{R}_{\text{co}}$, Root Mean Normalized - Subspace Squared Error (RMN-SSE) w.r.t. $\mathbf{Q}_{\text{co}}$, and Root Mean Squared Error (RMSE) w.r.t. Θ vs sample support for varying SNR $\in \{-4, 2\}$dB.

board in subspace estimation performance.

Next, for every value of sample support and realization r, we conduct DoA estimation by applying MUSIC on the estimate $\widehat{\mathbf{R}}_r$ which returns estimates $\{\widehat{\theta}_{k,r}\}_{k=1}^{K}$ [155]. Then, we measure the Root Mean Squared Error

$$\text{RMSE} = \sqrt{\frac{1}{\Gamma}\frac{1}{K}\sum_{r=1}^{\Gamma}\sum_{k=1}^{K}(\theta_k - \widehat{\theta}_{k,r})^2} \tag{4.65}$$

and illustrate the corresponding RMSE curves versus sample support Q, in Fig. 4.15c and Fig. 4.15f, for SNR $= -4$dB and 2dB, respectively. We include the Cramér Rao Lower Bound (CRLB) curves as benchmarks [165]. We notice that the the performances of standard counterparts (AM, PSR) deviate significantly from the CRLB. In contrast, the proposed coarray autocorrelation matrix estimate outperforms all counterparts by at least 0.3° and at most 2°. In addition, as Q increases, its performance curves approach the CRLB curves.

4.7 Conclusions

In this Chapter, we developed theory and novel algorithms for robust coprime array processing. Specifically: (i) We derived closed-form MSE expressions for the estimation error attained by selection and averaging autocorrelation combining and formally established the superiority of the latter. The derived expressions are validated by means of numerical simulations. (ii) We proposed a novel coprime array receiver that attains minimum MSE in coarray autocorrelation estimation, for any probability distribution of the source DoAs. Extensive numerical studies on various DoA distributions demonstrate that the proposed MMSE combining method consistently outperforms its existing counterparts in autocorrelation estimation performance with respect to the MSE metric. In turn, the proposed MMSE combiner enables lower RMSE in DoA estimation. (iii) We proposed an optimization framework which computes a structured coarray autocorrelation matrix estimate. The proposed algorithm is accompanied by convergence analysis and is guaranteed to return a coarray autocorrelation matrix estimate satisfying all structure properties of the true autocorrelation matrix. Numerical studies illustrate the enhanced performance of the proposed estimate compared to standard counterparts, both in autocorrelation matrix estimation error and DoA estimation.

Bibliography

[1] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966, publisher: Springer.

[2] M. B. Amin, W. Zirwas, and M. Haardt, "HOSVD-Based Denoising for Improved Channel Prediction of Weak Massive MIMO Channels," in *IEEE Veh. Technol. Conf.*, Sydney, Australia, Jun. 2017, pp. 1–5.

[3] D. C. Arajo, A. L. F. d. Almeida, J. P. C. L. D. Costa, and R. T. d. Sousa, "Tensor-Based Channel Estimation for Massive MIMO-OFDM Systems," *IEEE Access*, vol. 7, pp. 42 133–42 147, 2019.

[4] W. D. C. Freitas, G. Favier, and A. L. De Almeida, "Tensor-Based Joint Channel and Symbol Estimation for Two-Way MIMO Relaying Systems," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 227–231, 2019.

[5] P. R. Gomes, A. L. de Almeida, and J. P. C. da Costal, "Fourth-order tensor method for blind spatial signature estimation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2014, pp. 2992–2996.

[6] H. Q. Ngo and E. G. Larsson, "EVD-based channel estimation in multicell multiuser MIMO systems with very large antenna arrays," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3249–3252.

[7] P. Comon, "Tensors: a brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, 2014.

[8] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Technol.*, vol. 8, pp. 16:1–16:44, Jan. 2017.

[9] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, “Tensorly: Tensor learning in python,” *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 925–930, 2019.

[10] G. G. Calvi, A. Moniri, M. Mahfouz, Z. Yu, Q. Zhao, and D. P. Mandic, “Tucker Tensor Layer in Fully Connected Neural Networks,” *arXiv preprint arXiv:1903.06133*, 2019.

[11] J. Gusak, M. Kholyavchenko, E. Ponomarev, L. Markeeva, I. Oseledets, and A. Cichocki, “One time is not enough: iterative tensor decomposition for neural network compression,” *arXiv preprint arXiv:1903.09973*, 2019.

[12] N. D. Sidiropoulos, D. L. L, F. X, K. Huang, P. E. E, and F. C, “Tensor decomposition for signal processing and machine learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[13] D. Tao, X. Li, W. Hu, S. Maybank, and X. Wu, “Supervised tensor learning,” in *IEEE Int. Conf. Data Mining*, Houston, TX, Nov. 2005, pp. 8–pp.

[14] J. Kossaifi, Z. C. Lipton, A. Kolbeinsson, A. Khanna, T. Furlanello, and A. Anandkumar, “Tensor regression networks,” *J. Mach. Learn. Res.*, vol. 21, no. 123, pp. 1–21, 2020.

[15] A. Cichocki, D. Mandic, L. D. Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor Decompositions for Signal Processing Applications: From two-way to multiway component analysis,” *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.

[16] M. A. O. Vasilescu and D. Terzopoulos, “Multilinear analysis of image ensembles: Tensorfaces,” in *European conference on computer vision.* Springer, 2002, pp. 447–460.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1253–1278, 2000.

[18] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, “Two-dimensional PCA: a new approach to appearance-based face representation and recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, 2004.

[19] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, “TTHRESH: Tensor Compression for Multidimensional Visual Data,” *IEEE Trans. Vis. Comput. Graphics*, pp. 2891–2903, 2019.

[20] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, “Mutan: Multimodal tucker fusion for visual question answering,” in *Proc. IEEE Int. Conf. Comput. Vision*, Venice, Italy, Oct. 2017, pp. 2612–2620.

[21] M. Mrup, L. K. Hansen, and S. M. Arnfred, "Algorithms for sparse nonnegative Tucker decompositions," *Neural computation*, vol. 20, no. 8, pp. 2112–2131, 2008, publisher: MIT Press.

[22] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, and W. Qian, "Multivis: Content-based social network exploration through multi-way visual analysis," in *Proceedings of the 2009 SIAM International Conference on Data Mining.* SIAM, 2009, pp. 1064–1075.

[23] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *2008 Eighth IEEE international conference on data mining.* IEEE, 2008, pp. 363–372.

[24] C. M. Bishop, *Pattern recognition and machine learning.* Springer, 2006.

[25] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification.* John Wiley & Sons, 2012.

[26] A. Koochakzadeh and P. Pal, "On canonical polyadic decomposition of overcomplete tensors of arbitrary even order," in *Proc. IEEE Int. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, Curaao, Dutch Antilles, Dec. 2017, pp. 1–5.

[27] A. Prater, "Classification via tensor decompositions of echo state networks," in *Proc. IEEE Symp. Series Comput. Intell.*, Honolulu, HI, Dec. 2017, pp. 1–8.

[28] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, pp. 167–191, Nov. 2005.

[29] B. N. Sheehan and Y. Saad, "Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM," in *Proc. SIAM Int. Conf. Data Mining*, Minneapolis, MN, Apr. 2007, pp. 355–365.

[30] H. Kong, L. Wang, E. K. Teoh, X. Li, J.-G. Wang, and R. Venkateswarlu, "Generalized 2D principal component analysis for face image representation and recognition," *Neural Netw.*, vol. 18, no. 5-6, pp. 585–594, 2005.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.* Baltimore, MD: Johns Hopkins University Press, 1996.

[32] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r1,r2,...,rn) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

[33] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3198–3213, 2008.

[34] E. R. Balda, S. A. Cheema, J. Steinwandt, M. Haardt, A. Weiss, and A. Yeredor, "First-order perturbation analysis of low-rank tensor approximations based on the truncated HOSVD," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Nov. 2016, pp. 1723–1727.

[35] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," *SIAM J. Sci. Comput.*, vol. 34, no. 2, pp. A1027–A1052, 2012.

[36] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 4, pp. 2029–2054, 2010.

[37] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, "Efficient nonnegative tucker decompositions: Algorithms and uniqueness," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4990–5003, 2015.

[38] J. E. Cohen, P. Comon, and N. Gillis, "Some theory on Non-negative Tucker Decomposition," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, Grenoble, France, Feb. 2017, pp. 152–161.

[39] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: Theory and applications," *ACM Trans. Knowl. Discovery Data*, vol. 2, no. 3, pp. 1–37, 2008.

[40] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *ACM Int. Conf. Knowl. Discovery Data mining*, Philadelphia, PA, Aug. 2006, pp. 374–383.

[41] R. Yu, D. Cheng, and Y. Liu, "Accelerated online low rank tensor learning for multivariate spatiotemporal streams," in *Int. Conf. Machine Learn.*, Lille, France, Jul. 2015, pp. 238–247.

[42] Y. Sun, Y. Guo, C. Luo, J. Tropp, and M. Udell, "Low-rank tucker approximation of a tensor from streaming data," *SIAM J. Math. Data Sci.*, vol. 2, no. 4, pp. 1123–1150, 2020, publisher: SIAM.

[43] W. Austin, G. Ballard, and T. G. Kolda, "Parallel tensor compression for large-scale scientific data," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, Chicago, IL, May 2016, pp. 912–922.

[44] M. Che and Y. Wei, “Randomized algorithms for the approximations of Tucker and the tensor train decompositions,” *Advances Comput. Math.*, vol. 45, no. 1, pp. 395–428, 2019.

[45] O. A. Malik and S. Becker, “Low-Rank Tucker Decomposition of Large Tensors Using TensorSketch,” in *Proc. Advances Neural Inf. Process. Syst.*, Montreal, Canada, Dec. 2018, pp. 10 096–10 106.

[46] J. Oh, K. Shin, E. E. Papalexakis, C. Faloutsos, and H. Yu, “S-hot: scalable high-order tucker decomposition,” in *Proc. ACM Int. Conf. Web Search Data Mining*, Cambridge, UK, Feb. 2017, pp. 761–770.

[47] V. Barnett and T. Lewis, *Outliers in statistical data.*, 2nd ed. John Wiley & Sons Ltd., 1978.

[48] J. W. Tukey, “The future of data analysis,” *The annals of mathematical statistics*, vol. 33, no. 1, pp. 1–67, 1962, publisher: JSTOR.

[49] X. Fu, K. Huang, W. K. Ma, N. D. Sidiropoulos, and R. Bro, “Joint tensor factorization and outlying slab suppression with applications,” *IEEE Trans. Signal Process.*, vol. 63, pp. 6315–6328, Dec. 2015.

[50] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[51] X. Cao, X. Wei, Y. Han, and D. Lin, “Robust Face Clustering Via Tensor Decomposition,” *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2546–2557, 2015.

[52] Y. Pang, X. Li, and Y. Yuan, “Robust tensor analysis with L1-norm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 172–178, Feb. 2010.

[53] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, “Some options for L1-subspace signal processing,” in *Proc. Intern. Sym. Wireless Commun. Sys.*, Ilmenau, Germany, Aug. 2013, pp. 1–5.

[54] ——, “Optimal algorithms for L1-subspace signal processing,” *IEEE Trans. Signal Process.*, vol. 62, pp. 5046–5058, Oct. 2014.

[55] P. P. Markopoulos, S. Kundu, and D. A. Pados, “L1-fusion: Robust linear-time image recovery from few severely corrupted copies,” in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, Canada, Sep. 2015, pp. 1225–1229.

[56] Y. Liu and D. A. Pados, “Compressed-sensed-domain L1-PCA video surveillance,” *IEEE Trans. Multimedia*, vol. 18, pp. 351–363, Mar. 2016.

[57] P. P. Markopoulos and F. Ahmad, “Indoor human motion classification by L1-norm subspaces of micro-Doppler signatures,” in *Proc. IEEE Radar Conf.*, Seattle, WA, May 2017, pp. 1807–1810.

[58] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, “Efficient L1-norm Principal-Component Analysis via bit flipping,” *IEEE Trans. Signal Process.*, vol. 65, pp. 4252–4264, Aug. 2017.

[59] P. P. Markopoulos, D. G. Chachlakis, and E. E. Papalexakis, “The Exact Solution to Rank-1 L1-norm TUCKER2 Decomposition,” *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 511–515, Jan. 2018.

[60] P. P. Markopoulos, D. G. Chachlakis, and A. Prater-Bennette, “L1-norm Higher-Order Singular-value Decomposition,” in *Proc. IEEE Global Conf. Signal Inf. Process.*, Anaheim, CA, Nov. 2018, pp. 1353–1357.

[61] C. D. G and M. P. P, “Novel algorithms for exact and efficient L1-norm-based TUCKER2 decomposition,” in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, Canada, Apr. 2018, pp. 6294–6298.

[62] D. G. Chachlakis, M. Dhanaraj, A. Prater-Bennette, and P. P. Markopoulos, “Options for multimodal classification based on L1-Tucker decomposition,” in *Proc. SPIE Defense Commercial Sens.*, Apr. 2019, pp. 109 890O:1–109 890O:13, event-place: Baltimore, MD.

[63] D. G. Chachlakis, A. Prater-Bennette, and P. P. Markopoulos, “L1-Norm Tucker Tensor Decomposition,” *IEEE Access*, vol. 7, pp. 178 454–178 465, 2019.

[64] ——, “L1-Norm Higher-Order Orthogonal Iterations for Robust Tensor Analysis,” in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Barcelona, Spain (online), May 2020, pp. 4826–4830.

[65] D. G. Chachlakis, M. Dhanaraj, A. Prater-Bennette, and P. Markopoulos, “Dynamic l1-norm tucker tensor decomposition,” *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2021.

[66] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[67] Y. Xu, “On the convergence of higher-order orthogonal iteration,” *Linear Multilinear Algebra*, vol. 66, no. 11, pp. 2247–2265, 2018.

[68] J. Fan, F. Han, and H. Liu, “Challenges of big data analysis,” *Nature Sci. Rev.*, vol. 1, pp. 293–314, Jun. 2014.

[69] H. Xu, C. Caramanis, and S. Sanghavi, “Robust PCA via outlier pursuit,” in *Proc. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2010, pp. 2496–2504.

[70] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, “Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization,” in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, Las Vegas, NV, Jun. 2016, pp. 5249–5257.

[71] S. Li, W. Wang, H. Qi, B. Ayhan, C. Kwan, and S. Vance, “Low-rank tensor decomposition based anomaly detection for hyperspectral imagery,” in *Proc. IEEE Int. Conf. Image Process.*, Quebec City, Canada, Sep. 2015, pp. 4525–4529.

[72] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems.* Oxford, UK: Oxford University Press, 2004.

[73] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, “Robust principal component analysis with non-greedy L1-norm maximization,” in *Proc. Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, Jul. 2011, pp. 1433–1438.

[74] N. Kwak, “Principal component analysis based on L1-norm maximization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, 2008.

[75] M. McCoy and J. A. Tropp, “Two proposals for robust PCA using semidefinite programming,” *Electronic J. Statist.*, vol. 5, pp. 1123–1160, 2011.

[76] B. Minnehan and A. Savakis, “Grassmann Manifold Optimization for Fast L1-Norm Principal Component Analysis,” *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 242–246, 2019.

[77] P. P. Markopoulos, M. Dhanaraj, and A. Savakis, “Adaptive L1-norm principal-component analysis with online outlier rejection,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1131–1143, 2018.

[78] M. Pierantozzi, Y. Liu, D. A. Pados, and S. Colonnese, “Video background tracking and foreground extraction via L1-subspace updates,” in *Proc. SPIE Defense Commercial Sens.*, Baltimore, MD, Apr. 2016, pp. 985 708:1–985 708:16.

[79] N. Tsagkarakis, P. P. Markopoulos, G. Sklivanitis, and D. A. Pados, "L1-norm principal-component analysis of complex data," *IEEE Trans. Signal Process.*, vol. 66, pp. 3256–3267, Jun. 2018.

[80] M. Dhanaraj and P. P. Markopoulos, "Novel algorithm for incremental L1-norm principal-component analysis," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2020–2024.

[81] S. Papadimitriou, J.Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," in *Int. Conf. Very Large Databases*, Trondheim, Norway, Aug. 2005, pp. 697–708.

[82] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," in *IEEE Int. Symp. Inf. Theory*, Vail, CO, Jun. 2018, pp. 1–5.

[83] J. Feng, H. Xu, S. Mannor, and S. Yan, "Online PCA for contaminated data," in *Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, 2013, pp. 764–772.

[84] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *IEEE Conf. Comput. Vis. Pattern Recogn.*, Providence, RI, Jun. 2012, pp. 1568–1575.

[85] Y. Liu, K. Tountas, D. A. Pados, S. N. Batalama, and M. J. Medley, "L1-subspace tracking for streaming data," *Pattern Recogn.*, vol. 97, pp. 106 992:1–106 992:13, 2020, publisher: Elsevier.

[86] M. Baskaran, M. H. Langston, T. Ramananandro, D. Bruns-Smith, T. Henretty, J. Ezick, and R. Lethin, "Accelerated low-rank updates to tensor decompositions," in *IEEE High Perf. Extreme Comput. Conf.*, Waltham, MA, Sep. 2016, pp. 1–7.

[87] R. Pagh, "Compressed matrix multiplication," *ACM Trans. Comput. Theory*, vol. 5, no. 3, pp. 1–17, 2013.

[88] W. Hu, X. Li, X. Zhang, X. Shi, S. Maybank, and Z. Zhang, "Incremental tensor subspace learning and its applications to foreground segmentation and tracking," *Int. J. Comput. Vision*, vol. 91, no. 3, pp. 303–327, 2011.

[89] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," in *IEEE Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[90] A. Sobral, C. G. Baker, T. Bouwmans, and E.-h. Zahzah, "Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction," in *Int. Conf. Image Anal. Recogn.*, Algarve, Portugal, Oct. 2014, pp. 94–103.

[91] A. Sobral, S. Javed, S. Ki Jung, T. Bouwmans, and E.-h. Zahzah, “Online Tensor Decomposition for Background Subtraction in Multispectral Video Sequences,” in *IEEE Int. Conf. Comput. Vision*, Santiago, Chile, Dec. 2015, pp. 106–113. [Online]. Available: https://github.com/andrewssobral/ostd

[92] S. Smith, K. Huang, N. D. Sidiropoulos, and G. Karypis, “Streaming tensor factorization for infinite data sources,” in *SIAM Int. Conf. Data Mining*, San Diego, CA, May 2018, pp. 81–89.

[93] Y. Du, Y. Zheng, K.-c. Lee, and S. Zhe, “Probabilistic streaming tensor decomposition,” in *IEEE Int. Conf. Data Mining*, Singapore, Nov. 2018, pp. 99–108.

[94] M. Najafi, L. He, and S. Y. Philip, “Outlier-Robust Multi-Aspect Streaming Tensor Completion and Factorization.” in *Int. Joint Conf. Artifical Intell.*, Macao, China, Aug. 2019, pp. 3187–3194.

[95] P. Li, J. Feng, X. Jin, L. Zhang, X. Xu, and S. Yan, “Online robust low-rank tensor modeling for streaming data analysis,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 4, pp. 1061–1075, 2019.

[96] S. Mark, “Least squares optimization with L1-norm regularization,” *CS542B Project Report*, vol. 504, pp. 195–221, 2005.

[97] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “MPCA: Multilinear principal component analysis of tensor objects,” *IEEE transactions on Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008, publisher: IEEE.

[98] C. F. Van Loan, “The ubiquitous Kronecker product,” *Journal of computational and applied mathematics*, vol. 123, no. 1-2, pp. 85–100, 2000, publisher: Elsevier.

[99] “The Extended Yale Face Database B.” [Online]. Available: http://vision.ucsd.edu/iskwak/ExtYaleDatabase/ExtYaleB.html

[100] P. Orlik and H. Terao, *Arrangements of Hyperplanes.* New York, NY, USA: Springer-Verlag, Berlin-Heidelberg, 1992, vol. 300.

[101] R. Winder, “Partitions of N-space by hyperplanes,” *SIAM Journal on Applied Mathematics*, vol. 14, no. 4, pp. 811–818, 1966, publisher: SIAM.

[102] K. S. Brown and P. Diaconis, “Random walks and hyperplane arrangements,” *Annals of Probability*, pp. 1813–1854, 1998, publisher: JSTOR.

[103] G. N. Karystinos and A. P. Liavas, "Efficient computation of the binary vector that maximizes a rank-deficient quadratic form," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3581–3593, 2010.

[104] P. A. Huy and C. Andrzej, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *IEICE Nonlinear Theory Appl.*, vol. 1, no. 1, pp. 37–68, 2010.

[105] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[106] S. Smith, J. W. Choi, J. Li, R. Vuduc, J. Park, X. Liu, and G. Karypis, "FROSTT: The Formidable Repository of Open Sparse Tensors and Tools," 2017. [Online]. Available: http://frostt.io/

[107] M. Zhang and C. Ding, "Robust tucker tensor decomposition for effective image representation," in *Proc. IEEE Int. Conf. Comput. Vision*, Sydney, Australia, Dec. 2013, pp. 2448–2455.

[108] K. Tountas, D. G. Chachlakis, P. P. Markopoulos, and D. A. Pados, "Iteratively Re-weighted L1-PCA of Tensor Data," in *IEEE Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, 2019, pp. 1658–1661.

[109] K. Tountas, G. Sklivanitis, D. A. Pados, and M. J. Medley, "Tensor Data Conformity Evaluation for Interference-Resistant Localization," in *IEEE Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, 2019, pp. 1582–1586.

[110] "Context Aware Vision using Image-based Active Recognition (CAVIAR)." [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[111] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *IEEE Int. Joint Conf. Neural Net.*, Portland, OR, Jul. 2003, pp. 2685–2690.

[112] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *IEEE Conf. Comput. Vision Pattern Recogn. Workshops*, Providence, RI, Jun. 2012, pp. 1–8.

[113] J. Yuchen, C. Yanxi, and G. Yuantao, "Subspace change-point detection: A new model and solution," *IEEE J. Select. Top. Signal Process.*, vol. 12, no. 6, pp. 1224–1239, Oct. 2018, publisher: IEEE.

[114] I. Jolliffe, *Principal Component Analysis.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, publication Title: International Encyclopedia of Statistical Science.

[115] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.* Baltimore, MD: Johns Hopkins University Press, 1996.

[116] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, and F. Roli, "Is feature selection secure against training data poisoning?" in *Proc. Int. Conf. Mach. Learn.*, Lille, France, Jun. 2015, pp. 1689–1698.

[117] E. J. Cands, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[118] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2010, pp. 2496–2504.

[119] Q. Ke and T. Kanade, "Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Comp. Vision Patt. Recogn.*, vol. 1, San Diego, CA, Jun. 2005, pp. 739–746.

[120] F. Chierichetti, S. Gollapudi, R. Kumar, S. Lattanzi, R. Panigrahy, and D. P. Woodruff, "Algorithms for $\backslash ell_p $ low-rank approximation," in *International Conference on Machine Learning.* PMLR, 2017, pp. 806–814.

[121] A. Kyrillidis, "Simple and practical algorithms for ℓ_p-norm low-rank approximation," *Conference on Uncertainty in Artificial Intelligence (UAI-18)*, 2018.

[122] M. Dhanaraj and P. P. Markopoulos, "Novel algorithm for incremental l1-norm principal-component analysis," in *Proc. IEEE/EURASIP European Signal Process. Conf.*, Rome, Italy, Sep. 2018, pp. 2020–2024.

[123] ——, "Stochastic Principal Component Analysis via Mean Absolute Projection Maximization," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Ottawa, Canada, Nov. 2019, pp. 1–5.

[124] N. Tsagkarakis, P. P. Markopoulos, G. Sklivanitis, and D. A. Pados, "L1-norm principal-component analysis of complex data," *IEEE Trans. Signal Process.*, vol. 66, pp. 3256–3267, Jun. 2018.

[125] M. Dhanaraj, D. G. Chachlakis, and P. P. Markopoulos, "Incremental complex L1-PCA for direction-of-arrival estimation," in *2018 IEEE Western New York Image and Signal Processing Workshop (WNYISPW).* IEEE, 2018, pp. 1–5.

[126] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "Direction finding by complex L 1-principal-component analysis," in *Proc. IEEE Int. Workshop on Signal Proc. Adv. Wireless Comm.*, Stockholm, Sweden, Jun. 2015, pp. 475–479.

[127] D. G. Chachlakis, P. P. Markopoulos, R. J. Muchhala, and A. Savakis, "Visual tracking with L1-Grassmann manifold modeling," in *Proc. SPIE Defense and Commercial Sens.*, Anaheim, CA, Apr. 2017, pp. 1 021 102:1–1 021 102:12.

[128] D. G. Chachlakis, Y. Tsitsikas, E. E. Papalexakis, and P. P. Markopoulos, "Robust multi-relational learning with absolute projection rescal," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Ottawa, Canada, Nov. 2019, pp. 1–5.

[129] S. Park and N. Kwak, "Independent component analysis by lp-norm optimization," *Pattern Recognition*, vol. 76, pp. 752–760, 2018, publisher: Elsevier.

[130] Q. Ye, L. Fu, Z. Zhang, H. Zhao, and M. Naiem, "Lp-and Ls-norm distance based robust linear discriminant analysis," *Neural Networks*, vol. 105, pp. 393–404, 2018, publisher: Elsevier.

[131] C.-N. Li, Y.-H. Shao, Z. Wang, and N.-Y. Deng, "Robust bilateral Lp-norm two-dimensional linear discriminant analysis," *Information Sciences*, vol. 500, pp. 274–297, 2019, publisher: Elsevier.

[132] G. Tang and G. Lu, "Block Principle Component Analysis with Lp-norm for Robust and Sparse Modelling," *Journal of Shanghai Jiaotong University (Science)*, vol. 23, no. 3, pp. 398–403, 2018, publisher: Springer.

[133] T. Zhang, H. Wu, Y. Liu, L. Peng, C. Yang, and Z. Peng, "Infrared small target detection based on non-convex optimization with Lp-norm constraint," *Remote Sensing*, vol. 11, no. 5, p. 559, 2019, publisher: Multidisciplinary Digital Publishing Institute.

[134] N. Kwak, "Principal component analysis by *Lp*-norm maximization," *IEEE Trans. Cybernetics*, vol. 44, no. 5, pp. 594–609, 2013.

[135] B. Minnehan, N. Nagananda, and A. Savakis, "Grip-pca: Grassmann iterative p-norm principal component analysis," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 90–98, 2020.

[136] D. G. Chachlakis and P. P. Markopoulos, "Combinatorial Search for the Lp-Norm Principal Component of a Matrix," in *IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, 2019, pp. 1611–1615.

[137] D. G. Chachlakis and P. P. Markopoulos, “Novel algorithms for lp-quasi-norm principal-component analysis,” in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 1045–1049.

[138] D. G. Chachlakis and P. P. Markopoulos, “Lp-quasinorm principal-component analysis,” *(under submission)*, 2021.

[139] S. Foucart, A. Pajor, H. Rauhut, and T. Ullrich, “The Gelfand widths of Lp-balls for $0 < p \leq 1$,” *J. Complexity*, vol. 26, no. 6, pp. 629–640, 2010.

[140] S. J. Wright, *Primal-dual interior-point methods.* Siam, 1997, vol. 54.

[141] D. G. Luenberger, Y. Ye, and others, *Linear and nonlinear programming.* Reading, MA: Addison-wesley, 1984, vol. 2.

[142] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, 2014.

[143] M. C. Grant and S. P. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent advances in learning and control.* Springer, 2008, pp. 95–110.

[144] P. B. Yale, *Geometry and symmetry.* San Francisco, CA, USA: HoldenDay, 1968.

[145] R. Winder, “Partitions of n-space by hyperplanes,” *SIAM J. Appl. Math.*, vol. 14, no. 4, pp. 811–818, 1966.

[146] K. S. Brown and P. Diaconis, “Random walks and hyperplane arrangements,” *Ann. Probab.*, pp. 1813–1854, 1998.

[147] A. Ben-Tal and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications.* Philadelphia, PA: SIAM, 2001.

[148] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[149] M. Journe, Y. Nesterov, P. Richtrik, and R. Sepulchre, “Generalized power method for sparse principal component analysis,” *J. Machine Learn. Res.*, vol. 11, no. 2, 2010.

[150] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[151] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis,” in *Biomedical image processing and biomedical visualization*, vol. 1905. International Society for Optics and Photonics, 1993, pp. 861–870.

[152] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, “Breast cancer diagnosis and prognosis via linear programming,” *Operations Research*, vol. 43, no. 4, pp. 570–577, 1995.

[153] P. Pal and P. P. Vaidyanathan, “Coprime sampling and the music algorithm,” in *IEEE Digit. Signal Process. and Signal Process. Edu. Meeting (DSP/SPE)*, Sedona, AZ, Jan. 2011, pp. 289–294.

[154] ——, “Nested arrays: A novel approach to array processing with enhanced degrees of freedom,” *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4167–4181, 2010.

[155] D. G. Chachlakis, P. P. Markopoulos, and F. Ahmad, “The mean-squared-error of autocorrelation sampling in coprime arrays,” in *IEEE Int. Workshop on Comput. Advances in Multi-Sensor Adaptive Process. (CAMSAP)*, Curaçao, Dutch Antilles, Dec. 2017, pp. 1–5.

[156] ——, “MMSE-based autocorrelation sampling for coprime arrays,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Alberta, Canada, 2018, pp. 3474–3478.

[157] E. BouDaher, F. Ahmad, and M. Amin, “Sparsity-based direction finding of coherent and uncorrelated targets using active nonuniform arrays,” *IEEE Signal Process. Lett.*, vol. 22, pp. 1628–1632, Oct. 2015.

[158] S. Qin, Y. D. Zhang, and M. Amin, “Generalized coprime array configurations for direction-of-arrival estimation,” *IEEE Trans. Signal Process.*, vol. 63, pp. 1377–1390, Mar. 2015.

[159] ——, “DOA estimation of mixed coherent and uncorrelated targets exploiting coprime MIMO radar,” *Digit. Signal Process.*, vol. 61, pp. 26–34, Feb. 2017.

[160] P. P. Vaidyanathan and P. Pal, “Why does direct-MUSIC on sparse arrays work?” in *IEEE Asilomar Conf. on Signals, Syst. and Comput. (ACSSC)*, Pacific Grove, CA, Nov. 2014, pp. 2007–2011.

[161] ——, “Theory of sparse coprime sensing in multiple dimensions,” *IEEE Trans. Signal Procces.*, vol. 59, pp. 3592–3608, Aug. 2011.

[162] C.-L. Liu and P. P. Vaidyanathan, “Remarks on the spatial smoothing step in coarray MUSIC,” *IEEE Signal Process. Lett.*, vol. 22, pp. 1438–1442, Sep. 2015.

[163] ——, “Cramér-rao bounds for coprime and other sparse arrays, which find more sources than sensors,” *Digit. Signal Process.*, vol. 61, pp. 43–61, Feb. 2017.

[164] Z. Tan, Y. C. Eldar, and A. Nehorai, “Direction of arrival estimation using co-prime arrays: A super resolution viewpoint,” *IEEE Trans. Signal Process.*, vol. 62, pp. 5565–5576, Nov. 2014.

[165] M. Wang and A. Nehorai, “Coarrays, MUSIC, and the Cramér-Rao bound,” *IEEE Trans. Signal Process.*, vol. 65, pp. 933–946, Feb. 2017.

[166] Z. Shi, C. Zhou, Y. Gu, N. A. Goodman, and F. Qu, “Source estimation using coprime array: A sparse reconstruction perspective,” *IEEE Sensors*, vol. 17, no. 3, pp. 755–765, Feb. 2017.

[167] P. P. Vaidyanathan and P. Pal, “Sparse sensing with coprime arrays,” in *IEEE Asilomar Conf. on Signals, Syst. and Comput. (ACSSC)*, Pacific Grove, CA, Nov. 2010, pp. 1405–1409.

[168] C. Zhou, Z. Shi, and Y. Gu, “Coprime array adaptive beamforming with enhanced degrees-of-freedom capability,” in *IEEE Radar Conf. (RadarConf)*, Seattle, WA, May 2017, pp. 1–4.

[169] Y. Liu and J. R. Buck, “Gaussian source detection and spatial spectral estimation using a coprime sensor array with the min processor,” *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 186–199, 2018.

[170] M. Amin, X. Wang, Y. D. Zhang, F. Ahmad, and E. Aboutanios, “Sparse arrays and sampling for interference mitigation and DOA estimation in GNSS,” *Proc. IEEE*, vol. 104, pp. 1302–1317, Jun. 2016.

[171] C.-L. Liu and P. P. Vaidyanathan, “Coprime arrays and samplers for space-time adaptive processing,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Queensland, Australia, Apr. 2015, pp. 2364–2368.

[172] X. Wang, Z. Yang, and J. Huang, “Robust space-time adaptive processing for airborne radar with coprime arrays in presence of gain and phase errors,” *IET Radar, Sonar & Navigation*, vol. 15, no. 1, pp. 75–84, 2021.

[173] C.-L. Liu, P. P. Vaidyanathan, and P. Pal, “Coprime coarray interpolation for DOA estimation via nuclear norm minimization,” in *IEEE Int. Symp. on Circuits and Syst. (ISCAS)*, Montreal, Canada, May 2016, pp. 2639–2642.

[174] E. BouDaher, F. Ahmad, and M. G. Amin, “Performance analysis of sparsity-based interpolation for DOA estimation with non-uniform arrays,” in *SPIE Defense and Commercial. Sens.*, Anaheim, CA, May 2017, pp. 102 110F–1:102 110F–6.

[175] W. Zheng, X. Zhang, Y. Wang, J. Shen, and B. Champagne, "Padded coprime arrays for improved DOA estimation: Exploiting hole representation and filling strategies," *IEEE Trans. Signal Process.*, vol. 68, pp. 4597–4611, 2020.

[176] Z. Zheng, Y. Huang, W. Q. Wang, and H. C. So, "Direction-of-Arrival estimation of coherent signals via coprime array interpolation," *IEEE Signal Process. Lett.*, vol. 27, pp. 585–589, 2020.

[177] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang, "Direction-of-arrival estimation for coprime array via virtual array interpolation," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5956–5971, 2018.

[178] E. BouDaher, Y. Jia, F. Ahmad, and M. G. Amin, "Multi-frequency co-prime arrays for high-resolution direction-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3797–3808, 2015.

[179] C. Zhou, Y. Gu, X. Fan, Z. Shi, G. Mao, and Y. D. Zhang, "Direction-of-arrival estimation for coprime array via virtual array interpolation," *IEEE Trans. Signal Process.*, vol. 66, no. 22, pp. 5956–5971, 2018.

[180] C. Zhou, Z. Shi, Y. Gu, and Y. D. Zhang, "Coarray interpolation-based coprime array doa estimation via covariance matrix reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP 2018)*, Calgary, Alberta, Canada, Apr. 2018, pp. 3479–3483.

[181] Y. Li, J. Thomas, B. Ouyang, T.-C. Su, T. Zhou, and F. Ahmad, "Design and experimental study of underwater inflatable co-prime sonar array (uicsa)," in *OCEANS 2019 MTS/IEEE SEATTLE*, Seattle, WA, Oct. 2019, pp. 1–7.

[182] T. Zhou, F. Ahmad, B. Ouyang, and Y. Li, "Near-field source localization with an underwater inflatable sonar co-prime array," in *SPIE Defense and Commercial Sens.*, vol. 10989, Baltimore, MD, Apr. 2019, p. 1098905.

[183] A. Koochakzadeh and P. Pal, "Channel estimation for hybrid mimo communication with (non-) uniform linear arrays via tensor decomposition," in *IEEE Sensor Array Multichannel Signal Process. Workshop (SAM)*, Hangzhou, China, Jun. 2020, pp. 1–5.

[184] S. Qin, Y. D. Zhang, and M. G. Amin, "Improved two-dimensional DOA estimation using parallel coprime arrays," *Signal Process.*, vol. 172, p. 107428, 2020.

[185] G. Qin, M. G. Amin, and Y. D. Zhang, “Analysis of coprime arrays on moving platform,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Brighton, United Kingdom, May 2019, pp. 4205–4209.

[186] J. Ramirez, J. Odom, and J. Krolik, “Exploiting array motion for augmentation of co-prime arrays,” in *IEEE Sensor Array Mutlichannel Signal Process. Workshop (SAM)*, A Coruña, Spain, 2014, pp. 525–528.

[187] G. Qin, M. G. Amin, and Y. D. Zhang, “DOA estimation exploiting sparse array motions,” *IEEE Trans. Signal Process.*, vol. 67, no. 11, pp. 3013–3027, 2019.

[188] Z. Zheng, W.-Q. Wang, Y. Kong, and Y. D. Zhang, “Misc array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect,” *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1728–1741, 2019.

[189] R. Pribić, M. Coutino, and G. Leus, “Stochastic resolution analysis of co-prime arrays in radar,” in *IEEE Statistical Signal Process. Workshop (SSP)*, Palma de Mallorca, Spain, Jun. 2016, pp. 1–5.

[190] S. Qin, Y. D. Zhang, and M. Amin, “Multi-target localization using frequency diverse coprime arrays with coprime frequency offsets,” in *IEEE Radar Conf. (RadarConf)*, Philadelphia, PA, May 2016, pp. 1–5.

[191] P. P. Vaidyanathan and P. Pal, “Direct-MUSIC on sparse arrays,” in *Signal Procces. Commun. (SPCOM)*, Karnataka, India, Jul. 2012, pp. 1–5.

[192] C.-L. Liu and P. P. Vaidyanathan, “Tensor MUSIC in multidimensional sparse arrays,” in *IEEE Asilomar Conf. on Signals, Syst. and Comput. (ACSSC)*, Pacific Grove, CA, Nov. 2015, pp. 1783–1787.

[193] P. P. Vaidyanathan and P. Pal, “Sparse sensing with co-prime samplers and arrays,” *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 573–586, Oct. 2011.

[194] M. Guo, Y. D. Zhang, and T. Chen, “Doa estimation using compressed sparse array,” *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 4133–4146, 2018.

[195] M. Guo, T. Chen, and B. Wang, “An improved doa estimation approach using coarray interpolation and matrix denoising,” *Sensors*, vol. 17, no. 5, p. 1140, 2017.

[196] K. V. Mardia, “Statistics of directional data,” *Royal Statist. Society: Series B (Methodological)*, vol. 37, no. 3, pp. 349–371, 1975.

[197] A. Abdi, J. A. Barger, and M. Kaveh, “A parametric model for the distribution of the angle of arrival and the associated correlation function and power spectrum at the mobile station,” *IEEE Trans. Vehicular Technol.*, vol. 51, no. 3, pp. 425–434, 2002.

[198] D. G. Chachlakis and P. P. Markopoulos, “Structured autocorrelation matrix estimation for coprime arrays,” *Signal Processing*, vol. 183, p. 107987, 2021.

[199] D. G. Chachlakis, T. Zhou, F. Ahmad, and P. P. Markopoulos, “Minimum mean-squared-error autocorrelation processing in coprime arrays,” *accepted to appear, Digital Signal Processing*, 2021.

[200] A. Graham, *Kronecker Products and Matrix Calculus with Applications.* Chichester, U.K.: Ellis Horwood, 1981.

[201] P. Stoica and A. Nehorai, “MUSIC, maximum likelihood, and Cramér-Rao bound,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 5, pp. 720–741, 1989.

[202] C. J. Lam and A. C. Singer, “Bayesian beamforming for DOA uncertainty: Theory and implementation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4435–4445, 2006.

[203] Y. Yang, S. Dang, M. Wen, S. Mumtaz, and M. Guizani, “Mobile millimeter wave channel tracking: A bayesian beamforming framework against doa uncertainty,” in *IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, Dec. 2019, pp. 1–6.

[204] L. F. Shampine, “Vectorized adaptive quadrature in matlab,” *Comput. Appl. Math.*, vol. 211, no. 2, pp. 131–140, 2008.

[205] E. Isaacson and H. B. Keller, *Analysis of numerical methods.* Courier Corporation, 2012.

[206] G. Dahlquist and Å. Björck, *Numerical methods in scientific computing, volume I.* SIAM, 2008.

[207] L. C. Andrews, *Special functions of mathematics for engineers.* SPIE Press, 1998, vol. 49.

[208] N. M. Temme, *Special functions: An introduction to the classical functions of mathematical physics.* New York, NY: Wiley, 1996.

[209] M. Hirakawa, H. Tsuji, and A. Sano, “Computationally efficient DOA estimation based on linear prediction with Capon method,” in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Salt Lake City, UT, May 2001, pp. 3009–3012.

[210] X. Chen, J. Xin, N. Zheng, and A. Sano, "Direction-of-arrival estimation of coherent narrowband signals with arbitrary linear array," in *IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Lorient, France, Oct. 2017, pp. 1–5.

[211] L. Wang, R. C. de Lamare, and M. Haardt, "Direction finding algorithms based on joint iterative subspace optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 4, pp. 2541–2553, 2014.

[212] ——, "Reduced-rank DOA estimation based on joint iterative subspace recursive optimization and grid search," in *IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Dallas, TX, Mar. 2010, pp. 2626–2629.

[213] J. Lardies, H. Ma, and M. Berthillier, "Power estimation of acoustical sources by an array of microphones," in *Société Française d'Acoustique, Acoustics*, Nantes, France, Apr. 2012, pp. hal–00 810 871.

[214] Y. Shen, C. Zhou, Y. Gu, H. Lin, and Z. Shi, "Vandermonde decomposition of coprime coarray covariance matrix for DOA estimation," in *IEEE Int. Workshop Signal Process. Advances in Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 3–6.

[215] N. J. Higham, "Computing a nearest symmetric positive semi-definite matrix," *Linear Algebra Appl.*, vol. 103, pp. 102–118, 1988.

Appendices

Appendix A

Chapter 2

A.1 Proof of Convergence for (2.34)

Proof.

$$\|\mathbf{U}_{t-1}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\|_1 \tag{A.1}$$

$$= Tr(\mathbf{U}_{t-1}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\ \mathrm{sgn}(\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)^{\top}\mathbf{U}_{t-1})) \tag{A.2}$$

$$\leq Tr(\mathbf{U}_{t}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\mathrm{sgn}(\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)^{\top}\mathbf{U}_{t-1})) \tag{A.3}$$

$$\leq \|\mathbf{U}_{t}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\|_1. \tag{A.4}$$

At the same time, the metric of (2.30) is upper bounded by its exact solution [54]. Thus, the recursion in (2.34) is guaranteed to converge. In practice, iterations can be terminated when the metric-increase ratio $\|\mathbf{U}_{t}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\|_1\|\mathbf{U}_{t-1}{}^{\top}\mathtt{mat}(\boldsymbol{\mathcal{X}}, n)\|_1{}^{-1} - 1$ drops below a predetermined threshold $\tau > 0$, or when t exceeds a maximum number of permitted iterations. □

A.2 Proof of Lemma 2.4

Proof. For first inequality, it holds that

$$\left\|\mathbf{U}_n^{(q)}{}^{\top}\mathbf{A}_n^{(q)}\right\|_1 \tag{A.5}$$

$$= \left\|\mathbf{U}_n^{(q)}{}^{\top}\texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{k<n} \mathbf{U}_k^{(q)}{}^{\top} \times_{l>n} \mathbf{U}_l^{(q-1)}{}^{\top}, n\right)\right\|_1 \tag{A.6}$$

$$\geq \left\|\mathbf{U}_n^{(q-1)}{}^{\top}\texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{k<n} \mathbf{U}_k^{(q)}{}^{\top} \times_{l>n} \mathbf{U}_l^{(q-1)}{}^{\top}, n\right)\right\|_1 \tag{A.7}$$

$$= \left\|\mathbf{U}_{n-1}^{(q)}{}^{\top}\mathbf{A}_{n-1}^{(q)}\right\|_1. \tag{A.8}$$

By induction, for every $n > m$, $\left\|\mathbf{U}_n^{(q)}{}^{\top}\mathbf{A}_n^{(q)}\right\|_1 \geq \left\|\mathbf{U}_m^{(q)}{}^{\top}\mathbf{A}_m^{(q)}\right\|_1$. For the second inequality, it holds that

$$\left\|\mathbf{U}_1^{(q)}{}^{\top}\mathbf{A}_1^{(q)}\right\|_1 \tag{A.9}$$

$$= \left\|\mathbf{U}_1^{(q)}{}^{\top}\texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{k>1} \mathbf{U}_k^{(q-1)}{}^{\top}, 1\right)\right\|_1 \tag{A.10}$$

$$\overset{\text{Lemma 2.3}}{\geq} \left\|\mathbf{U}_1^{(q-1)}{}^{\top}\texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{k>1} \mathbf{U}_k^{(q-1)}{}^{\top}, 1\right)\right\|_1 \tag{A.11}$$

$$= \left\|\mathbf{U}_N^{(q-1)}{}^{\top}\texttt{mat}\left(\boldsymbol{\mathcal{X}} \times_{k<N} \mathbf{U}_k^{(q-1)}{}^{\top}, N\right)\right\|_1 \tag{A.12}$$

$$= \left\|\mathbf{U}_N^{(q-1)}{}^{\top}\mathbf{A}_N^{(q-1)}\right\|_1. \tag{A.13}$$

□

A.3 Proof of Proposition 2.2

Proof. It holds that

$$\begin{aligned}
\left\| \mathbf{U}_n^{(q')^\top} \mathbf{A}_n^{(q')} \right\|_1 &\overset{\text{Lemma 2.4}}{\geq} \left\| \mathbf{U}_1^{(q')^\top} \mathbf{A}_1^{(q')} \right\|_1 && \text{(A.14)} \\
&\overset{\text{Lemma 2.4}}{\geq} \left\| \mathbf{U}_N^{(q'-1)^\top} \mathbf{A}_N^{(q'-1)} \right\|_1 && \text{(A.15)} \\
&\overset{\text{Lemma 2.4}}{\geq} \left\| \mathbf{U}_1^{(q'-1)^\top} \mathbf{A}_1^{(q'-1)} \right\|_1 && \text{(A.16)} \\
&\geq \left\| \mathbf{U}_N^{(q)^\top} \mathbf{A}_N^{(q)} \right\|_1 && \text{(A.17)} \\
&\geq \left\| \mathbf{U}_n^{(q)^\top} \mathbf{A}_n^{(q)} \right\|_1 . && \text{(A.18)}
\end{aligned}$$

□

A.4 Proof of Lemma 2.5

Proof. Define $\mathbf{x} \overset{\triangle}{=} \text{vec}(\texttt{mat}\,(\boldsymbol{\mathcal{X}}, 1)) \in \mathbb{R}^P$ and $\mathbf{Z} \overset{\triangle}{=} \mathbf{U}_N \otimes \mathbf{U}_{N-1} \otimes \ldots \otimes \mathbf{U}_1 \in \mathbb{S}_{P \times p}$. Then,

$$\begin{aligned}
\|\boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{U}_n^\top\|_1 &= \|\mathbf{Z}^\top \mathbf{x}\|_1 && \text{(A.19)} \\
&\leq \sqrt{p}\|\mathbf{Z}^\top \mathbf{x}\|_2 && \text{(A.20)} \\
&\leq \sqrt{p}\|\mathbf{x}\|_2 = \sqrt{p}\|\boldsymbol{\mathcal{X}}\|_F . && \text{(A.21)}
\end{aligned}$$

□

Appendix B

Chapter 3

B.1 Proof of Lemma 3.1

Proof. Let $\mathbf{Q}_{\text{opt}}$ be the solution to (3.1). Without loss of generality, there exists $\mathbf{A} \in \mathbb{R}^{\rho \times K}$ and $\mathbf{B} \in \mathbb{R}^{D-\rho \times K}$ such that

$$\mathbf{Q}_{\text{opt}} = \mathbf{U}\mathbf{A} + \mathbf{U}^{\mathsf{c}}\mathbf{B}, \tag{B.1}$$

where $\mathbf{U}^{\mathsf{c}} \in \mathbb{R}^{D \times D-\rho}$ is such that $\mathbf{U}^{\mathsf{c}\top}\mathbf{U}^{\mathsf{c}} = \mathbf{I}_{D-\rho}$, $\text{span}(\mathbf{U}) \cap \text{span}(\mathbf{U}^{\mathsf{c}}) = \emptyset$, and $\dim(\mathbf{U}) + \dim(\mathbf{U}^{\mathsf{c}}) = D$. We notice that

$$\begin{aligned} \mathbf{X}^\top \mathbf{Q}_{\text{opt}} &= \mathbf{V\Sigma U}^\top \mathbf{U}\mathbf{A} + \mathbf{V\Sigma U}^\top \mathbf{U}^{\mathsf{c}}\mathbf{B} && \text{(B.2)} \\ &= \mathbf{V\Sigma U}^\top \mathbf{U}\mathbf{A} && \text{(B.3)} \\ &= \mathbf{X}^\top \mathbf{U}\mathbf{A}. && \text{(B.4)} \end{aligned}$$

It follows that $\mathbf{Q}_{\text{opt}} = \mathbf{U}\mathbf{A}$ for some $\mathbf{A} \in \mathbb{R}^{\rho \times K}$. Then, we define $\mathbf{Y} = \mathbf{\Sigma V}^\top$ and observe that, for any $\mathbf{S} \in \mathbb{R}^{\rho \times K}$, it holds

$$\begin{aligned} \mathbf{Y}^\top \mathbf{S} &= \mathbf{V\Sigma S} && \text{(B.5)} \\ &= \mathbf{V\Sigma U}^\top \mathbf{U}\mathbf{S} && \text{(B.6)} \\ &= \mathbf{X}^\top \mathbf{U}\mathbf{S}. && \text{(B.7)} \end{aligned}$$

By (B.4) and (B.7), if $\mathbf{A}_{\text{opt}}$ solves $\max._{\mathbf{A} \in \mathbb{S}_{\rho \times K}} \|\mathbf{Y}^\top \mathbf{A}\|_p^p$ then $\mathbf{Q}_{\text{opt}} = \mathbf{U}\mathbf{A}_{\text{opt}}$ solves (3.1). □

B.2 Proof of Lemma 3.2

Proof. Let $\mathbf{q}_{\mathrm{opt}} \in \mathbb{R}^D$ solve $\max._{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p$. Assume that $\|\mathbf{q}_{\mathrm{opt}}\|_2 < 1$ and consider the unit-norm vector $\widehat{\mathbf{q}}_{\mathrm{opt}} = \mathbf{q}_{\mathrm{opt}}\|\mathbf{q}_{\mathrm{opt}}\|_2^- 1 \in \mathbb{B}_{D\times 1}$. It holds that

$$\|\mathbf{X}^\top\widehat{\mathbf{q}}_{\mathrm{opt}}\|_p^p = \|\mathbf{q}_{\mathrm{opt}}\|_2^{-p}\|\mathbf{X}^\top\mathbf{q}_{\mathrm{opt}}\|_p^p > \|\mathbf{X}^\top\mathbf{q}_{\mathrm{opt}}\|_p^p, \tag{B.8}$$

which contradicts the assumption that $\|\mathbf{q}_{\mathrm{opt}}\|_2 < 1$. Then, $\mathbf{q}_{\mathrm{opt}}$ solves $\max._{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p$ and it must be such that $\|\mathbf{q}_{\mathrm{opt}}\|_2 = 1$. The latter implies that $\max_{\mathbf{q}\in\mathbb{S}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p = \max_{\mathbf{q}\in\mathbb{B}_{D\times 1}} \|\mathbf{X}^\top\mathbf{q}\|_p^p$. □

B.3 Proof of Lemma 3.3

Proof. Consider $\mathbf{b}, \mathbf{b}' \in \{\pm 1\}^N$ such that $\mathbf{b} \neq \mathbf{b}'$. Assume that there exist $\mathbf{q} \in \mathcal{C}(\mathbf{b})$ and $\mathbf{q}' \in \mathcal{C}(\mathbf{b}')$ such that $\mathbf{q} = \mathbf{q}'$. The latter implies that $\mathrm{sgn}(\mathbf{X}^\top\mathbf{q}) = \mathrm{sgn}(\mathbf{X}^\top\mathbf{q}')$ which can not be true since $\mathrm{sgn}(\mathbf{X}^\top\mathbf{q}) = \mathbf{b} \neq \mathbf{b}' = \mathrm{sgn}(\mathbf{X}^\top\mathbf{q}')$. We conclude that there is no $\mathbf{q} \in \mathcal{C}(\mathbf{b})$ and $\mathbf{q}' \in \mathcal{C}(\mathbf{b}')$ such that $\mathbf{q} = \mathbf{q}'$. Formally, $\mathcal{C}(\mathbf{b}) \cap \mathcal{C}(\mathbf{b}') = \emptyset$ for any $\mathbf{b} \neq \mathbf{b}'$. □

B.4 Proof of Lemma 3.5

Proof. Consider $\mathbf{b} \in \{\pm 1\}^N$ such that $\mathcal{C}(\mathbf{b}) \neq \emptyset$. Let $\mathbf{q}, \mathbf{q}' \in \mathcal{C}(\mathbf{b})$ –i.e., $\mathrm{sgn}(\mathbf{X}^\top\mathbf{q}) = \mathrm{sgn}(\mathbf{X}^\top\mathbf{q}') = \mathbf{b}$, $\|\mathbf{q}\|_2 \leq 1$, and $\|\mathbf{q}'\|_2 \leq 1$. First, notice that there exists $\mathbf{a} \in \mathbb{R}^N$ such that $[\mathbf{a}]_n > 0 \forall n \in [N]$ and

$$\mathbf{a} \odot \mathbf{X}^\top\mathbf{q} = \mathbf{X}^\top\mathbf{q}'. \tag{B.9}$$

Consider $\theta \in [0, 1]$ and let $\bar{\mathbf{q}} = \theta\mathbf{q} + (1-\theta)\mathbf{q}'$. It holds that

$$\begin{aligned}
\mathrm{sgn}(\mathbf{X}^\top\bar{\mathbf{q}}) &= \mathrm{sgn}(\theta\mathbf{X}^\top\mathbf{q} + (1-\theta)\mathbf{X}^\top\mathbf{q}') && \text{(B.10)}\\
&= \mathrm{sgn}(\theta\mathbf{X}^\top\mathbf{q} + (1-\theta)\mathbf{a}\odot\mathbf{X}^\top\mathbf{q}) && \text{(B.11)}\\
&= \mathrm{sgn}\left(\mathbf{X}^\top\mathbf{q}\odot(\theta\mathbf{1}_N + (1-\theta)\mathbf{a})\right) && \text{(B.12)}\\
&= \mathrm{sgn}(\mathbf{X}^\top\mathbf{q}) && \text{(B.13)}\\
&= \mathbf{b}. && \text{(B.14)}
\end{aligned}$$

Moreover, by Cauchy-Swarchz inequality we obtain that

$$\|\bar{\mathbf{q}}\|_2^2 = \theta^2\|\mathbf{q}\|_2^2 + (1-\theta)^2\|\mathbf{q}'\|_2^2 + 2\theta(1-\theta)\mathbf{q}^\top\mathbf{q}' \tag{B.15}$$

$$\leq \theta^2\|\mathbf{q}\|_2^2 + (1-\theta)^2\|\mathbf{q}\|_2^2 + 2\theta(1-\theta)\|\mathbf{q}\|_2^2 \tag{B.16}$$

$$= \|\mathbf{q}\|_2^2\left(\theta^2 + (1-\theta)^2 + 2\theta(1-\theta)\right) \tag{B.17}$$

$$= \|\mathbf{q}\|_2^2 \leq 1. \tag{B.18}$$

It follows that for any $\mathbf{q}, \mathbf{q}' \in \mathcal{C}(\mathbf{b})$ and $\theta \in [0,1]$, $\bar{\mathbf{q}} = \theta\mathbf{q} + (1-\theta)\mathbf{q}' \in \mathcal{C}(\mathbf{b})$. Thus, $\mathcal{C}(\mathbf{b})$ is a convex set. □

Proof of Lemma 3.6

Proof. Consider $\mathbf{b} \in \{\pm 1\}^N$ such that $\mathcal{C}(\mathbf{b}) \neq \emptyset$. Let $\mathbf{q}, \mathbf{q}' \in \mathcal{C}(\mathbf{b})$ –i.e., $\operatorname{sgn}(\mathbf{X}^\top\mathbf{q})) = \operatorname{sgn}(\mathbf{X}^\top\mathbf{q}') = \mathbf{b}$, $\|\mathbf{q}\|_2 \leq 1$, and $\|\mathbf{q}'\|_2 \leq 1$. For any $\bar{\mathbf{q}} = \theta\mathbf{q} + (1-\theta)\mathbf{q}'$, it holds

$$\left\|\mathbf{X}^\top\bar{\mathbf{q}}\right\|_p^p = \left\|\theta\mathbf{X}^\top\mathbf{q} + (1-\theta)\mathbf{X}^\top\mathbf{q}'\right\|_p^p \tag{B.19}$$

$$= \left\|\mathbf{X}^\top\mathbf{q} \odot (\theta\mathbf{1}_N + (1-\theta)\mathbf{a})\right\|_p^p \tag{B.20}$$

$$= \sum_{n\in[N]} \Big((\theta + (1-\theta)[\mathbf{a}]_n)|\mathbf{x}_n^\top\mathbf{q}|\Big)^p \tag{B.21}$$

$$\geq \sum_{n\in[N]} \theta^p|\mathbf{x}_n^\top\mathbf{q}|^p + (1-\theta)^p[\mathbf{a}]_n^p|\mathbf{x}_n^\top\mathbf{q}|^p \tag{B.22}$$

$$= \sum_{n\in[N]} \theta^p|\mathbf{x}_n^\top\mathbf{q}|^p + (1-\theta)^p|[\mathbf{a}]_n\mathbf{x}_n^\top\mathbf{q}|^p \tag{B.23}$$

$$\geq \sum_{n\in[N]} \theta|\mathbf{x}_n^\top\mathbf{q}|^p + (1-\theta)|[\mathbf{a}]_n\mathbf{x}_n^\top\mathbf{q}|^p \tag{B.24}$$

$$= \theta\|\mathbf{X}^\top\mathbf{q}\|_p^p + (1-\theta)\|\mathbf{X}^\top\mathbf{q}'\|_p^p. \tag{B.25}$$

□

B.5 Proof of Lemma 3.7

Proof. Let $\mathbf{q}_{\text{opt}}$ be the solution to (3.1). We find that

$$\begin{aligned} \|\mathbf{X}^\top \mathbf{q}_{\text{opt}}\|_p^p &= \sum_{n\in[N]} |\mathbf{x}_n^\top \mathbf{q}_{\text{opt}}|^p && \text{(B.26)} \\ &= \sum_{n\in[N]} \max_{b\in\{\pm1\}} \Re\{(b\mathbf{x}_n^\top \mathbf{q}_{\text{opt}})^p\}. && \text{(B.27)} \\ &\geq \sum_{n\in[N]} \Re\left\{(b_n\mathbf{x}_n^\top \mathbf{q})^p\right\}, && \text{(B.28)} \end{aligned}$$

where $b_n \in \{\pm1\}\forall n \in [N]$. Equality is attained if and only if $b_n = \text{sgn}(\mathbf{x}_n^\top \mathbf{q}_{\text{opt}})$ for every $n \in [N]$. □

B.6 Proof of Proposition 3.2

Proof. Let $\mathbb{B}^+_{D\times1} = \{\mathbf{q} \in \mathbb{R}^D : \|\mathbf{q}\|_2 \leq 1, [\mathbf{q}]_i \geq 0 \forall i \in [D]\}$. For any $\mathbf{q} \in \mathbb{B}_{D\times1}$, there exist $\mathbf{a} \in \mathbb{B}^+_{D\times1}$ and $\mathbf{w} \in \{\pm1\}^D$, such that $\mathbf{q} = \mathbf{a} \odot \mathbf{w}$. Accordingly, it holds

$$\begin{aligned} \max_{\mathbf{q}\in\mathbb{B}_{D\times1}} \|\mathbf{X}^\top \mathbf{q}\|_p^p &= \max_{\mathbf{q}\in\mathbb{B}_{D\times1}} \sum_{n\in[N]} |\mathbf{x}_n^\top \mathbf{q}|^p && \text{(B.29)} \\ &= \max_{\mathbf{a}\in\mathbb{B}^+_{D\times1}} \max_{\mathbf{w}\in\{\pm1\}^D} \sum_{n\in[N]} |(\mathbf{x}_n \odot \mathbf{a})^\top \mathbf{w}|^p && \text{(B.30)} \\ &\overset{\text{Lem. 3.6}}{\leq} \max_{\mathbf{a}\in\mathbb{B}^+_{D\times1}} \sum_{n\in[N]} \|(\mathbf{x}_n \odot \mathbf{a})\|_1^p && \text{(B.31)} \\ &= \max_{\mathbf{a}\in\mathbb{B}^+_{D\times1}} \sum_{n\in[N]} |\mathbf{x}_n^\top \mathbf{a}|^p. && \text{(B.32)} \end{aligned}$$

The upper bound is attained by $\mathbf{w} = \pm\mathbf{1}_D$. Moreover, we notice that $\mathbb{B}^+_{D\times1} \subseteq \mathcal{C}(\mathbf{1}_N)$. Accordingly, for non-negative $\mathbf{X}$

$$\max_{\mathbf{q}\in\mathbb{S}_{D\times1}} \|\mathbf{X}^\top \mathbf{q}\|_p^p = \max_{\mathbf{q}\in\mathbb{B}_{D\times1}} \|\mathbf{X}^\top \mathbf{q}\|_p^p = \max_{\mathbf{q}\in\mathcal{C}(\mathbf{1}_N)} \|\mathbf{X}^\top \mathbf{q}\|_p^p. \tag{B.33}$$

□

B.7 Proof of Lemma 3.9

Proof. Consider $\mathbf{B} \in \{\pm 1\}^{N \times K}$ such that $\bar{\mathcal{C}}(\mathbf{B}) \neq \emptyset$. Let $\mathbf{Q}, \mathbf{Q}' \in \bar{\mathcal{C}}(\mathbf{B})$ –i.e., $\text{sgn}(\mathbf{X}^\top \mathbf{Q}) = \text{sgn}(\mathbf{X}^\top \mathbf{Q}') = \mathbf{B}$, $\|\mathbf{Q}\|_2 \leq 1$, and $\|\mathbf{Q}'\|_2 \leq 1$. First, notice that there exists $\mathbf{A} \in \mathbb{R}^{N \times K}$ such that $[\mathbf{A}]_{n,k} > 0 \forall (n,k) \in [N] \times [K]$ and

$$\mathbf{A} \odot \mathbf{X}^\top \mathbf{Q} = \mathbf{X}^\top \mathbf{Q}'. \tag{B.34}$$

Consider $\theta \in [0,1]$ and let $\bar{\mathbf{Q}} = \theta \mathbf{Q} + (1-\theta)\mathbf{Q}'$. It holds that

$$\begin{aligned}
\text{sgn}(\mathbf{X}^\top \bar{\mathbf{Q}}) &= \text{sgn}(\theta \mathbf{X}^\top \mathbf{Q} + (1-\theta)\mathbf{X}^\top \mathbf{Q}') && \text{(B.35)}\\
&= \text{sgn}(\theta \mathbf{X}^\top \mathbf{Q} + (1-\theta)\mathbf{A} \odot \mathbf{X}^\top \mathbf{Q}) && \text{(B.36)}\\
&= \text{sgn}\left(\mathbf{X}^\top \mathbf{Q} \odot (\theta \mathbf{1}_{N \times K} + (1-\theta)\mathbf{A})\right) && \text{(B.37)}\\
&= \text{sgn}(\mathbf{X}^\top \mathbf{Q}) && \text{(B.38)}\\
&= \mathbf{Q}. && \text{(B.39)}
\end{aligned}$$

Moreover, the spectral norm satisfies the norm triangle inequality property. Therefore, it holds

$$\begin{aligned}
\|\bar{\mathbf{Q}}\|_2 =& \|\theta \mathbf{Q} + (1-\theta)\mathbf{Q}'\|_2 && \text{(B.40)}\\
&\leq \|\theta \mathbf{Q}\|_2 + \|(1-\theta)\mathbf{Q}'\|_2 && \text{(B.41)}\\
&= \theta \|\mathbf{Q}\|_2 + (1-\theta)\|\mathbf{Q}'\|_2 && \text{(B.42)}\\
&\leq \theta + (1-\theta) && \text{(B.43)}\\
&= 1. && \text{(B.44)}
\end{aligned}$$

It follows that for any $\mathbf{Q}, \mathbf{Q}' \in \bar{\mathcal{C}}(\mathbf{Q})$ and $\theta \in [0,1]$, $\bar{\mathbf{Q}} = \theta \mathbf{Q} + (1-\theta)\mathbf{Q}' \in \bar{\mathcal{C}}(\mathbf{B})$. Thus, $\bar{\mathcal{C}}(\mathbf{B})$ is a convex set. □

B.8 Proof of Lemma 3.10

Proof. Assume that there exists $k \in [K]$ such that $\mathbf{b} = [\bar{\mathbf{B}}_{\text{opt}}]_{:,k} \notin \mathcal{B}^\star$. Then, $\mathcal{C}(\mathbf{b})$ includes $\mathbf{q} = \mathbf{0}_D$ and $[\bar{\mathbf{Q}}_{\text{opt}}]_{:,k}$ has no contribution in the metric. Substituting $\mathbf{b}$ by any $\mathbf{b}' \in \mathcal{B}^\star$ not in $\bar{\mathbf{B}}_{\text{opt}}$, would offer a larger metric which contradicts that $\bar{\mathbf{Q}}_{\text{opt}}$ is the exact solution. □

B.9 Proof of Proposition 3.4

Proof. Let $\bar{\mathbf{Q}}_{\text{opt}} \in \mathbb{S}_{D\times K}$ be the exact solution to (3.25). Consider $\mathbf{Q}' \in \mathbb{S}_{D\times K}$ and assume that $\|\mathbf{X}^\top \bar{\mathbf{Q}}_{\text{opt}}\|_p^p > \|\mathbf{X}^\top \mathbf{Q}'\|_p^p$. This contradicts that $\bar{\mathbf{Q}}_{\text{opt}}$ is the exact solution to (3.25). Accordingly, there exists no matrix in Stiefel that offers a larger metric in (3.1) compared to $\bar{\mathbf{Q}}_{\text{opt}}$. □

Appendix C

Chapter 4

C.1 Proof of Lemma 4.1

Proof. Let $\mathbf{b}_q \overset{\triangle}{=} \mathbf{y}_q^* \otimes \mathbf{y}_q \; \forall q$ and define auxiliary variables $\dot{\mathbf{v}} = \mathbf{1}_L \otimes \mathbf{s}_L$, $\dot{\mathbf{p}} = \mathbf{p} \otimes \mathbf{1}_L$, $\ddot{\mathbf{v}} = \mathbf{s}_L \otimes \mathbf{1}_L$, and $\ddot{\mathbf{p}} = \mathbf{1}_L \otimes \mathbf{p}$.[1] For any $n \in \{1-L', 2-L', \dots L'-1\}$ and $j_n \in \mathcal{J}_n$, it holds that $[\mathbf{b}_q]_{j_n} = [\mathbf{y}_q^*]_{[\dot{\mathbf{v}}]_{j_n}} [\mathbf{y}_q]_{[\ddot{\mathbf{v}}]_{j_n}}$. Further, it holds

$$[\mathbf{y}_q]_{[\dot{\mathbf{v}}]_{j_n}} = \sum_{k=1}^{K} v(\theta_k)^{[\dot{\mathbf{p}}]_{j_n}} \xi_{q,k} + [\mathbf{n}_q]_{[\dot{\mathbf{v}}]_{j_n}}, \tag{C.1}$$

$$[\mathbf{y}_q]_{[\ddot{\mathbf{v}}]_{j_n}} = \sum_{k=1}^{K} v(\theta_k)^{[\ddot{\mathbf{p}}]_{j_n}} \xi_{q,k} + [\mathbf{n}_q]_{[\ddot{\mathbf{v}}]_{j_n}}. \tag{C.2}$$

Next, we compute $\mathbb{E}\{[\mathbf{b}_q]_{j_n}\} = \sum_{k=1}^{K} v(\theta_k)^{\omega_{j_n}} d_k + \sigma^2 [\mathbf{i}_L]_{j_n} = [\mathbf{r}]_{j_n}$,[2] which implies that $\mathbb{E}\{\hat{\mathbf{r}}\} = \mathbf{r}$. Then, for every $n \in \{1-L', 2-L', \dots, L'-1\}$ and $(i,j) \in \mathcal{J}_n$, we define $b^{(n)}_{p,q,i,j} \overset{\triangle}{=} [\mathbf{b}_q^*]_i [\mathbf{b}_p]_j$. After simple algebraic manipulations we find that

$$\mathbb{E}\left\{b^{(n)}_{p,q,i,j}\right\} = \left|\mathbf{g}_n^H \mathbf{d} + \delta(n)\sigma^2\right|^2 + \delta(q-p)\left|\mathbf{z}_{i,j}^H \mathbf{d} + \delta(i-j)\sigma^2\right|^2, \tag{C.3}$$

[1] Recall that for any $x \in \mathbb{N}_+$, $\mathbf{s}_x = [1, 2, \dots, x]^\top$.

[2] Recall that for any $i \in \{1, 2, \dots, L^2\}$, $\omega_i = [\dot{\mathbf{p}}]_i - [\ddot{\mathbf{p}}]_i$.

where $\mathbf{g}_n \overset{\triangle}{=} [v(\theta_1)^n,\ v(\theta_2)^n,\ \ldots,\ v(\theta_K)^n]^\top$. Next, we proceed as follows.

$$\begin{aligned}
e &= \mathbb{E}\left\{|[\mathbf{r}]_{j_n} - [\hat{\mathbf{r}}]_{j_n}|^2\right\} && \text{(C.4)}\\
&= |[\mathbf{r}]_{j_n}|^2 + \mathbb{E}\left\{[\hat{\mathbf{r}}]_{j_n}[\hat{\mathbf{r}}]^*_{j_n}\right\} - 2\mathbb{E}\left\{\Re\left\{[\mathbf{r}]_{j_n}[\hat{\mathbf{r}}]^*_{j_n}\right\}\right\} && \text{(C.5)}\\
&= \mathbb{E}\left\{[\hat{\mathbf{r}}]_{j_n}[\hat{\mathbf{r}}]^*_{j_n}\right\} - |[\mathbf{r}]_{j_n}|^2 && \text{(C.6)}\\
&= \mathbb{E}\left\{\frac{1}{Q^2}\sum_{q=1}^{Q}\sum_{p=1}^{Q}[\mathbf{b}_q]_{j_n}[\mathbf{b}_p]^*_{j_n}\right\} - |[\mathbf{r}]_{j_n}|^2 && \text{(C.7)}\\
&= \frac{1}{Q^2}\sum_{q=1}^{Q}\sum_{p=1}^{Q}\mathbb{E}\left\{b^{(n)}_{q,p,j_n,j_n}\right\} - |[\mathbf{r}]_{j_n}|^2 && \text{(C.8)}\\
&\overset{\text{(C.3)}}{=} \frac{(\mathbf{1}_K^\top\mathbf{d} + \sigma^2)^2}{Q}. && \text{(C.9)}
\end{aligned}$$

□

C.2 Proof of Lemma 4.2

Proof. By Lemma 4.1, $\text{err}_r(\hat{\mathbf{r}}_{\text{sel}}) = \mathbb{E}\{\|\mathbf{r}_{\text{co}} - \hat{\mathbf{r}}_{\text{sel}}\|_2^2\} = \sum_{n=1-L'}^{L'-1}\mathbb{E}\{|[\mathbf{r}]_{j_n} - [\hat{\mathbf{r}}]_{j_n}|^2\} = (2L'-1)e$. □

C.3 Proof of Proposition 4.1

Proof. Notice that $\mathbf{Z} = \mathbf{F}(\mathbf{I}_{L'} \otimes \mathbf{r}_{\text{co}}) = [\mathbf{F}_1\mathbf{r}_{\text{co}},\ \mathbf{F}_2\mathbf{r}_{\text{co}},\ \ldots,\ \mathbf{F}_{L'}\mathbf{r}_{\text{co}}]$. By definition, $\mathbf{F}_m$ is a selection matrix that selects the $\{L'-(m-1), L'-(m-2), \ldots, 2L'-m\}$th entries of the length-$(2L'-1)$ vector it multiplies, for every $m \in \{1, 2, \ldots, L'\}$. That is, $\mathbf{F}_m\mathbf{r}_{\text{co}} = [\mathbf{r}_{\text{co}}]_{L'-(m-1):2L'-m}$. Similarly, $\hat{\mathbf{Z}}_{\text{sel}} =$

$[\mathbf{F}_1\hat{\mathbf{r}}_{\text{sel}},\ \mathbf{F}_2\hat{\mathbf{r}}_{\text{sel}},\ \ldots,\ \mathbf{F}_{L'}\hat{\mathbf{r}}_{\text{sel}}]$ with $\mathbf{F}_m\hat{\mathbf{r}}_{\text{sel}} = [\hat{\mathbf{r}}_{\text{sel}}]_{L'-(m-1):2L'-m}$. In view of the above,

$$\begin{aligned}
\mathrm{err}_Z(\hat{\mathbf{Z}}_{\text{sel}}) &= \mathbb{E}\left\{\left\|\mathbf{Z} - \hat{\mathbf{Z}}_{\text{sel}}\right\|_F^2\right\} && \text{(C.10)}\\
&= \mathbb{E}\left\{\sum_{m=1}^{L'} \|\mathbf{F}_m\mathbf{r}_{\text{co}} - \mathbf{F}_m\hat{\mathbf{r}}_{\text{sel}}\|_2^2\right\} && \text{(C.11)}\\
&= \sum_{m=1}^{L'} \mathbb{E}\left\{\left\|[\mathbf{r}_{\text{co}}]_{L'-(m-1):2L'-m} - [\hat{\mathbf{r}}_{\text{sel}}]_{L'-(m-1):2L'-m}\right\|_2^2\right\} && \text{(C.12)}\\
&= \sum_{m=1}^{L'} \sum_{n=1-m}^{L'-m} \mathbb{E}\left\{|[\mathbf{r}_{\text{co}}]_{L'+n} - [\hat{\mathbf{r}}_{\text{sel}}]_{L'+n}|^2\right\} && \text{(C.13)}\\
&= \sum_{m=1}^{L'} \sum_{n=1-m}^{L'-m} e && \text{(C.14)}\\
&= L'^2 e. && \text{(C.15)}
\end{aligned}$$

□

C.4 Proof of Lemma 4.3

Proof.

$$\begin{aligned}
e_n &= \mathbb{E}\left\{\left|[\mathbf{r}]_{j_n} - \frac{1}{|\mathcal{J}_n|}\sum_{j\in\mathcal{J}_n}[\hat{\mathbf{r}}]_j\right|^2\right\} && \text{(C.16)}\\
&= |[\mathbf{r}]_{j_n}|^2 + \frac{1}{|\mathcal{J}_n|^2}\mathbb{E}\left\{\left|\sum_{j\in\mathcal{J}_n}[\hat{\mathbf{r}}]_j\right|^2\right\} - 2\mathbb{E}\left\{\Re\left\{[\mathbf{r}]_{j_n}^*\left(\frac{1}{|\mathcal{J}_n|}\sum_{j\in\mathcal{J}_n}[\hat{\mathbf{r}}]_j\right)\right\}\right\} && \text{(C.17)}\\
&= |[\mathbf{r}]_{j_n}|^2 + \frac{1}{|\mathcal{J}_n|^2}\sum_{j\in\mathcal{J}_n}\sum_{i\in\mathcal{J}_n}\frac{1}{Q^2}\sum_{q=1}^{Q}\sum_{p=1}^{Q}\mathbb{E}\left\{b_{q,p,i,j}^{(n)}\right\} - 2\Re\left\{[\mathbf{r}]_{j_n}^*\left(\frac{1}{|\mathcal{J}_n|}\sum_{j\in\mathcal{J}_n}[\mathbf{r}]_j\right)\right\} && \text{(C.18)}\\
&\overset{\text{(C.3)}}{=} \frac{1}{Q}\left(\frac{2\sigma^2\mathbf{1}_K^\top\mathbf{d} + \sigma^4}{|\mathcal{J}_n|} + \sum_{i\in\mathcal{J}_n}\sum_{j\in\mathcal{J}_n}\frac{|\mathbf{z}_{i,j}^H\mathbf{d}|^2}{|\mathcal{J}_n|^2}\right). && \text{(C.19)}
\end{aligned}$$

□

C.5 Proof of Lemma 4.4

Proof. By Lemma 4.3, $\mathrm{err}_r(\hat{\mathbf{r}}_{\mathrm{avg}}) = \mathbb{E}\{\|\mathbf{r}_{co} - \hat{\mathbf{r}}_{\mathrm{avg}}\|_2^2\} = \sum_{n=1-L'}^{L'-1} \mathbb{E}\{|[\mathbf{r}]_{j_n} - \frac{1}{|\mathcal{J}_n|}\sum_{j\in\mathcal{J}_n}[\hat{\mathbf{r}}]_j|^2\} = \sum_{n=1-L'}^{L'-1} e_n$. □

C.6 Proof of Proposition 4.2

Proof. We know that $\mathbf{Z} = [\mathbf{F}_1\mathbf{r}_{\mathrm{co}},\ \mathbf{F}_2\mathbf{r}_{\mathrm{co}},\ \dots\ \mathbf{F}_{L'}\mathbf{r}_{\mathrm{co}}]$. Thus, $\hat{\mathbf{Z}}_{\mathrm{avg}} = [\mathbf{F}_1\hat{\mathbf{r}}_{\mathrm{avg}},\ \mathbf{F}_2\hat{\mathbf{r}}_{\mathrm{avg}},\ \dots\ \mathbf{F}_{L'}\hat{\mathbf{r}}_{\mathrm{avg}}]$. By the definition of $\mathbf{F}_m$, for every $m \in \{1,2,\dots,L'\}$ it holds that $\mathbf{F}_m\mathbf{r}_{\mathrm{co}} = [\mathbf{r}_{\mathrm{co}}]_{L'-(m-1):2L'-m}$ and $\mathbf{F}_m\hat{\mathbf{r}}_{\mathrm{avg}} = [\hat{\mathbf{r}}_{\mathrm{avg}}]_{L'-(m-1):2L'-m}$. In view of the above,

$$\begin{aligned}
\mathrm{err}_Z(\hat{\mathbf{Z}}_{\mathrm{avg}}) &= \mathbb{E}\left\{\left\|\mathbf{Z} - \hat{\mathbf{Z}}_{\mathrm{avg}}\right\|_F^2\right\} && \text{(C.20)}\\
&= \mathbb{E}\left\{\sum_{m=1}^{L'} \|\mathbf{F}_m\mathbf{r}_{\mathrm{co}} - \mathbf{F}_m\hat{\mathbf{r}}_{\mathrm{avg}}\|_2^2\right\} && \text{(C.21)}\\
&= \sum_{m=1}^{L'} \mathbb{E}\left\{\left\|[\mathbf{r}_{\mathrm{co}}]_{L'-(m-1):2L'-m} - [\hat{\mathbf{r}}_{\mathrm{avg}}]_{L'-(m-1):2L'-m}\right\|_2^2\right\} && \text{(C.22)}\\
&= \sum_{m=1}^{L'} \sum_{n=1-m}^{L'-m} \mathbb{E}\left\{\left|[\mathbf{r}_{\mathrm{co}}]_{L'+n} - [\hat{\mathbf{r}}_{\mathrm{avg}}]_{L'+n}\right|^2\right\} && \text{(C.23)}\\
&= \sum_{m=1}^{L'} \sum_{n=1-m}^{L'-m} e_n. && \text{(C.24)}
\end{aligned}$$

□

C.7 Proof of Lemma 4.5

We recall that $\mathbf{w} = \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{x}_q^* \otimes \mathbf{x}_q$. Next, we notice that by utilizing the auxiliary variables[3] $\dot{\mathbf{u}} = \mathbf{1}_{K+L} \otimes \mathbf{s}_{K+L}$ and $\ddot{\mathbf{u}} = \mathbf{s}_{K+L} \otimes \mathbf{1}_{K+L}$, we obtain $[\mathbf{w}]_i = \frac{1}{Q}\sum_{q=1}^{Q}[\mathbf{x}_q^*]_{[\ddot{\mathbf{u}}]_i}[\mathbf{x}_q]_{[\dot{\mathbf{u}}]_i}$. Then, we define $\mathcal{I} \overset{\triangle}{=} \{i \in \{1,2,\dots,(K+L)^2\} \ : \ [\mathbf{i}]_i = 1\}$. We observe that $\mathbb{E}\{[\mathbf{x}_q^*]_{[\ddot{\mathbf{u}}]_i}[\mathbf{x}_q]_{[\dot{\mathbf{u}}]_i}\} = \delta([\dot{\mathbf{u}}]_i - [\ddot{\mathbf{u}}]_i) = 1$ if

[3] Recall that for any $x \in \mathbb{N}_+$, $\mathbf{s}_x = [1,2,\dots,x]^\top$.

$i \in \mathcal{I}$ and 0 if $i \notin \mathcal{I}$. The latter implies

$$\mathbb{E}\{\mathbf{w}\} = \mathbf{i}. \tag{C.25}$$

Next, for $(i, m) \in \{1, 2, \dots, (K+L)^2\}^2$, we define

$$\eta_{i,m} \stackrel{\triangle}{=} [\mathbf{x}_q^*]_{[\ddot{\mathbf{u}}]_i} [\mathbf{x}_q]_{[\dot{\mathbf{u}}]_i} [\mathbf{x}_p^*]_{[\dot{\mathbf{u}}]_m} [\mathbf{x}_p]_{[\ddot{\mathbf{u}}]_m}. \tag{C.26}$$

It holds $[\mathbf{w}]_i [\mathbf{w}^*]_m = \frac{1}{Q^2} \sum_{q=1}^{Q} \sum_{p=1}^{Q} \eta_{i,m}$. By the 2nd and 4th order moments of zero-mean independent normal variables, we find that $\mathbb{E}\{\eta_{i,m}\}$ is equal to $1 + \delta(p-q)\delta(i-m)$ if $(i,m) \in \mathcal{I}$ and 0 otherwise. The latter implies that $\mathbb{E}\{[\mathbf{w}]_i [\mathbf{w}^*]_m\} = \frac{1}{Q^2} \sum_{q=1}^{Q} \sum_{p=1}^{Q} \mathbb{E}\{\eta_{i,m}\}$ is equal to $1 + \frac{1}{Q}\delta(i,m)$, if $(i,m) \in \mathcal{I}$ and 0 otherwise. Altogether, we have

$$\mathbb{E}_{\mathbf{w}}\{\mathbf{w}\mathbf{w}^H\} = \mathbf{i}\mathbf{i}^\top + \frac{1}{Q}\mathbf{I}_{K+L}. \tag{C.27}$$

C.8 Proof of Lemma 4.6.

Proof. For $(i, m) \in \{1, 2, \dots, L^2\}^2$, $[\mathbf{H}]_{i,m} = [\mathbf{V}\mathbf{w}]_i [(\mathbf{V}\mathbf{i})^*]_m = \sum_{j=1,l=1}^{(K+L)^2} [\mathbf{V}]_{i,j} [\mathbf{w}]_j [\mathbf{V}^*]_{m,l} [\mathbf{i}]_l$. Accordingly, $[\mathbf{H}_{\mathbb{E}}]_{i,m} = \sum_{j=1,l=1}^{(K+L)^2} \mathbb{E}_{\Theta}\mathbb{E}_{\mathbf{w}}\{[\mathbf{V}]_{i,j} [\mathbf{w}]_j [\mathbf{V}]_{m,l}^* [\mathbf{i}]_l\}$. Considering that the random variables $\mathbf{w}$ and $\mathbf{V}$ are statistically independent from each other and that $\mathbb{E}_{\mathbf{w}}\{\mathbf{w}\} = \mathbf{i}$ (see Lemma 4.5), we obtain $[\mathbf{H}_{\mathbb{E}}]_{i,m} = \mathbb{E}_{\Theta}\{[\mathbf{V}\mathbf{i}]_i [(\mathbf{V}\mathbf{i})^*]_m\} = \mathbb{E}_{\Theta}\{[\mathbf{r}]_i [\mathbf{r}^*]_m\}$. Then, we substitute[4]

$$[\mathbf{r}]_i = \sum_{k=1}^{K} v(\theta_k)^{\omega_i} d_k + \sigma^2 [\mathbf{i}_L]_i \tag{C.28}$$

in $\mathbb{E}_{\Theta}\{[\mathbf{r}]_i [\mathbf{r}^*]_m\}$. After plain algebraic operations, we obtain

$$[\mathbf{H}_{\mathbb{E}}]_{i,m} = \|\mathbf{d}\|_2^2 \mathcal{I}(\omega_i - \omega_m) + \sigma^4 \delta(\omega_i)\delta(\omega_m) + \sigma^2 \left(\mathbf{1}_K^\top \mathbf{d}\right) \left(\delta(\omega_i)\mathcal{I}(-\omega_m) + \mathcal{I}(\omega_i)\delta(\omega_m)\right) \tag{C.29}$$

$$+ \mathcal{I}(\omega_i)\mathcal{I}(-\omega_m) \left((\mathbf{1}_K^\top \mathbf{d})^2 - \|\mathbf{d}\|_2^2\right). \tag{C.30}$$

□

[4] Recall that for any $i \in \{1, 2, \dots, L^2\}$, $\omega_i = [\dot{\mathbf{p}}]_i - [\ddot{\mathbf{p}}]_i$.

C.9 Proof of Lemma 4.7.

Proof. For $(i, m) \in \{1, 2, \dots, L^2\}^2$ it holds

$$[\mathbf{G}]_{i,m} = [\mathbf{V}\mathbf{w}]_i[(\mathbf{V}\mathbf{w})^*]_m = \sum_{j=1,l=1}^{(K+L)^2} [\mathbf{V}]_{i,j}[\mathbf{w}]_j[\mathbf{V}^*]_{m,l}[\mathbf{w}^*]_l. \tag{C.31}$$

Accordingly, $[\mathbf{G}_{\mathbb{E}}]_{i,m} = \sum_{j=1,l=1}^{(K+L)^2} \underset{\Theta\mathbf{w}}{\mathbb{E}\mathbb{E}}\{[\mathbf{V}]_{i,j}[\mathbf{w}]_j[\mathbf{V}^*]_{m,l}[\mathbf{w}^*]_l\}$. Next, we recall that the random variables Θ and $\mathbf{w}$ are statistically independent from each other. Thus,

$$[\mathbf{G}_{\mathbb{E}}]_{i,m} = \sum_{j=1,l=1}^{(K+L)^2} \underset{\Theta}{\mathbb{E}}\{[\mathbf{V}]_{i,j}\underset{\mathbf{w}}{\mathbb{E}}\{[\mathbf{w}]_j[\mathbf{w}^*]_l\}[\mathbf{V}^*]_{m,l}\}. \tag{C.32}$$

The latter is equivalent to $\mathbf{G}_{\mathbb{E}} = \underset{\Theta}{\mathbb{E}}\{\mathbf{V}\underset{\mathbf{w}}{\mathbb{E}}\{\mathbf{w}\mathbf{w}^H\}\mathbf{V}^H\}$. By Lemma 4.5, we obtain $\mathbf{G}_{\mathbb{E}} = \underset{\Theta}{\mathbb{E}}\{\mathbf{V}\mathbf{i}\mathbf{i}^\top\mathbf{V}^H + \frac{1}{Q}\mathbf{V}\mathbf{V}^H\}$. By Lemma 4.6, we find that $\mathbf{G}_{\mathbb{E}} = \mathbf{H}_{\mathbb{E}} + \frac{1}{Q}\tilde{\mathbf{V}}_{\mathbb{E}}$. □

C.10 Proof of Remark 4.3

Proof.

$$\begin{aligned}
\|\mathbf{T}\|_F^2 &= \sum_{i=1-D}^{D-1} \|\mathbf{d}_i(\mathbf{T})\|_2^2 &\text{(C.33)}\\
&= \sum_{i=1-D}^{D-1} \|\frac{1}{D-|i|}(\mathbf{1}_{D-|i|}^\top \mathbf{d}_i(\mathbf{X}))\mathbf{1}_{D-|i|}\|_2^2 &\text{(C.34)}\\
&= \sum_{i=1-D}^{D-1} \frac{(\mathbf{1}_{D-|i|}^\top \mathbf{d}_i(\mathbf{X}))^2}{(D-|i|)^2}\|\mathbf{1}_{D-|i|}\|_2^2 &\text{(C.35)}\\
&= \sum_{i=1-D}^{D-1} \frac{(\mathbf{1}_{D-|i|}^\top \mathbf{d}_i(\mathbf{X}))^2}{D-|i|} &\text{(C.36)}\\
&\leq \sum_{i=1-D}^{D-1} \frac{\|\mathbf{1}_{D-|i|}\|_2^2\|\mathbf{d}_i(\mathbf{X})\|_2^2}{D-|i|} &\text{(C.37)}\\
&= \|\mathbf{X}\|_F^2. &\text{(C.38)}
\end{aligned}$$

□

C.11 Proof of Remark 4.5

Proof. Consider Hermitian $\mathbf{T} \in \mathbb{C}^{D\times D}$ with EVD $\mathbf{T} = \mathbf{U}\text{diag}(\boldsymbol{\lambda}_T)\mathbf{U}^H$, $\mathbf{U}$ is unitary (i.e., $\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}_D$). Let $\mathbf{H} = \mathbf{U}^H\mathbf{P}_0\mathbf{U}$ which implies that $\mathbf{P}_0 = \mathbf{U}\mathbf{H}\mathbf{U}^H$. It holds

$$\begin{aligned}
\min_{\mathbf{P}_0\in\mathbb{S}_+^D} \|\mathbf{T}-\mathbf{P}_0\|_F^2 &= \min_{\mathbf{H}\in\mathbb{C}^{D\times D}} \|\text{diag}(\boldsymbol{\lambda}_T)-\mathbf{H}\|_F^2 && \text{(C.39)}\\
&= \min_{\mathbf{H}\in\mathbb{C}^{D\times D}} \sum_{\substack{i,j\\ i\neq j}}[\mathbf{H}]_{i,j}^2 + \sum_{i=1}^{D}([\boldsymbol{\lambda}_T]_i - [\mathbf{H}]_{i,i})^2 && \text{(C.40)}\\
&\geq \sum_{\substack{i,j\\ i\neq j}}[\mathbf{H}]_{i,j}^2 + \sum_{i=1}^{D}([\boldsymbol{\lambda}_T]_i - [\mathbf{H}]_{i,i}) && \text{(C.41)}\\
&\geq \sum_{i\in\{1,2,\dots,D|[\boldsymbol{\lambda}_T]_i<0\}} [\boldsymbol{\lambda}_T]_i^2. && \text{(C.42)}
\end{aligned}$$

Similar to [215], the lower bound in (C.42) is attained by matrix $\mathbf{H} = \text{diag}(\boldsymbol{\lambda}_P)$ for $\boldsymbol{\lambda}_P$ such that $[\boldsymbol{\lambda}_P]_i = \max\{[\boldsymbol{\lambda}_T]_i, 0\}$. □

C.12 Proof of Remark 4.7

Proof.

$$\begin{aligned}
\|\mathbf{A}\|_F^2 &= \sum_{i=1}^{D}[\boldsymbol{\lambda}_A]_i^2 && \text{(C.43)}\\
&= \sum_{i=1}^{D-\rho}[\boldsymbol{\lambda}_P]_i^2 + \rho(\frac{1}{\rho}\sum_{j=D-\rho+1}^{D}[\boldsymbol{\lambda}_P]_j)^2 && \text{(C.44)}\\
&\leq \sum_{i=1}^{D-\rho}[\boldsymbol{\lambda}_P]_i^2 + \sum_{j=D-\rho+1}^{D}[\boldsymbol{\lambda}_P]_j^2 && \text{(C.45)}\\
&= \|\mathbf{P}\|_F^2. && \text{(C.46)}
\end{aligned}$$

□