



HAL
open science

Estimation of structured tensor models and recovery of low-rank tensors

José Henrique de M Goulart

► **To cite this version:**

José Henrique de M Goulart. Estimation of structured tensor models and recovery of low-rank tensors. Other. Université Côte d'Azur, 2016. English. NNT : 2016AZUR4147 . tel-01466776

HAL Id: tel-01466776

<https://theses.hal.science/tel-01466776>

Submitted on 13 Feb 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ECOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Unité de recherche: Laboratoire d'Informatique,
Signaux et Systèmes de Sophia Antipolis (I3S)

Thèse de Doctorat

Présentée en vue de l'obtention du grade de

Docteur en Sciences

de l'UNIVERSITE COTE D'AZUR

Mention: Automatique, traitement du signal et des images

présentée et soutenue par

José Henrique DE MORAIS GOULART

Estimation of structured tensor models and recovery of low-rank tensors

Thèse dirigée par Gérard FAVIER

Soutenue le 15 décembre 2016

Devant le jury composé de:

M. Pierre COMON	Université Grenoble Alpes	<i>Examineur</i>
M. Gérard FAVIER	Université de Nice-Sophia Antipolis	<i>Directeur de thèse</i>
M. Cédric FEVOTTE	Institut de Recherche en Informatique de Toulouse	<i>Rapporteur</i>
M. Eric MOREAU	Université de Toulon	<i>Rapporteur</i>
M. Vicente ZARZOSO	Université de Nice-Sophia Antipolis	<i>Président</i>

*To my beloved wife,
Luana Batista Goulart*

Abstract

Tensor-based methods are by now well-established tools in many scientific and engineering domains, due to their capability of exploiting additional structure in comparison with matrix-based ones. Two quite common tasks in applications are the computation of a canonical polyadic decomposition (CPD) and the recovery of a tensor of interest from a reduced number of linear measurements under a low-rank assumption, known as low-rank tensor recovery (LRTR). General iterative methods exist for CPD computation, but they often suffer from slow convergence and are prone to delivering inaccurate estimates. With respect to LRTR, no approach has yet been proven optimal in terms of sampling requirements.

In the first part of this thesis, we formulate two methods for computing a CPD having linearly structured (e.g., Toeplitz or banded) matrix factors: a general constrained alternating least squares (CALS) algorithm and an algebraic solution for the case where all factors are circulant. Exact and approximate versions of the former method are studied. The latter method relies on a multidimensional discrete-time Fourier transform of the target tensor, which leads to a system of homogeneous monomial equations whose resolution provides the desired circulant factors. Our simulations show that combining these approaches yields a statistically efficient estimator, which is also true for other combinations of CALS in scenarios involving non-circulant factors.

The second part of the thesis concerns LRTR and, in particular, the tensor completion (TC) problem. We propose an efficient algorithm based on multilinear rank, called SeMPIHT, which employs sequentially optimal modal projections as its hard thresholding operator. Then, a performance bound is derived under usual restricted isometry conditions, which however yield suboptimal sampling bounds. Nevertheless, our simulations suggest SeMPIHT obeys optimal sampling bounds for Gaussian measurements. Step size selection and gradual rank increase heuristics are also elaborated in order to improve performance. In addition to SeMPIHT, we devise an imputation scheme for TC based on soft thresholding of a Tucker model core, named IFHST, relying on characteristics typically possessed by real-world tensors. The utility of such a scheme is illustrated in the completion of a road traffic data tensor acquired by an intelligent transportation system.

Keywords: canonical polyadic decomposition, structured matrices, structured tensors, alternating least-squares, circulant matrices, homogeneous monomial equations, low-rank tensor recovery, tensor completion, iterative hard thresholding, intelligent transportation system.

Résumé

Les méthodes basées sur les tenseurs sont aujourd'hui très utilisées dans différents domaines scientifiques et d'ingénierie, en raison de leur capacité à mieux exploiter l'aspect multidimensionnel des données et leur structure sous-jacente par rapport aux méthodes matricielles. Deux tâches importantes sont le calcul d'une décomposition polyadique canonique (CPD) et la récupération d'un tenseur de rang faible (LRTR) à partir d'un nombre réduit de mesures linéaires. Cependant, les méthodes itératives générales de calcul de CPD souvent convergent lentement et/ou produisent des estimées imprécises, et aucune approche au problème LRTR n'a à ce jour été démontrée optimale vis-à-vis du nombre de mesures nécessaires.

Dans la première partie de cette thèse, on formule deux méthodes pour le calcul d'une décomposition polyadique canonique ayant des facteurs matriciels structurés dans un sens linéaire (e.g., des facteurs de Toeplitz ou en bande): un algorithme général de moindres carrés alternés (CALS) et une solution algébrique dans le cas où tous les facteurs sont circulants. Des versions exacte et approchée de la méthode CALS sont étudiées. La solution algébrique fait appel à la transformée de Fourier multidimensionnelle à temps discret du tenseur considéré, ce qui conduit à un système d'équations monomiales homogènes dont la résolution fournit les facteurs circulants désirés. Nos simulations montrent que la combinaison de ces approches fournit un estimateur statistiquement efficace, ce qui reste vrai pour d'autres combinaisons de CALS dans des scénarios impliquant des facteurs non-circulants.

La seconde partie de la thèse porte sur le problème LRTR et, en particulier, celui de reconstruction tensorielle (TC). On propose un algorithme efficace basé sur le rang multilinéaire, noté SeMPIHT, qui emploie des projections séquentiellement optimales par mode comme opérateur de seuillage dur. Puis, une borne de performance est dérivée sous des conditions d'isométrie restreinte habituelles, ce qui fournit des bornes d'échantillonnage sous-optimales. Néanmoins, nos simulations suggèrent que SeMPIHT obéit à des bornes d'échantillonnage optimales pour des mesures Gaussiennes. Des heuristiques de sélection du pas et d'augmentation graduelle du rang sont aussi élaborées dans le but d'améliorer la performance. On propose aussi un schéma d'imputation pour TC basé sur un seuillage doux du cœur d'un modèle de Tucker, appelé IFHST, et on illustre son utilité pour la récupération de données réelles de trafic routier acquises par un système de transport intelligent.

Mots-clés: décomposition polyadique canonique, matrices structurées, tenseurs structurés, moindres carrés alternés, matrices circulantes, équations monomiales homogènes, récupération de tenseurs de rang faible, reconstruction tensorielle, seuillage dur itératif, système de transport intelligent.

Acknowledgments

I feel indebted to many people who had a direct participation or were somehow involved in this fruitful and long journey which I went through since I arrived in France.

First, I would like to express my very deep gratitude towards my advisor, professor Gérard Favier, who always trusted in my potential, giving me many opportunities for developing my skills, and with whom I learned very much. He was also a very generous person, offering me help in many times I needed.

I wish to thank also Eric Moreau, Cédric Févotte, Vicente Zarzoso and Pierre Comon for taking part in my thesis committee. I especially appreciate the efforts of Eric Moreau and Cédric Févotte, who as reviewers of my thesis gave valuable comments on my work and on how it could be improved.

I was fortunate to have the opportunity of discussing very interesting topics with many colleagues from I3S during my PhD, such as Vicente Zarzoso, Laure Blanc-Féraud, Emmanuel Soubies, Marc Antonini and Jerome Lebrun. I am grateful for the feedback they gave and questions they asked me about my work, which allowed me to better understand it and to improve it.

The Équipe SIS of I3S laboratory always gave me strong support, offering several possibilities for my development as a researcher. I thus wish to thank them, especially the team leaders Vicente Zarzoso and Sylvie Icart.

Collaborating with Pierre Comon, Maxime Boizard and Rémy Boyer was also an enriching experience, involving several interesting exchanges which allowed me to learn a lot.

I would like to thank also my former advisors, professors Phillip Burt, Leila Maciel de Almeida e Silva and Leonardo Nogueira Matos, who have encouraged me to pursue my PhD degree abroad. Everything I learned from them was very important in my formation.

The friends I made during this period in Nice were an essential part of this experience, for all the conversations, trips and laughs we had together. I thus would like to thank Leandro Ronchini Ximenes, Neetya Shrestha, Youssef Alj, Fernando Ireta Muñoz, Sergio Livi, Samuel Valduga, Nguyen Lam Hung, Howard Mahé, Arnaud Tanguy, Myriana Rifai, Jean-Marie Kai, Dino Lopez and Emmanuel Soubies.

I owe a very special “thank you” to my beloved wife, Luana Batista Goulart, for all the love and care that she always gives me in the most gracious way. Her support and partnership in my life fills me with joy and thankfulness.

Enfim, tenho muito a agradecer a toda minha família por todo o carinho e apoio que sempre me deram. Aos meus pais, especialmente, por terem sempre me incentivado a buscar minhas conquistas.

List of Figures

3.1	Block-diagram of the Wiener-Hammerstein model.	56
5.1	Scenario 1 of Section 5.1.1.1: empirical c.d.f. of the NSE attained by CALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.	91
5.2	Scenario 2 of Section 5.1.1.1: empirical c.d.f. of the NSE attained by CALS (top) and of the overall computing time it takes (bottom) under varying initial conditions and SNR = 50 dB.	92
5.3	Scenario 1 of Section 5.1.1.2: empirical c.d.f. of the NSE attained by SCALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.	93
5.4	Scenario 2 of Section 5.1.1.2: empirical c.d.f. of the NSE attained by SCALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.	94
5.5	Scenario 1 of Section 5.1.2: empirical c.d.f. of the NSE attained by CALS, under varying initial conditions, in the estimation of a non-symmetric SCPD.	95
5.6	Scenario 2 of Section 5.1.2: empirical c.d.f. of the NSE attained by SCALS, under varying initial conditions, in the estimation of a symmetric SCPD.	95
5.7	Scenario 1 of Section 5.2.3: estimated BMSE of several SCPD estimators and corresponding ECRB.	99
5.8	Scenario 2 of Section 5.2.3: estimated BMSE of several SCPD estimators and corresponding ECRB.	101
5.9	Symmetric setting of Section 5.2.4: estimated BMSE of several SCPD estimators and corresponding ECRB.	103
5.10	Wiener-Hammerstein identification scenario of Section 5.2.5: estimated BMSE of several SCPD estimators and corresponding ECRB.	104
7.1	Cost function value $J(\mathbf{X}_k)$ resulting from different choices of step size in one iteration of TIHT: μ_T denotes the step given by (7.16); while μ_N denotes the step given by (7.17). In this example, $\alpha = 0.5$ and $\beta = 0.7$. A similar behavior is observed for SeMPIHT.	137
7.2	Diagram of SeMPIHT algorithm with GRI heuristic.	138
7.3	Typical behavior of the considered random tensor models: (a) modal spectra; (b) row space coherence (see (7.55)).	139

7.4	Recovery performance of SeMPIHT with Gaussian sensing. The approximate lower bound in (b)–(d) is the average of $\text{NSE}(\mathcal{S}_r(\mathcal{X}^*); \mathcal{X}^*)$	141
7.5	Recovery performance of SeMPIHT with sampling operators (TC). The approximate lower bound in (b)–(d) is given by $\text{NMSE}(\mathcal{S}_r(\mathcal{X}^*); \mathcal{X}^*)$	142
7.6	Estimated (normalized) number of DOF which can be recovered by SeMPIHT for each level of ρ , using Gaussian MOs (GO) and SOs. Recovery was successful in 15 consecutive realizations for values of $\bar{\Phi}$ below or over the curve.	143
7.7	Estimated (normalized) number of DOF which can be recovered for each level of ρ , using (a) Gaussian MOs and (b) SOs. The target tensors are $20 \times 20 \times 20$. Recovery was successful in 15 consecutive realizations for values of $\bar{\Phi}$ below or over the curve.	144
7.8	Convergence of several algorithms in a Gaussian sensing scenario where $\rho = 0.25$ and $N_1 = N_2 = N_3 = 20$: (a) T1 tensors of mrank $\mathbf{r} = (3, 3, 3)$; (b) T2 tensors ($\varphi = 2.5$) modeled with mrank $\mathbf{r} = (9, 9, 9)$; (c) T2 tensors ($\varphi = 2.5$) modeled with mrank $\mathbf{r} = (13, 13, 13)$	145
7.9	Convergence of several algorithms in a TC scenario where $\rho = 0.2$ and $N_1 = N_2 = N_3 = 100$: (a) T1 tensors of mrank $\mathbf{r} = (10, 10, 10)$; (b) T2 tensors ($\varphi = 2$) modeled with mrank $\mathbf{r} = (30, 30, 30)$	146
8.1	Compressibility of the HOSVD core of T2 tensors for different spectral decay rates φ : (a) average sorted absolute values of its components; (b) empirical c.d.f. of $\Gamma(\text{vec}(\mathbf{S}))$. The minimum value of 1 is indicated by a red dashed line in (b).	153
8.2	Compressibility of the HOSVD core of tensors with varying number of modes having fast decaying singular spectra: (a) average sorted absolute values of its components; (b) empirical c.d.f. of $\Gamma(\text{vec}(\mathbf{S}))$; (c)–(e): average modal spectra of generated tensors. The minimum value of 1 is indicated by a red dashed line in (b).	154
8.3	Interpretation of Algorithm 8.1 (with null initial conditions) as a feedback control system.	159
8.4	Acceleration of convergence of IFHHT due to first-order feedback control mechanism with time constant ψ	161
8.5	Evolution of a reconstructed component of the target tensor during a run of IFHHT, for several values of ψ	162
8.6	Acceleration of convergence of IFHST due to feedback mechanism with forgetting factor ψ	162
8.7	Evolution of a reconstructed component of the target tensor during a run of IFHST, for several values of ψ	163

8.8	Evolution of (a) NSE and (b) sparsity of the thresholded HOSVD core along a run of the IFHST algorithm with $\psi = 0.85$ and $\eta_0 = 1.2$	164
8.9	Estimated (normalized) number of DOF which can be recovered by several algorithms for each level of ρ , using sampling MOs. The measured third-order tensors have dimensions $N_p = 20$ for all p . Recovery was successful in all 15 realizations for values of $\bar{\Phi}$ below or over the curve.	164
8.10	Evolution of NMSE as a function of time for several TC algorithms when applied to recover approximately (10, 10, 10)-mrank $100 \times 100 \times 100$ tensors, with $\rho = 0.2$	165
8.11	Evolution of NMSE as a function of time for several TC algorithms when applied to recover $100 \times 100 \times 100$ T2 tensors having a decay parameter of $\varphi = 3/2$, with $\rho = 0.15$	166
9.1	Grenoble's south ring from Meylan (point A) to Rondeau (point B) (image courtesy of Google Maps).	169
9.2	Proportion of daily missing data for each sensor: (a) time-mean speeds; (b) vehicle counts.	171
9.3	Portion of daily profiles measured by three different sensors on 12 th June 2015: (a) before preprocessing; (b) after preprocessing.	172
9.4	Speeds measured by all sensors on 7 th May 2015: (a) original data, (b) imputed data.	172
9.5	Possible placements of a missing data window of length $L = 3$ (represented by the blue brackets) within a daily profile. The squares denote the samples along a given mode-1 fiber (thus associated with time).	173
9.6	Examples of missing data patterns generated with the procedure of Section 9.4: (a) $\rho = 0.4$ and $L = 60$; (b) $\rho = 0.6$ and $L = 120$. The black dots indicate missing entries.	174
9.7	Experimental procedure: blue boxes indicate the usual stages of GTL, while the green ones represent the stages involved in our experiment.	178
9.8	Reconstruction of speeds on June 12 th 2015 achieved by all algorithms for $\rho = 0.4$ and $L = 120$: (a) reference data; (b) pattern of missing entries (in black color); (c)–(i) outputs of reconstruction algorithms.	180
9.9	Empirical c.d.f. of the absolute percentage errors committed by each algorithm with respect to travel times.	181
9.10	Estimation of travel times over the 26 th June 2015: (a) c.d.f. of absolute percentage errors; (b) estimated times and reference values.	182
9.11	Convergence plots of iterative algorithms for two selected scenarios.	183

List of Tables

3.1	Characteristics of $\mathbf{A}^{(p)}$ for some classes of matrices [19, 65].	54
5.1	Convergence in Scenarios 1 and 2 of Section 5.1.1.1: proportion of runs which converge to a minimum (bold); and proportion of runs which converge to a local minimum of NSE larger than -75 dB in Scenario 1 and larger than -30 dB in Scenario 2 (underlined).	90
5.2	Convergence in Scenario 1 of Section 5.1.1.2: proportion of runs which converge to a minimum (bold); and proportion of runs which converge to a local minimum of NSE larger than -75 dB (underlined).	93
5.3	Average computing time (in seconds) measured in Scenario 1 of Section 5.2.3.	100
5.4	Average computing time (in seconds) measured in Scenario 2 of Section 5.2.3.	102
5.5	Average computing time (in seconds) measured in the symmetric setting of Section 5.2.4.	103
5.6	Average computing time (in seconds) measured in Scenario 1 of Section 5.2.4.	105
9.1	Simulation results on the reconstruction of speeds data and subsequent estimation of travel times.	179

List of Acronyms

- AAS** *ad-hoc* algebraic solution
- ADMM** alternating direction method of multipliers
- ALS** alternating least squares
- AWGN** additive white Gaussian noise
- BCD** block coordinate descent
- BFGS** Broyden-Fletcher-Goldfarb-Shanno
- BMSE** Bayesian mean squared error
- CALS** constrained alternating least squares
- CCPD** circulant-constrained canonical polyadic decomposition
- c.d.f.** cumulative distribution function
- CPD** canonical polyadic decomposition
- CRB** Cramér-Rao lower bound
- CS** compressive sensing
- DOF** degrees of freedom
- ECRB** expected Cramér-Rao bound
- FIM** Fisher information matrix
- GRI** gradual rank increase
- GTL** Grenoble traffic lab
- HOSVD** higher-order singular value decomposition
- IFHHT** imputation scheme with feedback and HOSVD hard thresholding
- IFHST** imputation scheme with feedback and HOSVD soft thresholding
- IHT** iterative hard thresholding
- i.i.d.** independent and identically distributed

- ISS** improved step size
- LRMR** low-rank matrix recovery
- LRTR** low-rank tensor recovery
- MAPE** mean absolute percentage error
- MC** matrix completion
- MCALS** mixed constrained alternating least squares
- MDFT** multidimensional discrete Fourier transform
- MnRA** minimum n -rank approximation
- MO** measurement operator
- MPO** modal projection ordering
- MSCALS** mixed symmetric constrained alternating least squares
- MSE** mean squared error
- NIHT** normalized iterative hard thresholding
- NMSE** normalized mean squared error
- NNM** nuclear norm minimization
- NSE** normalized squared error
- NTIHT** normalized tensor iterative hard thresholding
- OTD** orthogonal Tucker decomposition
- p.d.f.** probability distribution function
- RIC** restricted isometry constant
- RIP** restricted isometry property
- RMSE** root mean squared error
- SBS** subspace-based solution
- SCALS** symmetric constrained alternating least squares
- SCPD** structured canonical polyadic decomposition

SeMP sequentially optimal modal projections

SeMPIHT SeMP-based iterative hard thresholding

SNN sum of nuclear norms

SNR signal-to-noise ratio

SO sampling operator

SVD singular value decomposition

TC tensor completion

TD Tucker decomposition

THOSVD truncated HOSVD

TIHT tensor iterative hard thresholding

TMac tensor completion by parallel matrix factorization

List of Symbols

Concept	Notation
Definition	\triangleq
Vectors	$\mathbf{x} = [x_n], \quad \boldsymbol{\lambda} = [\lambda_n]$
Component of a vector	$x_n, \quad [x + \lambda]_n$
Matrices	$\mathbf{X} = [x_{n_1, n_2}], \quad \boldsymbol{\Lambda} = [\lambda_{n_1, n_2}]$
Component of a matrix	$x_{n_1, n_2}, \quad [x + \lambda]_{n_1, n_2}$
Tensors (identified with their representations)	$\mathcal{A} = [a_{n_1, \dots, n_P}], \quad \boldsymbol{\mathcal{X}} = [x_{n_1, \dots, n_P}]$
Component of a tensor (representation)	$x_{n_1, \dots, n_P}, \quad [a + x]_{n_1, \dots, n_P}$
Complex conjugation	$\mathbf{v}^*, \mathbf{V}^*, \boldsymbol{\mathcal{V}}^*$
Transposition and Hermitian transposition	$\mathbf{X}^T, \quad \mathbf{X}^H \triangleq (\mathbf{X}^*)^T$
Sets and vector spaces	$\mathcal{V}, \mathcal{W}, \mathcal{N}$
Tensor spaces	$\mathcal{R}, \mathcal{S}, \mathcal{T}$
Operators	$\mathcal{A}, \mathcal{H}, \mathcal{S}$
Adjoint	\mathcal{A}^\dagger
Tensor product	$\otimes, \quad \bigotimes_{p=1}^P$
Kronecker product	\boxtimes
Hadamard product	\odot
Khatri-Rao product	\diamond
Mode- p (mode- n) product	$\times_p, \quad \times_{p=1}^P$
Mode- p matrix unfolding of tensor $\boldsymbol{\mathcal{X}}$	$\mathbf{X}_{\langle p \rangle}, \quad (\boldsymbol{\mathcal{X}})_{\langle p \rangle}, \quad (\mathcal{A} + \boldsymbol{\mathcal{X}})_{\langle p \rangle}$
Combined matrix unfolding of tensor $\boldsymbol{\mathcal{X}}$	$\mathbf{X}_{[p]}, \quad (\boldsymbol{\mathcal{X}})_{[p]}, \quad (\mathcal{A} + \boldsymbol{\mathcal{X}})_{[p]}$
Canonical polyadic decomposition	$\boldsymbol{\mathcal{X}} = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket$
Enumeration set	$\langle P \rangle \triangleq \{1, \dots, P\}$
Tensor space dimensions	$\bar{N} = N_1 \dots N_P, \quad \bar{N}_p = \bar{N}/N_p$
Orthogonality	$\mathbf{v} \perp \mathbf{w}, \quad \mathbf{v} \perp \mathcal{W}$
All-ones vector	$\mathbf{1}_N \triangleq [1 \ \dots \ 1]^T \in \mathbb{R}^N$
All-zeros vector	$\mathbf{0}_N \triangleq [0 \ \dots \ 0]^T \in \mathbb{R}^N$
All-zeros matrix	$\mathbf{0}_{N_1 \times N_2} \triangleq \mathbf{0}_{N_1} \mathbf{0}_{N_2}^T \in \mathbb{R}^{N_1 \times N_2}$
Positive definiteness	$\mathbf{X} \succ \mathbf{0}, \quad \mathbf{A} - \mathbf{B} \succ \mathbf{0} \implies \mathbf{A} \succ \mathbf{B}$
Mathematical expectation	$\mathbb{E}\{\cdot\}$

Contents

1	Introduction	1
1.1	Motivation	2
1.1.1	Structured canonical polyadic decomposition	2
1.1.2	Low-rank tensor recovery	2
1.2	Contributions	3
1.3	Publications and oral presentations	4
1.4	Structure of the thesis	5
2	Tensors and multilinear algebra	14
2.1	Tensors and tensor spaces	14
2.1.1	Tensor representations	17
2.1.2	Symmetric tensors	20
2.2	Isomorphisms among tensor spaces	20
2.3	Tensor norms	23
2.3.1	Induced scalar product and associated Hilbert-Schmidt norm	23
2.3.2	Other norms	24
2.4	Tensor-related algebraic operations	25
2.4.1	Kronecker product	26
2.4.2	Khatri-Rao product	27
2.4.3	Mode- p product	28
2.5	Tensor rank and the canonical polyadic decomposition (CPD)	31
2.5.1	The canonical polyadic decomposition and its properties	32
2.5.2	Properties of the tensor rank	36
2.6	Modal spaces, multilinear rank and the Tucker decomposition	39
2.6.1	Tucker decomposition	40
2.6.2	Best m rank- \mathbf{r} approximation	44
2.7	Final remarks and bibliographical notes	49
Part I	Structured canonical polyadic decomposition	51
3	Structured CPD estimation and Cramér-Rao bounds	52
3.1	Structured matrix factors and the structured CPD (SCPD)	52
3.1.1	Uniqueness of the SCPD	54
3.2	Application example: Wiener-Hammerstein model identification	55

3.3	Existing SCPD computation methods	56
3.3.1	Alternating least squares with structural constraints	57
3.3.2	Non-iterative solution for a SCPD having a banded circulant matrix	59
3.3.3	Subspace-based solution	61
3.4	Cramér-Rao bound for SCPD estimation	63
3.4.1	Information inequality and the Cramér-Rao bound (CRB)	63
3.4.2	SCPD model	64
3.4.3	Model identifiability	66
3.4.4	CRB for SCPD estimation under additive white Gaussian noise	67
3.4.5	Bayesian extension	69
4	Algorithms for SCPD computation	71
4.1	Constrained alternating least-squares (CALS) algorithm	71
4.1.1	Exact iterates	71
4.1.2	Approximate iterates	74
4.1.3	Symmetric CALS	74
4.2	Algebraic solution for a SCPD having circulant factors	76
4.2.1	Basic properties of circulant matrices	77
4.2.2	General system of monomial equations	78
4.2.3	Symmetric system of monomial equations	83
4.2.4	Summary of proposed algebraic solution	84
4.2.5	Illustrative examples	85
4.3	Final remarks	87
5	Numerical evaluation of SCPD algorithms	88
5.1	Evaluation of CALS schemes	88
5.1.1	Dependence of solution quality and computing time on initialization	89
5.1.2	Presence of scaling factors and normalization	94
5.2	Statistical performance of SCPD algorithms	96
5.2.1	Evaluated estimation algorithms	96
5.2.2	Applicability of the CRB	97
5.2.3	Non-symmetric SCPD	98
5.2.4	Symmetric SCPD	101
5.2.5	Wiener-Hammerstein model identification	103
5.3	Final remarks	105

Part II	Low-rank tensor recovery	106
6	From compressive sensing to low-rank tensor recovery	107
6.1	Compressive sensing	107
6.1.1	Problem statement and main involved issues	108
6.1.2	Measurement matrices and the restricted isometry property	108
6.1.3	Recovery via ℓ_1 norm minimization	110
6.2	Low-rank matrix recovery	110
6.2.1	Problem statement	110
6.2.2	Suitable measurement operators	111
6.2.3	Recovery via nuclear norm minimization	112
6.3	Low-rank tensor recovery	113
6.3.1	Problem statement	113
6.3.2	Suitable measurement operators	115
6.3.3	Review of main approaches	116
7	Iterative hard thresholding for low-rank tensor recovery	123
7.1	Iterative hard thresholding (IHT)	123
7.1.1	Application to compressive sensing and low-rank matrix recovery	125
7.1.2	Application to tensor recovery based on multilinear rank	126
7.2	An IHT algorithm based on sequentially optimal modal projections	127
7.2.1	Comparison with previous approaches	128
7.2.2	Performance bound	129
7.2.3	Computing cost per iteration	133
7.2.4	Stopping criteria	134
7.3	Step size selection	134
7.3.1	Step size selection in low-rank tensor recovery	135
7.3.2	A heuristic for improving step size selection	136
7.4	Gradual rank increase heuristic	137
7.4.1	Effect of gradual rank increase	138
7.5	Simulation results	142
7.5.1	Estimation of sampling bounds	142
7.5.2	Convergence and computational cost	144
7.6	Concluding remarks	146
8	Feedback-controlled imputation schemes for tensor completion	148
8.1	Single imputation schemes for tensor completion	148
8.2	Single imputation with ℓ_1 -regularized least squares	150

8.2.1	Sparse and compressible orthogonal Tucker models	150
8.2.2	Imputation scheme with ℓ_1 -regularized least squares	153
8.3	Imputation schemes with feedback	155
8.3.1	Nuclear norm minimization of a single unfolding	155
8.3.2	Interpretation as feedback-controlled system	157
8.3.3	Generalized imputation scheme for tensor completion	159
8.4	Experimental validation	160
8.4.1	Improvement of imputation scheme due to first-order feedback control mechanism	160
8.4.2	Evolution of Tucker core sparsity along iterations	163
8.4.3	Empirical sampling bounds	163
8.4.4	Convergence and computing cost	164
8.5	Conclusions	166
9	Traffic data reconstruction via tensor completion	167
9.1	Introduction	167
9.2	Grenoble Traffic Lab	168
9.3	Traffic data employed in our case study	170
9.3.1	Preprocessing	170
9.4	Simulation of missing data	173
9.5	Completion of speeds data	174
9.5.1	Description of evaluated algorithms	174
9.5.2	Experimental procedure	177
9.5.3	Simulation results	178
9.6	Final remarks	182
10	Conclusions and perspectives	184
10.1	Structured canonical polyadic decomposition	184
10.1.1	Perspectives	184
10.2	Low-rank tensor recovery	185
10.2.1	Perspectives	186
	Bibliography	192

Introduction

Tensors have interested mathematicians and physicists since the nineteenth century. In physics, they offer a convenient language for expressing certain natural laws. A famous example is Einstein's theory of general relativity, whose fundamental equations are expressed in terms of tensors. From the sixties onward, a growing interest in tensors has also been observed across many other scientific and engineering communities, in part spurred by pioneering works in psychometrics which applied tensor-based techniques with data analysis purposes [200, 201, 202, 100, 39]. Notably, blind source separation techniques were developed in the nineties by exploiting the tensor structure of higher-order cumulants [36, 38, 37, 50], while many works employing tensor models were surfacing also in the chemometrics literature [4, 25, 24, 130]. Nowadays, the ever expanding list of applications of tensors encompasses problems in telecommunications [179, 57, 84, 83, 81], signal processing [60, 180, 147, 154, 45], computer vision [177, 135, 42, 149], biomedical engineering [8, 171, 170, 140], dynamical system modeling and identification [114, 78, 79, 26] and data mining [6, 120, 145, 156].

This relatively recent surge of interest in tensor methods is mainly explained by their ability to exploit additional problem structure in comparison with more traditional matrix-based ones. The estimation of excitation/emission spectra from fluorescence data in chemometrics by means of high-order tensor decomposition techniques [25, 24] is a stereotypical example of such a superiority, which stems in this case from the uniqueness of these quantities under much milder conditions than are needed when matrix decompositions are used instead. More generally, the same idea applies to inverse problems whose sought quantities constitute multilinear models, such as, *e.g.*, in the estimation of directions of arrival in antenna array processing [174]. Tensor models are equally useful in many other problems because they often provide accurate and parsimonious representations of real-world multidimensional data, a fact that can be exploited for developing efficient storage, computation and estimation techniques. This line of thought is followed, for instance, in data compression [56, 7], nonlinear system modeling [79, 26] and low-rank tensor recovery [43, 167].

Concomitantly with the thriving resort to tensor-based methods in applied sciences, new and exciting research directions related to tensor properties, tensor representations and multilinear algebra have flourished. Significant advancement has been brought about by such an ample effort in recent times [95, 72, 73, 153, 87, 15, 44, 96, 162], and yet some

fundamental properties of high-order tensors still elude researchers to a great extent.

1.1 Motivation

1.1.1 Structured canonical polyadic decomposition

Tensor decompositions play a major part in the above described trend. Arguably, the canonical polyadic decomposition (CPD) [104] (also known as PARAFAC or Candecomp [100, 39]), which expresses a tensor as a minimal sum of rank-one tensors, is among the most important and popular tensor-related techniques. Such a relevance is primarily owed to its strong uniqueness properties, but also due to its connection with the tensor rank. As a consequence, much work has been devoted to the study and development of algorithms for CPD computation, a problem which is in general NP-hard [102].

General iterative algorithms exist for computing an approximate CPD [193, 2, 41], usually assuming knowledge of its rank. Owing to its simplicity, a quite popular one is alternating least squares (ALS) [100], which cyclically estimates the matrix factors by keeping all but one of them fixed. Yet, they are prone to suffering from well-known convergence difficulties, due to which they can fail to deliver accurate results or require a great computing effort. These difficulties are largely caused by the openness of the set of tensors having rank bounded by any $R > 1$, which implies that the approximate CPD computation problem may have no solution. Even when it does have one, convergence can be slowed down when traversing regions of the parameter space close to tensors having no best rank- R approximation [144, 155].

Tensors arising in some applications admit CPDs whose factor matrices are structured, whether in a linear sense (*e.g.*, having Toeplitz, circulant, Hankel, banded or block-Toeplitz structure) or not (*e.g.*, having Vandermonde structure) [198, 206, 9, 57, 84, 78, 186]. Therefore, CPD computation methods which exploit their structure have been proposed in an attempt to mitigate the above mentioned difficulties [116, 184, 187]. However, the treatment of structural constraints has been mostly *ad-hoc* and on a case-by-case basis. Algebraic methods are particularly interesting for they often provide a quite cheap approximate solution. Though this estimate is usually significantly disturbed in the presence of noise, it can be refined by an iterative algorithm, in many cases at a relatively small cost.

1.1.2 Low-rank tensor recovery

Recently, the low-rank tensor recovery (LRTR) problem was formulated as a logical extension of compressive sensing and low-rank matrix recovery, aiming at the estimation of higher-order tensors from a small number of linear measurements by exploiting parsimony with respect to some tensor model (or tensor representation). A particular case of LRTR which draws special interest, known as tensor completion, focus on recovering the missing entries of a partially

observed tensor. It finds applications in several fields where measured tensors can often be approximated by low-rank models, such as, *e.g.*, computer vision [134, 213], hyperspectral imaging [88, 181], seismic data processing [122] and road traffic data recovery [191, 163, 164].

The passage to higher-order tensors, however, is complicated by the fact that multiple non-equivalent definitions of rank exist in this case, each one corresponding to a different tensor model. Hence, various approaches exploiting these distinct rank notions have been proposed. One can argue that the strategy based on the Tucker model, whose supporting concept of parsimony is the multilinear rank, is adopted most of the time. The principal reason is that this approach rests upon classic matrix tools. In comparison, that based on tensor rank is harder to address, even though its underlying CPD model is more parsimonious.

In contrast with compressive sensing and matrix recovery, no LRTR method has yet been shown optimal or quasi-optimal in the sense of requiring a number of measurements growing roughly linearly with the complexity (in terms of the number of degrees of freedom) of its underlying tensor model. In fact, even systematic experimental studies on this point are lacking. Moreover, many existing methods are not suitable for recovering tensors which deviate considerably from the ideal assumption of low multilinear rank. Much work is still needed to bridge these gaps, both in the general case and in the tensor completion setting.

1.2 Contributions

This thesis addresses some of the foregoing issues involved in the problems of structured canonical polyadic decomposition (SCPD) estimation and LRTR. Its contributions are summarized below.

Formulation of SCPD algorithms and performance evaluation. We formulate a general constrained version of ALS, called constrained alternating least squares (CALs), in which the factors can be arbitrarily structured in a linear sense (*i.e.*, they lie in specified matrix subspaces) [65]. Also, (partially and completely) symmetric SCPDs are treated and approximate iterates with reduced cost are derived [64]. Though *ad-hoc* versions of ALS with structured factors have already been proposed, ours appears to be the first systematic treatment of the SCPD computation problem via this approach.

An algebraic solution for a SCPD having only circulant factors is developed as well, rooted in the special eigenstructure of square circulant matrices. By applying the multidimensional discrete-time Fourier transform to the target tensor, this approach reduces the SCPD computation to the resolution of a system of homogeneous monomial equations [64]. We characterize the resulting system of equations and study how its different solutions relate to one another.

The performance of the proposed **CALS** variants with regard to convergence to local minima and to total computing cost is studied by means of simulations. This investigation allows assessing the sensitivity of **CALS** *vis-à-vis* its initialization and also understanding under which conditions the use of approximate iterations can be of interest.

More broadly, we evaluate the statistical performance of several **SCPD** estimation algorithms by relying on closed-form expressions for the Cramér-Rao lower bound which were derived in [19, 63, 65, 20]. This is carried out in a Bayesian setting where random **SCPD** instances are drawn according to specified prior distributions. In particular, we assess the performance of estimators combining an algebraic (and non-iterative) stage with a subsequent refinement stage via some iterative algorithm [65].

Low-rank tensor recovery via iterative hard thresholding. An iterative hard thresholding (**IHT**) algorithm for **LRTR** based on multilinear rank is proposed [68, 67]. This algorithm employs a hard thresholding operator which is less costly than currently used alternatives and leads to superior or comparable performance. Furthermore, theoretical recovery results are derived by relying on its analytical properties and on restricted isometry assumptions. We also study the sampling requirements of our proposed algorithm and compare it with those of many others by means of systematic numerical experiments. Heuristics for step size selection and gradual (multilinear) rank increase are proposed, discussed and empirically evaluated [66].

Tensor completion algorithm based on soft Tucker core thresholding. Motivated by the typical modal singular spectra possessed by real-world data tensors, we develop an algorithm for tensor completion based on applying soft thresholding to the core of an orthogonal Tucker model. The rationale of this strategy lies in the connection between compressibility of that core and fast decay of modal singular spectra of the target tensor, which we show analytically and illustrate numerically. Links with existing approaches are elucidated and a control theory-based interpretation is given to a version of our algorithm comprising a feedback correction mechanism for accelerating convergence.

The suitability of this algorithm for completing real-world tensors is shown by means of a detailed application example. Namely, we deal with the problem of traffic data reconstruction, which is of great relevance in the context of intelligent transportation systems.

1.3 Publications and oral presentations

[64] J. H. de Morais Goulart and G. Favier, “An algebraic solution for the Candecomp/PARAFAC decomposition with circulant factors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 4, pp. 1543–1562, 2014.

- [63] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, “Statistical efficiency of structured CPD estimation applied to Wiener-Hammerstein modeling,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 948–952.
- [66] J. H. de Morais Goulart and G. Favier, “An iterative hard thresholding algorithm with improved convergence for low-rank tensor recovery,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 1701–1705.
- [20] M. Boizard, J. H. de Morais Goulart, R. Boyer, G. Favier, and P. Comon, “Performance asymptotique d’estimateurs de modèles CP avec facteurs structurés,” in *XXVème colloque GRETSI (GRETSI 2015)*, Lyon, France, Sep. 2015.
- [65] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, “Tensor CP decomposition with structured factor matrices: Algorithms and performance,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 757–769, 2016.
- [67] J. H. de Morais Goulart and G. Favier, “Iterative hard thresholding based algorithms for low-rank tensor recovery,” Symposium on Tensors for Signals and Systems at 20th Conference of the International Linear Algebra Society (ILAS), Leuven, Belgium, Jul. 2016.
- [68] J. H. de Morais Goulart and G. Favier, “Low-rank tensor recovery using sequentially optimal modal projections in iterative hard thresholding (SeMPIHT),” *SIAM Journal on Scientific Computing*, 2016, (submitted).

1.4 Structure of the thesis

This manuscript is divided into two parts, each one addressing one of the problems described in [Section 1.1](#). First, [Chapter 2](#) introduces basic mathematical definitions, algebraic operations and fundamental results which are required for the remaining of the thesis. The notation used in subsequent chapters is also established.

[Part I](#) starts with [Chapter 3](#), which describes the [SCPD](#) problem and illustrates its relevance by giving an application example. Existing specialized estimation algorithms are reviewed, and then expressions for the corresponding deterministic and Bayesian Cramér-Rao bounds are given.

Subsequently, our proposed methods for [SCPD](#) computation are presented in [Chapter 4](#). We start by working out a constrained version of [ALS](#) which takes the structure of the matrix factors into account. Exact and approximate iterates are developed, as well as a general formulation which deals with (partially or completely) symmetric [SCPDs](#). Next, an algebraic solution for handling circulant factors is elaborated by relying on multilinearity of the model and on basic properties of circulant matrices. Applying the multidimensional

Fourier transform to the target tensor, we essentially convert the SCPD problem into the resolution of a system of homogeneous monomial equations. The solutions of this system are studied and several illustrative examples are provided.

Chapter 5 then closes this part by presenting an empirical evaluation of the SCPD estimation algorithms proposed in Chapter 4. It first studies the performance of CALS variants with respect to convergence towards local minima and to computing cost. Then, the statistical performance of several SCPD estimators is numerically evaluated, by making use of the Cramér-Rao bounds given in Chapter 3.

Part II begins on Chapter 6, where we state the LRTR problem, discuss its main aspects and then survey some of the major existing approaches and their related recovery guarantees. This presentation is preceded by a brief review on compressive sensing and low-rank matrix recovery, with the goal of introducing and explaining in simpler settings some aspects which are also relevant to the LRTR problem.

The IHT framework is detailed in Chapter 7, where we describe existing algorithms based on this technique for compressive sensing, low-rank matrix recovery and LRTR. We then propose an IHT algorithm for LRTR and derive a recovery performance bound based on standard restricted isometry assumptions. Step size selection and gradual rank increase heuristics are developed for accelerating convergence, the latter being applicable to tensors having characteristics typically observed in real-world data. Numerical simulations are employed to estimate the sampling bounds of several algorithms both in the general setting (with Gaussian measurements) and in the context of tensor completion. This chapter also includes further simulation results concerning the comparison of LRTR algorithms with respect to convergence speed and computational efficiency.

We then proceed to present our proposed tensor completion algorithm based on soft thresholding in Chapter 8. To this end, a review on single imputation schemes for tensor completion is first given. Drawing upon this strategy, our proposed approach is developed, where the thresholding operation is meant to approximately solve an ℓ_1 -regularized least-squares problem formulated with the goal of estimating a parsimonious Tucker model. After discussing an interpretation of the alternating direction method of multipliers (ADMM) algorithm with exponentially growing penalty parameter in terms of a feedback control mechanism, we propose a version of our algorithm incorporating this idea for performance improvement. Lastly, simulation results are presented to validate the devised method.

The algorithm conceived in Chapter 8 is then applied to the problem of road traffic data reconstruction in Chapter 9. We first provide information on the application context, on the considered performance indices and on the real-world data used in our experiments. The experimental procedure is then described, followed by a discussion of the obtained results.

Finally, we present our general conclusions and elaborate on research perspectives.

Introduction

Les mathématiciens et physiciens s'intéressent aux tenseurs depuis le XIX^e siècle. En physique, les tenseurs offrent un langage qui convient pour exprimer certaines lois. La théorie de la relativité générale d'Einstein en est un exemple fameux, ses équations fondamentales étant écrites sous forme tensorielle. À partir des années soixante, un intérêt croissant pour les tenseurs est observé également dans beaucoup d'autres communautés scientifiques et d'ingénierie, ce qui est en partie dû aux travaux novateurs en psychométrie qui appliquaient des techniques basées sur les tenseurs à des fins d'analyse de données [200, 201, 202, 100, 39]. Notamment, des techniques de séparation aveugle de sources ont été développées dans les années quatre-vingt-dix en exploitant la structure tensorielle des cumulants d'ordre supérieur [36, 38, 37, 50], alors que plusieurs travaux employant des modèles tensoriels apparaissaient aussi en chimiométrie [4, 25, 24, 130]. Aujourd'hui, la liste toujours croissante des applications des tenseurs comprend des problèmes de télécommunications, [179, 57, 84, 83, 81], traitement du signal [60, 180, 147, 154, 45], vision par ordinateur [177, 135, 42, 149], ingénierie biomédicale [8, 171, 170, 140], modélisation et identification de systèmes dynamiques [114, 78, 79, 26] et fouille de données [6, 120, 145, 156].

Cet afflux d'intérêt relativement récent pour les méthodes tensorielles s'explique principalement par leur capacité à mieux exploiter la structure de certains problèmes par rapport à des méthodes matricielles traditionnelles. L'estimation de spectres d'excitation/émission à partir de données de fluorescence en chimétrie à l'aide de techniques de décomposition tensorielle [25, 24] est un exemple stéréotypé de leur supériorité, qui dans ce cas découle de l'unicité des quantités à estimer sous des conditions beaucoup moins contraignantes que celles qui s'appliquent lorsqu'on fait appel plutôt à des décompositions matricielles. De manière plus générale, cette même idée s'applique aux problèmes inverses dont les quantités d'intérêt satisfont des modèles multilinéaires, comme, *e.g.*, l'estimation de directions d'arrivée dans le traitement de réseaux d'antenne [174]. Les modèles tensoriels sont également utiles dans beaucoup d'autres problèmes car souvent ils fournissent des représentations précises et parcimonieuses de données réelles multidimensionnelles, ce qui peut être exploité par des techniques de stockage, calcul et estimation efficaces. Ce raisonnement est appliqué, par exemple, à la compression de données [56, 7], à la modélisation de systèmes [79, 26] et à la récupération de tenseurs de rang faible [43, 167].

Simultanément à l'application florissante de méthodes tensorielles aux sciences appliquées, de nouvelles directions de recherche passionnantes liées aux propriétés des tenseurs, aux représentations tensorielles et à l'algèbre multilinéaire ont prospéré. Des avancements significatifs ont été apportés récemment [95, 72, 73, 153, 87, 15, 44, 96, 162], et pour-

tant certaines propriétés fondamentales des tenseurs d'ordre supérieur échappent encore aux chercheurs dans une large mesure.

Motivation

Décomposition canonique polyadique structurée

Les décompositions tensorielles jouent un rôle majeur dans la tendance décrite ci-dessus. La décomposition canonique polyadique (CPD) [104] (aussi connue sous les noms PARAFAC et Candecomp [100, 39]), qui exprime un tenseur comme une somme minimale de tenseurs de rang un, se situe parmi les techniques tensorielles les plus importantes et populaires. Cette importance est principalement due à ses propriétés d'unicité, mais aussi à son lien avec le rang tensoriel. Par conséquent, un grand effort est consacré à l'étude et au développement d'algorithmes pour le calcul de la CPD, un problème qui est en général NP-difficile [102].

Des algorithmes itératifs généraux existent pour le calcul approximatif d'une CPD [193, 2, 41], normalement sous l'hypothèse de connaissance de son rang *a priori*. En raison de sa simplicité, un algorithme assez populaire est celui des moindres carrés alternés (ALS) [100], qui estime chaque facteur matriciel en gardant les autres facteurs fixés. Toutefois, ces algorithmes sont susceptibles de connaître des difficultés de convergence bien connues, qui peuvent produire des résultats imprécis ou bien exiger un temps de calcul prohibitif. Ces difficultés sont largement causées par l'ouverture de l'ensemble des tenseurs de rang borné par $R > 1$, qui entraîne la possibilité d'inexistence d'une solution au calcul approximatif d'une CPD. Même s'il y en a une, la convergence peut être ralentie lorsque l'algorithme traverse des régions de l'espace des paramètres incluant des tenseurs qui n'admettent pas de meilleure approximation de rang R [144, 155].

Les tenseurs qui apparaissent dans certaines applications admettent des CPDs à facteurs matriciels structurés, soit dans un sens linéaire (*e.g.*, ayant une structure Toeplitz, circulante, Hankel, en bande ou Toeplitz en bloc) ou non-linéaire (*e.g.*, ayant une structure Vandermonde) [198, 206, 9, 57, 84, 78, 186]. Donc, des méthodes de calcul de CPD qui exploitent leur structure ont été proposées afin d'essayer d'atténuer les difficultés susmentionnées [116, 184, 187]. Cependant, le traitement de contraintes structurelles est fait surtout de façon *ad-hoc* et au cas par cas. Les méthodes algébriques sont particulièrement intéressantes car souvent elles fournissent une solution approchée à un faible coût de calcul. Bien que cette estimée soit en général assez perturbée lorsqu'il y a du bruit, elle peut être raffinée par un algorithme itératif, à un coût relativement faible dans de nombreux cas.

Récupération de tenseurs de rang faible

Récemment, le problème de récupération de tenseurs de rang faible (LRTR) a été posé comme une extension logique de l'échantillonnage compressé et de la récupération de matrices de rang faible. Son but est l'estimation de tenseurs d'ordre supérieur à partir d'un nombre réduit de mesures linéaires en exploitant la parcimonie inhérente à un modèle (ou représentation) tensoriel(le). Un cas d'intérêt spécial de LRTR, connu sous le nom de reconstruction tensorielle, se concentre sur la récupération de composantes manquantes d'un tenseur de données partiellement observé. Ce problème s'applique à plusieurs domaines où les tenseurs mesurés peuvent souvent être approchés par des modèles à rang faible, comme, *e.g.*, en vision par ordinateur [134, 213], en imagerie hyperspectrale [88, 181], dans le traitement de données sismiques [122] et la récupération de données de trafic routier [191, 163, 164].

Néanmoins, le passage aux tenseurs d'ordre plus élevé est compliqué par le fait que de multiples définitions non-équivalentes de rang existent dans ce cas, chacune étant associée à un modèle tensoriel différent. Ainsi, plusieurs approches exploitant ces notions de rang distinctes ont été proposées. On peut argumenter que la stratégie basée sur le modèle de Tucker, dont la notion sous-jacente de parcimonie correspond au rang multilinéaire, est adopté la plupart du temps. La principale raison pour cela est que cette approche s'appuie sur des outils matriciels classiques. En comparaison, celle basée sur le rang tensoriel est plus difficile à aborder, malgré le fait que son modèle sous-jacent basé sur la CPD est plus parcimonieux.

Contrairement aux problèmes d'échantillonnage compressé et de récupération matricielle, aucune méthode de LRTR n'a encore été démontrée optimale ou quasi-optimale dans le sens d'exiger un nombre de mesures qui croît approximativement linéairement avec la complexité (en termes du nombre de degrés de liberté) de son modèle sous-jacent. En fait, même des études empiriques et systématiques sur cet aspect manquent. En plus, beaucoup de méthodes existantes ne sont pas adéquates pour la récupération de tenseurs s'éloignant considérablement de l'hypothèse idéale de rang multilinéaire faible. Beaucoup de travail est encore nécessaire pour que ces lacunes soient comblées, tant dans le cas général que dans le cadre de reconstruction tensorielle.

Contributions

Cette thèse aborde certaines des questions citées ci-dessus liées aux problèmes d'estimation d'une CPD structurée (SCPD) et de LRTR. Les contributions apportées sont résumées ci-dessous.

Formulation d’algorithmes de calcul d’une SCPD et évaluation de performance.

On formule une version contrainte générale de ALS, notée CALS, où les facteurs peuvent être structurés de façon arbitraire dans un sens linéaire (*i.e.*, ils se trouvent dans des sous-espaces matriciels spécifiés) [65]. Par ailleurs, les SCPDs (partiellement ou complètement) symétriques sont traitées et des itérées approchées à coût réduit sont dérivées [64]. Bien que des versions *ad-hoc* de ALS à facteurs structurés aient déjà été proposées, la nôtre est apparemment la première à traiter le problème de calcul d’une SCPD de manière systématique via l’approche ALS.

Une solution algébrique pour le calcul d’une SCPD n’ayant que des facteurs circulants est également développée, basée sur la structure spéciale de la décomposition en valeurs propres des matrices circulantes. Par l’application de la transformée de Fourier à temps discret multidimensionnelle au tenseur d’intérêt, cette approche ramène le problème de calcul d’une SCPD à la résolution d’un système d’équations monomiales homogènes [64]. On caractérise le système d’équations obtenu et on étudie comment ses différentes solutions sont liées entre elles.

La performance des variantes CALS proposées en ce qui concerne la convergence vers des minima locaux et le coût de calcul global est étudiée au moyen de simulations numériques. Cette investigation nous permet d’évaluer la sensibilité de CALS *vis-à-vis* de son initialisation et aussi de comprendre sous quelles conditions l’usage d’itérées approchées peut être intéressant.

Plus généralement, on évalue la performance statistique de plusieurs algorithmes d’estimation SCPD en s’appuyant sur des formules pour la borne inférieure de Cramér-Rao qui ont été dérivées dans [19, 63, 65, 20]. Ceci est réalisé dans un cadre Bayésien où des exemples aléatoires de modèles SCPD sont générés selon des distributions *a priori* spécifiées. En particulier, on évalue la performance d’estimateurs couplant une étape algébrique (et non-itérative) à une étape de raffinement mené par un algorithme itératif [65].

Récupération de tenseurs de rang faible via un seuillage itératif dur. Un algorithme de seuillage itératif dur (IHT) pour LRTR basé sur le rang multilinéaire est proposé [68, 67]. Cet algorithme emploie un opérateur de seuillage dur qui est moins coûteux que les alternatives actuellement utilisées et mène à une performance comparable ou supérieure. Par ailleurs, des résultats théoriques de récupération sont dérivés en s’appuyant sur ses propriétés analytiques et sur des hypothèses d’isométrie restreinte. On étudie également les besoins d’échantillonnage de notre algorithme proposé et on les compare à ceux de beaucoup d’autres algorithmes à l’aide d’expériences numériques systématiques. Des heuristiques pour le choix du pas et pour l’augmentation graduelle du rang (multilinéaire) sont proposées, discutées et empiriquement évaluées [66].

Algorithme de reconstruction tensorielle basée sur un seuillage doux du cœur d'un modèle de Tucker. Motivés par les spectres singuliers typiquement possédés par des tenseurs de données réelles, on développe un algorithme pour la reconstruction tensorielle basée sur l'application d'un seuillage doux au cœur d'un modèle de Tucker orthogonal. Le raisonnement central de cette stratégie repose sur le lien entre compressibilité du cœur et décroissance rapide des spectres singuliers modaux du tenseur d'intérêt, ce qu'on démontre analytiquement et illustre numériquement. Des parallèles avec d'autres approches existantes sont élaborés et une interprétation basée sur la théorie d'automatique est donnée à une version de notre algorithme comprenant un mécanisme de correction rétro-actif qui a comme but l'accélération de sa convergence.

La pertinence de cet algorithme pour la reconstruction de tenseurs de données réelles est montrée dans un exemple d'application détaillé. Cet exemple concerne la reconstruction de données de trafic routier, ce qui est d'une grande importance dans le contexte des systèmes de transport intelligents.

Publications et présentations orales

[64] J. H. de Morais Goulart and G. Favier, "An algebraic solution for the Candecomp/PARAFAC decomposition with circulant factors," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 4, pp. 1543–1562, 2014.

[63] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, "Statistical efficiency of structured CPD estimation applied to Wiener-Hammerstein modeling," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 948–952.

[66] J. H. de Morais Goulart and G. Favier, "An iterative hard thresholding algorithm with improved convergence for low-rank tensor recovery," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 1701–1705.

[20] M. Boizard, J. H. de Morais Goulart, R. Boyer, G. Favier, and P. Comon, "Performance asymptotique d'estimateurs de modèles CP avec facteurs structurés," in *XXVème colloque GRETSI (GRETSI 2015)*, Lyon, France, Sep. 2015.

[65] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, "Tensor CP decomposition with structured factor matrices: Algorithms and performance," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 757–769, 2016.

[67] J. H. de Morais Goulart and G. Favier, "Iterative hard thresholding based algorithms for low-rank tensor recovery," Symposium on Tensors for Signals and Systems at 20th Conference of the International Linear Algebra Society (ILAS), Leuven, Belgium, Jul. 2016.

[68] J. H. de Morais Goulart and G. Favier, “Low-rank tensor recovery using sequentially optimal modal projections in iterative hard thresholding (SeMPIHT),” *SIAM Journal on Scientific Computing*, 2016, (submitted).

Structure de la thèse

Le manuscrit est structuré en deux parties, chacune abordant l’un des deux problèmes y traités. D’abord, le Chapitre 2 introduit des définitions mathématiques basiques, des opérations algébriques et des résultats fondamentaux qui sont nécessaires dans le reste de la thèse. La notation utilisée dans les chapitres suivants est aussi établie.

Le Chapitre 3 décrit le problème de calcul d’une SCPD et illustre sa pertinence en donnant un exemple d’application. Des algorithmes d’estimation spécialisés sont rappelés, puis des expressions pour les bornes de Cramér-Rao déterministe et Bayésienne sont données.

Ensuite, nos méthodes proposées pour le calcul d’une SCPD sont présentées dans le Chapitre 4. On développe d’abord une version contrainte de ALS qui prend en compte la structure des facteurs matriciels. Des itérées exactes et approchées sont dérivées, ainsi qu’une formulation générale qui traite les SCPDs (partiellement ou complètement) symétriques. Puis, une solution algébrique pour l’estimation de facteurs circulants est élaborée en s’appuyant sur la multilinéarité du modèle et sur les propriétés fondamentales de matrices circulantes. À l’aide de l’application de la transformée de Fourier multidimensionnelle au tenseur d’intérêt, on réduit le problème du calcul d’une SCPD à la résolution d’un système d’équations monomiales homogènes. Les solutions de ce système sont étudiées et plusieurs exemples illustratifs sont fournis.

Le Chapitre 5 termine cette partie en présentant une évaluation empirique des algorithmes d’estimation d’une SCPD proposés dans le chapitre précédent. Premièrement, on étudie la performance des variantes CALS en ce qui concerne la convergence vers des minima locaux et le coût de calcul. Puis, la performance statistique de plusieurs estimateurs d’une SCPD est évaluée numériquement en faisant appel aux bornes de Cramér-Rao données dans le Chapitre 3.

Le Partie II commence au Chapitre 6, où on pose le problème LRTR et on discute de ses principaux aspects, puis on présente un survol d’un certain nombre d’approches majeures existantes et de leurs garanties de récupération. Cette présentation est précédée d’une succincte révision des problèmes d’échantillonnage compressé et de récupération de matrices de rang faible, dans le but d’introduire et d’expliquer dans ces cadres plus simples certains aspects qui sont aussi pertinents pour le problème LRTR.

L’approche IHT est détaillée dans le Chapitre 7, où on décrit des algorithmes existants basés sur cette technique pour l’échantillonnage compressé, pour la récupération de matrices

de rang faible et pour LRTR. Ensuite, on propose un algorithme IHT pour LRTR et on dérive une borne de performance basée sur des hypothèses usuelles d'isométrie restreinte. Des heuristiques de choix du pas et d'augmentation graduelle du rang sont développées pour accélérer la convergence, cette dernière s'appliquant à des tenseurs ayant des caractéristiques typiques de données réelles. Des simulations numériques sont réalisées pour estimer les bornes d'échantillonnage de plusieurs algorithmes à la fois dans un cadre plus général (avec mesures Gaussiennes) et dans le cadre d'une reconstruction tensorielle. Ce chapitre comprend également des résultats de simulation concernant la comparaison d'algorithmes LRTR quant à leur vitesse de convergence et leur efficacité de calcul.

On procède ensuite dans le Chapitre 8 à la présentation de notre algorithme de reconstruction tensorielle basé sur un seuillage doux. Pour cela, d'abord on fait une révision de schémas d'imputation simple pour la reconstruction tensorielle. En s'appuyant sur cette stratégie, notre approche proposée est développée, où l'opérateur de seuillage est destiné à résoudre approximativement un problème de moindres carrés régularisé par une norme ℓ_1 , qui est formulé dans le but d'estimer un modèle de Tucker parcimonieux. Après avoir fourni une interprétation de la méthode des multiplicateurs à directions alternées (ADMM) avec paramètre de pénalisation exponentiellement croissant en termes d'un mécanisme de contrôle rétro-actif, on propose une version de notre algorithme incorporant cette idée pour améliorer sa performance. Finalement, des résultats de simulation sont présentés pour valider la méthode développée.

L'algorithme conçu dans le Chapitre 8 est ensuite appliqué au problème de reconstruction de données de trafic routier dans le Chapitre 9. Premièrement, on décrit le contexte de l'application, les indices de performance considérés et les données réelles utilisées dans nos expériences. Puis, la procédure expérimentale est présentée et suivie par une discussion des résultats obtenus.

Enfin, on présente nos conclusions générales et on décrit quelques perspectives de recherche.

Tensors and multilinear algebra

In this chapter, we formally introduce tensors, define important algebraic operations and state fundamental results on multilinear algebra, laying the mathematical foundations for the subsequent chapters. In particular, we present and discuss the main properties of two of the most important tensor decompositions, the canonical polyadic and the Tucker decompositions, which play a central role in our contributions. Along this chapter, the mathematical notation used throughout the thesis will also be established.

Contents

2.1	Tensors and tensor spaces	14
2.2	Isomorphisms among tensor spaces	20
2.3	Tensor norms	23
2.4	Tensor-related algebraic operations	25
2.5	Tensor rank and the canonical polyadic decomposition (CPD) . . .	31
2.6	Modal spaces, multilinear rank and the Tucker decomposition . . .	39
2.7	Final remarks and bibliographical notes	49

2.1 Tensors and tensor spaces

We start by defining our mathematical objects of interest, tensors and tensor spaces. One way of introducing tensor spaces relies upon the notion of a quotient space [131, 95], which is recalled below. Vector spaces are basic building blocks in that construction. In this thesis, all vector spaces are defined either over the field of real numbers, \mathbb{R} , or over that of complex numbers, \mathbb{C} . Whenever a given definition or property applies to both fields, we use the symbol $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$.

Definition 2.1 (Quotient space). Let $\mathcal{N} \subset \mathcal{V}$ be a subspace of a vector space \mathcal{V} and define the equivalence relation for all $v, w \in \mathcal{V}$:

$$v \sim w \quad \text{if and only if} \quad v - w \in \mathcal{N}. \quad (2.1)$$

This relation induces, for each $v \in \mathcal{N}$, the equivalence class:

$$\underline{v} \triangleq \{w \in \mathcal{V} : v \sim w\}. \quad (2.2)$$

The *quotient space* with respect to this equivalence relation, denoted by \mathcal{V}/\mathcal{N} , consists of the set containing all equivalence classes (2.2), along with an addition and a scalar multiplication operations defined as $\underline{v} + \underline{w} = \underline{v + w}$ and $\alpha \underline{v} = \underline{\alpha v}$. These operations satisfy the axioms which characterize a vector space, hence the terminology “quotient space.”

A crucial point in the above definition is that all the elements in \mathcal{N} are equivalent to $0 \in \mathcal{V}$. One thus regards \mathcal{N} as a “null subspace” when dealing with \mathcal{V}/\mathcal{N} . For this reason, it is said that \mathcal{N} is “collapsed” into the zero vector.

Let us now turn to the definition of tensors. In essence, tensors are objects associated with *multilinear* transformations, whose coordinate representations themselves transform multilinearly under a change of basis [49]. We thus begin by defining multilinear transformations.

Definition 2.2 (Multilinear transformation). Let \mathcal{V}_p , with $p \in \langle P \rangle \triangleq \{1, \dots, P\}$, and \mathcal{W} be vector spaces defined over \mathbb{F} . A map $f : \mathcal{V}_1 \times \dots \times \mathcal{V}_P \mapsto \mathcal{W}$ is said to be *multilinear* if and only if for all $p \in \langle P \rangle$ it satisfies

$$f(v_1, \dots, v_{p-1}, \alpha v_p + \beta v'_p, v_{p+1}, \dots, v_P) = \alpha f(v_1, \dots, v_{p-1}, v_p, v_{p+1}, \dots, v_P) + \beta f(v_1, \dots, v_{p-1}, v'_p, v_{p+1}, \dots, v_P) \quad (2.3)$$

where $v_q \in \mathcal{V}_q$, $q \in \langle P \rangle$, $v'_p \in \mathcal{V}_p$ and $\alpha, \beta \in \mathbb{F}$. In particular, if $\mathcal{W} = \mathbb{F}$ (that is, if $\dim(\mathcal{W}) = 1$), then f is called a *multilinear form* or a *multilinear functional*.

Example 2.3. Let $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$ be vector spaces over \mathbb{F} equipped with scalar products $\langle \cdot, \cdot \rangle_{\mathcal{V}_1}$, $\langle \cdot, \cdot \rangle_{\mathcal{V}_2}$ and $\langle \cdot, \cdot \rangle_{\mathcal{V}_3}$, respectively. Define the following multilinear form over $\mathcal{V}_1 \times \mathcal{V}_2 \times \mathcal{V}_3$:

$$f_{(w_1, w_2, w_3)} : (v_1, v_2, v_3) \mapsto \langle v_1, w_1 \rangle_{\mathcal{V}_1} \langle v_2, w_2 \rangle_{\mathcal{V}_2} \langle v_3, w_3 \rangle_{\mathcal{V}_3} \in \mathbb{F}, \quad (2.4)$$

where $w_p \in \mathcal{V}_p$, $p \in \{1, 2, 3\}$. Clearly, for any scalars $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{F}$ satisfying $\alpha_1 \alpha_2 \alpha_3 = 1$, the multilinear form $f_{(\alpha_1 w_1, \alpha_2 w_2, \alpha_3 w_3)}$ is equivalent to $f_{(w_1, w_2, w_3)}$, in the sense that $f_{(\alpha_1 w_1, \alpha_2 w_2, \alpha_3 w_3)}(v_1, v_2, v_3) = f_{(w_1, w_2, w_3)}(v_1, v_2, v_3)$. ■

The above example shows the motivation for introducing corresponding equivalence classes. In order to define them, one must resort to the concept of *free vector space*. Essentially, a free vector space defined upon a set \mathcal{S} , denoted by $F(\mathcal{S})$, contains all formal finite linear combinations of elements of \mathcal{S} , disregarding details with respect to their structure [95]. Formally, an element $f \in F(\mathcal{S})$ can be thought of as a function $f : \mathcal{S} \mapsto \mathbb{F}$ with *finite* support. The elements $s \in \mathcal{S}$ for which $f(s) \neq 0$ are precisely those involved in the formal sum which is represented by f . For every $s \in \mathcal{S}$, there is a unique $f_s \in F(\mathcal{S})$ such that $f_s(t) = 1$ if $s = t$ and $f_s(t) = 0$ otherwise.

Definition 2.4 (Elementary tensor). Consider the free vector space $F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P)$, where each \mathcal{V}_p is a vector space. Every $(v_1, \dots, v_P) \in \mathcal{V}_1 \times \cdots \times \mathcal{V}_P$ generates an equivalence class $v_1 \otimes \cdots \otimes v_P$ on $F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P)$ which is induced by the following relation:

$$\forall g \in F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P), \quad v_1 \otimes \cdots \otimes v_P \sim g \text{ if and only if } v_1 \otimes \cdots \otimes v_P - g \in \mathcal{N}, \quad (2.5)$$

where \mathcal{N} is a subspace of $F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P)$ defined as [95]

$$\mathcal{N} \triangleq \text{span} \left\{ \sum_{i_1} \cdots \sum_{i_P} \alpha_{i_1}^{(1)} \cdots \alpha_{i_P}^{(P)} f_{(w_{i_1}^{(1)}, \dots, w_{i_P}^{(P)})} - f_{(\sum_{i_1} \alpha_{i_1}^{(1)} w_{i_1}^{(1)}, \dots, \sum_{i_P} \alpha_{i_P}^{(P)} w_{i_P}^{(P)})} : \alpha_{i_p}^{(p)} \in \mathbb{F}, w_{i_p}^{(p)} \in \mathcal{V}_p \right\}. \quad (2.6)$$

Each sum in the above expression has finitely many terms. The equivalence class $v_1 \otimes \cdots \otimes v_P$ is called *elementary* or *decomposable tensor*.

Given the definition of \mathcal{N} , we are ready to introduce tensor spaces. To avoid unnecessary complications regarding their topological closedness, our definition is limited to finite-dimensional tensor spaces, upon which we shall focus in this thesis.

Definition 2.5 (Tensor space [95]). Consider the free vector space $F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P)$, where each \mathcal{V}_p is a finite-dimensional vector space defined over \mathbb{F} . We define the *tensor space* $\mathcal{T} = \mathcal{V}_1 \otimes \cdots \otimes \mathcal{V}_P$ as

$$\mathcal{T} = \mathcal{V}_1 \otimes \cdots \otimes \mathcal{V}_P \triangleq F(\mathcal{V}_1 \times \cdots \times \mathcal{V}_P) / \mathcal{N}, \quad (2.7)$$

where \mathcal{N} is as defined by (2.6). The expression $\mathcal{V}_1 \otimes \cdots \otimes \mathcal{V}_P$ is said to be a tensor product (denoted by the symbol \otimes) of P vector spaces. The elements lying in $\mathcal{V}_1 \otimes \cdots \otimes \mathcal{V}_P$ are called *P -th order tensors*, and shall be represented by boldface calligraphic letters, as \mathbf{V} , \mathbf{X} and \mathbf{W} .

For conciseness, we shall adopt the notation $\bigotimes_{p=1}^P \mathcal{V}_p \triangleq \mathcal{V}_1 \otimes \cdots \otimes \mathcal{V}_P$. A mixed notation is also useful, as in $\mathcal{W} \otimes \left(\bigotimes_{p=1}^P \mathcal{V}_p \right) \otimes \mathcal{Z}$. An analogous notation shall be used for elementary tensors, as in $\bigotimes_{p=1}^P v_p \triangleq v_1 \otimes \cdots \otimes v_P$. Hence, for instance,

$$v_1 \otimes \cdots \otimes v_{p-1} \otimes w_p \otimes v_{p+1} \otimes \cdots \otimes v_P = \left(\bigotimes_{q=1}^{p-1} v_q \right) \otimes w_p \otimes \left(\bigotimes_{q=p+1}^P v_q \right). \quad (2.8)$$

Fundamentally, Definition 2.5 states that

$$\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p = \text{span} \left\{ \bigotimes_{p=1}^P v_p : v_p \in \mathcal{V}_p, p \in \langle P \rangle \right\}, \quad (2.9)$$

i.e., the elements of a P th-order tensor space are (finite) linear combinations of elementary tensors, each one composed by P vectors. \mathcal{T} is itself a vector space, in which all elements of the subspace \mathcal{N} are assimilated with the zero vector. Based on this observation, one can deduce multilinearity properties which together characterize a tensor space, as follows.

Proposition 2.6 (Multilinearity properties of tensor space [95, Lemma 3.10]). Let $\bigotimes_{p=1}^P \mathcal{V}_p$ be a tensor space over \mathbb{F} . Then, we have $\forall p \in \langle P \rangle, v'_q, v_q \in \mathcal{V}_q, \alpha \in \mathbb{F}$:

- (i) $\left(\bigotimes_{q=1}^{p-1} v_q\right) \otimes 0 \otimes \left(\bigotimes_{q=p+1}^P v_q\right) = 0$
- (ii) $\left(\bigotimes_{q=1}^{p-1} v_q\right) \otimes (\alpha v_p) \otimes \left(\bigotimes_{q=p+1}^P v_q\right) = \alpha \bigotimes_{q=1}^P v_q$
- (iii) $\left(\bigotimes_{q=1}^{p-1} v_q\right) \otimes (v_p + v'_p) \otimes \left(\bigotimes_{q=p+1}^P v_q\right) = \bigotimes_{q=1}^P v_q + \left(\bigotimes_{q=1}^{p-1} v_q\right) \otimes v'_p \otimes \left(\bigotimes_{q=p+1}^P v_q\right).$

Proof. Follows from the assimilation of the elements of \mathcal{N} with zero. \square

Although the tensor product \otimes has not been explicitly defined as an operator among vectors, it is characterized by the properties enumerated in [Proposition 2.6](#). It has precedence over and is distributive with respect to addition.

Proposition 2.7 (Induced basis of a tensor space). Let the vector spaces $\mathcal{V}_1, \dots, \mathcal{V}_P$ defined over \mathbb{F} , and let $\mathcal{B}_p \subset \mathcal{V}_p$ denote a basis of \mathcal{V}_p . Then, the set $\mathcal{B} = \left\{\bigotimes_{p=1}^P b_p : b_p \in \mathcal{B}_p\right\}$ constitutes a basis for $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$. Moreover, $\dim(\mathcal{T}) = \prod_{p=1}^P \dim(\mathcal{V}_p)$.

Proof. See [95, Proposition 3.21]. \square

In particular, when $\mathcal{V}_p = \mathbb{F}^{N_p}$, then the canonical bases $\mathcal{B}_p = \left\{\mathbf{e}_1^{(p)}, \dots, \mathbf{e}_{N_p}^{(p)}\right\}$ of $\mathcal{V}_1, \dots, \mathcal{V}_P$ induce the canonical basis $\mathcal{B} = \left\{\mathbf{e}_{n_1}^{(1)} \otimes \dots \otimes \mathbf{e}_{n_p}^{(P)} : n_p \in \langle N_p \rangle\right\}$ of $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$.

We finish this section by stating the fundamental universal factorization property which characterizes tensor spaces [131].

Proposition 2.8 (Universality of the tensor product). Let $f : \mathcal{V}_1 \times \dots \times \mathcal{V}_P \mapsto \mathcal{W}$ be any multilinear transformation defined over the vector spaces $\mathcal{V}_1, \dots, \mathcal{V}_P$. Then, there exists a unique linear transformation $\phi : \bigotimes_{p=1}^P \mathcal{V}_p \mapsto \mathcal{W}$ such that $\forall (v_1, \dots, v_P) \in \mathcal{V}_1 \times \dots \times \mathcal{V}_P, f(v_1, \dots, v_P) = \phi\left(\bigotimes_{p=1}^P v_p\right)$ holds.

Proof. See [95, Proposition 3.22]. \square

2.1.1 Tensor representations

Let $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$ be a finite-dimensional tensor space¹ and $\mathcal{B}_p = \{b_{n_p}^{(p)}\}_{n_p=1}^{N_p} \subset \mathcal{V}_p$ be a basis of \mathcal{V}_p , with $N_p \in \mathbb{N}$. From [Proposition 2.7](#), elementary tensors of the form $\bigotimes_{p=1}^P b_{n_p}^{(p)}$, where $b_{n_p}^{(p)} \in \mathcal{B}_p$, form a basis for \mathcal{T} . This means that any tensor $\mathbf{v} \in \mathcal{T}$ can be written in the form

$$\mathbf{v} = \sum_{n_1} \dots \sum_{n_P} v_{n_1, \dots, n_P} \bigotimes_{p=1}^P b_{n_p}^{(p)}, \quad (2.10)$$

¹Since all tensor spaces we consider have finite dimensions, we shall often omit the expression “finite-dimensional,” for brevity.

where the scalars v_{n_1, \dots, n_P} constitute the *coordinate representation* of \mathcal{V} with respect to the basis \mathcal{B} , formed as described by Proposition 2.7. These scalars can be thought as the components of a *hypermatrix* $[v_{n_1, \dots, n_P}]$ defined over \mathbb{F} , i.e., $[v_{n_1, \dots, n_P}] \in \mathbb{F}^{N_1 \times \dots \times N_P}$. Such an object can be regarded simply as a generalization of a matrix, in the sense that it is a *P-way array* of numbers indexed by P indices. Each one of these indices is said to be associated with one *mode* or geometric dimension of the multidimensional array. Needless to say, $\mathbb{F}^{N_1 \times \dots \times N_P}$ is a vector space itself with the obvious definitions of addition and multiplication by a scalar.

The above discussion makes a clear distinction between a tensor and its representation. As a matter of fact, a same object $[v_{n_1, \dots, n_P}] \in \mathbb{F}^{N_1 \times \dots \times N_P}$ can represent multiple tensors from different tensor spaces. Furthermore, as soon as we apply a change of basis to one of the vector spaces involved in the tensor space, the coordinate representation must evidently be transformed accordingly. This is analogous to the fact that a single linear transformation between finite-dimensional vector spaces can have multiple matrix representations and, conversely, a same matrix representation can represent different objects—in particular, it can also represent *bilinear* transformations.

Example 2.9. Let $\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3$ be vector spaces with corresponding bases $\mathcal{B}_p = \{b_n^{(p)}\}_{n=1}^{N_p} \subset \mathcal{V}_p$. Consider the vectors $v_p \in \mathcal{V}_p$ having representations $v_p = \sum_{n_p} \alpha_{n_p}^{(p)} b_{n_p}^{(p)}$, with $\alpha_{n_p} \in \mathbb{F}$. Then, using the properties described by Proposition 2.6 we have

$$\mathbf{v} = v_1 \otimes v_2 \otimes v_3 = \left(\sum_{n_1} \alpha_{n_1}^{(1)} b_{n_1}^{(1)} \right) \otimes \left(\sum_{n_2} \alpha_{n_2}^{(2)} b_{n_2}^{(2)} \right) \otimes \left(\sum_{n_3} \alpha_{n_3}^{(3)} b_{n_3}^{(3)} \right) \quad (2.11)$$

$$= \sum_{n_1} \sum_{n_2} \sum_{n_3} \alpha_{n_1}^{(1)} \alpha_{n_2}^{(2)} \alpha_{n_3}^{(3)} b_{n_1}^{(1)} \otimes b_{n_2}^{(2)} \otimes b_{n_3}^{(3)}. \quad (2.12)$$

Therefore, the representation of \mathcal{V} in the induced basis $\mathcal{B} = \{b_{n_1}^{(1)} \otimes b_{n_2}^{(2)} \otimes b_{n_3}^{(3)} : b_{n_p}^{(p)} \in \mathcal{B}_p\}$ is the $N_1 \times N_2 \times N_3$ hypermatrix $[v_{n_1, n_2, n_3}] = \alpha_{n_1}^{(1)} \alpha_{n_2}^{(2)} \alpha_{n_3}^{(3)}$. \blacksquare

The notion of tensor product is related to a more concrete operation, which is defined for vectors $\mathbf{x}^{(p)} = [x_{n_p}^{(p)}] \in \mathbb{F}^{N_p}$, $p \in \langle P \rangle$, as follows:

$$\otimes : \mathbb{F}^{N_1} \times \dots \times \mathbb{F}^{N_P} \mapsto \mathbb{F}^{N_1 \times \dots \times N_P} \quad (2.13)$$

$$\mathbf{x}^{(1)} \otimes \dots \otimes \mathbf{x}^{(P)} \mapsto [x_{n_1}^{(1)} \dots x_{n_P}^{(P)}]. \quad (2.14)$$

This operation is called *outer product* (or *Segre outer product* [131]). Its definition can be extended for hypermatrices, yielding

$$\otimes : \mathbb{F}^{N_1 \times \dots \times N_P} \times \mathbb{F}^{M_1 \times \dots \times M_Q} \mapsto \mathbb{F}^{N_1 \times \dots \times N_P \times M_1 \times \dots \times M_Q} \quad (2.15)$$

$$[v_{n_1, \dots, n_P}] \otimes [w_{m_1, \dots, m_Q}] \mapsto [v_{n_1, \dots, n_P} w_{m_1, \dots, m_Q}]. \quad (2.16)$$

It is straightforward to verify that it also satisfies the multilinearity properties of the tensor product which are described in Proposition 2.6.

It should be borne in mind that the symbol \otimes is used here in three different senses. First, it denotes the tensor product between vector spaces, which yields a tensor space. Second, it is employed to denote elementary equivalence classes from a tensor space \mathcal{T} whose linear combinations span \mathcal{T} . Third, it stands also for the outer product operation above defined.

Now, turning back to [Example 2.9](#), we observe that the coordinate representation of \mathcal{V} with respect to the basis \mathcal{B} satisfies $[v_{n_1, n_2, n_3}] = [\alpha_{n_1}^{(1)}] \otimes [\alpha_{n_2}^{(2)}] \otimes [\alpha_{n_3}^{(3)}]$. This holds of course generally: the coordinate representation $[v_{n_1, \dots, n_P}]$ of $\mathcal{V} = \sum_{r=1}^R \bigotimes_{p=1}^P v_r^{(p)}$ with respect to the basis \mathcal{B} induced by $\mathcal{B}_1, \dots, \mathcal{B}_P$ is such that

$$v_{n_1, \dots, n_P} = \sum_{r=1}^R \prod_{p=1}^P v_{n_p, r}^{(p)}, \quad (2.17)$$

where $\mathbf{v}_r^{(p)} = [v_{n_p, r}^{(p)}]$ is the representation of $v_r^{(p)}$ with respect to \mathcal{B}_p .

Example 2.10. Let $\mathcal{V}_1 = \mathcal{V}_2 = \mathbb{R}^2$, $\mathcal{V}_3 = \mathbb{R}^3$ and consider the vectors having coordinates

$$\mathbf{v}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \in \mathcal{V}_1, \quad \mathbf{v}_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \in \mathcal{V}_2, \quad \text{and} \quad \mathbf{v}_3 = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix} \in \mathcal{V}_3 \quad (2.18)$$

with respect to some chosen bases \mathcal{B}_p . Under the corresponding induced basis for $\bigotimes_{p=1}^3 \mathcal{V}_p$, the representation $[v_{n_1, n_2, n_3}]$ of the tensor $\mathcal{V} = \mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \mathbf{v}_3$ has coordinates

$$\begin{aligned} v_{1,1,1} &= 6, & v_{2,1,1} &= -6, & v_{1,2,1} &= 4, & v_{2,2,1} &= -4, & v_{1,1,2} &= -6, & v_{2,1,2} &= 6, \\ v_{1,2,2} &= -4, & v_{2,2,2} &= 4, & v_{1,1,3} &= 3, & v_{2,1,3} &= -3, & v_{1,2,3} &= 2, & v_{2,2,3} &= -2. \end{aligned}$$

In terms of the outer product, the representation of \mathcal{V} satisfies $[v_{n_1, n_2, n_3}] = \mathbf{v}_1 \otimes \mathbf{v}_2 \otimes \mathbf{v}_3$. ■

Example 2.11. In the familiar second-order case, a special notation is usually employed for the outer product. Namely, according to the rules of matrix multiplication, $\mathbf{v} \otimes \mathbf{w} = \mathbf{v}\mathbf{w}^T$ holds. So, if we consider, for instance, $\mathcal{V}_1 = \mathbb{R}^{N_1}$ and $\mathcal{V}_2 = \mathbb{R}^{N_2}$ and choose the canonical bases, then

$$\forall \mathbf{v}_r \in \mathbb{R}^{N_1}, \mathbf{w}_r \in \mathbb{R}^{N_2}, \quad \mathcal{V} = \sum_{r=1}^R \mathbf{v}_r \otimes \mathbf{w}_r \quad \text{is represented by} \quad \mathbf{V} = \sum_{r=1}^R \mathbf{v}_r \mathbf{w}_r^T. \quad \blacksquare$$

Despite the fundamental distinction that exists between tensors and their coordinate representations as hypermatrices, [Examples 2.9](#) to [2.11](#) suggest a natural assimilation of these concepts is conceivable when one is not necessarily interested in a particular interpretation in terms of their action (seen as multilinear transformations). This is the case in many practical applications (such as, *e.g.*, in signal processing) where usually what really matters

is the possibility of modeling some given quantities of interest as (finite-dimensional) objects possessing a tensorial structure, to which multilinear transformations are applied. This allows leveraging multilinear algebra results and tensor models for solving practical problems. In such a finite-dimensional setting, one can identify tensors with their representations, assuming some bases have been chosen for all involved vector spaces.

In this thesis, we adopt this viewpoint, with the implicit assumption that a basis has been specified, so that we refer to tensors via their representations. At any rate, most definitions and results we will discuss are invariant with respect to the basis chosen for each vector space, which legitimates attributing them to tensors, and not only to their representations.

2.1.2 Symmetric tensors

A particular case of interest concerns symmetric tensors. A necessary condition for symmetry is, of course, that all vector spaces involved in the tensor product be the same. In this case, we write $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V} = \mathcal{V}^{\otimes P}$ and say that the tensors in \mathcal{T} are *hypercubic*.

Definition 2.12. (Symmetric tensor [131]) A hypercubic tensor $\mathbf{V} \in \mathcal{T} = \mathcal{V}^{\otimes P}$ is said to be symmetric if and only if

$$\mathbf{V} = \sum_{r=1}^R v_r^{(1)} \otimes \cdots \otimes v_r^{(P)} = \sum_{r=1}^R v_r^{(\pi_1)} \otimes \cdots \otimes v_r^{(\pi_P)}, \quad (2.19)$$

for all possible permutations $\pi = (\pi_1, \dots, \pi_P)$ of $(1, \dots, P)$. The set of symmetric tensors is a subspace of \mathcal{T} , which we will denote by $\mathcal{S}(\mathcal{V}^{\otimes P})$.

Clearly, an elementary tensor $\mathbf{V} = \bigotimes_{p=1}^P v^{(p)}$ is symmetric if and only if $v = v^{(1)} = \cdots = v^{(P)}$, in which case we use the notation $\mathbf{V} = v^{\otimes P}$. For general tensors, however, $v_r^{(1)} = \cdots = v_r^{(P)}$ does not necessarily hold in (2.19). For instance, $\mathbf{V} = v \otimes v \otimes w + v \otimes w \otimes v + w \otimes v \otimes v$ is symmetric, but the elementary tensors which constitute it are not.

The above definition has an obvious implication with regard to the representation of a symmetric tensor: for any basis \mathcal{B} of \mathcal{V} , the representation $[v_{n_1, \dots, n_P}]$ of $\mathbf{V} \in \mathcal{S}(\mathcal{V}^{\otimes P})$ satisfies

$$[v_{n_1, \dots, n_P}] = [v_{n_{\pi_1}, \dots, n_{\pi_P}}] \quad (2.20)$$

for any permutation π of $(1, \dots, P)$. Note that this naturally holds provided that a same basis \mathcal{B} is used for all instances of \mathcal{V} appearing in the tensor product $\mathcal{V}^{\otimes P}$.

2.2 Isomorphisms among tensor spaces

Recall that two finite-dimensional vector spaces (over a same field) are isomorphic (denoted by the symbol \simeq) if and only if they have the same dimension. Thus, due to Proposition 2.7, we have for example

$$\mathbb{R}^4 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2 \simeq \mathbb{R}^{16} \simeq \mathbb{R}^4 \otimes \mathbb{R}^4 \simeq \mathbb{R}^{4 \times 4}. \quad (2.21)$$

In spite of these isomorphisms, the tensor structures of the listed spaces are clearly different. For instance, $\mathcal{V} = \mathbf{v}_1 \otimes (\mathbf{v}_2 \otimes \mathbf{v}_3 + \mathbf{w}_2 \otimes \mathbf{w}_3)$ is not an elementary tensor of $\mathbb{R}^4 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2$ in general, but $\mathbf{v}_1 \otimes \Phi(\mathbf{v}_2 \otimes \mathbf{v}_3 + \mathbf{w}_2 \otimes \mathbf{w}_3)$ is an elementary tensor of $\mathbb{R}^4 \otimes \mathbb{R}^4$ for any isomorphism $\Phi : \mathbb{R}^2 \otimes \mathbb{R}^2 \mapsto \mathbb{R}^4$.

When working with tensors, we routinely resort to certain (vector space) isomorphisms among different tensor spaces. This is done for multiple reasons:

- (i) *Analytical ease.* It can be easier or more convenient to manipulate tensor algebraic expressions by invoking isomorphic identities to work with “matricized” or “vectorized” tensors. Some examples of this expedient will be shown in [Section 2.4](#).
- (ii) *Storage and numerical calculus.* When conducting numerical calculations on a computer, it can be convenient to store a tensor (representation) as a long vector or a matrix. Furthermore, by resorting to isomorphic identities, certain tensor operations can be written in terms of matrix-matrix or matrix-vector operations, which allows the use of standard numerical linear algebra routines for their implementation (see [Section 2.4](#)).
- (iii) *Leveraging of matrix tools.* Exploiting isomorphic relations between tensors and matrices allows us to apply certain powerful matrix tools, such as the singular value decomposition (SVD), when performing tensor computations. Indeed, techniques such as matrix decompositions serve as building blocks for many tensor-related algorithms.

For a tensor space defined as $\bigotimes_{p=1}^P \mathbb{F}^{N_p}$, we shall resort to three main isomorphisms:

1. *Mode- p (“flat”) matrix unfolding.* This amounts to a isomorphic identity between a general tensor space and another one where the mode- p vector space is singled out, which can be denoted as

$$\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \simeq \mathbb{F}^{N_p} \otimes (\mathbb{F}^{N_p} \otimes \dots \otimes \mathbb{F}^{N_{p+1}} \otimes \mathbb{F}^{N_{p-1}} \otimes \dots \otimes \mathbb{F}^{N_1}) \quad (2.22)$$

$$\simeq \mathbb{F}^{N_p} \otimes \mathbb{F}^{\prod_{q \neq p} N_q} \simeq \mathbb{F}^{N_p \times \prod_{q \neq p} N_q}. \quad (2.23)$$

Throughout this thesis, we use the notational conventions $\bar{N} = \prod_p N_p$ and $\bar{N}_p = \bar{N}/N_p$, which allow us to write $\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \simeq \mathbb{F}^{N_p \times \bar{N}_p}$. Note that in (2.23) the tensor structure of the space between parentheses of (2.22) is disregarded. Exploiting the above isomorphisms, one can apply matrix techniques when working with tensors of $\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P}$. For instance, a QR decomposition can be employed to compute an orthogonal basis for the mode- p space (see [Section 2.6](#)).

Given a tensor space $\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P}$, we denote the mapping underlying the above described isomorphism by $\Phi_{(p)} : \mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \mapsto \mathbb{F}^{N_p \times \bar{N}_p}$. The image of a tensor \mathcal{V}

is denoted as

$$\Phi_{\langle p \rangle}(\mathbf{V}) = (\mathbf{V})_{\langle p \rangle} = \mathbf{V}_{\langle p \rangle} \in \mathbb{F}^{N_p \times \bar{N}_p}. \quad (2.24)$$

Concretely, given the representation $[v_{n_1, \dots, n_P}]$ of a tensor from $\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P}$, the isomorphism $\Phi_{\langle p \rangle}$ establishes the association

$$v_{n_1, \dots, n_P} = \bar{v}_{n_p, j}, \quad j = 1 + \sum_{q \in \langle P \rangle \setminus \{p\}} (n_q - 1) \prod_{s \in \langle q-1 \rangle \setminus \{p\}} N_s, \quad (2.25)$$

where $\mathbf{V}_{\langle p \rangle} = [\bar{v}_{n_p, j}] \in \mathbb{F}^{N_p \times \bar{N}_p}$. The column ordering implied by the above equation follows the same convention as that of [119]. Any other convention is equally acceptable, provided it is consistently used.

2. *Combined-mode matrix unfolding.* Instead of rearranging the coordinate representation of a tensor in a matrix where the row dimension is associated with a single mode, one can more generally consider the isomorphism

$$\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \simeq \left(\bigotimes_{p \in \mathcal{I}_1} \mathbb{F}^{N_p} \right) \otimes \left(\bigotimes_{p \in \mathcal{I}_2} \mathbb{F}^{N_p} \right), \quad (2.26)$$

where $\{\mathcal{I}_1, \mathcal{I}_2\} \subset 2^{\{1, \dots, P\}}$ is a partition of $\{1, \dots, P\}$. In the particular case where $\mathcal{I}_1 = \{1, \dots, p\}$ and $\mathcal{I}_2 = \{p+1, \dots, P\}$, the corresponding isomorphism is denoted by $\Phi_{[p]} : \mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \mapsto \mathbb{F}^{\prod_{q=1}^p N_q \times \prod_{q=p+1}^P N_q}$, and its image by

$$\Phi_{[p]}(\mathbf{V}) = (\mathbf{V})_{[p]} = \mathbf{V}_{[p]} \in \mathbb{F}^{\prod_{q=1}^p N_q \times \prod_{q=p+1}^P N_q}. \quad (2.27)$$

The adopted convention for indexing the resulting matrix is as follows. Given the representation $[v_{n_1, \dots, n_P}]$ of $\mathbf{V} \in \mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P}$, $\Phi_{[p]}$ implies the association:

$$v_{n_1, \dots, n_P} = \bar{v}_{j_1, j_2}, \quad (2.28)$$

where $\mathbf{V}_{[p]} = [\bar{v}_{j_1, j_2}] \in \mathbb{F}^{\prod_{q=1}^p N_q \times \prod_{q=p+1}^P N_q}$ and

$$j_1 = 1 + \sum_{q=1}^p (n_q - 1) \prod_{s=1}^{q-1} N_s, \quad j_2 = 1 + \sum_{q=p+1}^P (n_q - 1) \prod_{s=p+1}^{q-1} N_s. \quad (2.29)$$

3. *Tensor vectorization.* The simplest exploited isomorphism, written as

$$\mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \simeq \mathbb{F}^{\prod_{p=1}^P N_p} = \mathbb{F}^{\bar{N}}, \quad (2.30)$$

amounts to a vectorization of the tensor coordinate representation. In this case, we represent the mapping by the usual “vec” operator

$$\text{vec} : \mathbb{F}^{N_1} \otimes \dots \otimes \mathbb{F}^{N_P} \mapsto \mathbb{F}^{\bar{N}} \quad (2.31)$$

$$\text{vec}(\mathbf{V}) \mapsto \mathbf{w} = [w_j] \quad \text{such that} \quad v_{n_1, \dots, n_P} = w_j, \quad (2.32)$$

where

$$j = 1 + \sum_{p=1}^P (n_p - 1) \prod_{q=1}^{p-1} N_q.$$

One observation is now in order. Usually, one defines the vec operator in such a way that, given $\mathbf{A} = [\mathbf{a}_1 \ \dots \ \mathbf{a}_{N_2}] \in \mathbb{F}^{N_1 \times N_2}$, it yields $\text{vec}(\mathbf{A}) = [\mathbf{a}_1^T \ \dots \ \mathbf{a}_{N_2}^T]^T \in \mathbb{F}^{N_2 N_1}$. It can be checked that our above definition of vec is then equivalent to $\text{vec}(\mathbf{V}) = \text{vec}(\mathbf{V}_{\langle 1 \rangle})$.

Example 2.13. Let $[v_{n_1, n_2, n_3, n_4}] \in \mathbb{R}^{2 \times 2 \times 2 \times 2}$ be the representation of $\mathbf{V} \in \bigotimes_{p=1}^4 \mathbb{R}^2$. Then,

$$\mathbf{V}_{\langle 1 \rangle} = \begin{bmatrix} v_{1,1,1,1} & v_{1,2,1,1} & v_{1,1,2,1} & v_{1,2,2,1} & v_{1,1,1,2} & v_{1,2,1,2} & v_{1,1,2,2} & v_{1,2,2,2} \\ v_{2,1,1,1} & v_{2,2,1,1} & v_{2,1,2,1} & v_{2,2,2,1} & v_{2,1,1,2} & v_{2,2,1,2} & v_{2,1,2,2} & v_{2,2,2,2} \end{bmatrix},$$

$$\mathbf{V}_{[2]} = \begin{bmatrix} v_{1,1,1,1} & v_{1,1,2,1} & v_{1,1,1,2} & v_{1,1,2,2} \\ v_{2,1,1,1} & v_{2,1,2,1} & v_{2,1,1,2} & v_{2,1,2,2} \\ v_{1,2,1,1} & v_{1,2,2,1} & v_{1,2,1,2} & v_{1,2,2,2} \\ v_{2,2,1,1} & v_{2,2,2,1} & v_{2,2,1,2} & v_{2,2,2,2} \end{bmatrix} \quad \text{and} \quad \text{vec}(\mathbf{V}) = \begin{bmatrix} v_{1,1,1,1} \\ v_{2,1,1,1} \\ v_{1,2,1,1} \\ v_{2,2,1,1} \\ \vdots \\ v_{2,2,2,2} \end{bmatrix}.$$

■

2.3 Tensor norms

We describe now tensor norms which add further structure to tensor spaces.

2.3.1 Induced scalar product and associated Hilbert-Schmidt norm

In this thesis, we shall deal with iterative algorithms which perform sequences of numerical computations with tensor representations, and also with the approximation of tensors by tensor models. For these purposes, it is necessary to introduce a metric representing the notion of distance between elements of a tensor space. This can be addressed by profiting from the natural inner product space structure of the finite-dimensional vector spaces that we consider. More explicitly, we know that \mathbb{F}^{N_p} can be endowed with an operation $\langle \cdot, \cdot \rangle$ which satisfies the axioms characterizing a scalar product. These axioms are notably satisfied by the usual (Euclidean) scalar product defined for all $\mathbf{v} = [v_n]$, $\mathbf{w} = [w_n] \in \mathbb{F}^{N_p}$ as $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_n v_n w_n^*$ (where it is understood that $\alpha^* = \alpha$ for any $\alpha \in \mathbb{R}$). Jointly, the scalar products of the vector spaces \mathbb{F}^{N_p} induce a scalar product over $\bigotimes_{p=1}^P \mathbb{F}^{N_p}$ which is defined as follows.

Definition 2.14 (Induced scalar product). The *induced scalar product* of a tensor space $\bigotimes_{p=1}^P \mathcal{V}_p$ composed of finite-dimensional inner product spaces \mathcal{V}_p is defined for elementary

tensors as

$$\langle \mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_P, \mathbf{w}_1 \otimes \cdots \otimes \mathbf{w}_P \rangle = \prod_{p=1}^P \langle \mathbf{v}_p, \mathbf{w}_p \rangle. \quad (2.33)$$

Its extension to non-elementary tensors follows from combining the axioms $\langle \mathcal{V}, \mathcal{W} \rangle = (\langle \mathcal{W}, \mathcal{V} \rangle)^*$ and $\langle \alpha \mathcal{V} + \beta \mathcal{V}', \mathcal{W} \rangle = \alpha \langle \mathcal{V}, \mathcal{W} \rangle + \beta \langle \mathcal{V}', \mathcal{W} \rangle$ with (2.33).

The induced scalar product is unique [95]. Also, it is easy to verify that under its definition two elementary tensors $\bigotimes_{p=1}^P \mathbf{v}_p$ and $\bigotimes_{p=1}^P \mathbf{w}_p$ are orthogonal if and only if $\mathbf{v}_p \perp \mathbf{w}_p$ holds for at least one $p \in \langle P \rangle$.

Endowing a vector space with a scalar product naturally gives rise to a norm. In the case of a finite-dimensional tensor space $\bigotimes_{p=1}^P \mathcal{V}_p$, if one picks the scalar product induced by the Euclidean scalar products of the spaces \mathcal{V}_p , this norm is as follows.

Definition 2.15 (Tensor Frobenius norm). The norm associated to the induced Euclidean scalar product, usually called *Frobenius* or *Hilbert-Schmidt* norm, is given by

$$\|\mathcal{V}\|_F \triangleq \sqrt{\langle \mathcal{V}, \mathcal{V} \rangle}, \quad (2.34)$$

satisfying the usual axioms for all $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$.

For elementary tensors, it is easy to verify that $\|\mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_P\|_F = \prod_{p=1}^P \|\mathbf{v}_p\|_2$. The name ‘‘Frobenius norm’’ is used because it can be seen as an extension of the matrix Frobenius norm. Indeed, in terms of the coordinate representation $[v_{n_1, \dots, n_P}]$ of a tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ with respect to the induced canonical basis, we have

$$\|\mathcal{V}\|_F = \sqrt{\sum_{n_1} \cdots \sum_{n_P} |v_{n_1, \dots, n_P}|^2}. \quad (2.35)$$

Henceforth, every time we claim a certain tensor \mathcal{V} is ‘‘approximated’’ by another one $\hat{\mathcal{V}}$, denoted by $\mathcal{V} \approx \hat{\mathcal{V}}$, we refer to the fact that $\|\mathcal{V} - \hat{\mathcal{V}}\|_F$ is ‘‘sufficiently small’’ in some meaningful sense (*e.g.*, from the point of view of some application).

2.3.2 Other norms

We define now other notions of norm by identifying a finite-dimensional tensor with its coordinate representation under the induced canonical basis.

Definition 2.16 (Hölder p -norm [131]). Let \mathcal{V} be a finite dimensional P th-order tensor. A generalization of the Frobenius norm is the Hölder p -norm, given by

$$\|\mathcal{V}\|_{H,p} = \left(\sum_{n_1, \dots, n_P} |v_{n_1, \dots, n_P}|^p \right)^{\frac{1}{p}}, \quad (2.36)$$

for $p \in [1, \infty[$, and by

$$\|\mathbf{V}\|_{H,\infty} = \max_{n_1, \dots, n_P} |v_{n_1, \dots, n_P}| \quad (2.37)$$

for $p = \infty$. Note that, for $p = 2$ we have $\|\mathbf{V}\|_{H,2} = \|\mathbf{V}\|_F$.

From the above definition, it follows that the induced scalar product defined in the previous section satisfies the Hölder inequality

$$|\langle \mathbf{V}, \mathbf{W} \rangle| \leq \|\mathbf{V}\|_{H,p} \|\mathbf{W}\|_{H,q}, \quad \forall p, q \in [1, \infty] \text{ such that } \frac{1}{p} + \frac{1}{q} = 1, \quad (2.38)$$

which reduces to the Cauchy-Schwartz inequality when $p = q = 2$. This is easily shown by exploiting the evident fact that $\|\mathbf{V}\|_{H,p} = \|\text{vec}(\mathbf{V})\|_p$, where the latter is the ℓ_p norm.

When tensors are interpreted as multilinear functionals, one can define an *operator* or *spectral* norm in the usual sense used for linear operators.

Definition 2.17 (Spectral norm). Let $\mathbf{V} \in \mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$, where \mathcal{V}_p are finite-dimensional spaces. The spectral norm of \mathbf{V} is given by [133]

$$\|\mathbf{V}\|_2 = \max \left\{ \left| \left\langle \mathbf{V}, \bigotimes_{p=1}^P u_p \right\rangle \right| : u_p \in \mathcal{V}_p, \|u_p\|_2 = 1 \right\}. \quad (2.39)$$

Finally, we introduce the norm which is the dual of the spectral norm [133].

Definition 2.18 (Nuclear norm). The *tensor nuclear norm* (or Schatten 1-norm [131]) of a finite-dimensional P th-order tensor \mathbf{V} is defined as

$$\|\mathbf{V}\|_* = \min \left\{ \sum_{r=1}^R |\sigma_r| : \mathbf{V} = \sum_{r=1}^R \sigma_r \bigotimes_{p=1}^P u_r^{(p)}, \|u_r^{(p)}\|_2 = 1, R \in \mathbb{N} \right\}. \quad (2.40)$$

Using these definitions, the above mentioned duality property can be expressed as

$$\|\mathbf{V}\|_2 = \max\{|\langle \mathbf{V}, \mathbf{W} \rangle| : \|\mathbf{W}\|_* = 1\} \quad \text{and} \quad \|\mathbf{V}\|_* = \max\{|\langle \mathbf{V}, \mathbf{W} \rangle| : \|\mathbf{W}\|_2 = 1\}. \quad (2.41)$$

As a final remark, we point out that both the spectral and the nuclear norms are NP-hard² to compute [102, 87], which has important consequences for some applications.

2.4 Tensor-related algebraic operations

We now turn to the description of important tensor algebraic operations. To this end, the following ancillary mathematical definitions shall be used. The P th-order Kronecker delta

²The term ‘‘NP-hard,’’ where NP stands for ‘‘nondeterministic polynomial time,’’ comes from computational complexity theory. Informally speaking, the class of NP-hard problems encompasses all problems that are ‘‘as hard as any problem from NP,’’ in the sense that a solution of any NP-hard problem provides a solution to *all* problems from NP. In its turn, the class of NP problems contains the decision problems whose solution can be verified in polynomial time by a deterministic Turing machine. See, *e.g.*, [5] for formal definitions.

is defined as

$$\delta_{i_1, i_2, \dots, i_P} = \begin{cases} 1, & i_1 = \dots = i_P, \\ 0, & i_p \neq i_q \text{ for some } p, q \in \langle P \rangle. \end{cases} \quad (2.42)$$

For any $N_1, N_2 \in \mathbb{N}$, the symbol $\mathbf{I}_{N_1 \times N_2}$ shall denote the matrix $\mathbf{I}_{N_1 \times N_2} = [\delta_{n_1, n_2}] \in \mathbb{F}^{N_1 \times N_2}$. If $N_1 = N_2$, we write $\mathbf{I}_{N \times N} = \mathbf{I}_N$. When N_1 and N_2 are clear from the context, we use simply \mathbf{I} . The operator $\text{Diag} : \mathbb{F}^N \mapsto \mathbb{F}^{N \times N}$ is defined for any $\mathbf{b} = [b_n] \in \mathbb{F}^N$ as

$$\mathbf{b} \mapsto [\text{Diag}(\mathbf{b})_{n_1, n_2}] = [b_{n_1} \delta_{n_1, n_2}]. \quad (2.43)$$

2.4.1 Kronecker product

Many tensor identities can be isomorphically written as expressions involving Kronecker products of vectors and matrices. This is useful both for analysis and numerical computation purposes. So, let us recall this operation and its basic properties.

Definition 2.19 (Kronecker product). For any pair of matrix spaces $\mathbb{F}^{N_1 \times N_2}$ and $\mathbb{F}^{M_1 \times M_2}$, the Kronecker product is defined as

$$\boxtimes : \mathbb{F}^{N_1 \times N_2} \times \mathbb{F}^{M_1 \times M_2} \mapsto \mathbb{F}^{N_1 M_1 \times N_2 M_2} \quad (2.44)$$

$$\mathbf{A} \boxtimes \mathbf{B} = \begin{bmatrix} a_{1,1} \mathbf{B} & a_{1,2} \mathbf{B} & \dots & a_{1,N_2} \mathbf{B} \\ a_{2,1} \mathbf{B} & a_{2,2} \mathbf{B} & \dots & a_{2,N_2} \mathbf{B} \\ \vdots & \vdots & \dots & \vdots \\ a_{N_1,1} \mathbf{B} & a_{N_1,2} \mathbf{B} & \dots & a_{N_1,N_2} \mathbf{B} \end{bmatrix} \quad (2.45)$$

for all $\mathbf{A} \in \mathbb{F}^{N_1 \times N_2}$ and $\mathbf{B} \in \mathbb{F}^{M_1 \times M_2}$.

Proposition 2.20 (Properties of Kronecker product). The Kronecker product satisfies:

$$\text{(Distributivity w.r.t. addition)} \quad \mathbf{A} \boxtimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \boxtimes \mathbf{B} + \mathbf{A} \boxtimes \mathbf{C}, \quad (2.46)$$

$$\text{(Associativity)} \quad \mathbf{A} \boxtimes (\mathbf{B} \boxtimes \mathbf{C}) = (\mathbf{A} \boxtimes \mathbf{B}) \boxtimes \mathbf{C}, \quad (2.47)$$

$$\text{(Mixed-product property)} \quad (\mathbf{A} \boxtimes \mathbf{B})(\mathbf{C} \boxtimes \mathbf{D}) = (\mathbf{AC}) \boxtimes (\mathbf{BD}), \quad (2.48)$$

$$\text{(Relation with vectorization)} \quad \text{vec}(\mathbf{DEF}) = (\mathbf{F}^T \boxtimes \mathbf{D}) \text{vec}(\mathbf{E}), \quad (2.49)$$

$$\text{(Relation with transposition)} \quad (\mathbf{A} \boxtimes \mathbf{B})^T = \mathbf{A}^T \boxtimes \mathbf{B}^T, \quad (2.50)$$

$$\text{(Relation with inversion)} \quad (\mathbf{X} \boxtimes \mathbf{Y})^{-1} = \mathbf{X}^{-1} \boxtimes \mathbf{Y}^{-1} \quad (2.51)$$

$$\text{(Relation with conjugation)} \quad (\mathbf{A} \boxtimes \mathbf{B})^* = \mathbf{A}^* \boxtimes \mathbf{B}^*, \quad (2.52)$$

assuming the matrices in (2.48) and (2.49) have compatible dimensions allowing the displayed matrix products and that \mathbf{X}, \mathbf{Y} are square nonsingular matrices.

Proof. Properties (2.46), (2.47), (2.50) and (2.52) are obvious. Proofs for the other ones are found in [126, 148]. \square

Proposition 2.21 (Relation with isomorphisms). With regard to the isomorphisms introduced in Section 2.2, we have the following identities for $\mathbf{v}^{(p)} \in \mathbb{F}^{N_p}$, $p \in \langle P \rangle$:

$$\left(\mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(P)} \right)_{\langle p \rangle} = \mathbf{v}^{(p)} \left(\mathbf{v}^{(P)} \boxtimes \cdots \boxtimes \mathbf{v}^{(p+1)} \boxtimes \mathbf{v}^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}^{(1)} \right)^T, \quad (2.53)$$

$$\left(\mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(P)} \right)_{[p]} = \left(\mathbf{v}^{(p)} \boxtimes \cdots \boxtimes \mathbf{v}^{(1)} \right) \left(\mathbf{v}^{(P)} \boxtimes \cdots \boxtimes \mathbf{v}^{(p+1)} \right)^T, \quad (2.54)$$

$$\text{vec} \left(\mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(P)} \right) = \mathbf{v}^{(P)} \boxtimes \cdots \boxtimes \mathbf{v}^{(1)}. \quad (2.55)$$

Proof. Denote $\mathbf{v}^{(p)} = \left[v_{n_p}^{(p)} \right]$. By definition, the indexing convention adopted for $\Phi_{\langle p \rangle}$ implies that the (n_p, j) th element of $\left(\mathbf{v}^{(1)} \otimes \cdots \otimes \mathbf{v}^{(P)} \right)_{\langle p \rangle} \in \mathbb{F}^{N_p \times \tilde{N}_p}$ is given by $\prod_p v_{n_p}^{(p)}$, with

$$j = n_1 + \sum_{q=2}^{p-1} (n_q - 1) \prod_{r=1}^{q-1} N_r + (n_{p+1} - 1) \prod_{q=1}^{p-1} N_q + \sum_{q=p+2}^P (n_q - 1) \left(\prod_{r=1}^{p-1} N_r \right) \left(\prod_{r=p+1}^{q-1} N_r \right).$$

It can be checked that this equals the (n_p, j) th element of $\mathbf{v}^{(p)} \left(\mathbf{v}^{(P)} \boxtimes \cdots \boxtimes \mathbf{v}^{(p+1)} \boxtimes \mathbf{v}^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}^{(1)} \right)^T \in \mathbb{F}^{N_p \times \tilde{N}_p}$, which shows (2.53). The proofs of (2.54) and (2.55) are similar. \square

Corollary 2.22. Letting $\mathbf{V} = \sum_{r=1}^R \mathbf{v}_r^{(1)} \otimes \cdots \otimes \mathbf{v}_r^{(P)}$, we can write

$$\mathbf{V}_{\langle p \rangle} = \sum_{r=1}^R \mathbf{v}_r^{(p)} \left(\mathbf{v}_r^{(P)} \boxtimes \mathbf{v}_r^{(P-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(p+1)} \boxtimes \mathbf{v}_r^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(1)} \right)^T \quad (2.56)$$

$$\mathbf{V}_{[p]} = \sum_{r=1}^R \left(\mathbf{v}_r^{(p)} \boxtimes \mathbf{v}_r^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(1)} \right) \left(\mathbf{v}_r^{(P)} \boxtimes \mathbf{v}_r^{(P-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(p+1)} \right)^T \quad (2.57)$$

$$\text{vec}(\mathbf{V}) = \sum_{r=1}^R \mathbf{v}_r^{(P)} \boxtimes \mathbf{v}_r^{(P-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(1)}. \quad (2.58)$$

Proof. Follows from (2.53)–(2.55) via linearity of $\Phi_{\langle p \rangle}$, $\Phi_{[p]}$ and $\text{vec}(\cdot)$. \square

2.4.2 Khatri-Rao product

Definition 2.23 (Khatri-Rao product). For any pair of matrix spaces $\mathbb{F}^{N_1 \times N_2}$ and $\mathbb{F}^{M_1 \times N_2}$, we define the Khatri-Rao product as

$$\diamond : \mathbb{F}^{N_1 \times N_2} \times \mathbb{F}^{M_1 \times N_2} \mapsto \mathbb{F}^{N_1 M_1 \times N_2} \quad (2.59)$$

$$\mathbf{A} \diamond \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \boxtimes \mathbf{b}_1 & \mathbf{a}_2 \boxtimes \mathbf{b}_2 & \cdots & \mathbf{a}_{N_2} \boxtimes \mathbf{b}_{N_2} \end{bmatrix}, \quad (2.60)$$

where \mathbf{a}_n denotes the n th column of \mathbf{A} , and similarly for \mathbf{B} .

Due to the above definition, the Khatri-Rao product is referred to as a “columnwise Kronecker product.” Similarly to the Kronecker product, it has the following properties.

Proposition 2.24 (Properties of Khatri-Rao product). For matrices of appropriate dimensions, the Khatri-Rao product satisfies:

$$\text{(Distributivity w.r.t. addition)} \quad \mathbf{A} \diamond (\mathbf{B} + \mathbf{C}) = \mathbf{A} \diamond \mathbf{B} + \mathbf{A} \diamond \mathbf{C}, \quad (2.61)$$

$$\text{(Associativity)} \quad \mathbf{A} \diamond (\mathbf{B} \diamond \mathbf{C}) = (\mathbf{A} \diamond \mathbf{B}) \diamond \mathbf{C}, \quad (2.62)$$

$$\text{(Relation with Kronecker)} \quad \mathbf{A} \diamond \mathbf{B} = (\mathbf{A} \boxtimes \mathbf{B})(\mathbf{I}_{N_2} \diamond \mathbf{I}_{N_2}), \quad (2.63)$$

$$\text{(Relation with vectorization)} \quad \text{vec}(\mathbf{A} \text{Diag}(\mathbf{b})\mathbf{C}) = (\mathbf{C}^T \diamond \mathbf{A})\mathbf{b}, \quad (2.64)$$

where in (2.63), $\mathbf{A} \in \mathbb{F}^{N_1 \times N_2}$ and $\mathbf{B} \in \mathbb{F}^{M_1 \times N_2}$.

Proof. Properties (2.61) and (2.62) are straightforward; (2.63) follows from the structure of $(\mathbf{I}_{N_2} \diamond \mathbf{I}_{N_2})$; (2.64) follows from (2.49), (2.63) and the fact that $\text{vec}(\text{Diag}(\mathbf{b})) = (\mathbf{I}_N \diamond \mathbf{I}_N)\mathbf{b}$. \square

2.4.3 Mode- p product

The mode- p product consists in applying a linear transformation acting upon one of the vector spaces which constitute the underlying tensor space (*i.e.*, it acts upon one mode). The formal definition and the used notation are as follows.

Definition 2.25 (Mode- p product). Let $\mathcal{V}_p, \mathcal{W}_p$ be vector spaces and $\mathcal{L}(\mathcal{V}_p, \mathcal{W}_p)$ be the space of linear transformations from \mathcal{V}_p to \mathcal{W}_p . The mode- p product defined over the tensor space $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$ is the mapping

$$\times_p : \mathcal{T} \times \mathcal{L}(\mathcal{V}_p, \mathcal{W}_p) \mapsto \left(\bigotimes_{q=1}^{p-1} \mathcal{V}_q \right) \otimes \mathcal{W}_p \otimes \left(\bigotimes_{q=p+1}^P \mathcal{V}_q \right) \quad (2.65)$$

$$\mathbf{v} \times_p \mathbf{A}_p = \sum_{r=1}^R \left(\bigotimes_{q=1}^{p-1} v_r^{(q)} \right) \otimes \mathbf{A}_p \left(v_r^{(p)} \right) \otimes \left(\bigotimes_{q=p+1}^P v_r^{(q)} \right), \quad (2.66)$$

where $\mathbf{v} = \sum_{r=1}^R v_r^{(1)} \otimes \cdots \otimes v_r^{(P)} \in \mathcal{T}$ and $\mathbf{A}_p \in \mathcal{L}(\mathcal{V}_p, \mathcal{W}_p)$.

In a finite-dimensional setting where $\mathcal{V}_p = \mathbb{F}^{N_p}$ and $\mathcal{W}_p = \mathbb{F}^{M_p}$, letting $\mathbf{A}_p \in \mathbb{F}^{M_p \times N_p}$ be a matrix representing $\mathbf{A}_p \in \mathcal{L}(\mathcal{V}_p, \mathcal{W}_p)$ with respect to the chosen bases, one can write (2.66) more concretely as

$$\mathbf{v} \times_p \mathbf{A}_p = \sum_{r=1}^R \left(\bigotimes_{q=1}^{p-1} \mathbf{v}_r^{(q)} \right) \otimes \left(\mathbf{A}_p \mathbf{v}_r^{(p)} \right) \otimes \left(\bigotimes_{q=p+1}^P \mathbf{v}_r^{(q)} \right), \quad (2.67)$$

where $\mathbf{v} = \sum_{r=1}^R \bigotimes_{q=1}^P \mathbf{v}_r^{(q)} \in \mathcal{T}$, with $\mathbf{v}_r^{(q)} \in \mathbb{F}^{N_q}$. This is why this operation is referred to as “mode- p product.”

The mode- p product is an operation of central importance when dealing with tensors. In the first place, it can be used to carry out a change of basis with respect to one of the modes. Secondly, it is the key ingredient in the subspace-based representation of a tensor called Tucker decomposition, which we shall see ahead. We now derive its main properties.

Proposition 2.26 (Commutativity of mode- p products over distinct modes). For $p, q \in \langle P \rangle$ such that $p \neq q$, the mode- p product satisfies

$$(\mathbf{V} \times_p A_p) \times_q A_q = (\mathbf{V} \times_q A_q) \times_p A_p. \quad (2.68)$$

Proof. The proof follows easily from (2.66), by exploiting the fact that the operator \times_p acts only on the p th mode, preserving the vectors associated with the other modes. \square

Because of the above property, we usually do not employ parentheses as those of (2.68). For brevity, we shall also employ the notation

$$\mathbf{V} \times_{p=1}^P A_p \triangleq \mathbf{V} \times_1 A_1 \times_2 \cdots \times_P A_P. \quad (2.69)$$

Expression (2.69) is referred to as a *multilinear transformation* of \mathbf{V} because it consists of a joint application of P transformations, each one being linear with respect to one mode.

Proposition 2.27 (Composition of mode- p products over a same mode). For any $p \in \langle P \rangle$, the mode- p product satisfies

$$(\mathbf{V} \times_p A_p) \times_p A'_p = \mathbf{V} \times_p (A'_p \circ A_p), \quad (2.70)$$

where $A_p \in \mathcal{L}(\mathcal{V}_p, \mathcal{W}_p)$, $A'_p \in \mathcal{L}(\mathcal{W}_p, \mathcal{U}_p)$ and $A'_p \circ A_p \in \mathcal{L}(\mathcal{V}_p, \mathcal{U}_p)$ denotes the composite linear map $(A'_p \circ A_p)(v_p) = A'_p(A_p(v_p))$.

Proof. Similar to that of Proposition 2.26. \square

Proposition 2.28 (Distributivity of mode- p product with respect to addition). For any $p \in \langle P \rangle$, the mode- p product satisfies both

$$(\mathbf{V} + \mathbf{V}') \times_p A_p = \mathbf{V} \times_p A_p + \mathbf{V}' \times_p A_p, \quad (2.71)$$

$$\mathbf{V} \times_p (A_p + A'_p) = \mathbf{V} \times_p A_p + \mathbf{V} \times_p A'_p. \quad (2.72)$$

Proof. The proof of (2.71) follows trivially from Definition 2.25, while that of (2.72) is obtained by combining Definition 2.25 with the multilinearity of the tensor product. \square

It is worth noting that (2.72) implies that the expression (2.69) is multilinear with respect to A_1, \dots, A_P .

Proposition 2.29 (Relation between mode- p product and matrix unfolding isomorphism). Let $\mathcal{V} \in \bigotimes_{q=1}^P \mathbb{F}^{N_q}$. For any $p \in \langle P \rangle$ and $\mathbf{A}_p \in \mathbb{F}^{M_p \times N_p}$, we have the isomorphic association

$$\mathcal{W} = \mathcal{V} \times_p \mathbf{A}_p \quad \Leftrightarrow \quad \mathbf{W}_{\langle p \rangle} = \mathbf{A}_p \mathbf{V}_{\langle p \rangle}. \quad (2.73)$$

Proof. First, write $\mathcal{V} = \sum_{r=1}^R \mathbf{v}_r^{(1)} \otimes \cdots \otimes \mathbf{v}_r^{(P)}$. Then, by definition,

$$\mathcal{W} = \sum_{r=1}^R \left(\bigotimes_{q=1}^{p-1} \mathbf{v}_r^{(q)} \right) \otimes \left(\mathbf{A}_p \mathbf{v}_r^{(p)} \right) \otimes \left(\bigotimes_{q=p+1}^P \mathbf{v}_r^{(q)} \right). \quad (2.74)$$

By exploiting relation (2.53), we can write

$$\mathbf{W}_{\langle p \rangle} = \sum_{r=1}^R \left(\mathbf{A}_p \mathbf{v}_r^{(p)} \right) \left(\mathbf{v}_r^{(P)} \boxtimes \mathbf{v}_r^{(P-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(p+1)} \boxtimes \mathbf{v}_r^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(1)} \right)^T \quad (2.75)$$

$$= \mathbf{A}_p \left[\sum_{r=1}^R \mathbf{v}_r^{(p)} \left(\mathbf{v}_r^{(P)} \boxtimes \mathbf{v}_r^{(P-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(p+1)} \boxtimes \mathbf{v}_r^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{v}_r^{(1)} \right)^T \right] \quad (2.76)$$

$$= \mathbf{A}_p \mathbf{V}_{\langle p \rangle}. \quad (2.77)$$

□

Remark 2.30. The above property is clearly independent of the particular ordering adopted for the column indexing of the matrix unfolding.

Corollary 2.31. Let $\mathcal{V} \in \bigotimes_{q=1}^P \mathbb{F}^{N_q}$, $\mathcal{W} \in \bigotimes_{q=1}^P \mathbb{F}^{M_q}$ and $\mathbf{A}_q \in \mathbb{F}^{M_q \times N_q}$, $q \in \langle P \rangle$. Then, for any $p \in \langle P \rangle$, we have the relation

$$\mathcal{W} = \mathcal{V} \times_{q=1}^P \mathbf{A}_q \quad \Leftrightarrow \quad \mathbf{W}_{\langle p \rangle} = \mathbf{A}_p \mathbf{V}_{\langle p \rangle} \left(\mathbf{A}_P \boxtimes \cdots \boxtimes \mathbf{A}_{p+1} \boxtimes \mathbf{A}_{p-1} \boxtimes \cdots \boxtimes \mathbf{A}_1 \right)^T. \quad (2.78)$$

Proof. Follows from the properties of the Kronecker product. □

Example 2.32. It is easy to check that $\forall \mathcal{V} \in \mathbb{F}^{N_1} \otimes \mathbb{F}^{N_2}$, $\mathbf{V} \in \mathbb{F}^{N_1 \times N_2}$, $\mathbf{A}_p \in \mathbb{R}^{M_p \times N_p}$, with $p \in \{1, 2\}$, the natural isomorphic relation $\mathcal{V} \simeq \mathbf{V}$ implies $\mathcal{V} \times_1 \mathbf{A}_1 \times_2 \mathbf{A}_2 \simeq \mathbf{A}_1 \mathbf{V} \mathbf{A}_2^T$. ■

Finally, we note that the action of a tensor as a multilinear functional can be expressed in terms of mode- p products. Indeed, it can be easily checked that the expression

$$\mathcal{V}(\mathbf{w}_1, \dots, \mathbf{w}_P) = \mathcal{V} \times_{p=1}^P \mathbf{w}_p^T \quad (2.79)$$

defines a multilinear functional which is associated with $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$. In the above notation, it is understood that \mathcal{V} is represented under the same bases as the vectors $\mathbf{w}_p \in \mathbb{F}^{N_p}$.

2.5 Tensor rank and the canonical polyadic decomposition (CPD)

The concept of rank is of pivotal significance in linear algebra. It can be thought of as a measure of complexity of linear transformations and it is connected to certain properties of a matrix representation, such as its row and column spaces. When it comes to tensors, there are multiple ways of extending this concept, each one yielding a complexity measure with different properties. In this section, we review the tensor rank, a notion due to Hitchcock [104] which is intimately related to minimal representations³ of a tensor as a linear combination of elementary tensors.

Definition 2.33 (Polyadic decomposition). Any representation of a tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ having the form $\mathcal{V} = \sum_{s=1}^S v_s^{(1)} \otimes \cdots \otimes v_s^{(P)}$ is called a *polyadic decomposition* of \mathcal{V} .

Definition 2.34 (Tensor rank [104]). The rank of a tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$, denoted by $\text{rank}(\mathcal{V})$, is given by the smallest integer R such that a polyadic decomposition of \mathcal{V} with R terms exists, *i.e.*, such that we can write

$$\mathcal{V} = \sum_{r=1}^R v_r^{(1)} \otimes \cdots \otimes v_r^{(P)} \quad (2.80)$$

for some $v_r^{(p)} \in \mathcal{V}_p$, $p \in \langle P \rangle$.

It follows from [Definition 2.34](#) that a tensor is elementary if and only if it is a rank-one tensor. This is clearly consistent with the matrix rank, since a rank- R matrix is given by a sum of (not less than) R rank-one matrices. Despite this conceptual similarity, many properties of the tensor rank are in sharp contrast with those of the matrix rank, as we shall discuss in [Section 2.5.2](#).

The terminology “tensor rank” is justified because the rank is an intrinsic property of a tensors itself and not merely of one of its possible coordinate representations, since it is invariant with respect to changes of basis [133, 69]. This can be shown as follows.

Proposition 2.35 (Invariance of tensor rank with respect to coordinate bases). The identity

$$\text{rank} \left(\mathcal{V} \underset{p=1}{\times}^P \mathbf{A}_p \right) = \text{rank}(\mathcal{V}) \quad (2.81)$$

holds for every tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and any nonsingular matrix $\mathbf{A}_p \in \mathbb{F}^{N_p \times N_p}$.

Proof. The following argument is due to [95, Lemma 3.36]. From [Definition 2.25](#), it is seen that the multilinear transformation applied in (2.81) cannot increase the number of terms

³The word “representation” here refers to a way of writing the tensor as a sum of elementary ones, and not to the coordinate representation of a tensor.

in the sum (2.80). Hence, $\text{rank}\left(\mathbf{V} \times_{p=1}^P \mathbf{A}_p\right) \leq \text{rank}(\mathbf{V})$. But $\left(\mathbf{V} \times_{p=1}^P \mathbf{A}_p\right) \times_{p=1}^P \mathbf{A}_p^{-1} = \mathbf{V}$, and thus the chain of inequalities

$$\text{rank}(\mathbf{V}) = \text{rank}\left(\left(\mathbf{V} \times_{p=1}^P \mathbf{A}_p\right) \times_{p=1}^P \mathbf{A}_p^{-1}\right) \leq \text{rank}\left(\mathbf{V} \times_{p=1}^P \mathbf{A}_p\right) \leq \text{rank}(\mathbf{V}) \quad (2.82)$$

shows (2.81). \square

2.5.1 The canonical polyadic decomposition and its properties

We proceed now to the definition of the canonical polyadic decomposition (CPD), of which a constrained (structured) version will be the central subject of Part I of this thesis.

Definition 2.36 (Canonical polyadic decomposition [104, 39, 100]). A polyadic decomposition of a tensor $\mathbf{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ having $R = \text{rank}(\mathbf{V})$ terms is called⁴ a *canonical polyadic decomposition* of \mathbf{V} . It may be more convenient to write the CPD of a tensor as

$$\mathbf{V} = \sum_{r=1}^R \lambda_r a_r^{(1)} \otimes \cdots \otimes a_r^{(P)}, \quad (2.83)$$

where each vector $a_r^{(p)}$ satisfies $\|a_r^{(p)}\| = 1$ for some chosen norm (typically, the Euclidean norm) $\|\cdot\|$ of \mathcal{V}_p and $\lambda_r \in \mathbb{R}$ (or, without loss of generality, $\lambda_r \in \mathbb{R}_+$).

Because the CPD of a tensor exposes its rank, it is said to be a *rank-revealing decomposition* or *rank-retaining decomposition* [131].

If a finite-dimensional tensor $\mathbf{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ admits a CPD of rank R , then it is clear that any associated hypermatrix coordinate representation also admits an analogous decomposition written as a sum of outer products of vectors

$$[v_{n_1, \dots, n_P}] = \sum_{r=1}^R \lambda_r \mathbf{a}_r^{(1)} \otimes \cdots \otimes \mathbf{a}_r^{(P)}, \quad (2.84)$$

where, for all $p \in \langle P \rangle$, $\mathbf{a}_r^{(p)} \in \mathbb{F}^{N_p}$ is a coordinate representation of $a_r^{(p)} \in \mathcal{V}_p$ under the chosen basis for \mathcal{V}_p . When assimilating \mathbf{V} with $[v_{n_1, \dots, n_P}]$, we shall equal \mathbf{V} to expression (2.84). In scalar form, from (2.17) we have

$$v_{n_1, \dots, n_P} = \sum_{r=1}^R \lambda_r \prod_{p=1}^P a_{n_p, r}^{(p)}. \quad (2.85)$$

⁴The acronym CPD can also stand for Candecomp (from canonical decomposition) / PARAFAC (from parallel factors) decomposition. These two names were attributed to it when it was independently rediscovered in the seventies [39, 100].

Definition 2.37 (Matrix factor of a CPD). Given a CPD of the form (2.84), the vectors $\mathbf{a}_r^{(p)}$, $r \in \langle R \rangle$, can be thought of as columns of a *matrix factor* $\mathbf{A}^{(p)} = [\mathbf{a}_1^{(p)} \ \dots \ \mathbf{a}_R^{(p)}] \in \mathbb{F}^{N_p \times R}$. The following simplified notation (often referred to as Kruskal's notation) is then employed

$$\mathbf{V} = \llbracket \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket, \quad (2.86)$$

where $\boldsymbol{\lambda} = [\lambda_r] \in \mathbb{R}^R$. When no constraint is imposed over the norm of the vectors $\mathbf{a}^{(p)}$ and the CPD is written without the scaling factors λ_r , we denote it by $\mathbf{V} = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket$.

By resorting to Corollary 2.22, it can be easily shown that the mode- p matrix unfolding of \mathbf{V} in (2.86) is given by

$$\mathbf{V}_{\langle p \rangle} = \mathbf{A}^{(p)} \text{Diag}(\boldsymbol{\lambda}) \left(\mathbf{A}^{(P)} \diamond \dots \diamond \mathbf{A}^{(p+1)} \diamond \mathbf{A}^{(p-1)} \diamond \dots \diamond \mathbf{A}^{(1)} \right)^T. \quad (2.87)$$

Applying property (2.64) to (2.87) with $p = 1$, we can derive yet another useful identity:

$$\text{vec}(\mathbf{V}) = \text{vec}(\mathbf{V}_{\langle 1 \rangle}) = \left(\mathbf{A}^{(P)} \diamond \dots \diamond \mathbf{A}^{(1)} \right) \boldsymbol{\lambda}. \quad (2.88)$$

Finally, from Corollary 2.22 we have also the property

$$\mathbf{V}_{[p]} = \left(\mathbf{A}^{(p)} \diamond \dots \diamond \mathbf{A}^{(1)} \right) \text{Diag}(\boldsymbol{\lambda}) \left(\mathbf{A}^{(P)} \diamond \dots \diamond \mathbf{A}^{(p+1)} \right)^T. \quad (2.89)$$

2.5.1.1 Applications of the CPD

In applied domains, the computation of a CPD is typically sought with one of the following major goals:

1. *Solving multilinear inverse problems.* The CPD can be computed for estimating quantities of interest which are represented by the vectors involved in the outer products. In other words, it arises in inverse problems with multilinear models, which may involve additional constraints. Problems of this nature can be found in, *e.g.*, telecommunications [179, 84, 83], chemometrics [25, 183] and blind source separation [61, 151]. In this context, an attractive feature of the CPD for $P \geq 3$ is that its uniqueness properties are *stronger* than those of the dyadic decomposition (or bilinear decomposition) of a matrix, as we will discuss in Section 2.5.1.3.
2. *Exploratory data analysis.* The extraction of meaningful quantitative information in the form of constituents of a CPD model is an increasingly employed tool in a wide range of disciplines from psychology to web mining. We refer the reader to [3, 45, 145] and references therein for other examples.
3. *Memory complexity reduction.* For large tensors of sufficiently low rank (or which admit a low-rank approximation, an aspect which we shall discuss in Section 2.5.2.5), one can exploit the CPD in order to save storage memory. Indeed, assuming a tensor $\mathbf{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$

has a CPD with R terms, then it can be stored using $\mathcal{O}(R \sum_p N_p)$ memory, which represents a significant saving over \bar{N} when $R \ll \min_p N_p$. Examples of this use can be found in [56, 79].

4. *Computational complexity reduction.* When tensors are regarded as multilinear transformations or multilinear functionals, then a CPD can be exploited to reduce the computational complexity involved in its application, provided its rank is sufficiently low.

A celebrated example of item 4 concerns Strassen's algorithm for matrix multiplication. This bilinear operator can be regarded as a third-order tensor whose rank determines its fundamental arithmetic complexity. Strassen has developed an algorithm for computing the product of two 2×2 matrices using only 7 multiplications, instead of 8 [190]. It turns out that his algorithm is related to a CPD of the corresponding matrix multiplication tensor, whose rank is 7 [105].

A further example in the domain of nonlinear system modeling comes from the interpretation of a P th-order symmetric Volterra kernel [176] as a multilinear functional applied to the input signal, which appears repeated P times as its argument. In the discrete-time setting with finite memory N_p , this is expressed as

$$y_p(n) = \mathfrak{H}_p(\underbrace{\mathbf{u}(n), \dots, \mathbf{u}(n)}_{P \text{ times}}), \quad (2.90)$$

where $y_p(n)$ is the output signal and $\mathbf{u}(n) \in \mathbb{F}^{N_p}$ is a vector containing the most recent N_p samples of the input signal at (discrete) time n . If $\mathfrak{H}_p \in \bigotimes_{q=1}^p \mathbb{F}^{N_p}$ admits the decomposition $\mathfrak{H}_p = \sum_{s=1}^S \mathbf{v}_s^{\otimes p}$ with $S \ll N_p$, then it is more efficient to compute (2.90) by means of

$$y_p(n) = \sum_{s=1}^S \left(\langle \mathbf{u}(n), \mathbf{v}_s^* \rangle \right)^p, \quad (2.91)$$

which requires $\mathcal{O}(S(N_p + p))$ multiplications, than by using (2.79), which requires $\mathcal{O}(N_p^p)$ multiplications. This is the idea underlying the model proposed in [79]. Note that we have refrained from calling the considered decomposition a CPD and also from equating S with $\text{rank}(\mathfrak{H}_p)$. The reason shall be clear in the next section.

2.5.1.2 Symmetric CPD

For symmetric tensors, a naturally constrained form of the CPD can be defined as follows.

Definition 2.38 (Symmetric rank and symmetric CPD). Let $\mathbf{v} \in \mathcal{S}(\mathcal{V}^{\otimes P})$ and consider the polyadic decomposition composed only by symmetric elementary tensors

$$\mathbf{v} = \sum_{s=1}^S v_s^{\otimes P}. \quad (2.92)$$

The minimal S such that (2.92) holds is called *symmetric rank* of \mathcal{V} , denoted by $\text{srnk}(\mathcal{V})$. A decomposition of the form (2.92) with $S = \text{srnk}(\mathcal{V})$ is called a *symmetric CPD* of \mathcal{V} .

The fact that any symmetric tensor can be decomposed as in (2.92) is shown in [51, Lemma 4.2]. Then, it is clear from the above definition that $\text{rank}(\mathcal{V}) \leq \text{srnk}(\mathcal{V})$ holds for any symmetric tensor \mathcal{V} . An important open question is whether $\text{rank}(\mathcal{V}) = \text{srnk}(\mathcal{V})$ holds. A positive answer has been conjectured in [51].

2.5.1.3 Uniqueness of the CPD

While the existence of the CPD follows immediately from its definition (one can always take a sum of elementary tensors composed by basis vectors), its uniqueness properties are not equally evident. Right away, it can be seen that a trivial ambiguity arises from the fact that

$$\mathcal{V} = \llbracket \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(R)} \rrbracket = \llbracket \boldsymbol{\Pi} \boldsymbol{\lambda}; \mathbf{A}^{(1)} \boldsymbol{\Pi}, \dots, \mathbf{A}^{(R)} \boldsymbol{\Pi} \rrbracket \quad (2.93)$$

for any permutation matrix $\boldsymbol{\Pi} \in \mathbb{F}^{R \times R}$. This property is known as *permutation ambiguity* or *permutation indeterminacy*. Note that it can be partially alleviated if one imposes $|\lambda_r| \geq |\lambda_{r+1}|$ for all $r \in \langle R-1 \rangle$. Even so, the ambiguity persists if $|\lambda_r| = |\lambda_{r+1}|$ for at least one r .

Another (perhaps less) trivial ambiguity is called *scaling ambiguity* (or *indeterminacy*), and can be expressed in the same notation as

$$\mathcal{V} = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(R)} \rrbracket \quad (2.94)$$

$$= \llbracket \mathbf{A}^{(1)} \text{Diag}(\alpha_1^{(1)}, \dots, \alpha_R^{(1)}), \dots, \mathbf{A}^{(R)} \text{Diag}(\alpha_1^{(R)}, \dots, \alpha_R^{(R)}) \rrbracket, \quad (2.95)$$

where $\prod_{p=1}^R \alpha_r^{(p)} = 1$ for all $r \in \langle R \rangle$. It is thus related with the possibility of jointly rescaling the vectors of each elementary tensor, so that the result remains unchanged. When $\mathbb{F} = \mathbb{R}$, this indeterminacy is eliminated up to signs by normalizing these vectors and leaving the overall scaling factor in a constant λ_r , as in (2.83). For $\mathbb{F} = \mathbb{C}$, even imposing normalization there are infinitely many choices of constants $\alpha_r^{(p)} \in \mathbb{C}$ such that $|\alpha_r^{(p)}| = 1$ and $\prod_{p=1}^R \alpha_r^{(p)} = 1$.

The above discussion motivates the following definition.

Definition 2.39 (Essential uniqueness of the CPD). A CPD of a tensor \mathcal{V} is said to be *essentially unique* if it is unique up to the trivial scaling and permutation ambiguities.

When the CPD is applied to an inverse problem, essential uniqueness is vital for guaranteeing the recovery of the quantities of interest up to harmless indeterminacies. It is well-known that rank-retaining matrix decompositions can only be made unique by imposing strong constraints, such as orthogonality. But, often there is no physical basis for such constraints. Fortunately, in the tensor case much milder constraints suffice for essential uniqueness, as first proven by Kruskal [124] for third-order tensors. A generalization of this result due to Sidiropoulos and Bro is stated below.

Definition 2.40 (Kruskal rank [124]). The Kruskal rank (krank) of a matrix $\mathbf{A} \in \mathbb{F}^{N_1 \times N_2}$, denoted $\text{krank}(\mathbf{A})$, is the maximal integer such that every set containing $\text{krank}(\mathbf{A})$ columns of \mathbf{A} is linearly independent.

Theorem 2.41 (Sufficient condition for essential CPD uniqueness [178]). Let $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and assume its representation admits a CPD of the form (2.86), with $P > 2$ and $R > 1$. If the matrix factors $\mathbf{A}^{(p)}$ are such that

$$\sum_{p=1}^P \text{krank}(\mathbf{A}^{(p)}) \geq 2R + P - 1, \quad (2.96)$$

then the CPD of \mathcal{V} is essentially unique. The inequality (2.96) is known as *Kruskal condition* or *Kruskal bound*.

Theorem 2.41 is the best known CPD uniqueness result. The Kruskal condition is a sufficient one, but it is not necessary in general. Unfortunately, necessary and sufficient conditions are not known for the general case. Nonetheless, for practical applications the above result is already advantageous in comparison with the matrix setting. In particular, a matrix $\mathbf{A}^{(p)} \in \mathbb{R}^{N_p \times R}$ with $R \leq N_p$ having columns independently drawn from an absolutely continuous distribution has $\text{krank}(\mathbf{A}^{(p)}) = R$ with probability one [178].

The discussion of further CPD uniqueness results is beyond the scope of this thesis. For more recent results and a detailed account on this issue, the reader is referred to [72, 73] and references therein.

2.5.2 Properties of the tensor rank

We shall now briefly discuss some distinctive properties of the tensor rank.

2.5.2.1 The tensor rank depends on the underlying field

The first striking dissimilarity between the tensor and matrix ranks is the fact that, in general, the former depends on the underlying field. In other words, the rank of a tensor defined over \mathbb{R} can be greater than the rank it possesses when defined over \mathbb{C} . Examples of this phenomenon are given in [119, 49] and references therein.

2.5.2.2 The tensor rank can exceed the smallest dimension

Whereas the rank of a matrix cannot exceed its smallest dimension, the same does not hold for tensors. This property applies to real and complex tensors, as shown by the following result, whose proof can be found in [95].

Proposition 2.42 (Lemma 3.42 of [95]). Let \mathcal{V}_p , $p \in \langle P \rangle$, be vector spaces defined over \mathbb{F} such that $\dim(\mathcal{V}_p) \geq 2$ and suppose the vectors $v_p, w_p \in \mathcal{V}_p$ are linearly independent. Then,

$$\text{rank}(v_1 \otimes v_2 \otimes w_3 + v_1 \otimes w_2 \otimes v_3 + w_1 \otimes v_2 \otimes v_3) = 3. \quad (2.97)$$

Since Proposition (2.42) holds with $\mathcal{V}_p = \mathbb{F}^2$, the rank can indeed exceed the smallest dimension. This phenomenon is not restricted to tensor spaces involving vector spaces of dimension 2. Many examples are given in [52] of finite-dimensional tensor spaces having typical ranks (see the next subsection for a definition) which exceed the dimensions of all involved vector spaces.

This fact bears important implications with regard to the existence of certain special representations of tensors. More specifically, since there exist tensors $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ whose (any) minimal representation of the form (2.80) is such that $R > \dim(\mathcal{V}_p)$ for some $p \in \langle P \rangle$, then the set of vectors $\{v_1^{(p)}, \dots, v_R^{(p)}\} \subset \mathcal{V}_p$ which take part in this representation cannot be linearly independent. Consequently, (considering that each \mathcal{V}_p is endowed with a scalar product) there *cannot* be a minimal representation of \mathcal{V} with the property that for all $q \in \langle P \rangle$, $\{v_1^{(q)}, \dots, v_R^{(q)}\} \subset \mathcal{V}_q$ is a family of orthogonal vectors.

2.5.2.3 Multiple typical ranks exist for real tensors

When one randomly draws the elements of an $N_1 \times N_2$ matrix from an absolutely continuous distribution, the resulting matrix has maximal rank $R = \min\{N_1, N_2\}$ almost surely (*i.e.*, with probability one). This stems from the fact that in any matrix space, the set of rank-defective matrices (*i.e.*, having rank $R < \min\{N_1, N_2\}$) has Lebesgue measure zero.

This property partially carries over to tensors of higher orders when their spaces are defined over algebraic closed fields [54], like \mathbb{C} . Assuming for instance that some topology is defined for a tensor space $\bigotimes_{p=1}^P \mathcal{V}_p$ defined over \mathbb{C} , then it can be shown that there is a unique integer R such that the set of tensors from this space possessing rank R has nonzero volume. This rank is called the *generic* rank of $\bigotimes_{p=1}^P \mathcal{V}_p$. On the other hand, unlike the matrix case, the generic rank does not necessarily correspond to the maximal rank.

Now, for tensor spaces defined over \mathbb{R} , the dissimilarity is even greater: in *lieu* of a single generic rank, they possess multiple ranks which occur with positive probability. These are known as *typical* ranks. Also, the smallest typical rank of a tensor always matches the generic rank of the complex tensor space of same dimensions [49]. The interested reader can find many examples of generic ranks of finite-dimensional complex tensor spaces in [54].

2.5.2.4 The tensor rank is NP-hard

Given a tensor \mathcal{V} of unknown rank, a problem of significant practical interest is that of computing $\text{rank}(\mathcal{V})$. Yet, this turns out to be a very difficult problem for $P \geq 3$, which is

in contrast to the matrix case, where efficient algorithms exist. In [102], it was shown that many tensor-related problems are NP-hard, including computing the rank of a tensor.

2.5.2.5 The best rank- R approximation may not exist

Consider the set of tensors having rank bounded by $R \in \mathbb{N}$, defined as

$$\mathcal{R}_R(\mathcal{T}) = \left\{ \mathbf{V} \in \mathcal{T} : \exists v_r^{(p)} \in \mathcal{V}_p \text{ such that } \mathbf{V} = \sum_{r=1}^R \bigotimes_{p=1}^P v_r^{(p)} \right\} \subseteq \mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p. \quad (2.98)$$

Clearly, $\mathcal{R}_0 \subset \mathcal{R}_1 \subset \mathcal{R}_2 \subset \dots$ holds, where \mathcal{R}_0 contains only the null tensor. For finite-dimensional tensor spaces, there exists a maximal rank value R_{\max} satisfying $\mathcal{R}_0 \subset \mathcal{R}_1 \subset \mathcal{R}_2 \subset \dots \subset \mathcal{R}_{R_{\max}} = \mathcal{T}$ [95].

A problem of great importance for many practical applications asks for the best approximation (or one among the best approximations, if multiple ones exist) of a given tensor \mathbf{V} contained by some set \mathcal{R}_R . This approximation can be sought with any of the goals described in Section 2.5.1.1. In particular, note that in inverse problems, computing the CPD of low-rank tensors measured in real-world applications typically involves approximation, due to presence of measurement errors and noise. Hence, given a tensor \mathbf{V} from a (finite-dimensional) normed tensor space and a positive integer R , one is interested in finding

$$\mathbf{V}^* = \arg \min_{\hat{\mathbf{V}} \in \mathcal{R}_R} \|\mathbf{V} - \hat{\mathbf{V}}\|, \quad (2.99)$$

where $\|\cdot\|$ is some chosen norm.

A fundamental difficulty exists when trying to solve (2.99): because the set \mathcal{R}_R is *not* topologically closed (with respect to the topology induced by any norm $\|\cdot\|$) for $P > 2$ and $R > 1$, the sought minimizer may not exist [69]. Examples of this phenomenon can be found in, *e.g.*, [95, Proposition 9.10] and [49, 69, 131]. These examples essentially construct a tensor \mathbf{V} having rank strictly greater than R for which $\inf_{\mathbf{W} \in \mathcal{R}_R} \|\mathbf{V} - \mathbf{W}\| = 0$ is not attained by any $\mathbf{W} \in \mathcal{R}_R \subset \mathcal{T}$. Consequently, \mathbf{V} has no *best* rank- R approximation, as it can be arbitrarily well approximated by tensors in \mathcal{R}_R . This motivates the definition of the so-called *border rank* of a tensor, given by [131, 95]

$$\overline{\text{rank}}(\mathbf{V}) = \min \left\{ R : \inf_{\mathbf{W} \in \mathcal{R}_R} \|\mathbf{V} - \mathbf{W}\| = 0 \right\}. \quad (2.100)$$

In words, it is the minimal R such that either $\text{rank}(\mathbf{V}) = R$ or \mathbf{V} has no best R -rank approximation. In [69, Theorem 1.3], it is shown that for tensor spaces of the form $\mathbb{R}^{N_1} \otimes \mathbb{R}^{N_2} \otimes \mathbb{R}^{N_3}$ with $N_p \geq 2$, the set of tensors satisfying $\overline{\text{rank}}(\mathbf{V}) < \text{rank}(\mathbf{V})$ has positive volume. Hence, it is of practical relevance when one is interested in rank-reduced approximations. See [133] for a discussion on how to deal with this issue in practice.

2.6 Modal spaces, multilinear rank and the Tucker decomposition

Tensor subspace techniques are useful in many applications, as they enable the use of familiar matrix and numerical linear algebra tools. At their core lies the concept of multilinear rank, whose definition can also be traced back to the work of Hitchcock in the beginning of the twentieth century [103]. In order to define it, we first introduce modal spaces.

Definition 2.43 (Mode- p space [95]). Let $\mathcal{V} \in \bigotimes_{q=1}^P \mathcal{V}_q$ be a finite-dimensional tensor. The subspace of \mathcal{V}_p denoted by $\text{space}_p(\mathcal{V})$ satisfying

$$(i) \quad \mathcal{V} \in \left(\bigotimes_{q=1}^{p-1} \mathcal{V}_q \right) \otimes \text{space}_p(\mathcal{V}) \otimes \left(\bigotimes_{q=p+1}^P \mathcal{V}_q \right) \quad (2.101)$$

$$(ii) \quad \forall \mathcal{W}_p \subseteq \mathcal{V}_p, \mathcal{V} \in \left(\bigotimes_{q=1}^{p-1} \mathcal{V}_q \right) \otimes \mathcal{W}_p \otimes \left(\bigotimes_{q=p+1}^P \mathcal{V}_q \right) \implies \text{space}_p(\mathcal{V}) \subseteq \mathcal{W}_p \quad (2.102)$$

is called the *mode- p space* of \mathcal{V} with respect to mode p . In other words, $\text{space}_p(\mathcal{V})$ is the intersection of all subspaces $\mathcal{W}_p \subseteq \mathcal{V}_p$ such that $\mathcal{V} \in \left(\bigotimes_{q=1}^{p-1} \mathcal{V}_q \right) \otimes \mathcal{W}_p \otimes \left(\bigotimes_{q=p+1}^P \mathcal{V}_q \right)$.

It is easy to see from the above definition that the mode- p space of a tensor is unique, while its existence is shown in [95, Theorem 6.13]. The next result states that it can be seen as generalization of column and row spaces of a matrix. In the introduced notation, the mode-1 space (resp., mode-2 space) of a second-order finite-dimensional tensor equals its column space (resp., row space).

Theorem 2.44 (Relation between modal unfoldings and mode- p spaces). For any finite-dimensional tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$, we have $\text{colspace}(\mathbf{V}_{\langle p \rangle}) = \text{space}_p(\mathcal{V})$.

Proof. Due to (2.101), we can write $\mathcal{V} = \sum_{s=1}^S \bigotimes_{q=1}^P \mathbf{v}_s^{(q)}$ in such a way that all $\mathbf{v}_s^{(p)} \in \text{space}_p(\mathcal{V})$. Similarly to (2.87), we have also $\mathbf{V}_{\langle p \rangle} = \mathbf{A}^{(p)} \mathbf{B}^{(p)}$, where $\mathbf{A}^{(p)} = \begin{bmatrix} \mathbf{v}_1^{(p)} & \dots & \mathbf{v}_S^{(p)} \end{bmatrix}$ and the rows of $\mathbf{B}^{(p)} \in \mathbb{F}^{S \times \bar{N}_p}$ are Kronecker products of vectors from the spaces \mathcal{V}_q , $q \neq p$. Now, without loss of generality, we can assume $\mathbf{B}^{(p)}$ has linearly independent (l.i.) rows, for if this is not the case, then another representation of the form $\mathcal{V} = \sum_{r=1}^R \bigotimes_{q=1}^P \mathbf{w}_r^{(q)}$, with $R < S$ and $\mathbf{w}_r^{(p)} \in \text{space}_p(\mathcal{V})$, can be obtained by combining a number of rows of $\mathbf{B}^{(p)}$ and their corresponding columns in $\mathbf{A}^{(p)}$. Hence, $\text{colspace}(\mathbf{V}_{\langle p \rangle}) = \text{colspace}(\mathbf{A}^{(p)})$. On the other hand, we clearly have $\text{colspace}(\mathbf{A}^{(p)}) \subseteq \text{space}_p(\mathcal{V})$. But, $\text{space}_p(\mathcal{V})$ is minimal by definition, and thus $\text{space}_p(\mathcal{V}) = \text{colspace}(\mathbf{A}^{(p)}) = \text{colspace}(\mathbf{V}_{\langle p \rangle})$. \square

Let us now define the multilinear rank and clarify its relation with the tensor.

Definition 2.45 (Multilinear rank [104]). The multilinear rank (mrank) over a finite-dimensional tensor space $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$, with $\dim(\mathcal{V}_p) = N_p$, is the map defined by

$$\text{mrank} : \mathcal{T} \mapsto \langle N_1 \rangle \times \dots \times \langle N_P \rangle \quad (2.103)$$

$$[\text{mrank}(\mathcal{V})]_p = \dim(\text{space}_p(\mathcal{V})) = \text{rank}(\mathbf{V}_{\langle p \rangle}). \quad (2.104)$$

We shall also use the notation $\text{rank}_p(\mathcal{V}) \triangleq [\text{mrnk}(\mathcal{V})]_p$.

Lemma 2.46 (Relation between rank and mrnk). Every $\mathcal{V} \in \bigotimes_{p=1}^P \mathcal{V}_p$ satisfies the inequality $\text{rank}_p(\mathcal{V}) \leq \text{rank}(\mathcal{V})$ for all $p \in \langle P \rangle$.

Proof. Let $\mathcal{V} = \sum_{r=1}^R \bigotimes_{p=1}^P v_r^{(p)}$ be a CPD of \mathcal{V} . Clearly, $\mathcal{V} \in (\bigotimes_{q=1}^{p-1} \mathcal{V}_q) \otimes \mathcal{W}_p \otimes (\bigotimes_{q=p+1}^P \mathcal{V}_q)$, where $\mathcal{W}_p = \text{span} \{v_1^{(p)}, \dots, v_R^{(p)}\}$ is a subspace of \mathcal{V}_p . Thus, by definition, $\text{space}_p(\mathcal{V}) \subseteq \mathcal{W}_p$, and so we have $\text{rank}_p(\mathcal{V}) = \dim(\text{space}_p(\mathcal{V})) \leq \dim(\text{span} \{v_1^{(p)}, \dots, v_R^{(p)}\}) \leq R$. \square

In comparison to the tensor rank, the mrnk is “better behaved,” in the sense that strange phenomena such as those discussed in Section 2.5 do not take place. In particular, $\text{rank}_p(\mathcal{V}) \leq N_p$. A justification for defining the multilinear rank as a property of tensors rather than of their representations (in the spirit of Proposition 2.35) can be derived as follows.

Proposition 2.47 (Invariance of the mrnk with respect to coordinate bases). The identity

$$\text{mrnk} \left(\mathcal{V} \times_{p=1}^P \mathbf{A}_p \right) = \text{mrnk}(\mathcal{V}) \quad (2.105)$$

holds for every tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and all nonsingular matrices $\mathbf{A}_p \in \mathbb{F}^{N_p \times N_p}$.

Proof. Since $(\mathcal{V} \times_p \mathbf{A}_p)_{\langle p \rangle} = \mathbf{A}_p \mathbf{V}_{\langle p \rangle}$, we have the chain of equalities $\text{rank}_p(\mathcal{V} \times_p \mathbf{A}_p) = \text{rank}(\mathbf{A}_p \mathbf{V}_{\langle p \rangle}) = \text{rank}(\mathbf{V}_{\langle p \rangle}) = \text{rank}_p(\mathcal{V})$, where the second one follows from nonsingularity of \mathbf{A}_p . Applying this argument repeatedly for $p \in \langle P \rangle$ yields (2.105). \square

2.6.1 Tucker decomposition

In the same way in which the tensor rank is closely related to the CPD, the mrnk is closely related to the Tucker decomposition (TD). Below, we first define such a decomposition and present some of its properties. Then, we consider some particularizations of special interest.

Definition 2.48 (Tucker decomposition [103, 202]). Let $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$. A *Tucker decomposition* or *multilinear decomposition* of \mathcal{V} consists in writing it under the form

$$\mathcal{V} = \mathfrak{G} \times_{p=1}^P \mathbf{A}^{(p)}, \quad (2.106)$$

where $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times M_p}$ are again called *factor matrices* and $\mathfrak{G} \in \bigotimes_{p=1}^P \mathbb{F}^{M_p}$ is the *core tensor*.

It is easy to show that (2.106) is equivalent to saying that an element of the representation of \mathcal{V} can be written in terms of that of \mathfrak{G} as

$$v_{n_1, \dots, n_P} = \sum_{m_1=1}^{M_1} \cdots \sum_{m_P=1}^{M_P} g_{m_1, \dots, m_P} \prod_p a_{n_p, m_p}^{(p)}, \quad (2.107)$$

where $\mathbf{A}^{(p)} = [a_{n_p, m_p}^{(p)}]$. This is the scalar form of the Tucker model.

Without further restrictions, infinitely many TDs of a tensor exist. This comes as no surprise, given that (2.106) can be seen as a multilinear change of basis when the matrices $\mathbf{A}^{(p)}$ are nonsingular. More generally, if all matrix factors have linearly independent columns (but are not necessarily square), then (2.106) can be regarded as the representation of \mathcal{V} with respect to the bases associated with these matrices, in which case \mathcal{G} plays the role of the corresponding coordinate representation.

Proposition 2.49 (Relation between CPD and TD). Let $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and suppose it admits the CPD $\mathcal{V} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}]\!]$, with $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times R}$ and $\boldsymbol{\lambda} \in \mathbb{R}^R$. Then,

$$\mathcal{V} = \mathcal{L} \underset{p=1}{\times}^P \mathbf{A}^{(p)}, \quad (2.108)$$

where the representation of $\mathcal{L} \in \bigotimes_{p=1}^P \mathbb{R}^R$ satisfies $[l_{r_1, \dots, r_P}] = \lambda_{r_1} \delta_{r_1, \dots, r_P}$.

Proof. From (2.17), we have

$$v_{n_1, \dots, n_P} = \sum_r \lambda_r \prod_p a_{n_p, r}^{(p)} = \sum_{r_1} \cdots \sum_{r_P} \lambda_{r_1} \delta_{r_1, \dots, r_P} \prod_p a_{n_p, r_p}^{(p)} = \sum_{r_1} \cdots \sum_{r_P} l_{r_1, \dots, r_P} a_{n_p, r_p}^{(p)}.$$

The identity (2.108) then follows from (2.107). \square

Corollary 2.31 provides a useful expression for the mode- p unfolding of a tensor in terms of its TD components. In particular, it allows a visualization of the non-uniqueness of the TD: since $\mathbf{V}_{\langle p \rangle} = \mathbf{A}^{(p)} \mathbf{G}_{\langle p \rangle} (\mathbf{A}^{(P)} \boxtimes \cdots \boxtimes \mathbf{A}^{(p+1)} \boxtimes \mathbf{A}^{(p-1)} \boxtimes \cdots \boxtimes \mathbf{A}^{(1)})^T$, $\mathbf{A}^{(p)}$ and $\mathbf{G}_{\langle p \rangle}$ can be replaced by $\mathbf{A}^{(p)} \mathbf{B}$ and $\mathbf{B}^{-1} \mathbf{G}_{\langle p \rangle}$, respectively, without changing $\mathbf{V}_{\langle p \rangle}$, for any nonsingular $\mathbf{B} \in \mathbb{F}^{M_p \times M_p}$. Similarly to the CPD, by applying property (2.49) of the Kronecker product we can also derive the following identity:

$$\text{vec}(\mathcal{V}) = \left(\mathbf{A}^{(P)} \boxtimes \cdots \boxtimes \mathbf{A}^{(1)} \right) \text{vec}(\mathcal{G}). \quad (2.109)$$

We now formally state the link between the mrank and the TD.

Theorem 2.50 (Connection between Tucker decomposition and mrank). Let $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$. If $\text{mrank}(\mathcal{V}) = (R_1, \dots, R_P)$, then \mathcal{V} admits a Tucker decomposition of the form (2.106) with factors $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times R_p}$ and core tensor $\mathcal{G} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}$.

Proof. By definition, $\text{mrank}(\mathcal{V}) = (R_1, \dots, R_P)$ implies $\dim(\text{space}_p(\mathcal{V})) = R_p$. This means any basis of $\text{space}_p(\mathcal{V})$ has cardinality R_p . Let $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times R_p}$ hold the vectors of a basis as its columns, for all $p \in \langle P \rangle$. Due to Theorem 2.44, there is a matrix $\mathbf{B}^{(p)} \in \mathbb{F}^{R_p \times \bar{N}_p}$ having l.i. rows and such that $\mathbf{V}_{\langle p \rangle} = \mathbf{A}^{(p)} \mathbf{B}^{(p)}$. Moreover, as the columns of $\mathbf{A}^{(p)}$ are l.i.,

$\mathbf{A}^{(p)H} \mathbf{A}^{(p)}$ is nonsingular. Therefore, we have $\mathbf{V}_{\langle p \rangle} = \mathbf{A}^{(p)} (\mathbf{A}^{(p)})^+ \mathbf{V}_{\langle p \rangle}$, where $(\mathbf{A}^{(p)})^+ = (\mathbf{A}^{(p)H} \mathbf{A}^{(p)})^{-1} \mathbf{A}^{(p)H}$ is the left inverse of $\mathbf{A}^{(p)}$. Consequently,

$$\mathbf{V} = \mathbf{V} \times_{p=1}^P \left(\mathbf{A}^{(p)} (\mathbf{A}^{(p)})^+ \right) = \underbrace{\left[\mathbf{V} \times_{p=1}^P \left(\mathbf{A}^{(p)} \right)^+ \right]}_{\triangleq \mathcal{G}} \times_{p=1}^P \mathbf{A}^{(p)},$$

where \mathcal{G} is the representation of \mathbf{V} with respect to the basis matrices $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}$. As $\mathcal{G} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}$, the result holds. \square

Remark 2.51. Note that the converse of [Theorem 2.50](#) is not necessarily true. Indeed, we can at most say that the existence of such a Tucker decomposition implies $\text{rank}_p(\mathbf{V}) \leq R_p$ (e.g., the null tensor trivially admits a Tucker model of that form with a null core).

As with the CPD, tensor models based on the TD also find many practical applications. Among the goals described in [Section 2.5.1.1](#), items 2–4 are often also motivations for employing a TD. Examples of its use in data analysis are given in [[3](#), [45](#), [145](#), [158](#), [175](#)]. In this context, the particularizations described ahead in [Sections 2.6.1.1](#) and [2.6.1.2](#), one of them having properties reminiscent of the matrix SVD, are often exploited for dimensionality reduction and/or feature extraction. It is also used for compression, as in [[137](#)], by virtue of [Theorem 2.50](#). The motivation is that, for a $\text{mrank-}(R_1, \dots, R_P)$ tensor such that $R_p \ll N_p$, it is much less costly to store basis matrices $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times R_p}$ and a core tensor $\mathcal{G} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}$, which requires storing $\prod_p R_p + \sum_p N_p R_p$ numbers from \mathbb{F} , than storing $\bar{N} = \prod_p N_p$ numbers. Finally, its use for complexity reduction is exemplified by the so-called principal dynamic modes (PDM) model used in the modeling of (nonlinear) physiological systems [[141](#), [142](#)]. This link relies again on viewing Volterra kernels as multilinear functionals.

We finish this section by pointing out that [[110](#)] provides a thorough account of invariance relations between a tensor and the core of a TD. For instance, under the assumption that the factors $\mathbf{A}^{(p)}$ of a TD have l.i. columns, we have $\text{rank}(\mathbf{V}) = \text{rank}(\mathcal{G})$, which implies $\text{rank}(\mathbf{V}) \leq \prod_{p=1}^P \text{rank}_p(\mathbf{V})$. Also,

$$\alpha \|\mathcal{G}\|_F \leq \|\mathbf{V}\|_F \leq \beta \|\mathcal{G}\|_F, \quad (2.110)$$

with $\alpha = \prod_p \|\mathbf{A}^{(p)}\|_2$ and $\beta = \alpha / \left(\prod_p \kappa(\mathbf{A}^{(p)}) \right)$, where $\kappa(\mathbf{A})$ denotes the ratio between the largest and the smallest (nonzero) singular values of \mathbf{A} . A similar bound holds for the nuclear norm.

2.6.1.1 Orthogonal Tucker decomposition

When every factor of a TD belongs to the Stiefel manifold $\mathcal{V}_{R_p}(\mathbb{F}^{N_p})$ of $N_p \times R_p$ matrices having orthonormal columns, we call it an orthogonal Tucker decomposition (OTD). This

corresponds to an important case of the TD with special properties. For instance, with regard to property (2.110), we have $\alpha = \beta = 1$ for $\mathbf{A}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$, and thus $\|\mathcal{G}\|_F = \|\mathbf{V}\|_F$ [110]. Moreover, given a tensor $\mathcal{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ with $\text{mrank}(\mathcal{V}) = (R_1, \dots, R_P)$ and an orthonormal basis $\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$ for $\text{space}_p(\mathcal{V})$, it is easy to see that the corresponding core can be computed as $\mathcal{G} = \mathcal{V} \times_{p=1}^P \mathbf{U}^{(p)H}$.

OTDs are well-suited for numerical computations. In particular, they provide a convenient framework in which the problem of best subspace-based approximation, which we will discuss in Section 2.6.2, can be posed and numerically tackled. Clearly, given any TD of a tensor $\mathcal{V} = \mathcal{G} \times_{p=1}^P \mathbf{A}^{(p)}$ having $\text{mrank}(\mathcal{V}) = (R_1, \dots, R_P)$ and such that $\mathbf{A}^{(p)} \in \mathbb{F}^{N_p \times M_p}$, with $M_p \geq R_p$, an OTD can be derived by computing, *e.g.*, a QR decomposition $\mathbf{A}^{(p)} = \mathbf{Q}^{(p)} \mathbf{R}^{(p)}$, with $\mathbf{Q}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p \times R_p})$ and $\mathbf{R}^{(p)} \in \mathbb{F}^{R_p \times M_p}$, from which we have

$$\mathcal{V} = \mathcal{H} \times_{p=1}^P \mathbf{Q}^{(p)}, \quad \text{with} \quad \mathcal{H} = \mathcal{G} \times_{p=1}^P \mathbf{R}^{(p)} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}. \quad (2.111)$$

Alternatively, given a tensor \mathcal{V} having $\text{mrank}(\mathcal{V}) = (R_1, \dots, R_P)$, by computing P QR decompositions $\mathbf{V}_{\langle p \rangle} = \mathbf{Q}^{(p)} \mathbf{R}^{(p)}$, where $\mathbf{Q}^{(p)} \in \mathcal{V}_{R_p}(N_p)$ and $\mathbf{R}^{(p)} \in \mathbb{F}^{R_p \times \bar{N}_p}$, we can write the OTD $\mathcal{V} = \mathcal{H} \times_{p=1}^P \mathbf{Q}^{(p)}$, with $\mathcal{H} = \mathcal{V} \times_{p=1}^P \mathbf{Q}^{(p)H} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}$.

2.6.1.2 Higher-order singular value decomposition

We describe now a particularization of the OTD which plays an important role in many tensor-based techniques.

Proposition 2.52 (Higher-order singular value decomposition [58]). Every tensor $\mathcal{V} \in \mathcal{T} = \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ admits the OTD

$$\mathcal{V} = \mathcal{S} \times_{p=1}^P \mathbf{U}^{(p)} \quad (2.112)$$

where $\mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{F}^{N_p})$ is the matrix of left singular vectors of $\mathbf{V}_{\langle p \rangle}$ and $\mathcal{S} \in \mathcal{T}$. This OTD is called the higher-order singular value decomposition (HOSVD) of \mathcal{V} and its core satisfies

$$\mathbf{S}_{\langle p \rangle} \mathbf{S}_{\langle p \rangle}^H = \text{Diag} \left(\sigma_1^{(p)2}, \dots, \sigma_{N_p}^{(p)2} \right), \quad (2.113)$$

where $\sigma_{n_p}^{(p)}$ is the n_p th singular value of $\mathbf{V}_{\langle p \rangle}$, referred to as n_p th mode- p singular value of \mathcal{V} . In case of ambiguity, we use also the notation $\sigma_{n_p}^{(p)}(\mathcal{V})$.

Proof. Let $\mathbf{V}_{\langle p \rangle} = \mathbf{U}^{(p)} \mathbf{\Sigma}^{(p)} \mathbf{W}^{(p)H}$ be the SVD of $\mathbf{V}_{\langle p \rangle}$, where $\mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{F}^{N_p})$, $\mathbf{W}^{(p)} \in \mathcal{V}_{\bar{N}_p}(\mathbb{F}^{\bar{N}_p})$ and $\mathbf{\Sigma}^{(p)} \in \mathbb{R}^{N_p \times \bar{N}_p}$, with $[\mathbf{\Sigma}^{(p)}]_{n_p, n'_p} = \sigma_{n_p}^{(p)} \delta_{n_p, n'_p}$. Now, consider the tensor

$$\mathcal{S} = \mathcal{V} \times_{p=1}^P \mathbf{U}^{(p)H}. \quad (2.114)$$

As $\mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{F}^{N_p})$ is unitary, it is clear that (2.112) holds. From (2.114) and the SVD of $\mathbf{V}_{\langle p \rangle}$, we obtain

$$\mathbf{S}_{\langle p \rangle} = \mathbf{U}^{(p)H} \mathbf{V}_{\langle p \rangle} \left(\mathbf{U}^{(P)} \otimes \dots \otimes \mathbf{U}^{(p+1)} \otimes \mathbf{U}^{(p-1)} \otimes \dots \otimes \mathbf{U}^{(1)} \right)^H \quad (2.115)$$

$$= \boldsymbol{\Sigma}^{(p)} \mathbf{W}^{(p)H} \left(\mathbf{U}^{(P)} \otimes \dots \otimes \mathbf{U}^{(p+1)} \otimes \mathbf{U}^{(p-1)} \otimes \dots \otimes \mathbf{U}^{(1)} \right)^H. \quad (2.116)$$

Finally, using properties of the Kronecker product and the fact that $\mathbf{U}^{(p)}$ and $\mathbf{W}^{(p)}$ are unitary matrices, we can verify that (2.113) holds. \square

Remark 2.53. One can consider also an mrank-retaining “reduced” or “economical” HOSVD as follows. Letting $\text{mrank}(\mathbf{V}) = (R_1, \dots, R_P)$, the SVD of each unfolding $\mathbf{V}_{\langle p \rangle}$ can be computed in its “economical” form, yielding $\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$, $\mathbf{W}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$ and a positive definite $\boldsymbol{\Sigma}^{(p)} = \text{Diag}(\sigma_1^{(p)}, \dots, \sigma_{R_p}^{(p)}) \in \mathbb{R}^{R_p \times R_p}$. Then, (2.112)–(2.114) still hold with a reduced core $\mathbf{S} \in \bigotimes_{p=1}^P \mathbb{F}^{R_p}$, because $\text{colspace}(\mathbf{U}^{(p)}) = \text{space}_p(\mathbf{V}) = \text{colspace}(\mathbf{V}_{\langle p \rangle})$ (cf. Theorem 2.44). Property (2.113) is in this case expressed as $\mathbf{S}_{\langle p \rangle} \mathbf{S}_{\langle p \rangle}^H = \text{Diag}(\sigma_1^{(p)2}, \dots, \sigma_{R_p}^{(p)2})$.

2.6.2 Best mrank-r approximation

The computation of subspace-based tensor approximations is a problem with many applications. Given a tensor space $\mathcal{T} = \bigotimes_{p=1}^P \mathcal{V}_p$ and a tuple $\mathbf{r} = (R_1, \dots, R_P) \in \mathbb{N}^P$, by defining

$$\mathcal{L}_{\mathbf{r}} = \{ \mathbf{V} \in \mathcal{T} : \text{rank}_p(\mathbf{V}) \leq R_p, p \in \langle P \rangle \}, \quad (2.117)$$

one can formulate this problem as

$$\min_{\hat{\mathbf{V}} \in \mathcal{L}_{\mathbf{r}}} \|\mathbf{V} - \hat{\mathbf{V}}\| \quad (2.118)$$

for some norm $\|\cdot\|$. By far, the most common choice is the Frobenius norm, which means we look for “the closest tensor” $\hat{\mathbf{V}} \in \mathcal{L}_{\mathbf{r}}$ to \mathbf{V} in terms of Euclidean distance. In contrast to the tensor rank-based approximation discussed in Section 2.5.2.5, here one can show that in the finite-dimensional case, (2.118) *always* has a solution for any chosen norm, which is owed to the fact that $\mathcal{L}_{\mathbf{r}}$ is closed (and that all norms are equivalent). See, *e.g.*, [95, Theorem 10.8] for a formal statement of this fact in a more general setting. On the other hand, the solution can be nonunique, because $\mathcal{L}_{\mathbf{r}}$ is not a convex set.

Due to the relevance of the above problem, it has been studied by many authors. In particular, [59] has derived the following result.

Proposition 2.54. Let $\mathbf{V} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and $\mathbf{r} = (R_1, \dots, R_P) \in \mathbb{N}^P$, with $R_p \leq N_p$. Then,

$$\min_{\hat{\mathbf{V}} \in \mathcal{L}_{\mathbf{r}}} \|\mathbf{V} - \hat{\mathbf{V}}\|_F = \max_{\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})} \left\| \mathbf{V} \times_{p=1}^P \mathbf{U}^{(p)H} \right\|_F. \quad (2.119)$$

Proof. First note that $\|\mathbf{V} - \hat{\mathbf{V}}\|_F^2 = \|\mathbf{V}\|_F^2 - 2\text{Re}\{\langle \mathbf{V}, \hat{\mathbf{V}} \rangle\} + \|\hat{\mathbf{V}}\|_F^2$. Also, since $\hat{\mathbf{V}} \in \mathcal{L}_r$, it must admit an OTD $\hat{\mathbf{V}} = \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)}$. Now, whichever the matrices $\mathbf{U}^{(p)}$ are, it only makes sense to choose $\hat{\mathcal{G}} = \mathbf{V} \times_{p=1}^P \mathbf{U}^{(p)H}$, because it attains the minimum of

$$\begin{aligned} \min_{\mathcal{G}} \left\{ -2\text{Re} \left\{ \left\langle \mathbf{V}, \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)} \right\rangle \right\} + \left\| \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)} \right\|_F^2 \right\} &= \min_{\mathcal{G}} \left\{ -2\text{Re} \left\{ \left\langle \hat{\mathcal{G}}, \mathcal{G} \right\rangle \right\} + \|\mathcal{G}\|_F^2 \right\} \\ &= -\|\hat{\mathcal{G}}\|_F^2. \end{aligned}$$

The result thus follows from the definition of $\hat{\mathcal{G}}$. \square

Based on this result, [59] proposes the high-order orthogonal iteration (HOOI) algorithm, which is a multilinear generalization of the orthogonal iteration scheme used to compute dominant singular vectors of a matrix (see, *e.g.*, [90, Section 7.3.2]). The HOOI updates the matrices $\mathbf{U}^{(p)}$ of Proposition 2.54 in an alternating fashion. Fixing all of them but $\mathbf{U}^{(p)}$ for some $p \in \langle P \rangle$, $\mathbf{U}^{(p)}$ can be updated by computing the dominant mode- p subspace of dimension R_p by means of a SVD. Although its implementation is simple, the HOOI may require many iterations to converge, a fact which is aggravated by its relatively high computing cost per iteration.

Another approach to address this problem is proposed in [107, 108]. It consists in employing Riemannian optimization techniques in order to compute a solution to the best mr-rank-reduced approximation problem. These techniques exploit the properties of certain smooth manifolds to which the search for a solution is constrained. Despite the fact that convergence can be achieved in less iterations than HOOI with, *e.g.*, a Riemannian trust-region based scheme, the required computing effort is still quite high.

Often, instead of pursuing a solution close to the optimum (this is generally the best we can hope for, as this problem is NP-hard [102]), one is satisfied with a *quasi-optimal* solution $\tilde{\mathbf{V}}$ satisfying

$$\|\mathbf{V} - \tilde{\mathbf{V}}\|_F \leq C \min_{\hat{\mathbf{V}} \in \mathcal{L}_r} \|\mathbf{V} - \hat{\mathbf{V}}\|_F \quad (2.120)$$

for some (reasonable) constant C . There exist approximate non-iterative solutions which meet this requirement and whose computing cost is much milder than that of the above mentioned iterative methods. Below we describe two alternatives with these characteristics.

2.6.2.1 Truncated HOSVD

Owing to Proposition 2.54, the optimization problem of interest can be expressed as

$$\min_{\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})} \left\| \mathbf{V} - \left(\mathbf{V} \times_{p=1}^P \mathbf{U}^{(p)H} \right) \times_{p=1}^P \mathbf{U}^{(p)} \right\|_F^2 = \min_{\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})} \left\| \mathbf{V} - \mathbf{V} \times_{p=1}^P \mathbf{U}^{(p)} \mathbf{U}^{(p)H} \right\|_F^2.$$

The cost function $J(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(P)}) = \left\| \mathbf{v} - \mathbf{v} \times_{p=1}^P \mathbf{U}^{(p)} \mathbf{U}^{(p)H} \right\|_F^2$ can then be rewritten by introducing a telescoping sum, yielding

$$J(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(P)}) = \left\| \mathbf{v} - \sum_{p=1}^P \mathbf{v} \times_{q=1}^p \mathbf{P}^{(q)} + \sum_{p=1}^{P-1} \mathbf{v} \times_{q=1}^p \mathbf{P}^{(q)} \right\|_F^2, \quad (2.121)$$

where $\mathbf{P}^{(p)} = \mathbf{U}^{(p)} \mathbf{U}^{(p)H} \in \mathbb{F}^{N_p \times N_p}$ is the orthogonal projection onto $\text{colspace}(\mathbf{U}^{(p)})$. Regrouping the terms,

$$J(\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(P)}) = \left\| \sum_{p=1}^P \mathbf{v} \times_{q=1}^{p-1} \mathbf{P}^{(q)} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2 = \sum_{p=1}^P \left\| \mathbf{v} \times_{q=1}^{p-1} \mathbf{P}^{(q)} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2 \quad (2.122)$$

$$\leq \sum_{p=1}^P \left\| \mathbf{v} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2 = \sum_{p=1}^P \left\| \mathbf{P}^{(p)\perp} \mathbf{V}_{\langle p \rangle} \right\|_F^2, \quad (2.123)$$

where $\mathbf{P}^{(p)\perp} \triangleq \mathbf{I} - \mathbf{P}^{(p)}$, the second equality follows from orthogonality of the terms of the sum and the inequality follows from nonexpansiveness of $\mathbf{P}^{(q)}$, $q \in \langle P \rangle$.

The celebrated Eckart-Young theorem implies that the (separate) minimization of each term of the upper bound in (2.123) is achieved by taking $\mathbf{U}^{(p)}$ having the first R_p left singular vectors of $\mathbf{V}_{\langle p \rangle}$ as columns. This is the idea underlying the truncated HOSVD (THOSVD) method for approximating a solution of (2.119). Given a tensor $\mathbf{v} \in \bigotimes_{p=1}^P \mathbb{F}^{N_p}$ and a target approximation mrank $\mathbf{r} = (R_1, \dots, R_P)$, one first computes for each unfolding the “economical” SVD $\mathbf{V}_{\langle p \rangle} = \left[\bar{\mathbf{U}}^{(p)} \quad \tilde{\mathbf{U}}^{(p)} \right] \boldsymbol{\Sigma}^{(p)} \mathbf{W}^{(p)H}$, where $\bar{\mathbf{U}}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$, $\tilde{\mathbf{U}}^{(p)} \in \mathcal{V}_{M_p - R_p}(\mathbb{F}^{N_p})$, $\boldsymbol{\Sigma}^{(p)} \in \mathbb{R}^{M_p \times M_p}$ and $\mathbf{W}^{(p)} \in \mathcal{V}_{M_p}(\mathbb{F}^{\bar{N}_p})$, with $M_p = \min\{N_p, \bar{N}_p\}$. Then, the approximation

$$\bar{\mathbf{v}} = \mathbf{v} \times_{p=1}^P \bar{\mathbf{P}}^{(p)} \quad (2.124)$$

is computed, with $\bar{\mathbf{P}}^{(p)} = \bar{\mathbf{U}}^{(p)} \bar{\mathbf{U}}^{(p)H}$ for all $p \in \{1, \dots, P\}$.

In addition to its simplicity and to the fact that it only requires standard numerical linear algebra routines for its implementation, the truncated HOSVD is a quasi-optimal approximation in the sense of (2.120). To see this, let $\mathbf{v}^* \in \mathcal{L}_{\mathbf{r}}$ be a solution of (2.119). Because $\text{rank}_p(\mathbf{v}^*) = R_p$, then the Eckart-Young theorem implies

$$\left\| (\mathbf{I} - \bar{\mathbf{P}}^{(p)}) \mathbf{V}_{\langle p \rangle} \right\|_F^2 \leq \left\| \mathbf{V}_{\langle p \rangle} - \mathbf{V}_{\langle p \rangle}^* \right\|_F^2. \quad (2.125)$$

Therefore, from (2.123) we derive the bound

$$J(\bar{\mathbf{U}}^{(1)}, \dots, \bar{\mathbf{U}}^{(P)}) = \left\| \mathbf{v} - \bar{\mathbf{v}} \right\|_F^2 \leq P \left\| \mathbf{v} - \mathbf{v}^* \right\|_F^2 = P \min_{\hat{\mathbf{v}} \in \mathcal{L}_{\mathbf{r}}} \left\| \mathbf{v} - \hat{\mathbf{v}} \right\|_F^2. \quad (2.126)$$

This is a well-known property of the THOSVD [209, 91]. Another fact that can be directly derived from (2.123) is [59]

$$\|\mathbf{v} - \bar{\mathbf{v}}\|_F^2 \leq \sum_{p=1}^P \sum_{r_p=R_p+1}^{\min\{N_p, \bar{N}_p\}} \left[\sigma_{r_p}^{(p)}(\mathbf{v}) \right]^2, \quad (2.127)$$

which stems from the fact that the quadratic error of a truncated SVD equals the sum of the squared discarded singular values.

Due to the above described features, the THOSVD is often used either to obtain the final desired approximation or as starting point for iterative methods as those described in the previous section [59, 107].

2.6.2.2 Sequentially optimal modal approximations

In [209], an alternative to the THOSVD is proposed for the best mrank-r approximation problem. It is based on the fact that problem (2.119) is equivalent to minimizing (2.122) with the constraints $\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{F}^{N_p})$ and $\mathbf{P}^{(p)} = \mathbf{U}^{(p)} \mathbf{U}^{(p)H}$. Hence, omitting these constraints for simplicity, we can write it as [209]

$$\begin{aligned} & \min_{\mathbf{P}^{(1)}} \left(\left\| \mathbf{v} \times_1 \mathbf{P}^{(1)\perp} \right\|_F^2 + \min_{\mathbf{P}^{(2)}} \left(\left\| \mathbf{v} \times_1 \mathbf{P}^{(1)} \times_2 \mathbf{P}^{(2)\perp} \right\|_F^2 + \min_{\mathbf{P}^{(3)}} \left(\dots \right. \right. \right. \\ & \left. \left. \left. \dots + \min_{\mathbf{P}^{(P-1)}} \left(\left\| \mathbf{v} \times_{p=1}^{P-2} \mathbf{P}^{(p)} \times_{P-1} \mathbf{P}^{(P-1)\perp} \right\|_F^2 + \min_{\mathbf{P}^{(P)}} \left\| \mathbf{v} \times_{p=1}^{P-1} \mathbf{P}^{(p)} \times_P \mathbf{P}^{(P)\perp} \right\|_F^2 \dots \right) \dots \right) \right). \end{aligned} \quad (2.128)$$

An approximate solution can then be obtained by solving instead

$$\min_{\mathbf{P}^{(2)}} \left(\left\| \mathbf{v} \times_1 \hat{\mathbf{P}}^{(1)} \times_2 \mathbf{P}^{(2)\perp} \right\|_F^2 + \min_{\mathbf{P}^{(3)}} \left(\dots + \min_{\mathbf{P}^{(P)}} \left(\left\| \mathbf{v} \times_1 \hat{\mathbf{P}}^{(1)} \times_{p=2}^{P-1} \mathbf{P}^{(p)} \times_P \mathbf{P}^{(P)\perp} \right\|_F^2 \dots \right) \dots \right) \right),$$

where $\hat{\mathbf{P}}^{(1)} = \arg \min_{\mathbf{P}^{(1)}} \left\| \mathbf{v} \times_1 \mathbf{P}^{(1)\perp} \right\|_F^2$. The same principle is then applied for all $p \in \langle P \rangle$, leading to

$$\hat{\mathbf{P}}^{(p)} = \arg \min_{\mathbf{P}^{(p)}} \left\| \mathbf{v} \times_{q=1}^{p-1} \hat{\mathbf{P}}^{(q)} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2. \quad (2.129)$$

With respect to the minimum of the original problem, we can write

$$\min_{\hat{\mathbf{v}} \in \mathcal{L}_r} \|\mathbf{v} - \hat{\mathbf{v}}\|_F^2 \leq \sum_{p=1}^P \min_{\mathbf{P}^{(p)}} \left\| \mathbf{v} \times_{q=1}^{p-1} \hat{\mathbf{P}}^{(q)} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2 = \left\| \mathbf{v} - \mathbf{v} \times_{p=1}^P \hat{\mathbf{P}}^{(p)} \right\|_F^2. \quad (2.130)$$

Observe that $\hat{\mathbf{P}}^{(p)}$ depends on all $\hat{\mathbf{P}}^{(q)}$ with $q < p$. More precisely, the projectors $\hat{\mathbf{P}}^{(p)}$ result from a *sequential* optimization with respect to each mode, given the solution of all previous

ones. For this reason, we call⁵ this solution *sequentially optimal modal projections* (SeMP). It is not hard to see that the modal projections can just as well be applied in any different order (*i.e.*, any permutation of $(1, \dots, P)$), which generally yields different results.

Just like the THOSVD, it is easy to show that SeMP yields a quasi-optimal approximation, since the attained cost function value (cf. (2.130)) satisfies

$$\sum_{p=1}^P \min_{\mathbf{P}^{(p)}} \left\| \mathbf{V} \times_{q=1}^{p-1} \hat{\mathbf{P}}^{(q)} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2 \leq \sum_{p=1}^P \min_{\mathbf{P}^{(p)}} \left\| \mathbf{V} \times_p \mathbf{P}^{(p)\perp} \right\|_F^2. \quad (2.131)$$

Therefore, SeMP is subject to the same upper bound (2.126) derived for the THOSVD in Section 2.6.2.1, and is also quasi-optimal by a factor of \sqrt{P} . This holds regardless of how the modal projections are ordered.

In comparison with the THOSVD, though, SeMP possesses important advantages. The first one is computational, and stems from the fact that the procedure can be carried out in such a way that the dimensionality of the target tensor is reduced at each step (*i.e.*, for each mode) [209]. This is evidenced by the pseudocode presented in Algorithm 2.1, in which $L_p \triangleq \left(\prod_{q=1}^{p-1} R_q \right) \left(\prod_{q=p+1}^P N_q \right)$. Line 4 of this procedure is equivalent to

$$\bar{\mathbf{Z}}_p = \bar{\mathbf{Z}}_{p-1} \times_p \left(\bar{\mathbf{U}}^{(p)} \right)^H \in \mathbb{F}^{R_1 \times \dots \times R_p \times N_{p+1} \times \dots \times N_P}. \quad (2.132)$$

Therefore, the final outcome can be written as $\mathcal{S}_r(\mathbf{V}) = \mathbf{V} \times_{p=1}^P \bar{\mathbf{U}}^{(p)} \left(\bar{\mathbf{U}}^{(p)} \right)^H = \mathbf{V} \times_{p=1}^P \hat{\mathbf{P}}_p$. Note the similarity between this expression and (2.124). The fact that the matrix $\bar{\mathbf{U}}^{(p)}$ calculated in Algorithm 2.1 satisfies $\bar{\mathbf{U}}^{(p)} \left(\bar{\mathbf{U}}^{(p)} \right)^H = \hat{\mathbf{P}}_p$, with $\hat{\mathbf{P}}_p$ defined by (2.129), can be verified as follows. For brevity, let us denote $\mathbf{Z}_{p-1} \triangleq \mathbf{V} \times_{q=1}^{p-1} \hat{\mathbf{P}}_q$, with $\mathbf{Z}_0 = \mathbf{V}$. We need to show that $\bar{\mathbf{U}}^{(p)}$ contains the first left R_p singular vectors of $(\mathbf{Z}_{p-1})_{\langle p \rangle}$ as columns. For $p = 1$, this is clearly true, as $\mathbf{Z}_0 = \bar{\mathbf{Z}}_0 = \mathbf{V}$. For $p > 1$, we proceed by induction. Assume the claim holds for all $q \in \langle p-1 \rangle$, which implies $\bar{\mathbf{U}}^{(q)} \left(\bar{\mathbf{U}}^{(q)} \right)^H = \hat{\mathbf{P}}_q$. Then, it is easy to verify that it holds also for p , because the left singular vectors of

$$(\mathbf{Z}_{p-1})_{\langle p \rangle} = \mathbf{V}_{\langle p \rangle} \left(\bar{\mathbf{U}}^{(1)} \left(\bar{\mathbf{U}}^{(1)} \right)^H \boxtimes \dots \boxtimes \bar{\mathbf{U}}^{(p-1)} \left(\bar{\mathbf{U}}^{(p-1)} \right)^H \boxtimes \mathbf{I}_{N_{p+1}} \boxtimes \dots \boxtimes \mathbf{I}_{N_P} \right)^H \quad (2.133)$$

are the same as those of

$$(\bar{\mathbf{Z}}_{p-1})_{\langle p \rangle} = \mathbf{V}_{\langle p \rangle} \left(\left(\bar{\mathbf{U}}^{(1)} \right)^H \boxtimes \dots \boxtimes \left(\bar{\mathbf{U}}^{(p-1)} \right)^H \boxtimes \mathbf{I}_{N_{p+1}} \boxtimes \dots \boxtimes \mathbf{I}_{N_P} \right)^H. \quad (2.134)$$

where \mathbf{I}_N denotes the $N \times N$ identity matrix. Indeed,

$$\begin{aligned} (\mathbf{Z}_{p-1})_{\langle p \rangle} (\mathbf{Z}_{p-1})_{\langle p \rangle}^H &= \mathbf{V}_{\langle p \rangle} \left(\bar{\mathbf{U}}^{(1)} \left(\bar{\mathbf{U}}^{(1)} \right)^H \boxtimes \dots \boxtimes \bar{\mathbf{U}}^{(p-1)} \left(\bar{\mathbf{U}}^{(p-1)} \right)^H \boxtimes \mathbf{I}_{N_{p+1}} \boxtimes \dots \boxtimes \mathbf{I}_{N_P} \right)^H \mathbf{V}_{\langle p \rangle}^H \\ &= (\bar{\mathbf{Z}}_{p-1})_{\langle p \rangle} (\bar{\mathbf{Z}}_{p-1})_{\langle p \rangle}^H. \end{aligned}$$

⁵Though [209] uses the name “sequentially truncated HOSVD” we prefer to adopt “sequentially optimal modal projections,” because the resulting projection operators are not associated with the original dominant modal subspaces.

Algorithm 2.1 Sequentially optimal projections (**SeMP**) for low-mrank approximation [209].

Inputs: Tensor \mathcal{V} whose best approximation in \mathcal{L}_r is sought, target mrank $\mathbf{r} = (R_1, \dots, R_P)$

Output: An approximate projection $\mathcal{S}_r(\mathcal{V})$ of \mathcal{V} onto \mathcal{L}_r

1: $\tilde{\mathcal{Z}}_0 \leftarrow \mathcal{V}$

2: **for** $p = 1, \dots, P$ **do**

3: compute the SVD: $(\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle} = \begin{bmatrix} \bar{\mathbf{U}}^{(p)} & \tilde{\mathbf{U}}^{(p)} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\Sigma}}^{(p)} & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{\Sigma}}^{(p)} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{W}}^{(p)} & \tilde{\mathbf{W}}^{(p)} \end{bmatrix}^H,$

 where $\bar{\mathbf{U}}^{(p)} \in \mathbb{F}^{N_p \times R_p}$, $\bar{\boldsymbol{\Sigma}}^{(p)} \in \mathbb{R}^{R_p \times R_p}$, $\bar{\mathbf{W}}^{(p)} \in \mathbb{F}^{L_p \times R_p}$, $L_p = \left(\prod_{q=1}^{p-1} R_q \right) \left(\prod_{q=p+1}^P N_q \right)$

4: compute $\tilde{\mathcal{Z}}_p$ via: $(\tilde{\mathcal{Z}}_p)_{\langle p \rangle} \leftarrow \bar{\boldsymbol{\Sigma}}^{(p)} (\bar{\mathbf{W}}^{(p)})^H$

5: **return** $\mathcal{S}_r(\mathcal{V}) \leftarrow \tilde{\mathcal{Z}}_P \times_1 \bar{\mathbf{U}}^{(1)} \times_2 \dots \times_P \bar{\mathbf{U}}^{(P)}$

Let us calculate the resulting cost. When $N_p \ll L_p$, instead of computing the SVD of $(\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle} \in \mathbb{F}^{N_p \times L_p}$ as described by Algorithm 2.1, one can proceed as follows. First, the eigenvalue decomposition of $(\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle} (\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle}^H \in \mathbb{F}^{N_p \times N_p}$ provides (only) the left singular vectors of $(\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle}$, which are then used for the projection stage (2.132). The goal is decomposing a much smaller matrix. Though the overall cost of the decomposition stage remains $\mathcal{O}(N_p^2 L_p)$ because of the matrix product $(\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle} (\tilde{\mathcal{Z}}_{p-1})_{\langle p \rangle}^H$, it is much faster in practice, compensating for the increased effort of using (2.132) instead of Line 4 of Algorithm 2.1. Hence, assuming the modal projections are ordered as $(1, \dots, P)$, the cost of applying \mathcal{S}_r is

$$\mathcal{O} \left(\sum_{p=1}^P N_p L_p \min\{N_p, L_p\} \right) + \mathcal{O} \left(\sum_{p=1}^P R_1 \dots R_p N_p \dots N_P \right) + \mathcal{O} \left(\sum_{p=1}^P N_1 \dots N_p R_p \dots R_P \right).$$

The first term corresponds to the computation of $\bar{\mathbf{U}}_p$, $p \in \langle P \rangle$, while the other ones refer respectively to the costs of (2.132) and of Line 5 of Algorithm 2.1.

The second advantage is the fact that its error can be *exactly* quantified by [209]

$$\left\| \mathcal{V} - \mathcal{V} \times_{p=1}^P \hat{\mathbf{P}}^{(p)} \right\|_F^2 = \sum_{p=1}^P \sum_{r_p=R_p+1}^{\min\{N_p, \bar{N}_p\}} \left[\sigma_{r_p}^{(p)} \left(\mathcal{V} \times_{q=1}^{p-1} \hat{\mathbf{P}}^{(q)} \right) \right]^2. \quad (2.135)$$

While performing the approximations, this can be exploited in order to adjust the mrank components R_p in order to approximately meet a prescribed tolerated error level, potentially yielding a less complex (in terms of mrank) approximation than THOSVD for a same tolerance level. We refer the reader to [209] for more details.

2.7 Final remarks and bibliographical notes

We would like to close this chapter by pointing out that many important aspects related to tensor algebra were not discussed here, as they are not essential for our objectives. A notable

example is the distinction between covariant, contravariant and mixed tensors, crucial in physics for dealing with quantities which are invariant with respect to the chosen coordinate bases. The famous Einstein summation convention gives an elegant way of writing tensor equations which hides the involved summations, being based on the implicit assumption that the occurrence of a same index for a covariant and a contravariant modes implies summation (an operation which is called contraction). See, *e.g.*, [21]. More recently, [80] has applied a similar notation to express matrix identities involving Kronecker and Khatri-Rao products, which is useful to write matrix unfoldings of certain tensor models in a concise way.

The main bibliographical sources consulted for this chapter are [131, 95, 49]. Other relevant sources of background material on tensors are [119, 133]. The interested reader can find interesting and in-depth discussions in these works and references therein.

Part I

Structured canonical polyadic decomposition

Structured CPD estimation and Cramér-Rao bounds

When applied to qualify a mathematical model, the adjective “structured” generally implies some constraint reducing its number of degrees of freedom. Put in another way, structured models generally comprise interdependent components, thus forming a (proper) subset of the set of all possible models. We describe in this chapter a class of CPDs whose matrix factors are structured in a linear sense. Its relevance is illustrated by means of an application example in which an inverse problem can be solved by estimating a structured CPD. Subsequently, we recall existing specialized algorithms which perform this task. Finally, expressions for both deterministic and Bayesian Cramér-Rao bounds associated with the underlying estimation problem are given.

Contents

3.1	Structured matrix factors and the structured CPD (SCPD)	52
3.2	Application example: Wiener-Hammerstein model identification	55
3.3	Existing SCPD computation methods	56
3.4	Cramér-Rao bound for SCPD estimation	63

3.1 Structured matrix factors and the structured CPD (SCPD)

Recall that the CPD of a P th-order tensor can be expressed in terms of P matrix factors, as in (2.86). In this part of the thesis, we study a special case of this decomposition where one or more of its factors is (are) subject to structural constraints of the following kind.

Definition 3.1 (Structured matrix [184]). A matrix $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$ is said to be *structured* if it belongs to a proper subspace of $\mathbb{C}^{N_p \times R}$. In other words, it can be written as

$$\mathbf{A}^{(p)} = \sum_{u=1}^{U_p} \theta_u^{(p)} \mathbf{E}_u^{(p)}, \quad (3.1)$$

where $U_p < N_p R$ and the matrices $\mathbf{E}_u^{(p)} \in \mathbb{C}^{N_p \times R}$ are linearly independent, thus forming a basis $\mathcal{B}_p = \{\mathbf{E}_1^{(p)}, \dots, \mathbf{E}_{U_p}^{(p)}\}$ of a subspace having dimension U_p .

In terms of the vectorization operator discussed in Section 2.2, we have

$$\text{vec}(\mathbf{A}^{(p)}) = \mathbf{E}^{(p)} \boldsymbol{\theta}^{(p)}, \quad (3.2)$$

where

$$\mathbf{E}^{(p)} \triangleq \left[\text{vec}(\mathbf{E}_1^{(p)}) \quad \dots \quad \text{vec}(\mathbf{E}_{U_p}^{(p)}) \right] \in \mathbb{C}^{N_p \times U_p} \quad (3.3)$$

and $\boldsymbol{\theta}^{(p)} \triangleq \left[\theta_1^{(p)} \quad \dots \quad \theta_{U_p}^{(p)} \right]^T \in \mathbb{C}^{U_p}$ holds the coordinates of the representation of $\mathbf{A}^{(p)}$ with respect to the basis \mathcal{B}_p . Each column of $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$ can be written in terms of $\boldsymbol{\theta}^{(p)}$ as

$$\mathbf{a}_r^{(p)} = \mathbf{S}_r^{(p)} \boldsymbol{\theta}^{(p)}, \quad (3.4)$$

where $\mathbf{S}_r^{(p)} \in \mathbb{C}^{N_p \times U_p}$ is defined such that its u th column equals the r th column of $\mathbf{E}_u^{(p)}$.

As particular cases of Definition 3.1, we can mention banded, Hankel, Toeplitz and circulant matrices, as well as block-Hankel, block-Toeplitz and block-circulant ones. For convenience, we provide in Table 3.1 expressions for U_p , $\mathbf{S}_r^{(p)}$ and $\mathbf{E}_u^{(p)}$ considering some classes of structured matrices (which are not unique, since they depend on the chosen basis).

Example 3.2. According to Table 3.1, any 3×2 Toeplitz matrix $\mathbf{A}^{(p)}$ can be written as

$$\mathbf{A}^{(p)} = \begin{bmatrix} c & d \\ b & c \\ a & b \end{bmatrix} = a \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} + b \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} + c \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} + d \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = a\mathbf{E}_1^{(p)} + b\mathbf{E}_2^{(p)} + c\mathbf{E}_3^{(p)} + d\mathbf{E}_4^{(p)}.$$

Thus, $\boldsymbol{\theta}^{(p)} = [a \quad b \quad c \quad d]^T$ and, for instance, the first column of $\mathbf{A}^{(p)}$ is given by

$$\begin{bmatrix} c \\ b \\ a \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\theta}^{(p)} = \mathbf{S}_1^{(p)} \boldsymbol{\theta}^{(p)}. \quad (3.5)$$

■

Example 3.3. Matrices with banded circulant structure (cf. Table 3.1) have the form

$$\mathbf{A}^{(p)} = \begin{bmatrix} a_1 & & & & \\ a_2 & \ddots & & & \\ \vdots & \ddots & & & \\ a_{U_p} & & a_1 & & \\ & & a_2 & & \\ & & & \ddots & \\ & & & & a_{U_p} \end{bmatrix} \in \mathbb{C}^{N_p \times R}, \quad (3.6)$$

	U_p	$\mathbf{S}_r^{(p)}, r \in \langle R \rangle$	$\mathbf{E}_u^{(p)}, u \in \langle U_p \rangle$
Unstructured	RN_p	$\begin{bmatrix} \mathbf{0}_{N_p \times N_p(r-1)} & \mathbf{I}_{N_p} & \mathbf{0}_{N_p \times N_p(R-r)} \end{bmatrix}$	$\left[\mathbf{E}_u^{(p)} \right]_{i,j} = \delta_{u,(j-1)N_p+i}$
Hankel	$N_p + R - 1$	$\begin{bmatrix} \mathbf{0}_{N_p \times r-1} & \mathbf{I}_{N_p} & \mathbf{0}_{N_p \times R-r} \end{bmatrix}$	$\left[\mathbf{E}_u^{(p)} \right]_{i,j} = \delta_{u,i+j-1}$
Toeplitz	$N_p + R - 1$	$\begin{bmatrix} \mathbf{0}_{N_p \times r-1} & \mathbf{e}_{N_p} & \dots & \mathbf{e}_1 & \mathbf{0}_{N_p \times R-r} \end{bmatrix}$	$\left[\mathbf{E}_u^{(p)} \right]_{i,j} = \delta_{u,N_p+j-i}$
(Toeplitz) Circulant	N_p	$\mathbf{\Pi}_p^{r-1} = \begin{bmatrix} \mathbf{0}_{1 \times N_p-1} & 1 \\ \mathbf{I}_{N_p-1} & \mathbf{0}_{N_p-1 \times 1} \end{bmatrix}^{r-1}$	$\left[\mathbf{E}_u^{(p)} \right]_{i,j} = \delta_{u,1+((i-j) \bmod N_p)}$
Banded circulant	$N_p - R + 1$	$\begin{bmatrix} \mathbf{0}_{r-1 \times U_p} \\ \mathbf{I}_{U_p} \\ \mathbf{0}_{R-r \times U_p} \end{bmatrix}$	$\left[\mathbf{E}_u^{(p)} \right]_{i,j} = \delta_{u,(i-j)+1}$

Table 3.1: Characteristics of $\mathbf{A}^{(p)}$ for some classes of matrices [19, 65].

with $U_p = N_p - R + 1$, where all omitted elements are null. We can also write $\mathbf{A}^{(p)} = \begin{bmatrix} \mathbf{a} & \mathbf{\Pi}_p \mathbf{a} & \mathbf{\Pi}_p^2 \mathbf{a} & \dots & \mathbf{\Pi}_p^{R-1} \mathbf{a} \end{bmatrix}$, where $\mathbf{\Pi}_p$ is the permutation matrix defined in the fourth row of Table 3.1 and $\mathbf{a} = [a_1 \ \dots \ a_{U_p} \ 0 \ \dots \ 0]^T \in \mathbb{C}^{N_p}$. ■

We proceed now to the definition of the structured CPD.

Definition 3.4 (Structured CPD). A rank- R tensor $\mathcal{X} \in \mathcal{T} \triangleq \bigotimes_{p=1}^P \mathbb{C}^{N_p}$ is said to admit a structured canonical polyadic decomposition (SCPD) when it can be written in the form (2.86) and at least one of the matrix factors $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$ is structured.

3.1.1 Uniqueness of the SCPD

As the imposition of constraints over factors of a CPD is intended to aid in the resolution of inverse problems (in the sense discussed in Section 2.5.1.1), model uniqueness is a key concern. In general, the trivial scaling and permutation indeterminacies are more restricted when we consider an SCPD. We present next an example illustrating this point.

Example 3.5. Consider the SCPD $\mathcal{X} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket$, where $\mathbf{A}^{(1)} = [a_{i,j}] \in \mathbb{C}^{3 \times 2}$ is a Hankel matrix. In this case, the condition $a_{i,1} \neq a_{l,1}$ for $i \neq l$, together with the Hankel structure of $\mathbf{A}^{(1)}$, is sufficient to eliminate the permutation ambiguity. The scaling ambiguity persists but is more restricted, because the scaling matrix $\text{Diag}(\alpha_1^{(1)}, \alpha_2^{(1)})$ of (2.95) must satisfy $\alpha_1^{(1)} = \alpha_2^{(1)}$. ■

In practice, the existence of *non-trivial* ambiguities is actually the main concern. By relying on the concept of krank and on Theorem 2.41, it may be possible to derive simple uniqueness conditions for some SCPDs, as in the following lemma.

Lemma 3.6. Every nonzero banded circulant matrix $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$ of the form shown in (3.6) has full krank (*i.e.*, $\text{krank}(\mathbf{A}^{(p)}) = R$).

Proof. Since $a_u \neq 0$ for some $u \in \langle U_p \rangle$, then clearly $\text{rank}(\mathbf{A}^{(p)}) = \text{krank}(\mathbf{A}^{(p)}) = R$. \square

If all factors are (nonzero) banded circulant, then (2.96) is automatically satisfied, and hence essential uniqueness holds. More generally, if the factors $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(Q)}$, with $Q \in \langle P \rangle$, have that structure, then condition (2.96) can be rewritten as

$$\sum_{p=Q+1}^P \text{krank}(\mathbf{A}^{(p)}) \geq P - (Q - 2)R - 1. \quad (3.7)$$

When $Q = P - 1$, for example, it can be checked the above inequality is satisfied for $P = 3$ provided $\text{krank}(\mathbf{A}^{(3)}) \geq 2$, while it is always satisfied for $P \geq 4$.

3.2 Application example: Wiener-Hammerstein model identification

We now discuss the application of the SCPD to the problem of Wiener-Hammerstein (WH) model identification, as originally exploited by [78]. The well-known WH model is often used for representing nonlinear dynamical systems [94]. Its time-invariant discrete-time version is illustrated in Fig. 3.1, where $g(\cdot)$ is a memoryless nonlinearity and $W(z)$, $H(z)$ are linear time-invariant systems. Because of its “modular” structure consisting of simple blocks, the WH model is said to belong to the class of block-oriented models [94].

When applying the WH model to some real-world nonlinear system, a central problem is that of identifying its parameters. It consists in estimating the parameters which constitute the representation of that system under the WH model form, given a set of experimental data (*i.e.*, input and output signal samples). We describe in the following the connection between the identification of a WH model and the computation of a SCPD.

Let us assume that the components of a given WH model have the form $g(x) = \sum_{p=1}^P g_p x^p$, $W(z) = \sum_{u=0}^{U-1} w_u z^{-u}$ and $H(z) = \sum_{r=0}^{R-1} h_r z^{-r}$. Note that the polynomial form of $g(\cdot)$ is suitable for the approximation of a smooth function by its Taylor expansion, truncated at order P . Then, the resulting input/output relation is

$$y(n) = \sum_{p=1}^P g_p \sum_{r=0}^{R-1} h_r \left[\sum_{m=r}^{U+r-1} w_{m-r} x(n-m) \right]^p, \quad (3.8)$$

where $x(n)$ is the input signal and $y(n)$ its corresponding output. By expanding the term between brackets, rearranging the ordering of the summations and manipulating indices, it can be shown that this model admits an equivalent Volterra representation [114]

$$y(n) = \sum_{p=1}^P \sum_{n_1=0}^{N-1} \cdots \sum_{n_p=0}^{N-1} v^{(p)}(n_1, \dots, n_p) \prod_{q=1}^p x(n - n_q), \quad (3.9)$$

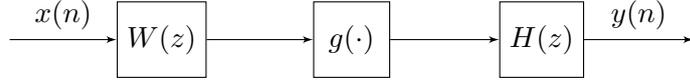


Figure 3.1: Block-diagram of the Wiener-Hammerstein model.

having symmetric discrete-time finite-memory Volterra kernels

$$v^{(p)}(n_1, \dots, n_p) = g_p \sum_{r=r_0}^{\tilde{R}} h_r \prod_{q=1}^p w_{n_q-r} \quad (3.10)$$

for $n_1, \dots, n_p \in \{0, \dots, N-1\}$, with $N = U + R - 1$, $r_0 = \max\{0, n_1 - U + 1, \dots, n_p - U + 1\}$ and $\tilde{R} = \min\{R - 1, n_1, \dots, n_p\}$.

Now, observe that (3.10) can be written in the same form as (2.85), provided we define $\lambda_r = h_{r-1}$ and $\mathbf{A}^{(1)} = \dots = \mathbf{A}^{(P)} = \mathbf{A} \in \mathbb{R}^{N \times R}$ such that

$$[\mathbf{A}]_{n_q, r} \triangleq \begin{cases} w_{n_q-r}, & r \leq n_q \leq U + r - 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3.11)$$

The above definition implies \mathbf{A} is banded circulant. Moreover, due to the scaling ambiguity of (3.10), one can assume, without loss of generality, that $g_p = 1$ (as long as $g_p \neq 0$). Also, assuming that $w_0 \neq 0$ (in other words, $W(z)$ does not comprise a pure delay), we can fix $w_0 = 1$, leaving the scaling in the vector $\boldsymbol{\lambda}$. So, associating $v^{(p)}(n_1, \dots, n_p)$ with a symmetric p th-order tensor $\mathbf{V}^{(p)} \in \mathcal{S} \left((\mathbb{R}^N)^{\otimes p} \right)$ such that $[\mathbf{V}^{(p)}]_{n_1, \dots, n_p} = v^{(p)}(n_1 - 1, \dots, n_p - 1)$, the parameters of $W(z)$ and $H(z)$ can be estimated by computing the SCPD

$$\mathbf{V}^{(p)} = \llbracket \boldsymbol{\lambda} ; \mathbf{A}, \dots, \mathbf{A} \rrbracket. \quad (3.12)$$

This reasoning underlies the approach described in [78], where it is assumed that the Volterra kernel $\mathbf{V}^{(p)}$ has been estimated from input-output samples (by using any available method, such as, *e.g.*, [199, 115]).

It is important to point out that estimates of the memory lengths of the filters, U and R , are necessary for applying the above described approach. This task can be performed by employing the structure identification algorithm proposed by [117] to the estimated Volterra kernels. Besides providing estimates of U and R , applying this technique allows one to validate the WH structure of the system of interest.

3.3 Existing SCPD computation methods

In this section, we present an overview of existing numerical methods aimed at computing the factors of an SCPD.

3.3.1 Alternating least squares with structural constraints

3.3.1.1 The alternating least squares method

By far, the most employed method to compute the CPD of a tensor in practical applications is that of alternating least squares (ALS). Its unparalleled popularity is evidenced by the frequent references to ALS as “the workhorse” of numerical CPD computation [203, 159]. This predominance can be basically attributed to its simplicity.

ALS is a rather general method for solving a least-squares problem involving a model which is multilinear with respect to certain blocks of parameters (such as, *e.g.*, matrix or vector components). It is based on the simple strategy of estimating these blocks in an alternating fashion, by fixing all but one of them, whose update is then given by a least-squares estimate. As such, it can be seen as a special case of the block-nonlinear Gauss-Seidel method (or cyclic block coordinate descent method) [203, 218, 204].

In the case of CPD estimation, one considers the problem

$$\min_{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}} J(\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}) = \min_{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}} \left\| \mathbf{y} - \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket \right\|_F^2, \quad (3.13)$$

where $\mathbf{A}^{(q)} \in \mathbb{C}^{N_q \times R}$, $q \in \langle P \rangle$, and $\mathbf{y} \in \mathcal{T}$ is typically a measured tensor which we wish to (approximately) decompose. By exploiting the isomorphic identity (2.87) and assuming estimates $\hat{\mathbf{A}}^{(q)}$ are available for all $\mathbf{A}^{(q)}$ with $q \in \langle P \rangle \setminus \{p\}$, one can rewrite the problem as

$$\min_{\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}} \left\| \mathbf{Y}_{\langle p \rangle} - \mathbf{A}^{(p)} \left(\hat{\mathbf{A}}^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}^{(p+1)} \diamond \hat{\mathbf{A}}^{(p-1)} \diamond \dots \diamond \hat{\mathbf{A}}^{(1)} \right)^T \right\|_F^2. \quad (3.14)$$

Assuming $\text{rank}(\hat{\mathbf{A}}^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}^{(p+1)} \diamond \hat{\mathbf{A}}^{(p-1)} \diamond \dots \diamond \hat{\mathbf{A}}^{(1)}) = R$, the solution to (3.14) is

$$\hat{\mathbf{A}}^{(p)} = \mathbf{Y}_{\langle p \rangle} \left(\hat{\mathbf{A}}^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}^{(p+1)} \diamond \hat{\mathbf{A}}^{(p-1)} \diamond \dots \diamond \hat{\mathbf{A}}^{(1)} \right)^+, \quad (3.15)$$

where $(\cdot)^+$ denotes the right inverse of the argument. This process is then repeated for all other $q \in \langle P \rangle$, and so on, until a stopping criterion is met. For instance, if one cycles through the modes in the order $1, \dots, P$, then the k th iteration of this scheme can be written as

$$\hat{\mathbf{A}}_k^{(1)} = \arg \min_{\mathbf{A}^{(1)} \in \mathbb{C}^{N_1 \times R}} J(\mathbf{A}^{(1)}, \hat{\mathbf{A}}_{k-1}^{(2)}, \dots, \hat{\mathbf{A}}_{k-1}^{(P)}) \quad (3.16)$$

$$\hat{\mathbf{A}}_k^{(2)} = \arg \min_{\mathbf{A}^{(2)} \in \mathbb{C}^{N_2 \times R}} J(\hat{\mathbf{A}}_k^{(1)}, \mathbf{A}^{(2)}, \hat{\mathbf{A}}_{k-1}^{(3)}, \dots, \hat{\mathbf{A}}_{k-1}^{(P)}) \quad (3.17)$$

⋮

$$\hat{\mathbf{A}}_k^{(P)} = \arg \min_{\mathbf{A}^{(P)} \in \mathbb{C}^{N_P \times R}} J(\hat{\mathbf{A}}_k^{(1)}, \dots, \hat{\mathbf{A}}_k^{(P-1)}, \mathbf{A}^{(P)}). \quad (3.18)$$

An often used stopping criterion is based on the (normalized) error between two consecutive tensor estimates [136, 193]: given values of the normalized squared error (NSE)

$$\text{NSE}(\hat{\mathbf{y}}_k; \mathbf{y}) = \frac{\left\| \mathbf{y} - \hat{\mathbf{y}}_k \right\|_F^2}{\left\| \mathbf{y} \right\|_F^2} \quad (3.19)$$

for two consecutive iterations $k-1$ and k , where $\hat{\mathbf{y}}_k = \llbracket \hat{\mathbf{A}}_k^{(1)}, \dots, \hat{\mathbf{A}}_k^{(P)} \rrbracket$, one checks whether

$$|\text{NSE}(\hat{\mathbf{y}}_k; \mathbf{y}) - \text{NSE}(\hat{\mathbf{y}}_{k-1}; \mathbf{y})| < \epsilon \quad (3.20)$$

holds for a given tolerance level ϵ . In other words, the algorithm stops whenever it does not achieve “sufficient progress” between two consecutive iterations.

The choice of R is clearly critical in the above scheme, but in practice it is generally unknown and NP-hard to compute (cf. Section 2.5.2.4). Even though in some applications the model rank is known beforehand, in practice this model is only an approximation of \mathbf{y} , due to the presence of noise [133]. Yet, as discussed in Section 2.5.2.5, a tensor may fail to have a best rank- R approximation. In that case, the ALS scheme can lead to factors which yield an arbitrarily low approximation error, although their columns contain meaningless elements with very large magnitude [155, 188].

3.3.1.2 Convergence of the alternating least squares method

By inspecting (3.16)–(3.18), one realizes that the cost function $J(\cdot)$ cannot increase from one iteration to another, as each estimated factor is conditionally optimal, given the previous estimates. On the surface, this may seem to guarantee convergence to a local minimum. It turns out, however, that even leaving aside existence issues (*i.e.*, assuming at least a global minimum exists for the chosen R), the convergence of ALS is a more complicated matter.

First, note that difficulties arise if no action is taken to suppress the scaling ambiguity of the iterates, which complicates the convergence analysis of the scheme (3.16)–(3.18) and can also cause an unstable behavior of the algorithm in practice [203]. To overcome this problem, a unit ℓ_2 norm can be imposed to every column of each factor; another option is to impose $\|\mathbf{a}_r^{(1)}\|_2 = \dots = \|\mathbf{a}_r^{(P)}\|_2$ for all r , where $\mathbf{a}_r^{(p)}$ denotes the r th column of $\mathbf{A}^{(p)}$ [203]. While this is sufficient in the real setting, since it leaves only sign ambiguities, it is not in the complex case, due to the possibility of rotations over the unit circle.

The particular case where $R = 1$ (over \mathbb{R}) has been studied in [204, 218]. In this special scenario, ALS can be seen as a generalization of the power method (see, *e.g.*, [90, Section 7.3.1]) and existence of solutions is not a concern, because every tensor admits a best rank-one approximation [69]. In [218] it is shown that in this case the ALS method converges linearly in a neighborhood of a local minimum, provided a certain matrix derived via the linearization of the cost function around that local minimum is nonsingular. A less restrictive global convergence result is derived in [204]. Concerning the general case ($R \geq 1$), [203] has shown the local convergence of ALS, also in the real setting, by relying on an assumption which implies essential uniqueness, but is stronger.

In practice, it is known that ALS often suffers from very slow convergence. Many researchers have reported that the algorithm is prone to traversing regions of very small

progress per iteration, which is known as a “swamp” and is related to the occurrence of quasi-collinear columns in the estimates. Besides increasing the computing time, this phenomenon can also induce a premature stop of the ALS scheme, since its stopping criterion is usually based on some measure of progress along the last iterations. The reader is referred to [155, 144] for interesting discussions on this issue and on ways of circumventing it.

3.3.1.3 Imposing structural constraints

Versions of ALS incorporating constraints have already been proposed in the literature. Yet, this is generally done in an *ad-hoc* manner, by exploiting special structure which is peculiar to a certain problem at hand. We briefly describe two examples in the following.

The authors of [84] came up with an ALS algorithm to compute a CPD of the form

$$\mathcal{Y} = \llbracket \mathbf{h}^* ; \mathbf{H}, \mathbf{H}^*, \mathbf{H} \rrbracket, \quad (3.21)$$

where $\mathbf{h} \in \mathbb{C}^R$ and $\mathbf{H} \in \mathbb{C}^{2R-1 \times R}$ is a Hankel matrix whose last column is given by $[\mathbf{h}^T \ 0 \ \dots \ 0]^T$. Using (2.88), \mathbf{h} is estimated at the first step of their algorithm via

$$\hat{\mathbf{h}}_k = \left[\left(\hat{\mathbf{H}}_{k-1} \diamond \hat{\mathbf{H}}_{k-1}^* \diamond \hat{\mathbf{H}}_{k-1} \right)^+ \text{vec}(\mathcal{Y}) \right]^*, \quad (3.22)$$

where $(\cdot)^+$ denotes here the left inverse of its argument. Subsequently, $\hat{\mathbf{H}}_k$ is constructed from $\hat{\mathbf{h}}_k$ according to its definition, and then the process is repeated. Despite preserving the model structure, the resulting sequence of iterates is no longer guaranteed to yield non-increasing cost function values, because the SCPD of (3.21) is *not* multilinear in \mathbf{h} and \mathbf{H} .

In [150], the same principle is employed to enforce the block-Toeplitz structure of a matrix factor, though the model in this case is not a CPD, but a TD. At each iteration, the parameters of this matrix are first estimated by exploiting the algebraic structure of the model, and then the structured matrix is used to estimate the other matrix components.

3.3.2 Non-iterative solution for a SCPD having a banded circulant matrix

The first work in which a special non-iterative algorithm is proposed to address a SCPD seems to be [116], where a third-order CPD comprising a banded circulant factor is considered.¹ The rationale behind that algorithm is explained in the following.

Assume that a given (nonzero) tensor $\mathcal{V} \in \mathbb{C}^{N_1} \otimes \mathbb{C}^{N_2} \otimes \mathbb{C}^{N_3}$ satisfies

$$\mathcal{V} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket, \quad (3.23)$$

where $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$. Without loss of generality, we suppose that the banded circulant factor in the above CPD is $\mathbf{A}^{(3)} = [a_{n,r}^{(3)}]$. So, according to Lemma 3.6, $\text{rank}(\mathbf{A}^{(3)}) = R$. The

¹Although it has the form shown in (3.6), that factor is referred to in [116] only as a Toeplitz matrix.

starting point is expression (2.87), which for $p = P = 3$ reads $\mathbf{V}_{\{3\}} = \mathbf{A}^{(3)} (\mathbf{A}^{(2)} \diamond \mathbf{A}^{(1)})^T$. With the further assumption that $\mathbf{B} = \mathbf{A}^{(2)} \diamond \mathbf{A}^{(1)} \in \mathbb{C}^{N_2 N_1 \times R}$ has full column rank, it can be verified that there exists a nonsingular matrix $\mathbf{D} \in \mathbb{C}^{R \times R}$ such that

$$\mathbf{U}\mathbf{D} = \mathbf{A}^{(3)} \quad \text{and} \quad \mathbf{W}\mathbf{\Sigma}\mathbf{D}^{-H} = \mathbf{B}^*, \quad (3.24)$$

where $\mathbf{V}_{\{3\}} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^H$ is the (economical) SVD of $\mathbf{V}_{\{3\}}$, with $\mathbf{U} \in \mathbb{C}^{N_3 \times R}$, $\mathbf{\Sigma} \in \mathbb{R}^{R \times R}$ and $\mathbf{W} \in \mathbb{C}^{N_2 N_1 \times R}$. In other words, \mathbf{U} provides a basis to $\text{colspace}(\mathbf{A}^{(3)})$ and likewise for \mathbf{W} and \mathbf{B}^* . The strategy of [116] consists in exploiting the structure of $\mathbf{A}^{(3)}$ and the knowledge of these bases in order to construct a system of linear equations in the elements of \mathbf{D} .

Let us assume also that $a_{1,1}^{(3)} \neq 0$. In that case, due to the scaling ambiguity in (3.23), one can impose $a_{1,1}^{(3)} = 1$. Recall that only the first $N_3 - R + 1$ components of the first column of $\mathbf{A}^{(3)}$ can be nonzero. Hence, $\left[a_{1,1}^{(3)} \ a_{N_3-R+2,1}^{(3)} \ a_{N_3-R+3,1}^{(3)} \ \cdots \ a_{N_3,1}^{(3)} \right]^T$ equals the canonical basis vector $\mathbf{e}_1^{(R)} \in \mathbb{C}^R$. One can thus construct a selection matrix

$$\mathbf{S}_1 = \left[\mathbf{e}_1^{(N_3)} \ \mathbf{e}_{N_3-R+2}^{(N_3)} \ \mathbf{e}_{N_3-R+3}^{(N_3)} \ \cdots \ \mathbf{e}_{N_3}^{(N_3)} \right]^T \in \mathbb{C}^{R \times N_3} \quad (3.25)$$

such that $\mathbf{S}_1 \mathbf{a}_1^{(3)} = \mathbf{e}_1^{(R)}$, where $\mathbf{a}_r^{(3)}$ denotes the r th column of $\mathbf{A}^{(3)}$. Now, as $\mathbf{A}^{(3)}$ is circulant, $\mathbf{a}_r^{(3)} = \mathbf{\Pi}_3^{r-1} \mathbf{a}_1^{(3)}$, where $\mathbf{\Pi}_3$ is as defined by Table 3.1. Defining $\mathbf{S}_r \triangleq \mathbf{S}_1 (\mathbf{\Pi}_3^{r-1})^T$, we have $\mathbf{S}_r \mathbf{a}_r^{(3)} = \mathbf{S}_1 (\mathbf{\Pi}_3^{r-1})^T \mathbf{\Pi}_3^{r-1} \mathbf{a}_1^{(3)} = \mathbf{e}_1^{(R)}$, because $\mathbf{\Pi}_3^r$ is orthogonal for all r . Due to (3.24), we can write $\mathbf{S}_r \mathbf{U} \mathbf{d}_r = \mathbf{e}_1^{(R)}$, where \mathbf{d}_r denotes the r th column of \mathbf{D} . This leads to the system of equations

$$\underbrace{\begin{bmatrix} \mathbf{S}_1 \mathbf{U} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \mathbf{U} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{S}_R \mathbf{U} \end{bmatrix}}_{\triangleq \mathbf{S}_U} \text{vec}(\mathbf{D}) = \begin{bmatrix} \mathbf{e}_1^{(R)} \\ \mathbf{e}_1^{(R)} \\ \vdots \\ \mathbf{e}_1^{(R)} \end{bmatrix} = \mathbf{1}_R \boxtimes \mathbf{e}_1^{(R)} \quad (3.26)$$

which comprehends R^2 equations in R^2 unknowns. As \mathbf{S}_U is nonsingular, \mathbf{D} can be estimated from it. Yet, more equations can be obtained by exploiting the structure of $\mathbf{A}^{(3)}$, as follows.

The idea is now to construct selection matrices which extract the parameters of $\mathbf{A}^{(3)}$ (*i.e.*, its elements that hold nonzero values) from each column $\mathbf{a}_r^{(3)}$. Concretely, by defining

$$\bar{\mathbf{S}}_r = \left[\mathbf{e}_r^{(N_3)} \ \mathbf{e}_{r+1}^{(N_3)} \ \cdots \ \mathbf{e}_{r+N_3-R}^{(N_3)} \right]^T \in \mathbb{C}^{N_3-R+1 \times N_3}, \quad (3.27)$$

we have $\bar{\mathbf{S}}_1 \mathbf{a}_1^{(3)} = \bar{\mathbf{S}}_2 \mathbf{a}_2^{(3)} = \cdots = \bar{\mathbf{S}}_R \mathbf{a}_R^{(3)}$. Thus, $\bar{\mathbf{S}}_r \mathbf{U} \mathbf{d}_r - \bar{\mathbf{S}}_{r+1} \mathbf{U} \mathbf{d}_{r+1} = \mathbf{0}$, which leads to the

system

$$\underbrace{\begin{bmatrix} \bar{\mathbf{S}}_1 \mathbf{U} & -\bar{\mathbf{S}}_2 \mathbf{U} & \mathbf{0} & \dots & \mathbf{0} \\ \bar{\mathbf{S}}_1 \mathbf{U} & \mathbf{0} & -\bar{\mathbf{S}}_3 \mathbf{U} & \dots & \mathbf{0} \\ \vdots & \vdots & & \ddots & \vdots \\ \bar{\mathbf{S}}_1 \mathbf{U} & \mathbf{0} & \dots & \mathbf{0} & -\bar{\mathbf{S}}_R \mathbf{U} \end{bmatrix}}_{\triangleq \bar{\mathbf{S}}_U} \text{vec}(\mathbf{D}) = \mathbf{0}_{(R-1)(N_3-R+1)}. \quad (3.28)$$

Hence, gathering together the above systems, we have

$$\underbrace{\begin{bmatrix} \mathbf{S}_U \\ \bar{\mathbf{S}}_U \end{bmatrix}}_{\triangleq \Phi} \text{vec}(\mathbf{D}) = \underbrace{\begin{bmatrix} \mathbf{1}_R \boxtimes \mathbf{e}_1^{(R)} \\ \mathbf{0} \end{bmatrix}}_{\triangleq \psi}, \quad (3.29)$$

which comprehends $R^2 + (R-1)(N_3 - R + 1) = R(N_3 + 2) - N_3 - 1$ equations in R^2 variables. Because Φ has full column rank [116], the least-squares solution is given by $\text{vec}(\mathbf{D}) = (\Phi^H \Phi)^{-1} \Phi^H \psi$. Once \mathbf{D} is estimated, $\mathbf{A}^{(3)}$ and \mathbf{B} can then be recovered from (3.24). Subsequently, the factors $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ can be computed from $\mathbf{b}_r = \text{vec}(\mathbf{a}_r^{(1)} \mathbf{a}_r^{(2)T}) = \mathbf{a}_r^{(2)} \boxtimes \mathbf{a}_r^{(1)}$ by appropriately rearranging the elements of \mathbf{b}_r in a $N_1 \times N_2$ matrix \mathbf{B}_r and computing its best rank-one approximation.

3.3.3 Subspace-based solution

The approach proposed in [184] is essentially a generalization of the previous one, being able to handle other structural constraints and allowing the estimation of multiple structured factors. Moreover, in certain cases a solution can be obtained in a non-iterative manner, as explained below.

Suppose $\mathcal{V} \in \bigotimes_{q=1}^P \mathbb{C}^{N_q}$ admits an SCPD with p structured factors. For simplicity of presentation, assume that the factors $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(p)}$ are structured, with $\mathbf{A}^{(q)} \in \mathbb{C}^{N_q \times R}$. So, defining $\mathbf{B} \triangleq \mathbf{A}^{(p)} \diamond \dots \diamond \mathbf{A}^{(p+1)}$, we have from (2.89)

$$\mathbf{V}_{[p]} = \left(\mathbf{A}^{(p)} \diamond \dots \diamond \mathbf{A}^{(1)} \right) \mathbf{B}^T \in \mathbb{C}^{\prod_{q=1}^p N_q \times \prod_{q=p+1}^P N_q}. \quad (3.30)$$

We assume here that both $\mathbf{A}^{(p)} \diamond \dots \diamond \mathbf{A}^{(1)}$ and \mathbf{B} have full column rank. For brevity, let us denote $M_1 = \prod_{q=1}^p N_q$ and $M_2 = \prod_{q=p+1}^P N_q$. The above matrix unfolding can also be written in terms of its (economical) SVD as $\mathbf{V}_{[p]} = \mathbf{U} \Sigma \mathbf{W}^H$, where $\mathbf{U} \in \mathbb{C}^{M_1 \times R}$, $\Sigma \in \mathbb{R}^{R \times R}$ and $\mathbf{W} \in \mathbb{C}^{M_2 \times R}$. Just as in the previous section, there is a nonsingular $\mathbf{D} \in \mathbb{C}^{R \times R}$ satisfying

$$\mathbf{A}^{(p)} \diamond \dots \diamond \mathbf{A}^{(1)} = \mathbf{U} \mathbf{D} \quad \text{and} \quad \mathbf{W} \Sigma \mathbf{D}^{-H} = \mathbf{B}^*. \quad (3.31)$$

Now, in order to exploit the structures of the factors $\mathbf{A}^{(q)}$, $q \in \langle p \rangle$, more generally than the approach of Section 3.3.2, that of [184] takes into account the fact that, according to

Definition 3.1, there exists a basis $\mathcal{B}_q = \{\mathbf{E}_{n_q}^{(q)}\}_{n_q=1}^{N_q}$ which characterizes the structure of $\mathbf{A}^{(q)}$, and under which one can write $\mathbf{A}^{(q)} = \sum_{n_q=1}^{N_q} \theta_{n_q}^{(q)} \mathbf{E}_{n_q}^{(q)}$. Vectorizing both sides of the first relation in (3.31) and applying property (2.49), we have $\text{vec}(\mathbf{UD}) = (\mathbf{I}_R \boxtimes \mathbf{U}) \text{vec}(\mathbf{D}) = \text{vec}(\mathbf{A}^{(p)} \diamond \cdots \diamond \mathbf{A}^{(1)})$. As by definition the r th column of $\mathbf{A}^{(p)} \diamond \cdots \diamond \mathbf{A}^{(1)}$ equals $\mathbf{a}_r^{(p)} \boxtimes \cdots \boxtimes \mathbf{a}_r^{(1)}$, then using (3.4) we obtain

$$(\mathbf{I}_R \boxtimes \mathbf{U}) \text{vec}(\mathbf{D}) = \begin{bmatrix} (\mathbf{S}_1^{(p)} \boldsymbol{\theta}^{(p)}) \boxtimes \cdots \boxtimes (\mathbf{S}_1^{(1)} \boldsymbol{\theta}^{(1)}) \\ \vdots \\ (\mathbf{S}_R^{(p)} \boldsymbol{\theta}^{(p)}) \boxtimes \cdots \boxtimes (\mathbf{S}_R^{(1)} \boldsymbol{\theta}^{(1)}) \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1^{(p)} \boxtimes \cdots \boxtimes \mathbf{S}_1^{(1)} \\ \vdots \\ \mathbf{S}_R^{(p)} \boxtimes \cdots \boxtimes \mathbf{S}_R^{(1)} \end{bmatrix} \boldsymbol{\zeta},$$

where $\boldsymbol{\zeta} \triangleq \boldsymbol{\theta}^{(p)} \boxtimes \cdots \boxtimes \boldsymbol{\theta}^{(1)} \in \mathbb{C}^{U_1 \cdots U_p}$. Finally, using the definition of $\mathbf{S}_r^{(q)} \in \mathbb{C}^{N_q \times U_q}$, we write

$$(\mathbf{I}_R \boxtimes \mathbf{U}) \text{vec}(\mathbf{D}) = \underbrace{\begin{bmatrix} \tilde{\mathbf{e}}_{1,1,\dots,1,1} & \tilde{\mathbf{e}}_{1,1,\dots,1,2} & \cdots & \tilde{\mathbf{e}}_{1,1,\dots,1,U_1} & \tilde{\mathbf{e}}_{2,1,\dots,1,1} & \tilde{\mathbf{e}}_{2,1,\dots,1,2} & \cdots & \tilde{\mathbf{e}}_{U_p,U_{p-1},\dots,U_2,U_1} \end{bmatrix}}_{\triangleq \tilde{\mathbf{E}}} \boldsymbol{\zeta}, \quad (3.32)$$

where $\tilde{\mathbf{e}}_{n_p,\dots,n_1} = \text{vec}(\mathbf{E}_{n_p}^{(p)} \diamond \cdots \diamond \mathbf{E}_{n_1}^{(1)})$.

The above development yields the following system of $R \prod_{q=1}^p N_q$ linear equations

$$\mathbf{G} \begin{bmatrix} \text{vec}(\mathbf{D}) \\ \boldsymbol{\zeta} \end{bmatrix} = \mathbf{0}, \quad \text{with} \quad \mathbf{G} = \begin{bmatrix} -\mathbf{I}_R \boxtimes \mathbf{U} & \tilde{\mathbf{E}} \end{bmatrix} \quad (3.33)$$

in $R^2 + \prod_{q=1}^p U_q$ variables. If this system admits a unique solution, then it can be sought by computing the right singular vector of \mathbf{G} associated to its smallest singular value. As this expedient does not take the Kronecker structure of $\boldsymbol{\zeta}$ into account, for $p > 1$ it is necessary to construct from the estimate $\hat{\boldsymbol{\zeta}}$ a p th-order tensor (using (2.55)) whose best rank-one approximation then yields an estimate for each $\boldsymbol{\theta}^{(q)}$, $q \in \langle p \rangle$. When $p = 2$, such an approximation can be computed with an SVD, but for $p > 2$ iterative algorithms such as those proposed in [218, 59] are required.

The computation of $\mathbf{A}^{(p+1)}, \dots, \mathbf{A}^{(P)}$ follows the same logic. Namely, after recovering \mathbf{B} from (3.31) and the obtained estimate of \mathbf{D} , each column \mathbf{b}_r of \mathbf{B} is then rearranged as a $(P-p)$ th-order tensor, whose best rank-one approximation gives estimates of $\mathbf{a}_r^{(p+1)}, \dots, \mathbf{a}_r^{(P)}$. One can therefore conclude that this method is particularly attractive when either $P = 3$ or $P = 4$ and $p = 2$, because then only standard numerical linear algebra routines are needed.

A similar procedure is also given in [184] specifically for banded matrices with no further structure. In this case, R systems of linear equations are derived, each one related to one column of $\mathbf{A}^{(p)} \diamond \cdots \diamond \mathbf{A}^{(1)}$. These systems can be solved independently, yielding estimates for the respective columns of the structured factors (as long as their solutions are unique).

As mentioned above, a key concern in this method is whether the solution of the constructed system(s) of equation is (are) unique. Sufficient conditions for guaranteeing so are derived in [184], but covering only the cases where the factors are banded and possibly also Toeplitz or Hankel. Yet, there are other scenarios to which this method applies, such as when the structured factors are tall circulant matrices.

The main drawbacks of this approach are:

1. It cannot handle certain structures. For instance, if $p \geq 2$ and the structured factors are Toeplitz but neither banded nor circulant, then some $\tilde{\mathbf{e}}_{u_p, \dots, u_1}$ vectors in (3.32) are necessarily null. Consequently, \mathbf{G} possesses multiple null singular values, with an associated subspace of right singular vectors, and hence (3.33) admits infinitely many solutions. In general, uniqueness is not guaranteed whenever² $\text{rank}(\mathbf{G}) < R^2 + \prod_{q=1}^p U_q - 1$. We have observed that this also happens in scenarios where square circulant factors are considered.
2. It cannot take into account the structure of all P factors jointly, since $\mathbf{V}_{[p]}$ must be such that $p \leq P - 1$.

3.4 Cramér-Rao bound for SCPD estimation

We review in this section the material presented in [19, 65, 63], which concerns the derivation of formulas for the Cramér-Rao bound associated with the SCPD estimation problem. These formulas shall be employed in Chapter 5, where we numerically compare several algorithms in terms of their statistical estimation performance by means of Monte Carlo experiments.

First, we provide some basic definitions and notation related to estimation theory. For the sake of simplicity, our presentation will be restricted to the real setting, as in [65]. Yet, an extension to the complex case can be derived by relying, *e.g.*, on [113, Chap. 15].

3.4.1 Information inequality and the Cramér-Rao bound (CRB)

The information inequality is a mathematical result rooted in estimation theory and often invoked for evaluating estimators. Essentially, it states that (under certain conditions) the mean squared error (MSE) achieved by any unbiased estimator cannot be smaller than the trace of the inverse of the Fisher information matrix (FIM) associated with the estimation problem. We now recall this result, which will require some definitions.

Let $\mathbf{y}(\boldsymbol{\eta})$ be a random variable containing information about some parameter vector $\boldsymbol{\eta}$, which we wish to estimate, and let $p(\mathbf{y}; \boldsymbol{\eta})$ be its probability distribution function (p.d.f.).

²Of course, when noise is present \mathbf{G} has full column rank almost surely, but then the unique solution of (3.33) is merely an artifact produced by the noise.

Given a measured \mathbf{y} , we denote an estimator of $\boldsymbol{\eta}$ by $\hat{\boldsymbol{\eta}}(\mathbf{y})$. A natural performance criterion for the evaluation of $\hat{\boldsymbol{\eta}}(\mathbf{y})$ is the **MSE**, defined as

$$\text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) = \mathbb{E} \left\{ \|\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}(\mathbf{y})\|_2^2 \right\}, \quad (3.34)$$

where \mathbb{E} denotes mathematical expectation (with respect to $p(\mathbf{y}; \boldsymbol{\eta})$). Note that the true parameter vector $\boldsymbol{\eta}$ is assumed here to be fixed in a deterministic way.

In this context, the *Fisher information* is a valuable tool, because it quantifies the amount of information on $\boldsymbol{\eta}$ which is provided by a sample of \mathbf{y} . This measure is directly related to how strongly $p(\mathbf{y}; \boldsymbol{\eta})$ depends on $\boldsymbol{\eta}$, as formalized next.

Definition 3.7. (Fisher information matrix) Let $\mathbf{y} \in \mathbb{R}^{\bar{N}}$ be a random variable whose p.d.f. $p(\mathbf{y}; \boldsymbol{\eta})$ depends on the parameter vector $\boldsymbol{\eta} \in \mathbb{R}^M$ and assume that $p(\mathbf{y}; \boldsymbol{\eta})$ is continuously differentiable with respect to $\boldsymbol{\eta}$. The Fisher information matrix (**FIM**) of \mathbf{y} is defined as

$$\mathbf{F}(\boldsymbol{\eta}) \triangleq \mathbb{E} \left\{ \left(\frac{\partial \log p(\mathbf{y}; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right) \left(\frac{\partial \log p(\mathbf{y}; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right)^T \right\} \in \mathbb{R}^{M \times M}. \quad (3.35)$$

If $p(\mathbf{y}; \boldsymbol{\eta})$ is twice continuously differentiable with respect to $\boldsymbol{\eta}$,

$$\mathbb{E} \left\{ \frac{\partial \log p(\mathbf{y}; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}} \right\} = \mathbf{0}, \quad \text{for all } \boldsymbol{\eta}, \quad (3.36)$$

and some further mild regularity conditions hold (see, *e.g.*, [128, Section 2.6]), then the components of $\mathbf{F}(\boldsymbol{\eta})$ can be written as $[\mathbf{F}(\boldsymbol{\eta})]_{m_1, m_2} = \mathbb{E} \left\{ -\frac{\partial^2 \log p(\mathbf{y}; \boldsymbol{\eta})}{\partial [\boldsymbol{\eta}]_{m_1} \partial [\boldsymbol{\eta}]_{m_2}} \right\}$.

By definition, $\mathbf{F}(\boldsymbol{\eta})$ is symmetric and positive semidefinite. When it is positive definite, then the information inequality can be shown under certain conditions, as stated below.

Theorem 3.8 (Information inequality [113, 189]). Let $\boldsymbol{\eta}$ be the (deterministic) parameter vector on which the random variable \mathbf{y} depends. Assume that the **FIM** of $p(\mathbf{y}; \boldsymbol{\eta})$ exists and is positive definite and that $p(\mathbf{y}; \boldsymbol{\eta})$ fulfills the regularity condition (3.36). If $\hat{\boldsymbol{\eta}}(\mathbf{y})$ is an unbiased estimator of $\boldsymbol{\eta}$, then the covariance matrix of its produced estimates satisfies $\mathbb{E} \left\{ (\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}(\mathbf{y})) (\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}(\mathbf{y}))^T \right\} \succeq \mathbf{F}^{-1}(\boldsymbol{\eta})$. Defining $\mathbf{C}(\boldsymbol{\eta}) \triangleq \mathbf{F}^{-1}(\boldsymbol{\eta})$ and taking the trace of both sides, this implies

$$\text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) \geq \text{trace}(\mathbf{C}(\boldsymbol{\eta})). \quad (3.37)$$

The right-hand side of the above equation is referred to as Cramér-Rao lower bound (**CRB**).

Proof. See, *e.g.*, [113, 128]. □

3.4.2 SCPD model

In practice, the assumption that a measured tensor admits a **SCPD** is at best a model which approximates the available data. First, because it is generally corrupted by noise.

Also, modeling errors are usually present, due to deviations from assumed ideal conditions. Hence, a modeled tensor $\mathbf{Y} \in \mathcal{T} = \bigotimes_{p=1}^P \mathbb{C}^{N_p}$ can be written as

$$\mathbf{Y} = \mathbf{X} + \mathcal{N}, \quad (3.38)$$

where $\mathbf{X} \in \mathcal{T}$ admits a SCPD and $\mathcal{N} \in \mathcal{T}$ is a random variable accounting for the aforementioned imperfections.

In order to present the derivation of CRBs associated with the SCPD estimation problem, it will be helpful to introduce a vectorized form of (3.38). By exploiting (3.4) together with (2.58), the SCPD of \mathbf{X} can be expressed as

$$\mathbf{x} \triangleq \text{vec}(\mathbf{X}) = \sum_{r=1}^R \lambda_r \left(\mathbf{a}_r^{(P)} \boxtimes \cdots \boxtimes \mathbf{a}_r^{(1)} \right) = \sum_{r=1}^R \lambda_r \left[\left(\mathbf{S}_r^{(P)} \boldsymbol{\theta}^{(P)} \right) \boxtimes \cdots \boxtimes \left(\mathbf{S}_r^{(1)} \boldsymbol{\theta}^{(1)} \right) \right]. \quad (3.39)$$

Defining $\Phi_r \triangleq \mathbf{S}_r^{(P)} \boxtimes \cdots \boxtimes \mathbf{S}_r^{(1)}$ and using the mixed-product property (2.48) of the Kronecker product, we can rewrite \mathbf{x} as a function of the parameter vector $\boldsymbol{\nu} \triangleq [\boldsymbol{\theta}^T \quad \boldsymbol{\lambda}^T]^T \in \mathbb{R}^K$, where $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}^{(1)T} \quad \cdots \quad \boldsymbol{\theta}^{(P)T}]^T$, having the form

$$\mathbf{x}(\boldsymbol{\nu}) = \underbrace{\left(\sum_{r=1}^R \lambda_r \Phi_r \right)}_{\triangleq \Phi(\boldsymbol{\lambda})} \underbrace{\left(\boldsymbol{\theta}^{(P)} \boxtimes \cdots \boxtimes \boldsymbol{\theta}^{(1)} \right)}_{\triangleq \mathbf{f}(\boldsymbol{\theta})} \in \mathbb{R}^{\bar{N}}. \quad (3.40)$$

Note that $K = \sum_p U_p + R$. Finally, defining $\mathbf{n} \triangleq \text{vec}(\mathcal{N}) \in \mathbb{R}^{\bar{N}}$, we have the vectorized model

$$\mathbf{y}(\boldsymbol{\nu}) \triangleq \text{vec}(\mathbf{Y}) = \Phi(\boldsymbol{\lambda}) \mathbf{f}(\boldsymbol{\theta}) + \mathbf{n} \in \mathbb{R}^{\bar{N}}. \quad (3.41)$$

Here, we consider that the noise tensor \mathcal{N} is composed of independent and identically distributed (i.i.d.) zero-mean Gaussian elements of variance σ^2 . Under such an additive white Gaussian noise (AWGN) assumption, it is well known that maximum likelihood estimates are obtained by solving the (nonlinear) least-squares problem

$$\min_{\boldsymbol{\nu}} \|\mathbf{y} - \mathbf{x}(\boldsymbol{\nu})\|_2^2. \quad (3.42)$$

Depending on the structure of the SCPD factors, different algorithms can be employed for addressing problem (3.42). Clearly, the ALS method is appropriate since its target cost function is precisely a least-squares one. We shall formulate a constrained version of this method in Chapter 4. Even though the other (algebraic) approaches presented in Section 3.3 are not based on a least-squares criterion, they are often able to provide a relatively accurate approximate solution, provided the signal-to-noise ratio (SNR) is sufficiently high. Another algebraic approach, aimed at estimating circulant factors, shall also be proposed in Chapter 4.

3.4.3 Model identifiability

Before proceeding to the derivation of **CRB** formulas, a discussion on identifiability of our model is necessary. The parameter vector $\boldsymbol{\nu}$ is said to be locally identifiable when any point $\boldsymbol{\nu}_0$ has a neighborhood in which the mapping $\boldsymbol{\nu} \mapsto \mathbf{x}(\boldsymbol{\nu})$ is injective. This property is necessary for existence of a finite **CRB**, since otherwise the **FIM** is singular. To see this, note from the zero-mean **AWGN** hypothesis we have

$$\frac{\partial \log p(\mathbf{y}; \boldsymbol{\nu})}{\partial \boldsymbol{\nu}} = \frac{\partial}{\partial \boldsymbol{\nu}} \log \left[(2\pi\sigma^2)^{-\frac{\bar{N}}{2}} \exp \left(-\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{x}(\boldsymbol{\nu})\|_2^2 \right) \right] \quad (3.43)$$

$$= -\frac{\partial}{\partial \boldsymbol{\nu}} \left(\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{x}(\boldsymbol{\nu})\|_2^2 \right) = \frac{1}{\sigma^2} \mathbf{J}(\boldsymbol{\nu})^T \mathbf{n}, \quad (3.44)$$

where $\mathbf{J}(\boldsymbol{\nu}) = \frac{\partial}{\partial \boldsymbol{\nu}} \mathbf{x}(\boldsymbol{\nu}) \in \mathbb{R}^{\bar{N} \times K}$ is the Jacobian of the (vectorized) model with respect to $\boldsymbol{\nu}$. Hence, since $\mathbb{E}\{\mathbf{n}\mathbf{n}^T\} = \sigma^2 \mathbf{I}_{\bar{N}}$, we have from (3.35)

$$\mathbf{F}(\boldsymbol{\nu}) = \frac{1}{\sigma^2} \mathbf{J}(\boldsymbol{\nu})^T \mathbf{J}(\boldsymbol{\nu}). \quad (3.45)$$

Therefore, $\mathbf{F}(\boldsymbol{\nu})$ is nonsingular only if $\boldsymbol{\nu}$ is locally identifiable, for otherwise $\text{rank}(\mathbf{J}(\boldsymbol{\nu})) < K$ (by the inverse function theorem), implying $\text{rank}(\mathbf{F}(\boldsymbol{\nu})) < K$. Global identifiability, in its turn, means $\boldsymbol{\nu} \mapsto \mathbf{x}(\boldsymbol{\nu})$ is injective over all \mathbb{R}^K , and is not required for existence of the **CRB**.

As discussed in Section 2.5.1.3, the **CPD** is inherently subject to permutation and scaling ambiguities. It is easy to see that the former has no influence over the injectivity of $\mathbf{x}(\boldsymbol{\nu})$, despite the fact that it must be taken into account when assessing performances through computer simulations, for correctly measuring estimation errors. On the other hand the scaling ambiguity implies the existence of infinitely many vectors $\boldsymbol{\nu}$ yielding the same $\mathbf{x}(\boldsymbol{\nu})$, thus precluding (3.40) from being (locally or globally) injective.

For suppressing the scaling indeterminacy, degrees of freedom (**DOF**) “in excess” must be eliminated, according to the structure of the factors. If, *e.g.*, a factor $\mathbf{A}^{(p)}$ is circulant, imposing $\theta_u^{(p)} = 1$ for some u is sufficient to fix its scaling, due to its structure (this excludes, though, the case in which $\theta_u^{(p)} = 0$). Another option would be to impose unit norm for each column, which also eliminates one **DOF** (as one $\theta_u^{(p)}$ becomes a function of the others). Alike measures must be taken for the other factors, so that their scaling is absorbed by $\boldsymbol{\lambda}$.

Henceforth, we consider that for each $\boldsymbol{\theta}^{(p)}$, a reduced version $\bar{\boldsymbol{\theta}}^{(p)}$ including only the minimal necessary number of **DOF** has been appropriately defined. For simplicity, we assume that the V_p first elements of $\boldsymbol{\theta}^{(p)}$ are fixed and denote $\bar{U}_p \triangleq U_p - V_p$. Thus, $\bar{\boldsymbol{\theta}}^{(p)} = \mathbf{B}_p \boldsymbol{\theta}^{(p)}$, where $\mathbf{B}_p = \begin{bmatrix} \mathbf{e}_{V_p+1}^{(U_p)} & \dots & \mathbf{e}_{U_p}^{(U_p)} \end{bmatrix}^T \in \mathbb{R}^{\bar{U}_p \times U_p}$ is a selection matrix. A vector $\bar{\boldsymbol{\theta}}$ is defined analogously to $\boldsymbol{\theta}$, so that we can consider the reduced vector of parameters $\boldsymbol{\eta} \triangleq [\bar{\boldsymbol{\theta}}^T \boldsymbol{\lambda}^T]^T \in \mathbb{R}^M$, with $M \triangleq \sum_p \bar{U}_p + R$. Its corresponding model will be denoted by

$$\mathbf{y}(\boldsymbol{\eta}) = \mathbf{x}(\boldsymbol{\eta}) + \mathbf{n} \in \mathbb{R}^{\bar{N}} \quad (3.46)$$

and the **FIM** and Jacobian matrices are from now on denoted by $\mathbf{F}(\boldsymbol{\eta})$ and $\mathbf{J}(\boldsymbol{\eta})$, respectively.

3.4.4 CRB for SCPD estimation under additive white Gaussian noise

Applying the Cramér-Rao inequality to our setting, we have

$$\text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) \geq \text{trace}(\mathbf{C}(\boldsymbol{\eta})) = \sum_{p=1}^P \sum_{u=1}^{\bar{U}_p} \text{CRB}(\bar{\theta}_u^{(p)}) + \sum_{r=1}^R \text{CRB}(\lambda_r), \quad (3.47)$$

where $\text{CRB}(\bar{\theta}_u^{(p)}) = [\mathbf{C}(\boldsymbol{\eta})]_{v,v}$, with $v = \sum_{p'=1}^{p-1} \bar{U}_{p'} + u$, and $\text{CRB}(\lambda_r) = [\mathbf{C}(\boldsymbol{\eta})]_{s,s}$, with $s = \sum_p \bar{U}_p + r$. In order to derive these diagonal elements of $\mathbf{C}(\boldsymbol{\eta})$, we can employ (3.45) and the definition of $\mathbf{C}(\boldsymbol{\eta})$, which together imply

$$\mathbf{C}(\boldsymbol{\eta}) = \sigma^2 (\mathbf{J}(\boldsymbol{\eta})^T \mathbf{J}(\boldsymbol{\eta}))^{-1}. \quad (3.48)$$

Hence, the key for deriving the CRB of a SCPD model consists in calculating its corresponding Jacobian, which depends on the structure of the factors and also on some model assumptions. We briefly describe below the formulas derived in [65] for different cases.

3.4.4.1 No identical factors

We assume here that $\mathbf{A}^{(p)} \neq \mathbf{A}^{(q)}$ for all $p \neq q$. Two cases are considered, as discussed below.

1) *Arbitrary $\boldsymbol{\lambda}$* . Here, the Jacobian matrix is given by

$$\mathbf{J}(\boldsymbol{\eta}) = \left[\underbrace{\mathbf{J}_{\bar{\boldsymbol{\theta}}^{(1)}}(\boldsymbol{\eta}) \quad \dots \quad \mathbf{J}_{\bar{\boldsymbol{\theta}}^{(P)}}(\boldsymbol{\eta})}_{\triangleq \mathbf{J}_{\bar{\boldsymbol{\theta}}}} \quad \mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}) \right], \quad (3.49)$$

where $\mathbf{J}_{\bar{\boldsymbol{\theta}}^{(p)}}(\boldsymbol{\eta})$ contains the derivatives of $\mathbf{x}(\boldsymbol{\eta})$ with respect to $\bar{\boldsymbol{\theta}}^{(p)} = \mathbf{B}_p \boldsymbol{\theta}^{(p)}$, and $\mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta})$ is likewise. From (3.40), we can derive

$$\mathbf{J}_{\bar{\boldsymbol{\theta}}^{(p)}}(\boldsymbol{\eta}) = \boldsymbol{\Phi}(\boldsymbol{\lambda}) \left(\boldsymbol{\theta}^{(P)} \boxtimes \dots \boxtimes \boldsymbol{\theta}^{(p+1)} \boxtimes \mathbf{B}_p^T \boxtimes \boldsymbol{\theta}^{(p-1)} \boxtimes \dots \boxtimes \boldsymbol{\theta}^{(1)} \right), \quad p \in \langle P \rangle, \quad (3.50)$$

$$\mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}) = \left[\boldsymbol{\Phi}_1 \mathbf{f}(\boldsymbol{\theta}) \quad \dots \quad \boldsymbol{\Phi}_R \mathbf{f}(\boldsymbol{\theta}) \right]. \quad (3.51)$$

For notational simplicity, we shall omit the argument of $\mathbf{J}(\boldsymbol{\eta})$ whenever convenient. Relying on these formulas, the following result, shown in [65], can be derived.

Proposition 3.9. The CRB for the u th element of $\bar{\boldsymbol{\theta}}^{(p)}$ in (3.47) is given by

$$\text{CRB}(\bar{\theta}_u^{(p)}) = \frac{\sigma^2}{\|\mathbf{g}_{u,p}\|_2^2 - \|\mathbf{P}_{u,p} \mathbf{g}_{u,p}\|_2^2}, \quad (3.52)$$

where $\mathbf{g}_{u,p}$ is the column of $\mathbf{J}_{\bar{\boldsymbol{\theta}}}$ containing the derivatives with respect to $\bar{\theta}_u^{(p)}$, $\mathbf{G}_{u,p}$ contains all the other columns of $\mathbf{J}_{\bar{\boldsymbol{\theta}}}$ (in any order) and $\mathbf{P}_{u,p}$ is the orthogonal projector onto the column space of $[\mathbf{G}_{u,p} \quad \mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta})]$. Similarly, the CRB for the r th element of $\boldsymbol{\lambda}$ is

$$\text{CRB}(\lambda_r) = \frac{\sigma^2}{\|\mathbf{d}_r\|_2^2 - \|\mathbf{P}_r \mathbf{d}_r\|_2^2}, \quad (3.53)$$

where \mathbf{d}_r is the column of \mathbf{J}_λ associated with λ_r , \mathbf{D}_r holds all other columns (in any order) and \mathbf{P}_r is the orthogonal projector onto the column space of $[\mathbf{J}_{\bar{\theta}}(\boldsymbol{\eta}) \ \mathbf{D}_r]$.

- 2) $\boldsymbol{\lambda}$ with identical elements. If we assume now that $\lambda_1 = \dots = \lambda_R = \lambda$, then we can impose $\lambda = 1$ without loss of generality,³ due to the scaling ambiguity. The parameter vector is then $\boldsymbol{\eta} = \bar{\boldsymbol{\theta}}$. Under this assumption, we have the following result, also from [65].

Proposition 3.10. The closed-form expression of the lower bound on the MSE is

$$\text{MSE}(\hat{\boldsymbol{\eta}}) \geq \text{trace}(\mathbf{C}(\boldsymbol{\eta})) = \sum_{p=1}^P \sum_{u=1}^{\bar{U}_p} \text{CRB} \left(\bar{\theta}_u^{(p)} \right), \quad (3.54)$$

in which $\text{CRB} \left(\bar{\theta}_u^{(p)} \right)$ is given by (3.52), where $\mathbf{g}_{u,p}$ and $\mathbf{G}_{u,p}$ are as defined in Proposition 3.9 and $\mathbf{P}_{u,p}$ is now the orthogonal projector onto the column space of $\mathbf{G}_{u,p}$.

3.4.4.2 Symmetric (or partially symmetric) model

If identical factors exist, then the formulas (3.52)–(3.53) are still valid, but the structure of the resulting Jacobian becomes more complex. Assume that only $Q \leq P$ distinct matrix factors are involved in the SCPD of \mathcal{X} . Without loss of generality, this can be expressed as

$$\mathcal{X} = \left[\boldsymbol{\lambda}; \underbrace{\mathbf{A}^{(k_1)}, \dots, \mathbf{A}^{(k_1)}}_{l_1 \text{ times}}, \underbrace{\mathbf{A}^{(k_2)}, \dots, \mathbf{A}^{(k_2)}}_{l_2 \text{ times}}, \dots, \underbrace{\mathbf{A}^{(k_Q)}, \dots, \mathbf{A}^{(k_Q)}}_{l_Q \text{ times}} \right], \quad (3.55)$$

where $l_q > 0$, $l_1 + \dots + l_Q = P$ and $k_q = \sum_{s=1}^{q-1} l_s + 1$. The parameter vector can thus be written as $\boldsymbol{\eta} = \left[\bar{\boldsymbol{\theta}}^{(k_1)}, \dots, \bar{\boldsymbol{\theta}}^{(k_Q)}, \boldsymbol{\lambda} \right]^T$, and the Jacobian is given by $\mathbf{J}(\boldsymbol{\eta}) =$

$$\left[\mathbf{J}_{\bar{\boldsymbol{\theta}}^{(k_1)}}(\boldsymbol{\eta}) \ \dots \ \mathbf{J}_{\bar{\boldsymbol{\theta}}^{(k_Q)}}(\boldsymbol{\eta}) \ \mathbf{J}_\lambda(\boldsymbol{\eta}) \right], \text{ with } \mathbf{J}_{\bar{\boldsymbol{\theta}}^{(k_q)}}(\boldsymbol{\eta}) = \left[\mathbf{w}_1^{(k_q)} \ \dots \ \mathbf{w}_{\bar{U}_{k_q}}^{(k_q)} \right], \text{ where}$$

$$\mathbf{w}_u^{(k_q)} = \left[\left(\boldsymbol{\theta}^{(k_1)} \right)^{\boxtimes l_1} \right] \boxtimes \dots \boxtimes \left[\left(\boldsymbol{\theta}^{(k_{q-1})} \right)^{\boxtimes l_{q-1}} \right] \boxtimes \mathbf{z}_u^{(k_q)} \boxtimes \left[\left(\boldsymbol{\theta}^{(k_{q+1})} \right)^{\boxtimes l_{q+1}} \right] \boxtimes \dots \boxtimes \left[\left(\boldsymbol{\theta}^{(k_Q)} \right)^{\boxtimes l_Q} \right]$$

and

$$\mathbf{z}_u^{(k_q)} = \mathbf{e}_t \boxtimes \left[\left(\boldsymbol{\theta}^{(k_q)} \right)^{\boxtimes l_q - 1} \right] + \boldsymbol{\theta}^{(k_q)} \boxtimes \mathbf{e}_t \boxtimes \left[\left(\boldsymbol{\theta}^{(k_q)} \right)^{\boxtimes l_q - 2} \right] + \dots + \left[\left(\boldsymbol{\theta}^{(k_q)} \right)^{\boxtimes l_q - 1} \right] \boxtimes \mathbf{e}_t,$$

with $t = V_{k_q} + u$. The block $\mathbf{J}_\lambda(\boldsymbol{\eta})$ is as described by (3.51), but now with

$$\mathbf{f}(\boldsymbol{\theta}) = \left[\left(\boldsymbol{\theta}^{(k_Q)} \right)^{\boxtimes l_Q} \right] \boxtimes \dots \boxtimes \left[\left(\boldsymbol{\theta}^{(k_1)} \right)^{\boxtimes l_1} \right]. \quad (3.56)$$

Example 3.11. Consider a third-order tensor whose SCPD is such that $\mathbf{A}^{(2)} = \mathbf{A}^{(3)}$. Hence, $Q = 2$, $k_1 = 1$ and $k_2 = l_2 = 2$. The parameter vector and the Jacobian are written as

$$\boldsymbol{\eta} = \left[\left(\bar{\boldsymbol{\theta}}^{(1)} \right)^T \ \left(\bar{\boldsymbol{\theta}}^{(2)} \right)^T \ \boldsymbol{\lambda}^T \right]^T \quad \text{and} \quad \mathbf{J}(\boldsymbol{\eta}) = \left[\mathbf{J}_{\bar{\boldsymbol{\theta}}^{(1)}}(\boldsymbol{\eta}) \ \mathbf{J}_{\bar{\boldsymbol{\theta}}^{(2)}}(\boldsymbol{\eta}) \ \mathbf{J}_\lambda(\boldsymbol{\eta}) \right], \quad (3.57)$$

³In this case, though, only $P - 1$ factors should have the scaling of their columns fixed.

with

$$\mathbf{J}_{\bar{\theta}^{(1)}}(\boldsymbol{\eta}) = \Phi(\boldsymbol{\lambda}) \left(\boldsymbol{\theta}^{(2)} \boxtimes \boldsymbol{\theta}^{(2)} \boxtimes \mathbf{B}_1^T \right), \quad \mathbf{J}_{\bar{\theta}^{(2)}}(\boldsymbol{\eta}) = \Phi(\boldsymbol{\lambda}) \begin{bmatrix} \mathbf{w}_1^{(2)} & \dots & \mathbf{w}_{\bar{U}_2}^{(2)} \end{bmatrix}, \quad (3.58)$$

where $\mathbf{w}_u^{(2)} = \left(\mathbf{e}_{V_2+u} \boxtimes \boldsymbol{\theta}^{(2)} \boxtimes \boldsymbol{\theta}^{(1)} \right) + \left(\boldsymbol{\theta}^{(2)} \boxtimes \mathbf{e}_{V_2+u} \boxtimes \boldsymbol{\theta}^{(1)} \right)$ and

$$\mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}) = \begin{bmatrix} \Phi_1 \mathbf{f}(\boldsymbol{\theta}) & \dots & \Phi_R \mathbf{f}(\boldsymbol{\theta}) \end{bmatrix}, \quad \text{with } \mathbf{f}(\boldsymbol{\theta}) = \left(\boldsymbol{\theta}^{(2)} \boxtimes \boldsymbol{\theta}^{(2)} \boxtimes \boldsymbol{\theta}^{(1)} \right). \quad (3.59)$$

■

This somewhat intricate notation can be greatly simplified if the SCPD is completely symmetric, meaning $\mathbf{A}^{(p)} = \mathbf{A}$, $\boldsymbol{\theta}^{(p)} = \boldsymbol{\theta}$, $\bar{U}_p = \bar{U}$, $V_p = V$ and $U_p = U$ for all $p \in \langle P \rangle$. Indeed, under such an assumption the Jacobian $\mathbf{J}(\boldsymbol{\eta}) = [\mathbf{J}_{\bar{\theta}}(\boldsymbol{\eta}) \quad \mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta})]$ has blocks given by

$$\mathbf{J}_{\bar{\theta}}(\boldsymbol{\eta}) = \Phi(\boldsymbol{\lambda}) \begin{bmatrix} \mathbf{w}_1 & \dots & \mathbf{w}_{\bar{U}} \end{bmatrix}, \quad (3.60)$$

$$\mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}) = \begin{bmatrix} \Phi_1(\boldsymbol{\theta} \boxtimes \boldsymbol{\theta} \boxtimes \boldsymbol{\theta}) & \dots & \Phi_R(\boldsymbol{\theta} \boxtimes \boldsymbol{\theta} \boxtimes \boldsymbol{\theta}) \end{bmatrix}, \quad (3.61)$$

now with $\mathbf{w}_u = \mathbf{e}_{V+u} \boxtimes \left(\boldsymbol{\theta}^{\boxtimes(P-1)} \right) + \boldsymbol{\theta} \boxtimes \mathbf{e}_{V+u} \boxtimes \left(\boldsymbol{\theta}^{\boxtimes(P-2)} \right) + \dots + \left(\boldsymbol{\theta}^{\boxtimes(P-1)} \right) \boxtimes \mathbf{e}_{V+u}$.

In practical applications, when \mathcal{X} represents a symmetric quantity (such as, *e.g.*, a symmetric Volterra kernel), \mathcal{Y} may also be symmetric, because elements which should be identical are not repeatedly estimated or observed. As this implies that \mathcal{N} is also symmetric, the preceding results are no longer valid, because we have assumed its elements are *i.i.d.* Nonetheless, to accommodate this additional constraint it suffices to introduce a selection matrix $\Psi \in \mathbb{R}^{L \times N^P}$ containing as rows every Kronecker product of canonical basis vectors $\mathbf{e}_{n_P}^T \boxtimes \dots \boxtimes \mathbf{e}_{n_1}^T$ such that $(n_1, \dots, n_P) \in D = \{(n_1, \dots, n_P) : 1 \leq n_1 \leq \dots \leq n_P \leq N\}$, in such a way that the product $\Psi \text{vec}(\mathcal{Y})$ no longer contains redundant components due to symmetry. Note that $L = |D| = \binom{N^P + P - 1}{P}$. Then, we redefine the vector model as

$$\mathbf{y} \triangleq \Psi \text{vec}(\mathcal{Y}) = \Psi \Phi(\boldsymbol{\lambda}) \mathbf{f}(\boldsymbol{\theta}) + \mathbf{n}, \quad (3.62)$$

where now $\mathbf{n} \triangleq \Psi \text{vec}(\mathcal{N})$. As this \mathbf{n} is again zero-mean Gaussian *i.i.d.*, (3.48) remains valid. The Jacobian, however, must now be multiplied from left by Ψ . A similar adjustment applies to partially symmetric SCPDs. Also, it should be noted that the particular ordering of the rows of Ψ is irrelevant in the above reasoning.

3.4.5 Bayesian extension

An MSE-based comparison with a single fixed $\boldsymbol{\eta}$ is insufficient for a sound statistical evaluation of estimators, for there is no reason to choose a particular value of $\boldsymbol{\eta}$ over another. Hence, we consider now the case where $\boldsymbol{\eta}$ is itself a random variable with some specified

prior p.d.f., which incorporates available information about the values it typically assumes in a given application. This is the basic principle underlying the Bayesian estimation framework [13]. An appropriate performance criterion in this case is the Bayesian mean squared error (BMSE), defined as [10]

$$\text{BMSE} = \mathbb{E}_{\boldsymbol{\eta}} \{ \text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) \}, \quad (3.63)$$

now with

$$\text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) = \mathbb{E}_{\mathbf{y}|\boldsymbol{\eta}} \left\{ \|\hat{\boldsymbol{\eta}}(\mathbf{y}) - \boldsymbol{\eta}\|_2^2 \right\}, \quad (3.64)$$

where $\mathbb{E}_v\{\cdot\}$ denotes expectation with respect to the p.d.f. of v and, similarly, $\mathbb{E}_{v|u}\{\cdot\}$ denotes expectation with respect to the conditional distribution $p(v|u)$. Expression (3.64) therefore defines the conditional MSE given $\boldsymbol{\eta}$.

As discussed in [65], two possibilities exist for the extension of the CRB to the Bayesian setting: Van Trees' Bayesian Cramér-Rao bound (BCRB) [207] and the expected Cramér-Rao bound (ECRB) based on the Bayesian-deterministic connection [207, 10]. The latter is adopted here for two reasons, as discussed in [65]:

- (i) The regularity conditions required by the BCRB are stricter than those required by the ECRB. For instance, a uniform prior is ruled out by the regularity conditions of BCRB, whereas it is admissible for the ECRB.
- (ii) For any statistical priors $p(\boldsymbol{\eta})$, the ECRB is tighter than the BCRB when both are applied to our estimation problem. A proof of this fact is given in [65, Result 5].

Analogously to expression (3.47), in the Bayesian setting we consider the inequality

$$\mathbb{E}_{\boldsymbol{\eta}} \{ \text{MSE}(\hat{\boldsymbol{\eta}}; \boldsymbol{\eta}) \} \geq \sum_{p=1}^P \sum_{u=1}^{\bar{U}_p} \text{ECRB}(\bar{\boldsymbol{\theta}}_u^{(p)}) + \sum_{r=1}^R \text{ECRB}(\lambda_r), \quad (3.65)$$

where the terms constituting the lower bound are as follows:

- for arbitrary $\boldsymbol{\lambda}$, the ECRB for parameters $\bar{\boldsymbol{\theta}}_u^{(p)}$ and λ_r are

$$\text{ECRB}(\bar{\boldsymbol{\theta}}_u^{(p)}) = \mathbb{E}_{\bar{\boldsymbol{\theta}}}\mathbb{E}_{\boldsymbol{\lambda}} \left\{ \text{CRB}(\bar{\boldsymbol{\theta}}_u^{(p)}) \right\}, \quad (3.66)$$

$$\text{ECRB}(\lambda_r) = \mathbb{E}_{\bar{\boldsymbol{\theta}}}\mathbb{E}_{\boldsymbol{\lambda}} \left\{ \text{CRB}(\lambda_r) \right\}, \quad (3.67)$$

where $\text{CRB}(\bar{\boldsymbol{\theta}}_u^{(p)})$ and $\text{CRB}(\lambda_r)$ are given by (3.52) and (3.53), respectively;

- when $\lambda_r = \lambda$ for all $r \in \langle R \rangle$, the ECRB for parameters $\bar{\boldsymbol{\theta}}_u^{(p)}$ is given by

$$\text{ECRB}(\bar{\boldsymbol{\theta}}_u^{(p)}) = \mathbb{E}_{\bar{\boldsymbol{\theta}}} \left\{ \text{CRB}(\bar{\boldsymbol{\theta}}_u^{(p)}) \right\} \quad (3.68)$$

where $\text{CRB}(\bar{\boldsymbol{\theta}}_u^{(p)})$ is given by (3.52).

Algorithms for SCPD computation

This chapter presents our proposed solutions for the SCPD computation problem. The first one consists of a systematic way of constraining the ALS algorithm to take into account the structure of each factor, with either exact or approximate iterates, the latter being less costly. The second solution is algebraic and applies when all factors are circulant. It is based on computing the multidimensional Fourier transform of the tensor of interest, which leads to a system of homogeneous monomial equations whose resolution provides the factor estimates.

Contents

4.1	Constrained alternating least-squares (CALS) algorithm	71
4.2	Algebraic solution for a SCPD having circulant factors	76
4.3	Final remarks	87

4.1 Constrained alternating least-squares (CALS) algorithm

As discussed in Section 3.3.1, adapted implementations of the ALS method have been proposed in the literature for dealing with constrained factors, but always in an *ad-hoc* manner. In this section, by relying upon the characterization of an SCPD given in Section 3.1, we develop a general constrained alternating least squares (CALS) algorithm for taking into account the linear structure of the desired factors. Two variants of this algorithm are formulated: one with exact iterates [65] and another with approximate iterates [64].

4.1.1 Exact iterates

We assume that a data tensor $\mathcal{Y} \in \mathcal{T} = \bigotimes_{p=1}^P \mathbb{C}^{N_p}$ has been given, and that one wishes to compute its SCPD by minimizing the least-squares criterion (3.13). The structure of the decomposition factors $\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}$, $p \in \langle P \rangle$, is assumed to be known and characterized by their corresponding matrix bases $\mathcal{B}_p = \{\mathbf{E}_1^{(p)}, \dots, \mathbf{E}_{U_p}^{(p)}\}$, as described in Section 3.1. The coordinate representation of $\mathbf{A}^{(p)}$ under \mathcal{B}_p is denoted by $\boldsymbol{\theta}^{(p)} \in \mathbb{C}^{U_p}$.

Our starting point is the basic subproblem (3.14) which must be solved for estimating each factor. Let us denote by $\hat{\mathbf{A}}_k^{(q)}$ the estimate of $\mathbf{A}^{(q)}$ computed at iteration k and define

$$\mathbf{W}_k^{(p)} \triangleq \left(\hat{\mathbf{A}}_{k-1}^{(P)} \diamond \cdots \diamond \hat{\mathbf{A}}_{k-1}^{(p+1)} \diamond \hat{\mathbf{A}}_k^{(p-1)} \diamond \cdots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^T, \quad (4.1)$$

so that this subproblem can be written as

$$\min_{\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}} \left\| \mathbf{Y}_{\langle p \rangle} - \mathbf{A}^{(p)} \mathbf{W}_k^{(p)} \right\|_F^2. \quad (4.2)$$

For the sake of generality, we may include the scaling factors λ_r of (2.86), thus obtaining

$$\min_{\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}} \left\| \mathbf{Y}_{\langle p \rangle} - \mathbf{A}^{(p)} \hat{\mathbf{\Lambda}}_{k-1} \mathbf{W}_k^{(p)} \right\|_F^2, \quad (4.3)$$

where $\hat{\mathbf{\Lambda}}_{k-1} = \text{Diag}(\hat{\boldsymbol{\lambda}}_{k-1})$ is our current estimate of $\mathbf{\Lambda} = \text{Diag}(\boldsymbol{\lambda})$, with $\boldsymbol{\lambda} = [\lambda_1 \ \dots \ \lambda_R]^T$.

By employing property (2.49) of the Kronecker product, it is easy to see that (4.3) is equivalent to the problem

$$\min_{\mathbf{A}^{(p)} \in \mathbb{C}^{N_p \times R}} \left\| \text{vec}(\mathbf{Y}_{\langle p \rangle}) - \left[\left(\hat{\mathbf{\Lambda}}_{k-1} \mathbf{W}_k^{(p)} \right)^T \boxtimes \mathbf{I}_{N_p} \right] \text{vec}(\mathbf{A}^{(p)}) \right\|_2^2. \quad (4.4)$$

Moreover, as $\mathbf{A}^{(p)}$ is structured in the sense of Definition 3.1, it obeys the identity $\text{vec}(\mathbf{A}^{(p)}) = \mathbf{E}^{(p)} \boldsymbol{\theta}^{(p)}$, where $\mathbf{E}^{(p)} \in \mathbb{C}^{RN_p \times U_p}$ is defined by (3.3). So, we can rewrite (4.4) as

$$\min_{\boldsymbol{\theta}^{(p)} \in \mathbb{C}^{U_p}} \left\| \text{vec}(\mathbf{Y}_{\langle p \rangle}) - \boldsymbol{\Phi}_k^{(p)} \boldsymbol{\theta}^{(p)} \right\|_2^2, \quad (4.5)$$

where $\boldsymbol{\Phi}_k^{(p)} \triangleq \left[\left(\hat{\mathbf{\Lambda}}_{k-1} \mathbf{W}_k^{(p)} \right)^T \boxtimes \mathbf{I}_{N_p} \right] \mathbf{E}^{(p)} \in \mathbb{C}^{\bar{N} \times U_p}$. Hence, assuming $\boldsymbol{\Phi}_k^{(p)}$ has full column rank, the least-squares estimate of $\boldsymbol{\theta}^{(p)}$ is given by [65]

$$\hat{\boldsymbol{\theta}}_k^{(p)} = \left(\boldsymbol{\Phi}_k^{(p)} \right)^+ \text{vec}(\mathbf{Y}_{\langle p \rangle}), \quad (4.6)$$

where $(\cdot)^+$ denotes the left inverse of its argument. Once all estimates $\hat{\boldsymbol{\theta}}_k^{(p)}$ are obtained and their respective factors $\hat{\mathbf{A}}_k^{(p)}$ are constructed, $\boldsymbol{\lambda}$ can be estimated by exploiting (2.88):

$$\hat{\boldsymbol{\lambda}}_k = \left(\hat{\mathbf{A}}_k^{(P)} \diamond \cdots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^+ \text{vec}(\mathbf{y}) = \left(\hat{\mathbf{A}}_k^{(P)} \diamond \cdots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^+ \text{vec}(\mathbf{Y}_{\langle 1 \rangle}). \quad (4.7)$$

These update equations lead to the scheme shown in Algorithm 4.1, where we also indicate the computational complexity of each step. Some comments are in order:

- At line 5, we take into account the fact that $\mathbf{X} \boxtimes \mathbf{I}$ can be calculated (for any \mathbf{X}) without performing any actual multiplications, by just arranging the elements of \mathbf{X} into the resulting matrix of known structure.
- At line 6, it takes only $\mathcal{O}(R)$ operations (rather than $\mathcal{O}(RN_p)$) to calculate each component of the result, because $\bar{\boldsymbol{\Phi}}_k^{(p)}$ has at most R nonzero entries per row.

Algorithm 4.1 CALS algorithm with exact iterates [65].

Inputs: $\mathbf{y} \in \mathcal{T}$, basis matrices $\mathbf{E}^{(p)}$ and initial parameter vectors $\hat{\boldsymbol{\lambda}}_0, \hat{\boldsymbol{\theta}}_0^{(p)}, p \in \langle P \rangle$
Outputs: Estimated parameter vectors $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\theta}}^{(p)}, p \in \langle P \rangle$

```

1:  $k \leftarrow 1$ 
2: repeat
3:   for  $p = 1, \dots, P$  do
4:      $\mathbf{W}_k^{(p)} \leftarrow \left( \hat{\mathbf{A}}_{k-1}^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}_{k-1}^{(p+1)} \diamond \hat{\mathbf{A}}_k^{(p-1)} \diamond \dots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^T$  //  $\mathcal{O}(R\bar{N}_p)$ 
5:      $\bar{\boldsymbol{\Phi}}_k^{(p)} \leftarrow \left( \hat{\mathbf{A}}_{k-1} \mathbf{W}_k^{(p)} \right)^T \boxtimes \mathbf{I}_{N_p}$  //  $\mathcal{O}(R\bar{N}_p)$ 
6:      $\boldsymbol{\Phi}_k^{(p)} \leftarrow \bar{\boldsymbol{\Phi}}_k^{(p)} \mathbf{E}^{(p)}$  //  $\mathcal{O}(R\bar{N}_p U_p)$ 
7:      $\hat{\boldsymbol{\theta}}_k^{(p)} \leftarrow \left( \boldsymbol{\Phi}_k^{(p)} \right)^+ \text{vec}(\mathbf{Y}_{\langle p \rangle})$  //  $\mathcal{O}(\bar{N}U_p^2)$ 
8:      $\hat{\mathbf{A}}_k^{(p)} \leftarrow \text{unvec} \left( \mathbf{E}^{(p)} \hat{\boldsymbol{\theta}}_k^{(p)} \right)$  //  $\mathcal{O}(R N_p U_p)$ 
9:     normalize  $\hat{\mathbf{A}}_k^{(p)}$  //  $\mathcal{O}(R N_p)$ 
10:     $\hat{\boldsymbol{\lambda}}_k \leftarrow \left( \hat{\mathbf{A}}_k^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^+ \text{vec}(\mathbf{y})$  //  $\mathcal{O}(R^2 \bar{N})$ 
11:     $\hat{\mathbf{A}}_k \leftarrow \text{Diag}(\hat{\boldsymbol{\lambda}}_k)$  //  $\mathcal{O}(1)$ 
12:     $k \leftarrow k + 1$ 
13: until the stopping criterion is fulfilled
14: return  $\hat{\boldsymbol{\lambda}}$  and  $\hat{\boldsymbol{\theta}}^{(p)} = \hat{\boldsymbol{\theta}}_{k-1}^{(p)}, p \in \langle P \rangle$ 

```

- At lines 7 and 10, in practice it is not advisable to explicitly calculate a left inverse when solving a least-squares problem, for numerical accuracy reasons. Instead, standard procedures such as the method of normal equations or that which employs a QR factorization should be preferred (see [90, Section 5.3]).

An important feature of the above described CALS algorithm is that it preserves the fundamental property of ALS of always yielding non-increasing cost function values. This is owed to the fact that it essentially solves a sequence of least-squares subproblems having convex cost functions over matrix subspaces.

As a final remark, note that for certain SCPDs the estimation of $\boldsymbol{\lambda}$ must be omitted in order to meet the model constraints. For instance, if $\mathbf{y} = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket$ and all factors $\mathbf{A}^{(p)}$ are Toeplitz (or Hankel), then a model of the form

$$\llbracket \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \text{Diag}(\boldsymbol{\lambda}) \rrbracket \quad (4.8)$$

is actually more general than necessary, as $\mathbf{A}^{(P)} \text{Diag}(\boldsymbol{\lambda})$ is not necessarily Toeplitz (or Hankel) for an arbitrary $\boldsymbol{\lambda}$. A simple solution in this case is to estimate at line 10 of Algorithm 4.1 a single scaling factor $\lambda \in \mathbb{R}$ which applies to all rank-one terms of the decomposition, instead of a vector-valued $\boldsymbol{\lambda} \in \mathbb{R}^R$. Another option would be to eliminate λ altogether and then normalize all but one factor.

4.1.2 Approximate iterates

Instead of deriving an exact least-squares solution to each subproblem, one can alternatively consider *approximate* iterates obtained by solving them in a two-step fashion. First, we compute an *unconstrained* estimate of $\mathbf{A}^{(p)}$ via the standard update equation

$$\tilde{\mathbf{A}}_k^{(p)} = \mathbf{Y}_{\langle p \rangle} \left(\hat{\mathbf{\Lambda}}_{k-1} \mathbf{W}_k^{(p)} \right)^+, \quad (4.9)$$

from which the parameter vector is then estimated in the least-squares sense via [64]

$$\hat{\boldsymbol{\theta}}_k^{(p)} = \left(\mathbf{E}^{(p)} \right)^+ \text{vec} \left(\tilde{\mathbf{A}}_k^{(p)} \right). \quad (4.10)$$

Since we can assume without loss of generality that the basis matrices $\mathbf{E}_u^{(p)}$, $u \in \langle U_p \rangle$, are mutually orthogonal (*i.e.*, $\langle \mathbf{E}_u^{(p)}, \mathbf{E}_v^{(p)} \rangle = 0$ if $u \neq v$), the above left inverse is given simply by $\left(\mathbf{E}^{(p)} \right)^+ = \text{Diag} \left(\left\| \mathbf{E}_1^{(p)} \right\|_F^{-2}, \dots, \left\| \mathbf{E}_{U_p}^{(p)} \right\|_F^{-2} \right) \left(\mathbf{E}^{(p)} \right)^H$. This applies, in particular, to all bases described by Table 3.1.

Algorithm 4.2 provides a CALS scheme with approximate iterates, describing also the computational complexity of each step. In particular, at line 5 we account for the fact that, though each row of $\tilde{\mathbf{A}}_k^{(p)}$ is the solution of a least-squares problem of size $R \times \bar{N}_p$, only $\mathcal{O}(R^2(\bar{N}_p + N_p))$ total operations are necessary rather than $\mathcal{O}(R\bar{N}_pN_p)$. For instance, if a QR factorization of $\hat{\mathbf{\Lambda}}_{k-1} \mathbf{W}_k^{(p)}$ is employed, it is computed only once and then used to solve for the N_p rows. By comparing the complexity expressions of Algorithm 4.2 with those given in Algorithm 4.1, it can be seen that the approximate iterates are less costly to compute. Indeed, the complexity of an iteration in the exact version is dominated by lines 6 and 7 of Algorithm 4.1, which cost $\mathcal{O}(R\bar{N}U_p)$ and $\mathcal{O}(\bar{N}U_p^2)$ operations, respectively, and are repeated for all $p \in \langle P \rangle$. By contrast, in Algorithm 4.2 the costliest step is that of line 9, whose complexity is $\mathcal{O}(R^2\bar{N})$, and which is also present in Algorithm 4.1.

On the other hand, as the approximate iterates are suboptimal, the algorithm may suffer from convergence problems. A combined strategy can be sought by starting off with approximate iterates and switching to the exact ones after some criterion is met. Our simulations of Chapter 5 show that, under certain circumstances, such a mixed approach can reduce the total computing effort and can also improve the chances of attaining a global minimum.

4.1.3 Symmetric CALS

It is straightforward to adapt Algorithms 4.1 and 4.2 to handle a (partially or totally) symmetric SCPD, by estimating each distinct factor only once per iteration. We note that this is a standard technique employed in the unconstrained case [52, 79, 84]. This adapted version is called symmetric constrained alternating least squares (SCALS). In order to provide its formulation, it will be useful to resort to the notation introduced in Section 3.4.4.2

Algorithm 4.2 CALS algorithm with approximate iterates [64].

Inputs: $\mathbf{Y} \in \mathcal{T}$, basis matrices $\mathbf{E}^{(p)}$ and initial parameter vectors $\hat{\boldsymbol{\lambda}}_0, \hat{\boldsymbol{\theta}}_0^{(p)}, p \in \langle P \rangle$
Outputs: Estimated parameter vectors $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\theta}}^{(p)}, p \in \langle P \rangle$

```

1:  $k \leftarrow 1$ 
2: repeat
3:   for  $p = 1, \dots, P$  do
4:      $\mathbf{W}_k^{(p)} \leftarrow \left( \hat{\mathbf{A}}_{k-1}^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}_{k-1}^{(p+1)} \diamond \hat{\mathbf{A}}_k^{(p-1)} \diamond \dots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^T$  //  $\mathcal{O}(R\bar{N}_p)$ 
5:      $\tilde{\mathbf{A}}_k^{(p)} \leftarrow \mathbf{Y}_{\langle p \rangle} \left( \hat{\mathbf{A}}_{k-1} \mathbf{W}_k^{(p)} \right)^+$  //  $\mathcal{O}(R^2(\bar{N}_p + N_p))$ 
6:      $\hat{\boldsymbol{\theta}}_k^{(p)} \leftarrow \frac{1}{R} \left( \mathbf{E}^{(p)} \right)^H \text{vec} \left( \tilde{\mathbf{A}}_k^{(p)} \right)$  //  $\mathcal{O}(RN_p U_p)$ 
7:      $\hat{\mathbf{A}}_k^{(p)} \leftarrow \text{unvec} \left( \mathbf{E}^{(p)} \hat{\boldsymbol{\theta}}_k^{(p)} \right)$  //  $\mathcal{O}(RN_p U_p)$ 
8:     normalize  $\hat{\mathbf{A}}_k^{(p)}$  //  $\mathcal{O}(RN_p)$ 
9:      $\hat{\boldsymbol{\lambda}}_k \leftarrow \left( \hat{\mathbf{A}}_k^{(P)} \diamond \dots \diamond \hat{\mathbf{A}}_k^{(1)} \right)^+ \text{vec}(\mathbf{Y})$  //  $\mathcal{O}(R^2 \bar{N})$ 
10:     $\hat{\boldsymbol{\Lambda}}_k \leftarrow \text{Diag}(\hat{\boldsymbol{\lambda}}_k)$  //  $\mathcal{O}(1)$ 
11:     $k \leftarrow k + 1$ 
12: until the stopping criterion is fulfilled
13: return  $\hat{\boldsymbol{\lambda}}$  and  $\hat{\boldsymbol{\theta}}^{(p)} = \hat{\boldsymbol{\theta}}_{k-1}^{(p)}, p \in \langle P \rangle$ 

```

for describing a P th-order SCPD having Q distinct factors $\mathbf{A}^{(k_1)}, \dots, \mathbf{A}^{(k_Q)}, Q < P$. Recall that the k_q th factor appears l_q times in the decomposition, with $k_q = \sum_{s=1}^{q-1} l_s$. By defining $\hat{\mathbf{B}}_k^{(k_q)} \triangleq \left(\hat{\mathbf{A}}_k^{(k_q)} \right)^{\diamond l_q}$ and

$$\mathbf{W}_k^{(k_q)} \triangleq \left[\hat{\mathbf{B}}_{k-1}^{(k_Q)} \diamond \dots \diamond \hat{\mathbf{B}}_{k-1}^{(k_{q+1})} \diamond \left(\hat{\mathbf{A}}_{k-1}^{(k_q)} \right)^{\diamond l_q - 1} \diamond \hat{\mathbf{B}}_k^{(k_{q-1})} \diamond \dots \diamond \hat{\mathbf{B}}_k^{(k_1)} \right]^T, \quad (4.11)$$

we can write each least-squares subproblem as

$$\min_{\mathbf{A}^{(k_q)} \in \mathbb{C}^{N_{k_q} \times R}} \left\| \mathbf{Y}_{\langle k_q \rangle} - \mathbf{A}^{(k_q)} \hat{\boldsymbol{\Lambda}}_{k-1} \mathbf{W}_k^{(k_q)} \right\|_2^2, \quad (4.12)$$

for all $q \in \langle Q \rangle$. Hence, at iteration k , one estimates all $\hat{\boldsymbol{\theta}}_k^{(k_q)}$ and $\hat{\mathbf{A}}_k^{(k_q)}$ by employing either the exact update equations of Algorithm 4.1 or the approximate ones of Algorithm 4.2. For convenience, we explicitly give this adapted scheme in Algorithm 4.3.

A significant drawback of the SCALS is the fact that the cost function is no longer guaranteed to decrease or stay the same along the iterations, unlike happens with ALS and CALS. This stems from the fact that the solution $\hat{\boldsymbol{\theta}}_k^{(k_q)}$ is suboptimal when $l_q > 1$, since the solved subproblem does not account for the multiple (l_q) occurrences of $\mathbf{A}^{(k_q)}$ in the SCPD of \mathbf{Y} . Consequently, SCALS is more sensitive to its initialization and more prone to suffering from convergence problems than CALS.

We must also note that estimating a scaling factor λ is of vital importance when dealing with symmetric SCPDs of the form $[[\mathbf{A}, \dots, \mathbf{A}]]$, because only then normalization can be applied with no loss of generality. Without normalization, estimation errors undergo a feedback effect which quickly destabilizes the iterates, causing the algorithm to diverge. The

Algorithm 4.3 CALS algorithm with partial or total symmetry, SCALS.**Inputs:** $\mathcal{Y} \in \mathcal{T}$, basis matrices $\mathbf{E}^{(k_q)}$ and initial parameter vectors $\hat{\boldsymbol{\lambda}}_0, \hat{\boldsymbol{\theta}}_0^{(k_q)}$ **Outputs:** Estimated parameter vectors $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\theta}}^{(k_q)}$

```

1:  $k \leftarrow 1$ 
2: repeat
3:   for  $q = 1, \dots, Q$  do
4:      $\mathbf{W}_k^{(k_q)} \leftarrow \left[ \hat{\mathbf{B}}_{k-1}^{(k_Q)} \diamond \dots \diamond \hat{\mathbf{B}}_{k-1}^{(k_{q+1})} \diamond \left( \hat{\mathbf{A}}_{k-1}^{(k_q)} \right)^{\diamond l_{q-1}} \diamond \hat{\mathbf{B}}_{k-1}^{(k_{q-1})} \diamond \dots \diamond \hat{\mathbf{B}}_{k-1}^{(k_1)} \right]^T$  //  $\mathcal{O}(RN\bar{N}_{k_q})$ 
5:     Compute exact or approximate estimates  $\hat{\boldsymbol{\theta}}_k^{(k_q)}$  and  $\hat{\mathbf{A}}_k^{(k_q)}$  // See Algs. 4.1 and 4.2
6:     normalize  $\hat{\mathbf{A}}_k^{(k_q)}$  //  $\mathcal{O}(RN_{k_q})$ 
7:      $\hat{\mathbf{B}}_k^{(k_q)} \leftarrow \left( \hat{\mathbf{A}}_k^{(k_q)} \right)^{\diamond l_q}$  //  $\mathcal{O}(RN_{k_q}^{l_q})$ 
8:      $\hat{\boldsymbol{\lambda}}_k \leftarrow \left( \hat{\mathbf{B}}_k^{(k_Q)} \diamond \dots \diamond \hat{\mathbf{B}}_k^{(k_1)} \right)^+ \text{vec}(\mathcal{Y})$  //  $\mathcal{O}(R^2\bar{N})$ 
9:      $\hat{\boldsymbol{\Lambda}}_k \leftarrow \text{Diag}(\hat{\boldsymbol{\lambda}}_k)$  //  $\mathcal{O}(1)$ 
10:     $k \leftarrow k + 1$ 
11: until the stopping criterion is fulfilled
12: return  $\hat{\boldsymbol{\lambda}}$  and  $\hat{\boldsymbol{\theta}}^{(k_q)} = \hat{\boldsymbol{\theta}}_{k-1}^{(k_q)}, q \in \langle Q \rangle$ 

```

same is not true for non-symmetric (or partially symmetric) SCPDs, whose estimation via CALS (or SCALS) is generally not compromised by the absence of normalization.

The above observations will be illustrated by means of simulation results in Chapter 5.

4.2 Algebraic solution for a SCPD having circulant factors

In this section, we address the estimation SCPDs having only circulant factors. Let us first formally define this structure.

Definition 4.1 (Circulant matrix). A matrix $\mathbf{C} \in \mathbb{C}^{N \times R}$ is said to be *circulant* with *generating vector* $\mathbf{c} = [c_0 \ c_1 \ \dots \ c_{N-1}]^T \in \mathbb{C}^N$ if $\mathbf{C} = \text{circ}_R(\mathbf{c}) \triangleq \mathbf{C} = \begin{bmatrix} \mathbf{c} & \mathbf{\Pi}_N \mathbf{c} & \dots & \mathbf{\Pi}_N^{R-1} \mathbf{c} \end{bmatrix}$, where $\mathbf{\Pi}_N \in \mathbb{C}^{N \times N}$ denotes the same permutation matrix of Table 3.1, *i.e.*,

$$\mathbf{\Pi}_N = \begin{bmatrix} \mathbf{0}_{1 \times N-1} & 1 \\ \mathbf{I}_{N-1} & \mathbf{0}_{N-1 \times 1} \end{bmatrix}. \quad (4.13)$$

Here, we focus only on hypercubic tensors from $\mathcal{T} \triangleq \bigotimes_{p=1}^P \mathbb{C}^N$. A circulant-constrained canonical polyadic decomposition (CCPD) of $\mathcal{X} \in \mathcal{T}$ is a SCPD of the form

$$\mathcal{X} = \llbracket \mathbf{C}^{(1)}, \dots, \mathbf{C}^{(P)} \rrbracket \in \mathcal{T}, \quad (4.14)$$

where the factors $\mathbf{C}^{(p)} \in \mathbb{C}^{N \times R}$ are circulant. Our goal is to compute these factors from \mathcal{X} , up to trivial (scaling and permutation) ambiguities. In order to accomplish it, we now present the approach we developed in [64], which is based on the resolution of a system of monomial equations obtained through the multidimensional Fourier transform of \mathcal{X} .

A distinguishing feature of the CCPD (4.14) is that its rank is upper bounded by N , as opposed to the general CPD, whose rank can exceed all dimensions of the tensor (see Section 2.5.2.2). Indeed, if the factors $\mathbf{C}^{(p)}$ in (4.14) were such that $R > N$, then from Definition 4.1 (and Proposition 4.4 shown below) we would have $\mathbf{c}_{N+r}^{(p)} = \mathbf{c}_r^{(p)}$ for all p and $1 \leq r \leq R - N$, where $\mathbf{c}_r^{(p)}$ denotes the r th column of $\mathbf{C}^{(p)}$. This would clearly imply the existence of linearly dependent terms in the decomposition, and therefore R would not be minimal. Because of this fact, in what follows we assume that $R \leq N$, without loss of generality.

4.2.1 Basic properties of circulant matrices

Let us start by recalling properties of circulant matrices. In particular, square circulant matrices possess a well known special property, as expressed by the following result.

Lemma 4.2. Every square circulant matrix $\mathbf{C} \in \mathbb{C}^{N \times N}$ possesses a complete set of orthogonal eigenvectors, which are the columns of the DFT matrix

$$\mathbf{F} \triangleq \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \dots & \omega_N^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \dots & \omega_N^{(N-1)(N-1)} \end{bmatrix} \in \mathbb{C}^{N \times N}, \quad (4.15)$$

where¹ $\omega_N \triangleq \exp(j2\pi/N)$. It thus follows that any circulant matrix $\mathbf{C} \in \mathbb{C}^{N \times N}$ can be diagonalized by \mathbf{F} , or, equivalently,

$$\mathbf{C} = \mathbf{F}\mathbf{\Lambda}\mathbf{F}^H, \quad (4.16)$$

where $\mathbf{\Lambda} = \text{Diag}(\boldsymbol{\lambda}) \in \mathbb{C}^{N \times N}$ contains the eigenvalues of \mathbf{C} .

Proof. See [93, Section 3.1] (in which a different ordering of the eigenvectors is adopted). \square

A direct consequence of Lemma 4.2 is that, if we rewrite (4.16) as $\mathbf{F}^H\mathbf{C} = \mathbf{\Lambda}\mathbf{F}^H$, then from the first column of the latter identity we have $\mathbf{F}^H\mathbf{c} = \frac{1}{\sqrt{N}}\boldsymbol{\lambda}$. In other words, \mathbf{c} and $\frac{1}{\sqrt{N}}\boldsymbol{\lambda}$ form a DFT pair.² But, since this result applies only to square matrices, we now introduce a definition which will facilitate the treatment of tall circulant factors.

Definition 4.3 (Circulant completion). Let $\mathbf{C} = \text{circ}_R(\mathbf{c}) \in \mathbb{C}^{N \times R}$ with $R \leq N$. We define the circulant completion of \mathbf{C} as the square matrix $\check{\mathbf{C}} \in \mathbb{C}^{N \times N}$ given by $\check{\mathbf{C}} = \text{circ}_N(\mathbf{c})$. Note that, if $N = R$, then $\check{\mathbf{C}} = \mathbf{C}$.

¹According to our definition, $\mathbf{F}^H\mathbf{x}$ yields the DFT of \mathbf{x} . Some authors adopt instead the convention $\omega_N \triangleq -\exp(j2\pi/N)$, in which case the DFT of \mathbf{x} is given by $\mathbf{F}\mathbf{x}$.

²In [93, Section 3.1], it is rather the first row of \mathbf{C} and $(1/\sqrt{N})\boldsymbol{\lambda}$ that form a DFT pair, because of the different ordering they assume for the eigenvalues.

From the above definition, it follows that any $N \times R$ circulant matrix with $R \leq N$ is such that its circulant completion can be decomposed as in (4.16). By noting that $\mathbf{\Pi}_N$ itself is circulant, we can also state the following useful property.

Proposition 4.4. Let $\mathbf{\Pi}_N$ be as defined by (4.13). Then, $\mathbf{\Pi}_N^k = \mathbf{\Pi}_N^{(k)_N}$ for all $k \in \mathbb{Z}$, where $(\cdot)_N$ denotes the modulo N operator.³ Consequently, there exist only N distinct matrices of the form $\mathbf{\Pi}_N^k$, namely, $\mathbf{\Pi}_N^0, \mathbf{\Pi}_N^1, \dots, \mathbf{\Pi}_N^{N-1}$. Moreover, $\mathbf{\Pi}_N^k$ admits the eigendecomposition

$$\mathbf{\Pi}_N^k = \mathbf{F} \text{Diag} \left(1, \omega_N^{-k}, \dots, \omega_N^{-k(N-1)} \right) \mathbf{F}^H. \quad (4.17)$$

Proof. It is easy to check that, for any $k \in \mathbb{Z}$,

$$\mathbf{\Pi}_N^k = \begin{bmatrix} \mathbf{0}_{(k)_N \times N-(k)_N} & \mathbf{I}_{(k)_N} \\ \mathbf{I}_{N-(k)_N} & \mathbf{0}_{N-(k)_N \times (k)_N} \end{bmatrix} = \text{circ}_N(\mathbf{e}_{(k)_N+1}), \quad (4.18)$$

and thus $\mathbf{\Pi}_N^k = \mathbf{\Pi}_N^{(k)_N}$. Hence, applying Lemma 4.2, we can calculate its eigenvalues via $\sqrt{N} \mathbf{F}^H \mathbf{e}_{(k)_N+1} = \left[1 \ \omega_N^{-k} \ \dots \ \omega_N^{-k(N-1)} \right]^T$, from which (4.17) follows. \square

4.2.2 General system of monomial equations

4.2.2.1 Derivation of the monomial equations

Our approach consists in exploiting property (4.16) to derive a set of monomial equations from \mathfrak{X} , by relying on the sole assumption that it admits a CCPD. Taking the multidimensional discrete Fourier transform (MDFT) of \mathfrak{X} , we obtain

$$\mathbf{y} = \mathfrak{X} \times_{p=1}^P \mathbf{F}^H \in \mathcal{T}. \quad (4.19)$$

Applying Propositions 2.27 and 2.49, it can be deduced from (4.14) and (4.19) that

$$\mathbf{y} = \left[\mathbf{F}^H \mathbf{C}^{(1)}, \dots, \mathbf{F}^H \mathbf{C}^{(P)} \right] \in \mathcal{T}. \quad (4.20)$$

But, since each $\mathbf{C}^{(p)} \in \mathbb{C}^{N \times R}$ is a circulant matrix, we can decompose its circulant completion as $\check{\mathbf{C}}^{(p)} = \mathbf{F} \mathbf{\Lambda}^{(p)} \mathbf{F}^H$, where $\mathbf{\Lambda}^{(p)} = \text{Diag}(\boldsymbol{\lambda}^{(p)})$ is a diagonal matrix containing the eigenvalues of $\check{\mathbf{C}}^{(p)}$. Therefore, as \mathbf{F} is both unitary and symmetric, we have

$$\mathbf{F}^H \check{\mathbf{C}}^{(p)} = \text{Diag}(\boldsymbol{\lambda}^{(p)}) \mathbf{F}^H = \left[\boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_1^* \ \boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_2^* \ \dots \ \boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_R^* \right], \quad (4.21)$$

where \mathbf{f}_n is the n th column of \mathbf{F} and \odot denotes the Hadamard product, *i.e.*, entry-wise multiplication. Thus, considering only the first R columns of (4.21),

$$\mathbf{F}^H \mathbf{C}^{(p)} = \left[\boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_1^* \ \boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_2^* \ \dots \ \boldsymbol{\lambda}^{(p)} \odot \mathbf{f}_R^* \right]. \quad (4.22)$$

³The modulo N operator is defined as follows: $\forall k \in \mathbb{Z}, N \in \mathbb{N}, (k)_N = m$ if and only if $k = lN + m$ for some $l \in \mathbb{Z}$, with $m \in \{0\} \cup \langle N-1 \rangle$.

In particular, as $\mathbf{f}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N$, the first column of (4.22) implies, as already mentioned,

$$\text{DFT} \left\{ \mathbf{c}^{(p)} \right\} = \frac{1}{\sqrt{N}} \boldsymbol{\lambda}^{(p)} \iff \text{IDFT} \left\{ \frac{1}{\sqrt{N}} \boldsymbol{\lambda}^{(p)} \right\} = \mathbf{c}^{(p)}, \quad (4.23)$$

where $\mathbf{c}^{(p)}$ is the generating vector of $\mathbf{C}^{(p)}$.

Now, by substituting (4.22) into (4.20) and resorting to the scalar form (2.85) of the CPD, the elements of \mathbf{Y} are seen to satisfy the P th order monomial equations

$$y_{n_1, \dots, n_P} = N^{-\frac{P}{2}} \left(\sum_{r=1}^R \omega_N^{-(r-1)(n_1 + \dots + n_P - P)} \right) \prod_{p=1}^P \lambda_{n_p}^{(p)}, \quad (4.24)$$

whose unknowns are the eigenvalues $\lambda_n^{(p)} = [\boldsymbol{\lambda}^{(p)}]_n$ of the circulant completions of the desired matrix factors.

Remark 4.5. The above development can be generalized to handle any CPD $\mathbf{X} = \llbracket \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \rrbracket$ such that each $\mathbf{A}^{(p)} \in \mathbb{C}^{N \times R}$ (with $R \leq N$) can be completed to form a square matrix $\tilde{\mathbf{A}}^{(p)} \in \mathbb{C}^{N \times N}$ satisfying $\tilde{\mathbf{A}}^{(p)} = \mathbf{V}^{(p)} \boldsymbol{\Lambda}^{(p)} \mathbf{Z}^{(p)}$, where $\mathbf{V}^{(p)}, \mathbf{Z}^{(p)} \in \mathbb{C}^{N \times N}$ are known *a priori*, $\mathbf{V}^{(p)}$ is nonsingular and the unknown $\boldsymbol{\Lambda}^{(p)}$ is diagonal. In that case, one computes $\mathbf{Y} = \mathbf{X} \times_{p=1}^P (\mathbf{V}^{(p)})^{-1}$, whose elements are then given by

$$y_{n_1, \dots, n_P} = \left(\sum_{r=1}^R \prod_{p=1}^P [\mathbf{Z}^{(p)}]_{n_p, r} \right) \prod_{p=1}^P \lambda_{n_p}^{(p)}. \quad (4.25)$$

4.2.2.2 Characterization of the derived equations and their solutions

In order to exploit the set of equations (4.24) for computing the factors of the CCPD of \mathbf{X} , it is important to first determine which equations are relevant, as the summation of complex exponentials can vanish for some multi-indices (n_1, \dots, n_P) . The next result establishes a necessary and sufficient condition for the non-nullity of that summation.

Proposition 4.6. Let $N, R \in \mathbb{N}$ such that $R \leq N$ and $N > 1$, $n_1, \dots, n_P \in \langle N \rangle$ and

$$Q \triangleq \frac{N}{\text{gcd}(N, n_1 + \dots + n_P - P)}, \quad (4.26)$$

where $\text{gcd}(\cdot, \cdot)$ yields the greatest common divisor of its arguments. Then, we have

$$\sum_{r=1}^R \omega_N^{-(r-1)(n_1 + \dots + n_P - P)} \neq 0 \quad (4.27)$$

if and only if one of the following (mutually exclusive) conditions are met: (i) $Q = 1$; (ii) Q does not divide R .

Proof. Defining $v \triangleq (n_1 + \dots + n_P - P)/\gcd(N, n_1 + \dots + n_P - P) \in \mathbb{N}$ and using (4.26), we can obtain from (4.27) the equivalent relation

$$\sum_{r=1}^R \omega_Q^{-(r-1)v} \neq 0. \quad (4.28)$$

It is now evident that (i) is a sufficient condition for (4.28), since $\omega_1 = 1$. On the other hand, if (ii) is true, then we necessarily have $Q > 1$. In this case, (4.28) corresponds to a sum of R Q th roots of unity raised to a power v that is co-prime with Q by definition, which yields zero if and only if $R = lQ$ for some positive integer $l < R$. Hence, (ii) also implies (4.27). Note that the last argument also establishes the only if part of the proof, since in this part we have to show that the sum in (4.27) vanishes if $Q > 1$ and Q divides R (*i.e.*, when both (i) and (ii) above are false). \square

Proposition 4.7. If $N = R$, condition (i) of Proposition 4.6 is equivalent to (4.27) and can be alternatively written as

$$\exists l \in \mathbb{N} \text{ such that } n_1 + \dots + n_P - P = lN. \quad (4.29)$$

Proof. Condition (ii) of Proposition 4.6 is immediately ruled out when $N = R$, since Q divides N by definition, whereas condition (i) and (4.29) are trivially equivalent. \square

We now focus on the characterization of the solutions of the system of equations (4.24). First, we claim that it provides no “spurious” solution, in the sense that every solution is associated with a CCPD. This is assured by the following lemma.

Lemma 4.8. Let $\mathbf{X} \in \mathcal{T}$ and assume the elements of $\mathbf{Y} = \text{MDFT}\{\mathbf{X}\}$ satisfy (4.24), for some N, R such that $R \leq N$ and some set of N -tuples $\{(\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)})\}_{p=1}^P$. Then, \mathbf{X} admits a CCPD having as factors P circulant $N \times R$ matrices such that the eigenvalues of the circulant completion of the p th factor are $\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)}$.

Proof. First, note that every solution $\{(\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)})\}_{p=1}^P$ can be (injectively) associated with P circulant $N \times R$ matrices $\mathbf{G}^{(1)}, \dots, \mathbf{G}^{(P)}$ whose circulant completions are given by $\check{\mathbf{G}}^{(p)} = \mathbf{F} \text{Diag}(\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)}) \mathbf{F}^H$. By construction, those matrices are such that $\text{MDFT}\{\llbracket \mathbf{G}^{(1)}, \dots, \mathbf{G}^{(P)} \rrbracket\} = \mathbf{Y}$, because the eigenvalues of $\check{\mathbf{G}}^{(1)}, \dots, \check{\mathbf{G}}^{(P)}$ jointly satisfy the system of equations (4.24). Now, taking the inverse multidimensional discrete Fourier transform of both sides, we obtain $\llbracket \mathbf{G}^{(1)}, \dots, \mathbf{G}^{(P)} \rrbracket = \mathbf{X}$, as claimed. \square

In general, (4.24) admits infinitely many solutions. In light of the above result, this is of course expected, because of the ambiguities that are inherent to the CPD. In particular, if the factors $\mathbf{C}^{(p)}$ in (4.14) satisfy Kruskal’s uniqueness condition, Theorem 2.41 tells us that the CPD is essentially unique, *i.e.*, with only column scaling and permutation ambiguities on

its factor matrices. This implies that the different solutions of (4.24) should also be related with each other accordingly. Our next result shows that such a relation is rather simple.

Theorem 4.9. Suppose $\mathfrak{X} \in \mathcal{T}$, with $P \geq 3$ and $N > 1$, admits a CCPD whose factors $\mathbf{C}^{(p)} \in \mathbb{C}^{N \times R}$, with $1 < R \leq N$, satisfy Kruskal's condition $\sum_{p=1}^P \text{krank}(\mathbf{C}^{(p)}) \geq 2R + P - 1$. Let $\{(\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)})\}_{p=1}^P$ denote the solution of (4.24) corresponding to the eigenvalues of $\check{\mathbf{C}}^{(p)}$, $p \in \langle P \rangle$, and assume that $\{(\mu_1^{(p)}, \mu_2^{(p)}, \dots, \mu_N^{(p)})\}_{p=1}^P$ is another solution. Then,

- (i) there are P complex scalars α_p satisfying $\prod_{p=1}^P \alpha_p = 1$ and an integer $r \in \{0\} \cup \langle R - 1 \rangle$ such that, for all $p \in \langle P \rangle$ and $n \in \langle N \rangle$, we have

$$\mu_n^{(p)} = \alpha_p \omega_N^{-r(n-1)} \lambda_n^{(p)}; \quad (4.30)$$

- (ii) defining $\mathbf{G}^{(p)} \in \mathbb{C}^{N \times R}$ as the circulant matrix whose circulant completion $\check{\mathbf{G}}^{(p)}$ has eigenvalues $\mu_1^{(p)}, \mu_2^{(p)}, \dots, \mu_N^{(p)}$, we have for all $p \in \langle P \rangle$

$$\mathbf{G}^{(p)} = \mathbf{C}^{(p)} \mathbf{\Pi}_R^r \mathbf{\Delta}_p, \quad (4.31)$$

where $\mathbf{\Delta}_p \in \mathbb{C}^{R \times R}$ is a diagonal matrix such that $\mathbf{\Delta}_1 \dots \mathbf{\Delta}_P = \mathbf{I}$ and $\alpha_p = [\mathbf{\Delta}_p]_{1,1}$.

Proof. By virtue of Lemma 4.8, \mathfrak{X} admits the CCPD $\mathfrak{X} = \llbracket \mathbf{G}^{(1)}, \dots, \mathbf{G}^{(P)} \rrbracket$. But, since we assume that the factors satisfy Kruskal's uniqueness condition, we have

$$\mathbf{G}^{(p)} = \mathbf{C}^{(p)} \mathbf{\Pi} \mathbf{\Delta}_p \quad (4.32)$$

for some permutation matrix $\mathbf{\Pi} \in \mathbb{C}^{R \times R}$ and some diagonal matrix $\mathbf{\Delta}_p \in \mathbb{C}^{R \times R}$, with the constraint $\mathbf{\Delta}_1 \dots \mathbf{\Delta}_P = \mathbf{I}$. Since by definition every column of $\mathbf{C}^{(p)}$ is of the form $\mathbf{\Pi}_N^r \mathbf{c}^{(p)}$, where $\mathbf{\Pi}_N$ is given by (4.13) and $\mathbf{c}^{(p)}$ is the generating vector of $\mathbf{C}^{(p)}$, we have from (4.32) that there exists $r \in \{0\} \cup \langle R - 1 \rangle$ such that

$$\mathbf{g}^{(p)} = \alpha_p \mathbf{\Pi}_N^r \mathbf{c}^{(p)}, \quad (4.33)$$

where $\mathbf{g}^{(p)}$ is the generating vector of $\mathbf{G}^{(p)}$ and $\alpha_p = [\mathbf{\Delta}_p]_{1,1}$. Due to the circulant structure of $\mathbf{G}^{(p)}$, we can also write

$$\mathbf{G}^{(p)} = \alpha_p \mathbf{\Pi}_N^r \mathbf{C}^{(p)}. \quad (4.34)$$

Now, substituting (4.17) in (4.33) and multiplying both sides from the left by $\sqrt{N} \mathbf{F}^H$, we obtain $\sqrt{N} \mathbf{F}^H \mathbf{g}^{(p)} = \alpha_p \sqrt{N} \mathbf{F}^H \mathbf{F} \text{Diag} \left(1, \omega_N^{-r}, \dots, \omega_N^{-r(N-1)} \right) \mathbf{F}^H \mathbf{c}^{(p)}$. But, since $\mathbf{G}^{(p)}$ and $\mathbf{C}^{(p)}$ are circulant, from (4.23) we have $\sqrt{N} \mathbf{F}^H \mathbf{g}^{(p)} = \boldsymbol{\mu}_p$ and $\sqrt{N} \mathbf{F}^H \mathbf{c}^{(p)} = \boldsymbol{\lambda}_p$, where $\boldsymbol{\mu}_p = [\mu_1^{(p)} \dots \mu_N^{(p)}]^T$ and $\boldsymbol{\lambda}_p = [\lambda_1^{(p)} \dots \lambda_N^{(p)}]^T$. Using this property and recalling that \mathbf{F} is unitary, we thus deduce $\boldsymbol{\mu}_p = \alpha_p \text{Diag} \left(1, \omega_N^{-r}, \dots, \omega_N^{-r(N-1)} \right) \boldsymbol{\lambda}_p$ or, in scalar form, $\mu_n^{(p)} = \alpha_p \omega_N^{-r(n-1)} \lambda_n^{(p)}$. To complete the first part of the proof, we observe that the constraint

$\Delta_1 \dots \Delta_P = \mathbf{I}_R$ implies $\prod_{p=1}^P \alpha_p = 1$ and that the constant r is the same for all $p \in \langle P \rangle$, since the same permutation $\mathbf{\Pi}$ applies to all factors in (4.32).

For the second part, we use the fact that $\text{krank}(\mathbf{C}^{(q)}) > 1$ must hold for at least some $q \in \langle P \rangle$, otherwise Kruskal's condition would not be satisfied. Because of that, we claim that $\mathbf{\Pi} = \mathbf{\Pi}_R^s$ for some $s \in \{0\} \cup \langle R-1 \rangle$. In other words, only a circulant permutation of the columns is allowed. To see why this is so, note that if $\mathbf{\Pi}$ is not circulant then $\mathbf{G}^{(q)}$ has two consecutive columns $\mathbf{g}_k^{(q)}$ and $\mathbf{g}_{k+1}^{(q)}$ which are (possibly rescaled versions of) non-consecutive columns in $\mathbf{C}^{(q)}$, *i.e.*,

$$\mathbf{g}_k^{(q)} = \beta_1 \mathbf{\Pi}_N^{k_1} \mathbf{c}^{(q)} \quad \text{and} \quad \mathbf{g}_{k+1}^{(q)} = \beta_2 \mathbf{\Pi}_N^{k_2} \mathbf{c}^{(q)} \quad (4.35)$$

such that $k_2 \neq (k_1 + 1)_R$, with $k_1, k_2 \in \{0\} \cup \langle R-1 \rangle$, and β_1, β_2 are nonzero. But, $\mathbf{g}_{k+1}^{(q)} = \mathbf{\Pi}_N \mathbf{g}_k^{(q)}$, and hence

$$\beta_2 \mathbf{\Pi}_N^{k_2} \mathbf{c}^{(q)} = \beta_1 \mathbf{\Pi}_N^{k_1+1} \mathbf{c}^{(q)}. \quad (4.36)$$

Now, assuming $k_2 > k_1 + 1$, the above equation implies

$$\mathbf{\Pi}_N^{k_2-k_1-1} \mathbf{c}^{(q)} = \frac{\beta_1}{\beta_2} \mathbf{c}^{(q)}. \quad (4.37)$$

As $0 < k_2 - k_1 - 1 \leq R-2$, this means that $\mathbf{C}^{(q)}$ has a column which is proportional to the first one (*i.e.*, to its generating vector), contradicting $\text{krank}(\mathbf{C}^{(q)}) > 1$. If instead $k_2 < k_1 + 1$, a similar observation holds, because

$$\mathbf{\Pi}_N^{k_1+1-k_2} \mathbf{c}^{(q)} = \frac{\beta_2}{\beta_1} \mathbf{c}^{(q)} \quad (4.38)$$

and $0 < k_1 + 1 - k_2 \leq R-1$. (Note that we cannot have $k_1 + 1 - k_2 = R$, because then that would mean $k_1 = R-1$ and $k_2 = 0$, *i.e.*, $k_2 = (k_1 + 1)_R$.) This shows that $\mathbf{\Pi} = \mathbf{\Pi}_R^s$ for some $s \in \{0\} \cup \langle R-1 \rangle$. By exploiting again the fact that $\text{krank}(\mathbf{C}^{(q)}) > 1$, it is easy to see that such an s must be unique, for otherwise $\mathbf{C}^{(q)}$ would have collinear columns. Therefore, since $\mathbf{\Pi} = \mathbf{\Pi}_R^s$ implies $\mathbf{g}^{(p)} = \alpha_p \mathbf{\Pi}_N^s \mathbf{c}^{(p)}$, a comparison with (4.33) shows $s = r$, where r is the same integer as in (4.30) and (4.33). \square

Example 4.10. Consider a tensor \mathfrak{X} which admits the CCPDs:

$$\mathfrak{X} = \left\| \left\| \begin{bmatrix} a & 0 \\ 0 & a \\ -a & 0 \\ 0 & -a \end{bmatrix}, \begin{bmatrix} b & 0 \\ 0 & b \\ b & 0 \\ 0 & b \end{bmatrix}, \begin{bmatrix} c & 0 \\ 0 & c \\ -c & 0 \\ 0 & -c \end{bmatrix} \right\| \right\| = \left\| \left\| \begin{bmatrix} 0 & a \\ -a & 0 \\ 0 & -a \\ a & 0 \end{bmatrix}, \begin{bmatrix} 0 & b \\ b & 0 \\ 0 & b \\ b & 0 \end{bmatrix}, \begin{bmatrix} 0 & c \\ -c & 0 \\ 0 & -c \\ c & 0 \end{bmatrix} \right\| \right\|. \quad (4.39)$$

When a, b, c are all nonzero, Kruskal's condition is fulfilled, and so the factors $\mathbf{G}^{(p)}$ of the second decomposition must be related to those of the first one, $\mathbf{C}^{(p)}$, as in (4.31). This is indeed true, with $R = 2$, $r = 1$, $\Delta_2 = \mathbf{I}_R$ and $\Delta_1 = \Delta_3 = \text{Diag}(-1, 1)$. On the other hand, (4.34) must be also satisfied. Note that it holds with $r = 1$, $\alpha_1 = \alpha_3 = -1$ and $\alpha_2 = 1$. \blacksquare

Corollary 4.11. If all conditions of Theorem 4.9 hold and $N = R$, then

- (i) the permutation matrix appearing in (4.32) is the same as that of (4.33), *i.e.*, $\mathbf{\Pi} = \mathbf{\Pi}_R^r$;
- (ii) $\mathbf{\Delta}_p = \alpha_p \mathbf{I}_R$ for all p .

Proof. Since all square circulant matrices commute (because they share the same set of orthonormal eigenvectors), we have from (4.34) that $\mathbf{G}^{(p)} = \alpha_p \mathbf{C}^{(p)} \mathbf{\Pi}_N^r$. Comparing this equation with (4.31), and recalling that r is unique, (i) follows.

To prove (ii), we note that all circulant permutations of the generating vector are present when $N = R$. As $\mathbf{\Pi} = \mathbf{\Pi}_N^r$, the product $\mathbf{C}^{(p)} \mathbf{\Pi}_N^r$ is also circulant. Therefore, the same scaling factor must be applied to all columns, otherwise $\mathbf{G}^{(p)}$ could not be circulant. Hence, from $\mathbf{G}^{(p)} = \alpha_p \mathbf{C}^{(p)} \mathbf{\Pi}_N^r$ we conclude that $\mathbf{\Delta}_p = \alpha_p \mathbf{I}_R$ for all p . \square

Remark 4.12. As seen above, when multiple solutions exist by virtue of the permutation ambiguity, then the corresponding permutation matrices must be circulant. When $N = R$, the converse is true: all the N distinct circulant permutations of the original factors yield equivalent CCPDs. In other words, if $\{(\lambda_1^{(p)}, \lambda_2^{(p)}, \dots, \lambda_N^{(p)})\}_{p=1}^P$ is a solution for (4.24), then so is any set $\{(\omega_N^0 \lambda_1^{(p)}, \omega_N^{-r} \lambda_2^{(p)}, \dots, \omega_N^{-r(N-1)} \lambda_N^{(p)})\}_{p=1}^P$ with $r \in \{0\} \cup \langle N-1 \rangle$. Indeed, from Proposition 4.7 we know that, when $N = R$, every nonzero equation of the form (4.24) is associated with indices n_1, \dots, n_P that satisfy $n_1 + \dots + n_P - P = lN$ for some $l \in \mathbb{N}$. Consequently, $\omega_N^{-r(n_1 + \dots + n_P - P)} = 1$ holds for such indices, and hence

$$\prod_{p=1}^P \lambda_{n_p}^{(p)} = \omega_N^{-r(n_1 + \dots + n_P - P)} \prod_{p=1}^P \lambda_{n_p}^{(p)} = \prod_{p=1}^P \omega_N^{-r(n_p - 1)} \lambda_{n_p}^{(p)}, \quad (4.40)$$

which shows that the equations (4.24) are invariant to this introduction of the factors $\omega_N^{-r(n_p - 1)}$ in the corresponding eigenvalues.

4.2.3 Symmetric system of monomial equations

The symmetric case can be derived from the previous subsection by considering that $\mathbf{C}^{(1)} = \dots = \mathbf{C}^{(P)} = \mathbf{C} \in \mathbb{C}^{N \times R}$, which yields

$$\mathbf{x} = \llbracket \mathbf{C}, \dots, \mathbf{C} \rrbracket \in \mathcal{S}(\mathcal{T}) \subset \mathcal{T}. \quad (4.41)$$

This simplifies the equations (4.24), which can be rewritten as

$$y_{n_1, \dots, n_P} = N^{-\frac{P}{2}} \left(\sum_{r=1}^R \omega_N^{-(r-1)(n_1 + \dots + n_P - P)} \right) \prod_{p=1}^P \lambda_{n_p}, \quad (4.42)$$

where the superscript of the eigenvalues has been dropped, since they all refer now to the same circulant completion $\check{\mathbf{C}}$. As the MDFT of a symmetric tensor is also symmetric, some of the equations (4.42) are redundant. Moreover, assuming Kruskal's condition holds (*i.e.*, $\text{krank}(\mathbf{C}) \geq (2R + P - 1)/P$), we can show that (4.42) admits at most PN solutions.

Theorem 4.13. Let \mathfrak{X} be given by (4.41), where $\mathbf{C} \in \mathbb{C}^{N \times R}$ is circulant, with $N \geq R > 1$. If $\text{krank}(\mathbf{C}) \geq (2R+P-1)/P$, then the equations (4.42) admit at most PN different solutions.

Proof. The proof is similar to that of Theorem 4.9, with the additional constraint $\alpha_1 = \dots = \alpha_P = \alpha$ due to symmetry. Since the scaling matrices must satisfy $\mathbf{\Delta}_1 \dots \mathbf{\Delta}_P = \mathbf{I}_R$, the only possible distinct values for α are $\alpha = \exp(2\pi p/P)$ for $p \in \{0\} \cup \langle P-1 \rangle$. Combining this fact with the existence of only N distinct values for the integer r in (4.34), we are left with at most PN distinct solutions for (4.42). \square

Remark 4.14. From the result of Corollary 4.11 we conclude that, if a symmetric CCPD with square factors is essentially unique, then (4.42) admits *exactly* PN solutions, since every combination of one of the P distinct matrices $\exp(2\pi p/P)\mathbf{I}_R$ with one of the N distinct matrices $\mathbf{\Pi}_N^r$ yields an equivalent symmetric CCPD, and no other equivalent CCPDs exist.

4.2.4 Summary of proposed algebraic solution

Algorithm 4.4 summarizes our proposed algebraic solution for the CCPD. Complexity expressions are also given for steps 1, 2 and 4. With regard to the first one, we point out that \mathfrak{Y} can be efficiently computed with a multilinear fast Fourier transform algorithm [75]. The second step can be performed by verifying for all combinations of indices $n_1, \dots, n_P \in \langle N \rangle$ which of them satisfy one of the conditions of Proposition 4.6, if $N > R$, or the condition of Proposition 4.7, if $N = R$. Evidently, since for each combination of indices these conditions depend only on P , R and N , their assessment can be done *a priori* and reused for several different hypercubic tensors of same order, rank and dimension (hence the $\mathcal{O}(1)$ cost). Step 3 is the core of the method, being discussed below. Finally, step 4 consists in simply reconstructing the desired factors from the estimated eigenvalues of their circulant completions. This can be done by computing the inverse fast Fourier transform of each obtained vector of eigenvalues, $\boldsymbol{\lambda}^{(p)}$, which yields its corresponding generating vector.

As we have shown, solving (4.24) provides an *exact* solution for the CCPD in the noiseless case. In the presence of noise, evidently, only an approximate solution can be sought. Discussing computational methods for the resolution of polynomial systems, which have been employed to solve practical problems (see, *e.g.*, [127] and references therein), is outside the scope of this thesis. Here, we consider only some simple examples of the systems (4.24) and (4.42), whose solutions can be calculated by employing easily derived *ad-hoc* procedures. This derivation is done in Section 4.2.5, in order to illustrate the approach. The derived procedures will also be used for numerical evaluation purposes in Chapter 5. This kind of *ad-hoc* approach, to which we shall refer as *ad-hoc* algebraic solution (AAS), is of low computational cost and can always be derived, as long as certain eigenvalues are nonzero. As this is generically true for hypercubic tensors \mathfrak{X} of the form (4.14), little generality is lost.

Algorithm 4.4 Algebraic solution for computing a CCPD.

Inputs: Hypercubic tensor $\mathcal{X} \in \mathcal{T}$ to be decomposed and the rank R of the CCPD.

Outputs: Circulant $N \times R$ complex matrix factors $\mathbf{C}^{(1)}, \dots, \mathbf{C}^{(P)}$ (or \mathbf{C} , in the symmetric case).

- 1: $\mathcal{Y} \leftarrow \text{MDFT}\{\mathcal{X}\}$ // $\mathcal{O}(PN^P \log N)$
 - 2: Find the relevant equations using Proposition 4.6 (or Proposition 4.7, if $N = R$) // $\mathcal{O}(1)$
 - 3: Solve the relevant equations (4.24) (resp., (4.42)) for $\lambda_n^{(p)}$ (resp., λ_n)
 - 4: **return** $\mathbf{C}^{(p)} \leftarrow N^{-1/2} \text{circ}_R(\text{IDFT}\{\lambda^{(p)}\})$ (resp., $\mathbf{C} \leftarrow N^{-1/2} \text{circ}_R(\text{IDFT}\{\lambda\})$) // $\mathcal{O}(PN \log N)$
-

A major drawback of these AAS procedures is the fact that they are neither robust to noise nor numerically well-behaved. For certain combinations of P , N and R , multiple procedures of that kind can be derived and applied, so that one can keep the candidate solution which yields best reconstruction in a least-squares sense. This strategy allows, to a certain level, a mitigation of the degradation due to noise. Moreover, an AAS estimate can be effectively and efficiently refined by iterative algorithms, as will be illustrated by the experimental results presented in Chapter 5.

4.2.5 Illustrative examples

We now consider some simple examples.

Example 4.15. Letting $\mathcal{X} = \llbracket \mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{C}^{(3)} \rrbracket$ with $\mathbf{C}^{(p)} = \text{circ}_R(\mathbf{c}^{(p)}) \in \mathbb{C}^{3 \times 2}$, it can be checked that all 27 elements of $\mathcal{Y} = \text{MDFT}\{\mathcal{X}\}$ are generally nonzero. In particular,

$$\lambda_1^{(1)} \lambda_1^{(2)} \lambda_1^{(3)} = \frac{\tilde{y}_{1,1,1}}{2}, \quad (4.43)$$

where $\tilde{y}_{n_1, n_2, n_3} \triangleq 3\sqrt{3} y_{n_1, n_2, n_3}$. Although infinitely many solutions exist due to the scaling ambiguity, we can eliminate that ambiguity by imposing the values of all but one $\lambda_1^{(p)}$, since they determine the scaling of their corresponding factors (due to (4.23)). However, this can only be done when $\lambda_1^{(1)} \lambda_1^{(2)} \lambda_1^{(3)} \neq 0$. For instance, if we choose $\lambda_1^{(2)} = \lambda_1^{(3)} = 1$, then (4.43) can be directly solved for $\lambda_1^{(1)}$. Next, we can use, for example, the equations

$$\begin{aligned} \lambda_1^{(1)} \lambda_1^{(2)} \lambda_2^{(3)} &= \frac{\tilde{y}_{1,1,2}}{1 + \omega_N^{-1}} \rightarrow \lambda_2^{(3)}, & \lambda_2^{(1)} \lambda_1^{(2)} \lambda_1^{(3)} &= \frac{\tilde{y}_{2,1,1}}{1 + \omega_N^{-1}} \rightarrow \lambda_2^{(1)}, \\ \lambda_1^{(1)} \lambda_3^{(2)} \lambda_1^{(3)} &= \frac{\tilde{y}_{1,3,1}}{1 + \omega_N^{-2}} \rightarrow \lambda_3^{(2)}, & \lambda_1^{(1)} \lambda_2^{(2)} \lambda_1^{(3)} &= \frac{\tilde{y}_{1,2,1}}{1 + \omega_N^{-1}} \rightarrow \lambda_2^{(2)}, \\ \lambda_1^{(1)} \lambda_1^{(2)} \lambda_3^{(3)} &= \frac{\tilde{y}_{1,1,3}}{1 + \omega_N^{-2}} \rightarrow \lambda_3^{(3)}, & \lambda_3^{(1)} \lambda_1^{(2)} \lambda_1^{(3)} &= \frac{\tilde{y}_{3,1,1}}{1 + \omega_N^{-2}} \rightarrow \lambda_3^{(1)} \end{aligned}$$

to calculate the values of the other eigenvalues, as indicated. This scheme amounts to an AAS procedure for the non-symmetric setting with $P = N = 3$ and $R = 2$. ■

Example 4.16. For $P = 3$, $N = 4$ and $R = 3$, all 64 equations are generally nonzero. So, if $\lambda_1^{(p)} \neq 0$ for all p , then we can assume $\lambda_1^{(2)} = \lambda_1^{(3)} = 1$ without loss of generality and compute

the remaining eigenvalues from $\lambda_1^{(1)}\lambda_1^{(2)}\lambda_1^{(3)} = \frac{8}{3}y_{1,1,1}$ and

$$\begin{aligned} \lambda_1^{(1)}\lambda_2^{(2)}\lambda_1^{(3)} &= 8jy_{1,2,1}, & \lambda_2^{(1)}\lambda_1^{(2)}\lambda_1^{(3)} &= 8jy_{2,1,1}, & \lambda_1^{(1)}\lambda_1^{(2)}\lambda_2^{(3)} &= 8jy_{1,1,2}, \\ \lambda_1^{(1)}\lambda_3^{(2)}\lambda_1^{(3)} &= 8y_{1,3,1}, & \lambda_3^{(1)}\lambda_1^{(2)}\lambda_1^{(3)} &= 8y_{3,1,1}, & \lambda_1^{(1)}\lambda_1^{(2)}\lambda_3^{(3)} &= 8y_{1,1,3}, \\ \lambda_1^{(1)}\lambda_4^{(2)}\lambda_1^{(3)} &= -8jy_{1,4,1}, & \lambda_4^{(1)}\lambda_1^{(2)}\lambda_1^{(3)} &= -8jy_{4,1,1}, & \lambda_1^{(1)}\lambda_1^{(2)}\lambda_4^{(3)} &= -8jy_{1,1,4}. \end{aligned} \quad (4.44)$$

Alternatively, if $\lambda_2^{(1)}\lambda_2^{(2)}\lambda_2^{(3)} \neq 0$, then a similar solution consists in exploiting instead the equations for $y_{2,2,2}, y_{2,2,1}, y_{2,2,3}, y_{2,2,4}, y_{2,1,2}, y_{2,3,2}, y_{2,4,2}, y_{1,2,2}, y_{3,2,2}, y_{4,2,2}$, after imposing $\lambda_2^{(2)} = \lambda_2^{(3)} = 1$. Note that these equations are all different from those in (4.44). Thus, if the elements of \mathcal{Y} contain i.i.d. noise, it makes sense to apply in practice both derived AAS procedures and then keep the solution with the best least-squares reconstruction error with respect to \mathcal{Y} , which can reduce degradation of the results. ■

Example 4.17. When $R < N$, the permutation ambiguity is only present if the factors do not have full column rank. This can be grasped by considering the monomial equations together with the result of Theorem 4.9. For instance, if $\{(\mu_1^{(p)}, \mu_2^{(p)}, \mu_3^{(p)}, \mu_4^{(p)})\}_{p=1}^3$ provides another solution for the system (4.44), then $\mu_n^{(p)} = \alpha_p \omega_N^{-r(n-1)} \lambda_n^{(p)}$ for all $n \in \langle N \rangle$ and $p \in \langle P \rangle$. On the other hand, this requires, for instance, that the equation $\lambda_2^{(1)}\lambda_2^{(2)}\lambda_4^{(3)} = \mu_2^{(1)}\mu_2^{(2)}\mu_4^{(3)} = \lambda_2^{(1)}\lambda_2^{(2)}\lambda_4^{(3)}\omega_N^{-5r}$ holds, where we have used the fact that $\alpha_1\alpha_2\alpha_3 = 1$. Now, if there is a permutation indeterminacy, then $r \in \langle R-1 \rangle$. But, since this implies $\omega_N^{-5r} \neq 1$, there must be at least one null eigenvalue among $\lambda_2^{(1)}, \lambda_2^{(2)}, \lambda_4^{(3)}$. ■

The simple approach followed in Examples 4.15 and 4.16 for deriving AAS procedures can be employed in general, but may involve more complicated procedures and may require the non-nullity of other eigenvalues. Furthermore, disjoint sets of equations may not be available for deriving multiple alternative AAS procedures. When $N = R$, for instance, the pattern of vanishing equations generally leaves N^{P-1} exploitable equations out of N^P .

Example 4.18. For $P = N = R = 3$, only $3^2 = 9$ equations (in 9 unknowns) are available, having the form $\lambda_{n_1}^{(1)}\lambda_{n_2}^{(2)}\lambda_{n_3}^{(3)} = 3\sqrt{3}y_{n_1,n_2,n_3}$, with $n_p \in \langle 3 \rangle$. Here, no two disjoint sets of equations can be exploited in order to derive alternative AAS procedures. One possible procedure is to assume $\lambda_1^{(2)} = \lambda_1^{(3)} = 1$, determine $\lambda_1^{(1)}$ from $y_{1,1,1}$ and then compute $\beta_1 = y_{2,1,3}/y_{1,2,3}$, $\beta_2 = y_{2,3,1}/y_{1,3,2}$ and $\beta_3 = 3\sqrt{3}(\lambda_1^{(1)})^2\beta_1\beta_2y_{2,2,2} = (\lambda_2^{(1)})^3$. This gives us three different solutions for $\lambda_2^{(1)}$, each of which can be exploited to yield (different) solutions for all eigenvalues. This multiplicity stems from the permutation ambiguity, as $(\lambda_2^{(1)}\omega_N^r)^3 = (\lambda_2^{(1)})^3$ for all $r \in \{0, 1, 2\}$. ■

Example 4.19. For the symmetric case where $P = N = R = 3$, defining $\tilde{y}_{n_1,n_2,n_3} \triangleq 3\sqrt{3}y_{n_1,n_2,n_3}$ and disregarding redundant equations due to symmetries, we have the following equations: $\lambda_1^3 = \tilde{y}_{1,1,1}$, $\lambda_2^3 = \tilde{y}_{2,2,2}$, $\lambda_3^3 = \tilde{y}_{3,3,3}$ and $\lambda_1\lambda_2\lambda_3 = \tilde{y}_{1,2,3}$. Here, the coupled

equation in $\lambda_1\lambda_2\lambda_3$ constrains the possible solutions, so that there can be at most $PN = 9$ instead of 27 solutions, as stated by [Theorem 4.13](#). ■

Remark 4.20. In the previous example, if it is known *a priori* that $\mathbf{C} \in \mathbb{R}^{N \times R}$ and $\lambda_1 \neq 0$, we can choose a real solution for the equation involving λ_1^P , which is guaranteed to exist (in the noiseless case) due to (4.23). This allows suppressing the scaling ambiguity whilst obtaining a real factor \mathbf{C} .

Example 4.21. When $N = 4$, $R = 3$, all factors are identical and $\lambda_1 \neq 0$, then one possible straightforward procedure for computing a solution is to calculate

$$\lambda_1 = 2 \sqrt[3]{\frac{1}{3}y_{1,1,1}}, \quad \lambda_2 = 8j \frac{y_{2,1,1}}{\lambda_1^2}, \quad \lambda_3 = 8 \frac{y_{3,1,1}}{\lambda_1^2}, \quad \lambda_4 = -8j \frac{y_{4,1,1}}{\lambda_1^2}. \quad (4.45)$$

Other subsets of equations can be similarly exploited. Due to the scaling ambiguity, the generating vector \mathbf{c} can only be estimated up to a complex factor $e^{j\frac{2\pi}{3}p}$, with $p \in \{0, 1, 2\}$. When \mathbf{c} is known to be real, this can be exploited in order to cancel such a factor out. ■

4.3 Final remarks

We have proposed in this chapter two methods for the problem of [SCPD](#) estimation.

The first one is a general constrained formulation of [ALS](#) for handling the estimation of linearly structured factors, which generalizes previously proposed *ad-hoc* algorithms. We derived versions with exact and with approximate iterates, which can help finding a good compromise between accuracy and computing time, and gave also an explicit formulation for the case with partial or total symmetry. This [CALs](#) approach can be used in conjunction with a non-iterative approach (such as [AAS](#) or the methods of [Sections 3.3.2](#) and [3.3.3](#)).

The second solution brings the possibility of employing non-iterative procedures (such as the [AAS](#) procedures) for estimating [CCPDs](#). Though its estimates can be degraded by noise and numerical errors, they are quite cheap (in terms of computing complexity) approximate solutions which can be further refined. It can thus be used alongside an iterative algorithm, allowing to reduce the total computing time. In comparison with the approach of [Section 3.3.2](#), our algebraic solution allows simultaneously estimating multiple factors having a more general circulant structure. In comparison with the method reviewed in [Section 3.3.3](#), it is less robust numerically, but permits taking the structure of all factors into account and is able to handle the symmetric case. Furthermore, square non-banded circulant factors can also be estimated with our algebraic approach, while the method of [Section 3.3.3](#) does not seem to apply in this case, due to non-uniqueness of the solution of (3.33).

Numerical evaluation of SCPD algorithms

In the first part of this chapter, we investigate properties of the **CALS** algorithm variants proposed in [Section 4.1](#) by means of computer experiments. Then, a numerical evaluation of the statistical performance of several **SCPD** estimators is conducted in the second part, by resorting to the **CRB** expressions given in [Section 3.4](#). Finally, based on the insights gained from the experimental outcomes, we draw concluding remarks for this part of the thesis.

Contents

5.1	Evaluation of CALS schemes	88
5.2	Statistical performance of SCPD algorithms	96
5.3	Final remarks	105

We note that all simulations reported in this chapter, and also on the upcoming ones, were performed in Matlab R2013a running on a Intel Xeon ES-2630v2 2.60 GHz with 32 GB of 1866 MHz RAM memory.

5.1 Evaluation of CALS schemes

In this section, we empirically study the behavior of **CALS** in several simulation scenarios. This study is focused on three aspects:

1. *Solution quality.* Since it tackles a nonlinear least-squares problem, convergence to a local minimum is the best we can generally expect from **CALS**. Thus, the choice of the initial solution naturally influences the quality of its estimate. It is therefore important to study how sensitive **CALS** is to degradation caused by a poorly chosen initial point. This degradation can be due to convergence to a local minimum, premature stopping induced by slow convergence or failure to converge within a maximum number of iterations. In particular, a relevant question is how the exact and approxi-

mate variants formulated in Sections 4.1.1 and 4.1.2 compare with respect to solution quality, assuming both start from a same initial point.

2. *Computing time.* The initialization also affects the amount of iterations required by CALS to converge (and whether it converges at all). Hence, it is reasonable to ask how the total computing time varies as a function of the distance between the initial solution and a global minimum. We have shown that the approximate iterates of Section 4.1.2 have reduced computing cost. Yet, as such an approximation might slow down convergence speed and thus increase the total number of iterations, it is crucial to investigate whether it can bring an effective overall economy.
3. *Presence of scaling factors and normalization.* In Section 4.1.1, we claimed that unnecessary scaling factors should not be included in an SCPD. Also, in Section 4.1.3 it was argued that normalization is critical when all factors are identical, which requires inclusion of a (scalar) scaling factor λ . These remarks will be illustrated by means of experimental results.

5.1.1 Dependence of solution quality and computing time on initialization

We now describe some numerical simulations aimed at studying aspects 1 and 2 listed above.

5.1.1.1 Non-symmetric case

Let us start by considering the computation of non-symmetric SCPDs. Here, the decomposed tensors belong to $\mathcal{T} = \mathbb{R}^{20} \otimes \mathbb{R}^{12} \otimes \mathbb{R}^{15}$. Two simulation scenarios are considered. The tensor to be decomposed in Scenario 1 is

$$\mathbf{y} = \llbracket \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket \in \mathcal{T}, \quad (5.1)$$

where $\mathbf{A}^{(1)} \in \mathbb{R}^{20 \times 10}$ is Toeplitz, $\mathbf{A}^{(2)} \in \mathbb{R}^{12 \times 10}$ is circulant and $\mathbf{A}^{(3)} \in \mathbb{R}^{15 \times 10}$ is banded circulant. In Scenario 2, we decompose $\mathbf{y} = \mathbf{y}_0 + \sigma_n \mathbf{N}$, where \mathbf{y}_0 is given by the same SCPD as in (5.1), $\mathbf{N} \in \mathcal{T}$ has its i.i.d. elements drawn from a standard Gaussian distribution, and the scalar σ_n is chosen so that $\text{SNR} = \frac{\|\mathbf{y}_0\|_F^2}{\|\sigma_n \mathbf{N}\|_F^2} = 50$ dB. The elements of $\boldsymbol{\lambda}$ and $\boldsymbol{\theta}^{(p)}$ are also drawn from a standard Gaussian distribution. The matrix factors are constructed according to (3.1), and then normalized so that the first column has unit norm.

The experimental procedure is as follows:

- (i) 100 realizations of \mathbf{y} are generated as described above.
- (ii) For each model realization, 10 random initial points corresponding to perturbed solutions are generated via $\hat{\mathbf{A}}_0^{(p)} = (1 - \alpha)\mathbf{A}^{(p)} + \alpha \boldsymbol{\Delta}_p$ and $\hat{\boldsymbol{\lambda}}_0 = (1 - \alpha)\boldsymbol{\lambda} + \alpha \boldsymbol{\delta}$, where $\boldsymbol{\Delta}_p$ and $\boldsymbol{\delta}$ are generated as $\mathbf{A}^{(p)}$ and $\boldsymbol{\lambda}$, respectively, and $\alpha \in \{0.1, 0.5, 0.9\}$.

Scenario	Exact			Approximate			Mixed		
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$
1	1.00 <u>0.00</u>	0.88 <u>0.22</u>	0.70 <u>0.51</u>	0.56 <u>0.00</u>	0.44 <u>0.00</u>	0.41 <u>0.00</u>	0.98 <u>0.13</u>	0.97 <u>0.27</u>	0.96 <u>0.30</u>
2	1.00 <u>0.00</u>	0.88 <u>0.22</u>	0.75 <u>0.54</u>	0.00 <u>0.00</u>	0.00 <u>0.00</u>	0.00 <u>0.00</u>	0.99 <u>0.15</u>	0.96 <u>0.26</u>	0.96 <u>0.30</u>

Table 5.1: Convergence in Scenarios 1 and 2 of Section 5.1.1.1: proportion of runs which converge to a minimum (bold); and proportion of runs which converge to a local minimum of NSE larger than -75 dB in Scenario 1 and larger than -30 dB in Scenario 2 (underlined).

(iii) Three variants of CALS are employed: CALS with exact iterates (Algorithm 4.1), CALS with approximate iterates (Algorithm 4.2), and a “mixed” version in which a number of approximate iterates is performed before switching to exact ones. As stopping criteria, we use (3.20) with tolerance $\epsilon = 10^{-12}$ and set the maximum number of iterations to $K = 1500$. In mixed CALS, at most 300 of the K iterations are allowed to be approximate; if (3.20) is met after $K_1 < 300$ iterations, then the algorithm switches to exact iterates right away.

(iv) To verify that a minimum has been reached,¹ we check whether the gradient of the cost function (see (5.3)) has norm smaller than 10^{-5} .

Fig. 5.1 shows, for each variant, the empirical cumulative distribution functions (c.d.f.) of the NSE and of the computing time spent at each run. Table 5.1 displays (i) the proportion of runs which reach a minimum (in boldface) and (ii) the proportion of runs which reach a (local) minimum having final NSE larger than -75 dB (underlined). It can be seen that, when the initialization is quite close ($\alpha = 0.1$) to the global optimum ($\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$), exact iterates reach that optimum almost certainly. For approximate CALS, this is true at only 56% of the realizations (though about 60% attain a small error), while none of the remaining runs reached a minimum. The gap with respect to the exact version is reduced when the mixed approach is used, so that 85% of the runs attain the global minimum. The performances of approximate CALS are highly similar for $\alpha = 0.5$ and $\alpha = 0.9$, and likewise for mixed CALS. The same is not true for the exact version, whose share of runs trapped by local minima rises from 22% ($\alpha = 0.5$) to 51% ($\alpha = 0.9$). Exact CALS is outperformed by mixed CALS for these values of α , and even by approximate CALS when $\alpha = 0.9$.

Fig. 5.1(d)–(f) show that, despite the smaller cost per iteration of approximate CALS, it takes longer than exact CALS for $\alpha = 0.1$, as it may require many more iterations. For instance, almost all realizations of exact CALS take no more than 0.2 seconds, the same

¹The condition (3.20) does not imply convergence to a minimum, as the algorithm may have entered a region of slow progress or may even have stagnated far from a minimum, since convergence is not guaranteed.

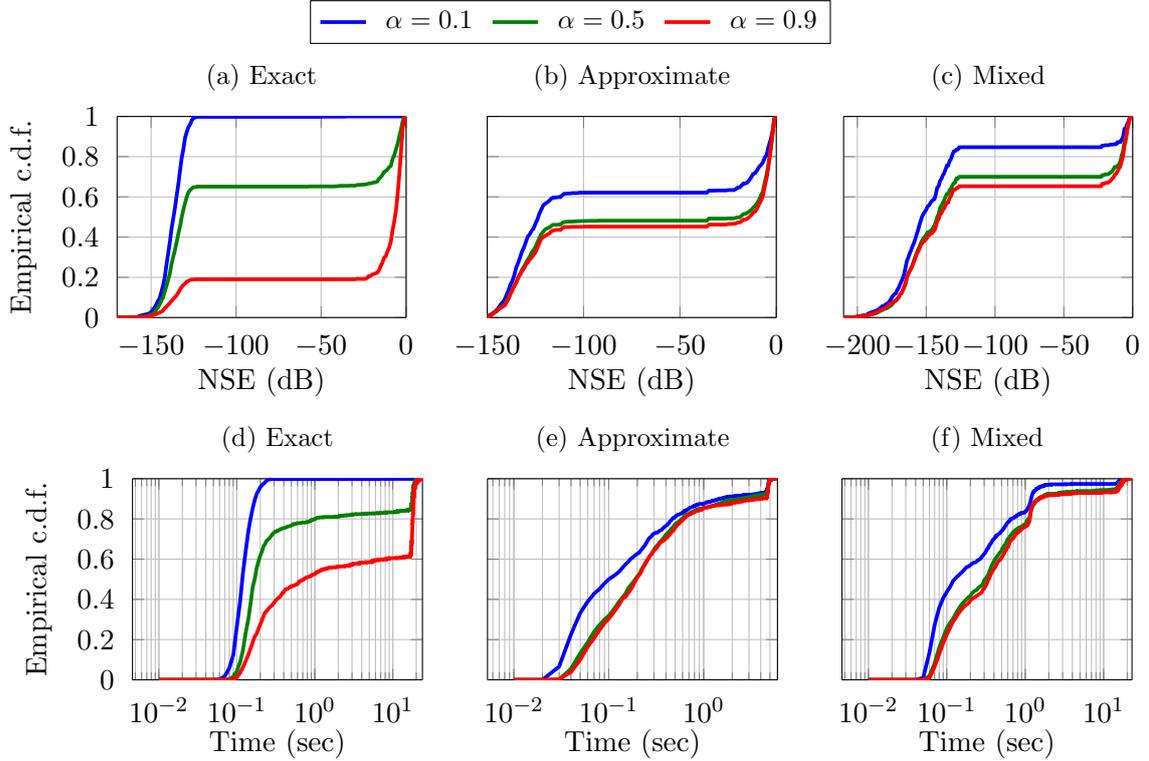


Figure 5.1: Scenario 1 of Section 5.1.1.1: empirical c.d.f. of the NSE attained by CALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.

being true for about 60% of the runs of approximate CALS. Yet, the computing cost of exact CALS varies considerably with α . So, when $\alpha = 0.5$ or $\alpha = 0.9$, approximate and mixed CALS enjoy a visible computational advantage in comparison with the exact version.

Fig. 5.2 shows the results for Scenario 2, where noise is present with an SNR of 50 dB. The behavior is similar to that seen in Scenario 1, but now the added noise prevents the NSE from going below -50 dB and approximate CALS always stops before reaching a minimum.

The above outcomes demonstrate that exact CALS is the best choice when the initial solution is close to a global optimum, as expected. On the other hand, if the initial solution is sufficiently far from the sought global optimum, the performance of exact CALS tends to degrade, because it is more likely to get trapped by local minima or to fail to converge. Approximate CALS, it seems, does not get stuck around other minima than the desired solution, and thus is more robust than exact CALS with respect to initialization. This apparently comes from the fact that the approximate updates are not locally optimal. Indeed, we have observed that the NSE often increases or oscillates before convergence to the global optimum. Even when approximate CALS does not converge to the optimum, it often reaches some close enough estimate from where exact CALS can attain the desired solution, which is what frequently happens in the mixed CALS approach.

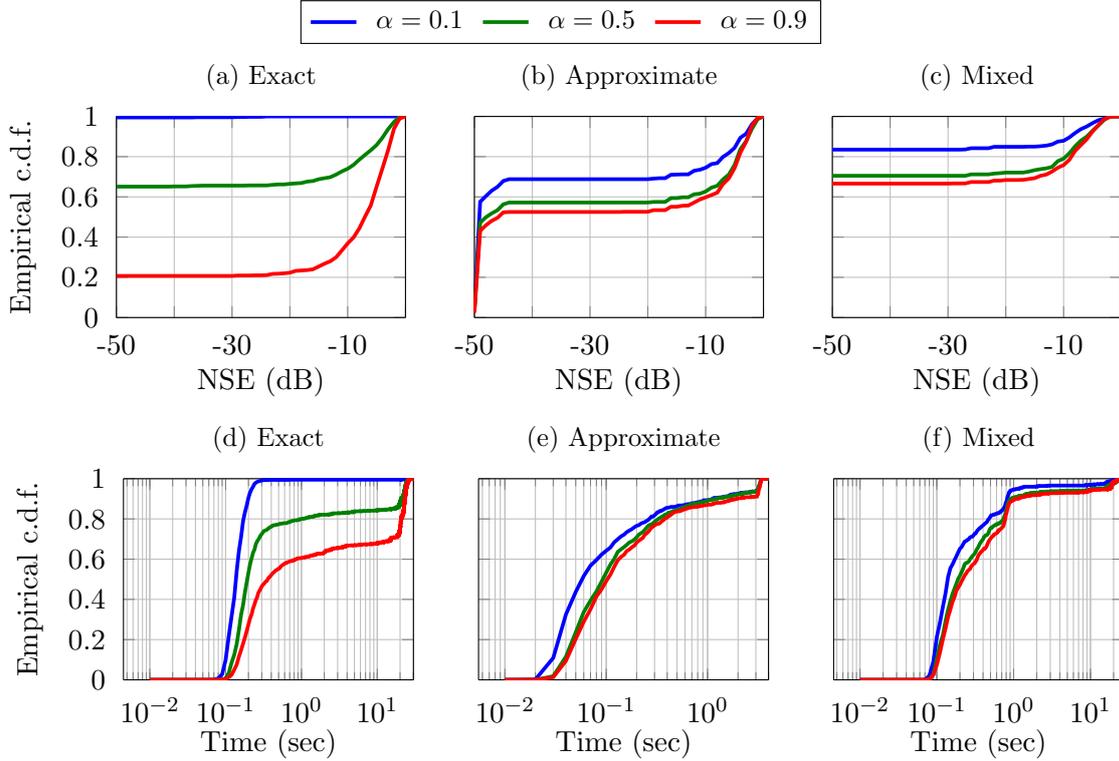


Figure 5.2: Scenario 2 of Section 5.1.1.1: empirical c.d.f. of the NSE attained by CALS (top) and of the overall computing time it takes (bottom) under varying initial conditions and $\text{SNR} = 50$ dB.

5.1.1.2 Symmetric case

We now turn to the case where the SCPD is partially or completely symmetric. The experimental procedure is the same as in the previous section, but now employing the SCALS algorithm (see Section 4.1.3). Two scenarios are again simulated for $P = 3$. Scenario 1 concerns the case where all three factors are given by a same circulant matrix $\mathbf{A} \in \mathbb{R}^{20 \times 10}$. In Scenario 2, $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ are still identical and circulant, but $\mathbf{A}^{(3)} \in \mathbb{R}^{15 \times 10}$ is a distinct matrix having banded circulant structure.

We plot in Fig. 5.3 the results of Scenario 1 and display the proportion of runs converging to a local minimum on Table 5.2. For $\alpha = 0.1$, the results are similar to those seen in the non-symmetric setting, with “exact” SCALS² achieving a quasi-perfect success rate. For the other values of α , though, its performance is not as good before. As the iterates are necessarily of approximate nature in the symmetric case, the algorithm suffers from convergence difficulties more often than in the non-symmetric case. This also hampers the performance of mixed SCALS, because the additional “exact” iterations often do not bring much improvement over

²Recall from Section 4.1.3 that the subproblem solution for repeated factors is never really exact in the symmetric case.

Scenario	Exact			Approximate			Mixed		
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.9$
1	1.00 <u>0.00</u>	0.49 <u>0.00</u>	0.13 <u>0.00</u>	0.61 <u>0.00</u>	0.30 <u>0.00</u>	0.28 <u>0.00</u>	0.78 <u>0.00</u>	0.43 <u>0.00</u>	0.43 <u>0.00</u>
2	1.00 <u>0.00</u>	0.70 <u>0.00</u>	0.25 <u>0.00</u>	0.69 <u>0.00</u>	0.54 <u>0.00</u>	0.51 <u>0.00</u>	0.90 <u>0.00</u>	0.72 <u>0.00</u>	0.69 <u>0.00</u>

Table 5.2: Convergence in Scenario 1 of Section 5.1.1.2: proportion of runs which converge to a minimum (bold); and proportion of runs which converge to a local minimum of NSE larger than -75 dB (underlined).

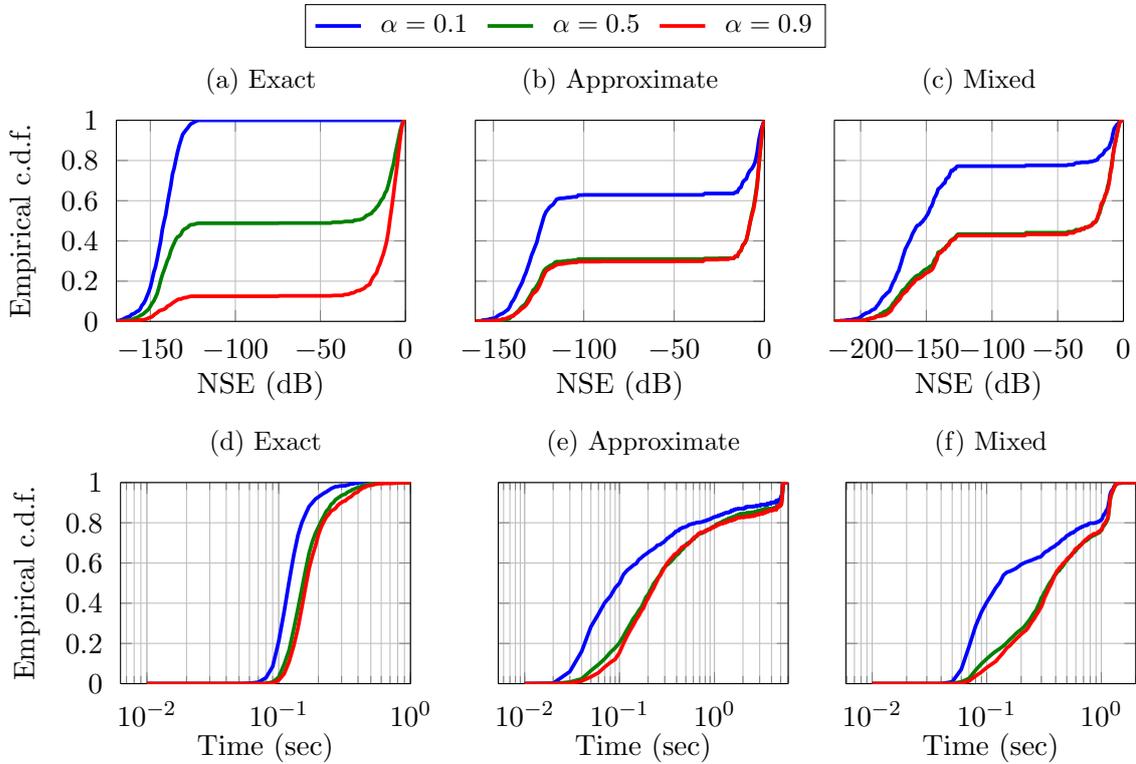


Figure 5.3: Scenario 1 of Section 5.1.1.2: empirical c.d.f. of the NSE attained by SCALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.

the initial approximate solution. Moreover, we notice that all runs with NSE larger than -75 dB either stagnate or attain the maximum number of iterations without reaching a local minimum. This behavior is similar to that of approximate CALS in the non-symmetric case. In Scenario 2, whose results are shown in Fig. 5.4, the same deterioration is not observed, because the estimate of the third factor is now exact. Indeed, the performances are even a little better than those of the non-symmetric case.

With respect to computing time, the exact version is in average more efficient in both scenarios and for all α . This stems from the economy brought by estimating identical factors

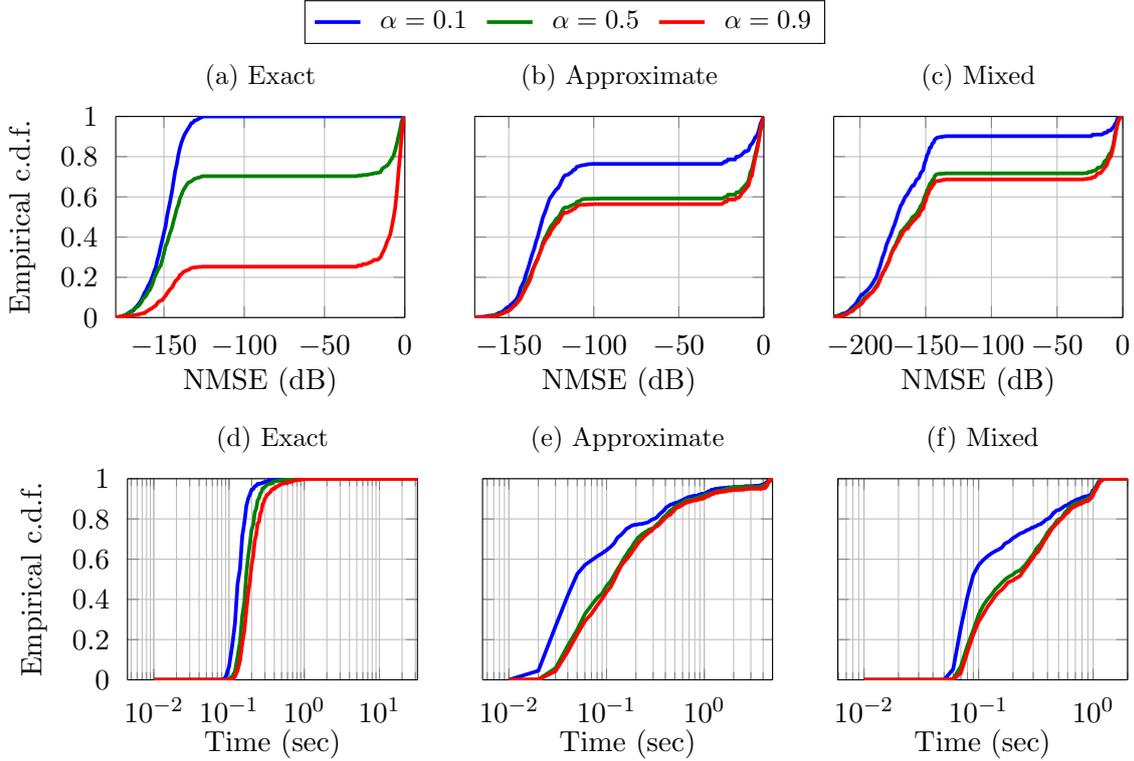


Figure 5.4: Scenario 2 of Section 5.1.1.2: empirical c.d.f. of the NSE attained by SCALS (top) and of the overall computing time it takes (bottom) under varying initial conditions.

only once, together with the smaller number of iterations which are usually needed.

5.1.2 Presence of scaling factors and normalization

We now study the impact of estimating vector-valued instead of scalar-valued scaling factors and also of applying normalization during the iterations of CALS and of SCALS. Two study cases are considered: Scenario 1 concerns a non-symmetric SCPD, while Scenario 2 is focused on a symmetric one. In the former, 100 realizations of $\mathcal{Y} = [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}]$ are constructed, where the factors $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$ are exactly as in Scenario 1 of Section 5.1.1.1. Given \mathcal{Y} , we apply CALS (with exact iterates) with the same stopping criteria as in Section 5.1.1.1. Now, the algorithm is run with three types of iterates: (i) exactly as described by Algorithm 4.1, which includes the estimation of $\boldsymbol{\lambda}$ absorbing the scaling factors of the model; (ii) as described by Algorithm 4.1 but with a single scaling factor λ and (iii) without the normalization step of line 9 and without the estimation of $\boldsymbol{\lambda}$ ($\hat{\boldsymbol{\Lambda}}_{k-1}$ is absent at line 5). Fig. 5.5 shows the computed c.d.f. of the NSE attained by each one of the applied versions. It is clear that, in comparison with the version using scalar-valued scaling or with the unnormalized version, that estimating a vector-valued $\boldsymbol{\lambda}$ has a slightly worse performance, due to the reasons discussed in Section 4.1.1.

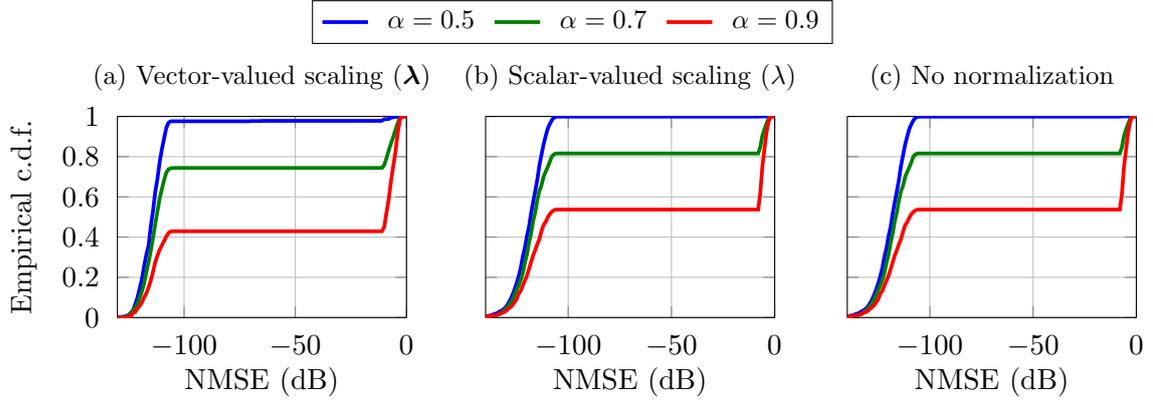


Figure 5.5: Scenario 1 of Section 5.1.2: empirical c.d.f. of the NSE attained by CALS, under varying initial conditions, in the estimation of a non-symmetric SCPD.

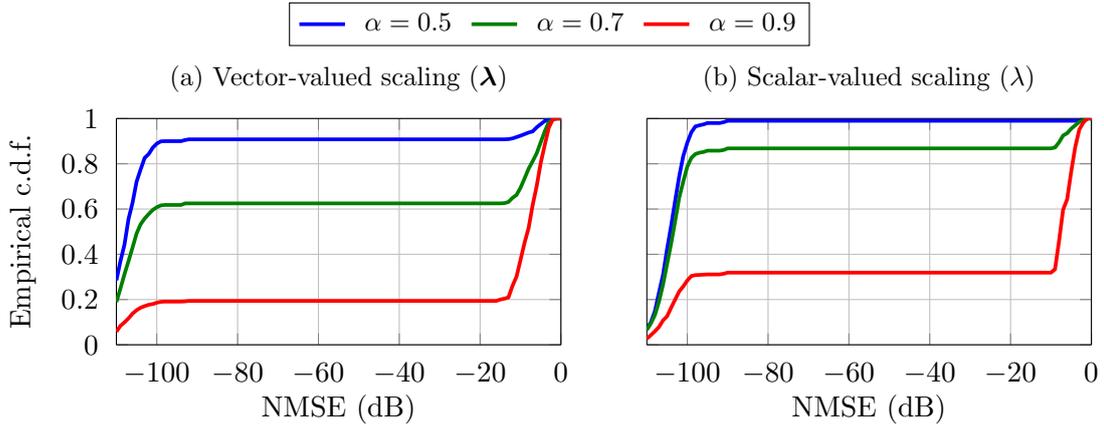


Figure 5.6: Scenario 2 of Section 5.1.2: empirical c.d.f. of the NSE attained by SCALS, under varying initial conditions, in the estimation of a symmetric SCPD.

In Scenario 2, we repeated the same procedure in a symmetric setting where $\mathbf{y} = [\mathbf{A}^{(1)}, \mathbf{A}^{(1)}, \mathbf{A}^{(1)}]$ and $\mathbf{A}^{(1)}$ is constructed exactly as in the previous scenario. This leads to the results shown in Fig. 5.6, where the curves given by the unnormalized version are not shown because it always diverges due to the resulting numerical instability which is discussed in Section 4.1.3. Similarly to the previous scenario, it can be seen that the performance is degraded by the introduction of additional (and unnecessary) degrees of freedom brought by a vector-valued λ .

We thus conclude that (i) estimating an overcomplex model can cause a significant degradation of the results and (ii) normalization is crucial in the (completely) symmetric setting, otherwise the algorithm diverges. The first conclusion is explained by the potential introduction of local minima, as the overcomplex model is more general than the sought SCPD.

5.2 Statistical performance of SCPD algorithms

Although a wide variety of algorithms have been developed for estimating CPDs, very few statistical assessments exist in the literature. The first one was [136], which derived the Cramér-Rao bound (CRB) for unstructured third- and fourth-order tensors, and applied it to evaluate the performance of the standard ALS algorithm. Apart from [136], both [173] and [23] have derived CRBs for the estimation of CPDs having Vandermonde factors, motivated, respectively, by the estimation of the directions of arrival of multiple source signals and by the estimation of the multidimensional harmonic model.

In Chapter 3, we saw closed-form expressions for the CRB of the SCPD estimation problem which were derived in [65]. We now apply them for evaluating the statistical performance of estimators by means of Monte Carlo simulations. The evaluation is performed in a Bayesian framework: prior distributions are assigned to the parameters of interest, and then the BMSE of each algorithm is estimated by computing the ensemble average of the measured MSE. The CRB is also averaged with respect to the parameter vector realizations, yielding the ECRB. All ensemble averages are calculated by taking the 2% trimmed (or truncated) mean of the data, *i.e.*, the 2% largest and 2% smallest values are discarded before computing the average. This is done in order to attenuate the degrading effect of a few realizations whose results are outstandingly poor.

5.2.1 Evaluated estimation algorithms

In Chapters 3 and 4, four different SCPD estimation methods have been described. Here, we evaluate:

- A1) the subspace-based solution (SBS) described in Section 3.3.3;
- A2) *ad-hoc* procedures based on the algebraic solution of Section 4.2, which are referred to as AAS in Section 4.2.4;
- A3) the CALS algorithm proposed in Section 4.1, either using only exact iterates or adopting the mixed approach described in Section 5.1;
- A4) combinations of the above non-iterative methods (*i.e.*, items A1 and A2) with iterative ones, which are applied for refining the estimates provided by the former.

We note that the method described in Section 3.3.2 is not included because its rationale is essentially the same as that of SBS, whilst the latter is more general.

The last item on the above list is naturally relevant because, despite bearing a small computational cost in comparison with iterative methods, the non-iterative approaches of algorithms A1 and A2 often fall short of precision, especially in the presence of noise. Their

estimates can be refined, *e.g.*, by **CALS** or by a gradient, Newton or quasi-Newton descent. As our **SCPD** model involves **AWGN** (cf. Section 3.4.2), the maximum likelihood estimate is the solution of the nonlinear least-squares problem tackled by **CALS**. Using (3.40) and (3.42), this problem can be rewritten in vector form as

$$\min_{\boldsymbol{\eta}} \|\mathbf{y} - \mathbf{x}(\boldsymbol{\eta})\|_2^2 = \min_{\boldsymbol{\eta}} \|\mathbf{y} - \Phi(\boldsymbol{\lambda})\mathbf{f}(\boldsymbol{\theta})\|_2^2. \quad (5.2)$$

The gradient of its cost function is

$$\nabla_{\boldsymbol{\eta}} = -\mathbf{J}^T(\boldsymbol{\eta})(\mathbf{y} - \Phi(\boldsymbol{\lambda})\mathbf{f}(\boldsymbol{\theta})), \quad (5.3)$$

where \mathbf{J} denotes the Jacobian of $\mathbf{x}(\boldsymbol{\eta})$ with respect to $\boldsymbol{\eta}$, which can be calculated by using the expressions derived at Section 3.4.4. When both \mathfrak{X} and \mathfrak{N} are (partially or completely) symmetric, as discussed in Section 3.4.4.2, we have the problem

$$\min_{\boldsymbol{\eta}} \|\Psi(\mathbf{y} - \Phi(\boldsymbol{\lambda})\mathbf{f}(\boldsymbol{\theta}))\|_F^2, \quad (5.4)$$

where Ψ is a selection matrix meant to eliminate redundancies arising due to symmetry. Thus, the gradient reads

$$\nabla_{\boldsymbol{\eta}} = -\mathbf{J}^T(\boldsymbol{\eta})\Psi^T\Psi(\mathbf{y} - \Phi(\boldsymbol{\lambda})\mathbf{f}(\boldsymbol{\theta})). \quad (5.5)$$

In the next sections, we shall employ the above expressions to refine estimates produced by non-iterative schemes with the use of the Broyden-Fletcher-Goldfarb-Shanno (**BFGS**) algorithm [27], which is a quasi-Newton optimization method. Essentially, instead of employing the exact Hessian matrix at each iteration, it resorts to an approximation which takes into account the information provided by the sequence of computed gradient vectors. This is done for computational reasons, since computing the Hessian is usually a quite time-consuming task. The reader is referred to [27, 152] for a detailed explanation of this algorithm. We employ here the Fortran implementation **L-BFGS-B** [219], for which a Matlab interface is available at <http://github.com/pcarbo/lbfgsb-matlab>.

5.2.2 Applicability of the CRB

The validity of our statistical evaluation can be justified along the same lines of [136]. Under the assumption that \mathbf{y} is disturbed by **AWGN**, algorithms which approach a global minimum of problem (5.2) deliver maximum likelihood estimates, which are asymptotically unbiased and asymptotically attain the **CRB** under mild conditions. In signal-in-noise problems, this is approximately true even for a small sample size, as long as the **SNR** is sufficiently high [113]. Under such an **SNR** condition, it is thus legitimate to compare the **MSE** of these algorithms with the **CRB**. Though the same cannot be claimed for **SBS** and **AAS** without a refinement step, we are mostly interested in the case where such a refinement is performed by an iterative method for solving (5.2).

5.2.3 Non-symmetric SCPD

Two different scenarios are considered for the evaluation of non-symmetric SCPD estimators. The first one concerns the CCPD of a hypercubic tensor (see Section 4.2), and thus all algorithms enumerated in Section 5.2.1 apply. In the second scenario, we address a more general SCPD, which cannot be handled by AAS.

5.2.3.1 Scenario 1: circulant-constrained CPD

Here, the SCPD model is constructed as $\mathcal{X} = [\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \mathbf{C}^{(3)}] \in \mathcal{T} \triangleq (\mathbb{R}^N)^{\otimes 3}$, where each $\mathbf{C}^{(p)} \in \mathbb{R}^{N \times R}$ is circulant with $N = 4$ and $R = 3$. We fix $\theta_1^{(1)} = \theta_1^{(2)} = 1$ to avoid identifiability issues, as discussed in Section 3.4.3. Hence, applying the notation of that section, we have $\boldsymbol{\eta} = [(\bar{\boldsymbol{\theta}}^{(1)})^T \ (\bar{\boldsymbol{\theta}}^{(2)})^T \ (\boldsymbol{\theta}^{(3)})^T]^T$. A Gaussian prior distribution is assumed for the parameter vector. So, several joint realizations of $\bar{\boldsymbol{\theta}}^{(1)}, \bar{\boldsymbol{\theta}}^{(2)}, \bar{\boldsymbol{\theta}}^{(3)} = \boldsymbol{\theta}^{(3)}$ and $\bar{\mathbf{N}} \in \mathcal{T}$ are generated by drawing their elements from the standard Gaussian distribution, and then the AWGN tensor is obtained as $\mathbf{N} = \sigma \bar{\mathbf{N}}$, with σ varying to simulate different SNR conditions.

Given one realization $\mathcal{Y} = \mathcal{X} + \mathbf{N}$, we apply the following estimators:

1. **AAS**: The estimate $\hat{\boldsymbol{\eta}}$ is computed by solving (4.24). To reduce degradation due to noise, we employ three different *ad-hoc* procedures like those shown in Example 4.16 and keep the solution which yields the lowest quadratic error with respect to \mathcal{Y} . As some imaginary residual is generally present in $\hat{\boldsymbol{\eta}}$, we take its real part.
2. **N_i -CALs**: Corresponds to Algorithm 4.1 with a multi-initialization scheme. It consists in running CALs N_i times with different random initializations and keeping the solution yielding the lowest quadratic error with respect to \mathcal{Y} . As stopping criteria, we employ (3.20) with $\epsilon = 10^{-10}$ and a maximum number of iterations $K_{\max} = 2000$.
3. **N_i -MCALS**: The mixed CALs scheme of Section 5.1, here referred to as mixed constrained alternating least squares (MCALS), is used with the same stopping criteria as N_i -CALs. At most 10% of the total of K_{\max} iterates are allowed to be approximated. This proportion can be even smaller, if the stopping criteria (3.20) is met by the approximate iterates, in which case the algorithm immediately switches to exact iterates. The same multi-initialization strategy as that of N_i -CALs is employed.
4. **SBS**: The approach of Section 3.3.3 is applied to jointly compute circulant factors $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$, as well as an unstructured first estimate of the third factor, $\mathbf{A}^{(3)}$. Then, $\boldsymbol{\theta}^{(3)}$ is estimated from $\mathbf{A}^{(3)}$ in the least-squares sense, by resorting to the update equation (4.10) of approximate CALs (applied only once).
5. **AAS-CALs**: The estimates given by AAS are taken (after normalization) as initial points for CALs.

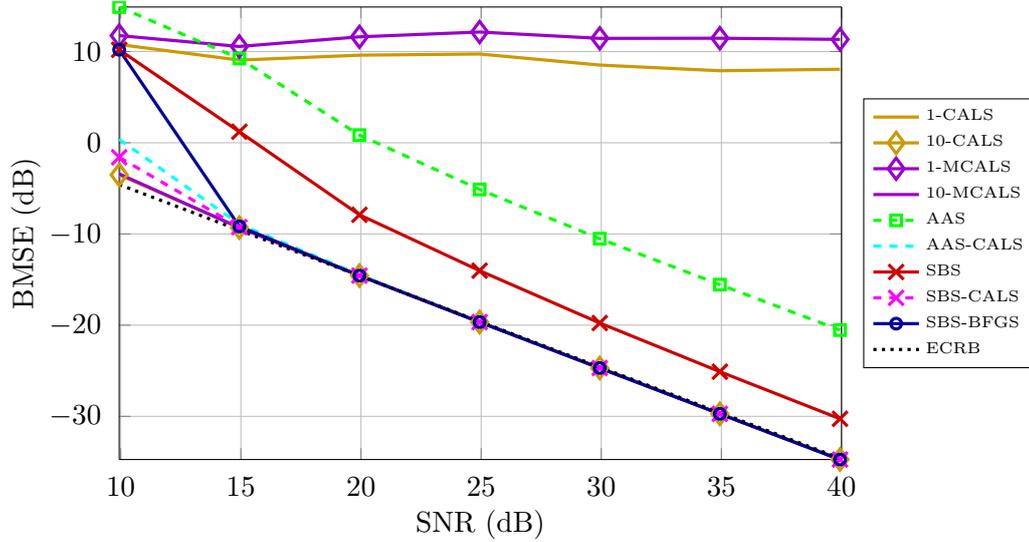


Figure 5.7: Scenario 1 of Section 5.2.3: estimated BMSE of several SCPD estimators and corresponding ECRB.

6. **SBS-CALS**: After obtaining (normalized) initial estimates with SBS, the CALS algorithm is used for refining them.
7. **SBS-BFGS**: Instead of using CALS to refine the estimates given by SBS, the quasi-Newton algorithm BFGS [27] is employed. The stopping criterion in this case checks whether the norm of the gradient is below a given tolerance level. As recommended in [219], the tolerance should be about as large as the square root of machine precision. We thus set it to 10^{-8} . As a second stopping criterion, we set $K_{\max} = 2000$.

Note that, as no normalization is imposed in AAS and SBS, the parameter vectors must be normalized *a posteriori*, by dividing $\boldsymbol{\theta}^{(p)}$ by $\theta_1^{(p)}$ for $p \in \{1, 2\}$ and absorbing these scaling factors in $\boldsymbol{\theta}^{(3)}$. In CALS and MCALS, normalization is imposed during the iterations in a similar fashion.

We have also performed simulations including an AAS-BFGS estimator, but its results, unlike those of SBS-BFGS, were generally not satisfactory. This is owed to the fact that AAS often provides starting points which are not sufficiently precise in order to initialize a (quasi-)Newton algorithm. These results are therefore not shown in the following.

In Fig. 5.7, we show the BMSE of each estimator for $N_r = 500$ realizations of \mathcal{Y} , as well as the estimated ECRB, for multiple SNR levels (in dB), which is defined here as

$$\text{SNR} = 10 \log_{10} \frac{\frac{1}{N_r} \sum_{n_r=1}^{N_r} \|\boldsymbol{x}_{n_r}\|_F^2}{\sigma^2 N^3}, \quad (5.6)$$

where \boldsymbol{x}_{n_r} stands for the n_r th realization of \boldsymbol{x} . The estimated ECRB corresponds to the sample average of the CRBs of all realizations, which are computed by applying formula

	N_i -CALS	N_i -MCALS	AAS		SBS		
SNR	$N_i = 10$	$N_i = 10$	AAS	AAS-CALS	SBS	SBS-CALS	SBS-BFGS
9.9	$1.65 \cdot 10^2$	$1.32 \cdot 10^2$	$1.53 \cdot 10^{-3}$	$1.59 \cdot 10^{-2}$	$4.32 \cdot 10^{-3}$	$1.88 \cdot 10^{-2}$	$1.04 \cdot 10^{-1}$
14.9	$1.47 \cdot 10^2$	$1.19 \cdot 10^2$	$1.30 \cdot 10^{-3}$	$1.33 \cdot 10^{-2}$	$4.00 \cdot 10^{-3}$	$1.60 \cdot 10^{-2}$	$7.38 \cdot 10^{-2}$
19.9	$1.43 \cdot 10^2$	$1.15 \cdot 10^2$	$1.28 \cdot 10^{-3}$	$1.21 \cdot 10^{-2}$	$3.96 \cdot 10^{-3}$	$1.47 \cdot 10^{-2}$	$6.91 \cdot 10^{-2}$
24.9	$1.40 \cdot 10^2$	$1.08 \cdot 10^2$	$1.28 \cdot 10^{-3}$	$1.13 \cdot 10^{-2}$	$3.91 \cdot 10^{-3}$	$1.38 \cdot 10^{-2}$	$6.37 \cdot 10^{-2}$
29.9	$1.40 \cdot 10^2$	$1.07 \cdot 10^2$	$1.27 \cdot 10^{-3}$	$1.07 \cdot 10^{-2}$	$3.92 \cdot 10^{-3}$	$1.31 \cdot 10^{-2}$	$6.12 \cdot 10^{-2}$
34.9	$1.40 \cdot 10^2$	$1.05 \cdot 10^2$	$1.27 \cdot 10^{-3}$	$1.02 \cdot 10^{-2}$	$3.89 \cdot 10^{-3}$	$1.23 \cdot 10^{-2}$	$5.77 \cdot 10^{-2}$
39.9	$1.42 \cdot 10^2$	$1.03 \cdot 10^2$	$1.27 \cdot 10^{-3}$	$9.56 \cdot 10^{-3}$	$3.92 \cdot 10^{-3}$	$1.18 \cdot 10^{-2}$	$5.51 \cdot 10^{-2}$

Table 5.3: Average computing time (in seconds) measured in Scenario 1 of Section 5.2.3.

(3.52) to each parameter. Similarly, the estimate of **BMSE** for each method is given by the sample average of the computed **MSEs**. The average computing time spent by each method for each **SNR** level is shown in Table 5.3. We omit, though, the computing times of **1-CALS** and **1-MCALs**, due to their unsatisfactory estimation performance.

It can be seen that both **CALS** and **MCALS** perform quite poorly with a single random initialization, due to frequent early termination or inability to converge, while all other iterative estimators approximately reach the **ECRB** for $\text{SNR} \geq 15$ dB. With regard to the non-iterative ones, **SBS** performs better than **AAS**, thanks to its better numerical properties. By inspecting also Table 5.3, we conclude that **AAS-CALS** and **SBS-CALS** provide the best compromise between precision and computing effort, with a computational advantage for the former. Another relevant observation is that **10-MCALs** is slightly less costly than **10-CALS**, while their **BMSE** curves are indistinguishable in Fig. 5.7.

5.2.3.2 Scenario 2: factors with different structures

Let us consider now the **SCPD** $\mathbf{X} = \llbracket \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)} \rrbracket$, where $\mathbf{A}^{(1)}, \mathbf{A}^{(2)} \in \mathbb{R}^{5 \times 4}$ are banded Toeplitz matrices having the forms

$$\mathbf{A}^{(1)} = \begin{bmatrix} \theta_1^{(1)} & 0 & 0 & 0 \\ \theta_2^{(1)} & \theta_1^{(1)} & 0 & 0 \\ \theta_3^{(1)} & \theta_2^{(1)} & \theta_1^{(1)} & 0 \\ \theta_4^{(1)} & \theta_3^{(1)} & \theta_2^{(1)} & \theta_1^{(1)} \\ 0 & \theta_4^{(1)} & \theta_3^{(1)} & \theta_2^{(1)} \end{bmatrix}, \quad \mathbf{A}^{(2)} = \begin{bmatrix} \theta_1^{(2)} & \theta_6^{(2)} & 0 & 0 \\ \theta_2^{(2)} & \theta_1^{(2)} & \theta_6^{(2)} & 0 \\ \theta_3^{(2)} & \theta_2^{(2)} & \theta_1^{(2)} & \theta_6^{(2)} \\ \theta_4^{(2)} & \theta_3^{(2)} & \theta_2^{(2)} & \theta_1^{(2)} \\ \theta_5^{(2)} & \theta_4^{(2)} & \theta_3^{(2)} & \theta_2^{(2)} \end{bmatrix} \quad (5.7)$$

and $\mathbf{A}^{(3)} \in \mathbb{R}^{4 \times 4}$ is unstructured. Note that the set of basis matrices associated with each one of $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ is simply a subset of the Toeplitz basis described by Table 3.1. Again, we fix $\theta_1^{(1)} = \theta_1^{(2)} = 1$ in order to eliminate the scaling ambiguity. For the other parameter vector components, a uniform distribution over the interval $[-1, 1]$ is chosen as prior distribution.

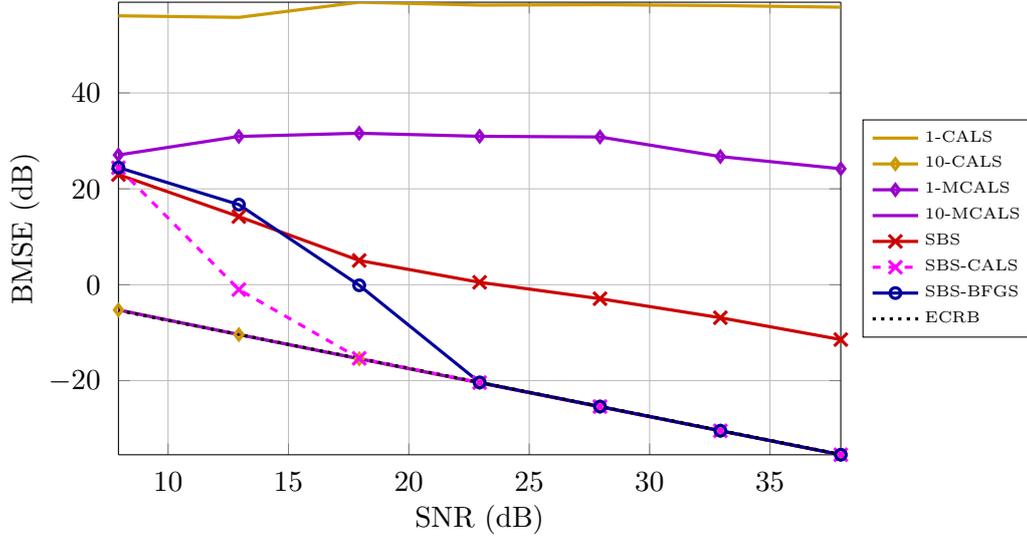


Figure 5.8: Scenario 2 of Section 5.2.3: estimated BMSE of several SCPD estimators and corresponding ECRB.

The experimental procedure here employed is exactly akin to that of considered in Section 5.2.3.1, and we also evaluate the same estimators, except for AAS and its combinations with iterative algorithms, which do not apply.

Fig. 5.8 and Table 5.4 display the obtained results. Just as in Scenario 1, 1-CALS and 1-MCALs do not deliver accurate estimates in average, for the same reasons. With 10 random initializations, their performances are greatly enhanced, corresponding to the best ones at SNR levels below 17 dB. With respect to computing time, 10-MCALs is again less costly than 10-CALS and this time by a larger margin, which indicates that the use of approximate iterates brings important computational savings in this setting. When initialized by SBS, CALS is able to reach quite close to the ECRB for $\text{SNR} \geq 18$ dB, whereas the same is true for BFGS only after about 23 dB. So, for a sufficiently high SNR level, SBS-CALS provides the best compromise between cost and precision, as it is about 4 orders of magnitude less costly than 10-MCALs and almost one order of magnitude less costly than SBS-BFGS.

5.2.4 Symmetric SCPD

We consider here one scenario involving a completely symmetric SCPD. In this scenario, \mathcal{X} admits a SCPD of the form $\mathcal{X} = \llbracket \mathbf{C}, \mathbf{C}, \mathbf{C} \rrbracket$, with $\mathbf{C} \in \mathbb{R}^{4 \times 3}$ circulant. The generation of model realizations is similar to that of Section 5.2.3.1, but now there is a single $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta} = \boldsymbol{\eta}$ (note that no component θ_u is fixed here, because there is no scaling ambiguity to suppress). Also, the noise tensor is symmetric (cf. Section 3.4.4.2), being generated as follows: the elements $[\mathcal{N}]_{n_1, n_2, n_3}$ such that $n_1 \leq n_2 \leq n_3$ are drawn from a zero-mean Gaussian distribution of variance σ^2 , while all the others are determined by symmetry.

	N_i -CALs	N_i -MCALS	SBS		
SNR	$N_i = 10$	$N_i = 10$	SBS	SBS-CALS	SBS-BFGS
7.9	$2.29 \cdot 10^3$	$3.98 \cdot 10^2$	$6.06 \cdot 10^{-3}$	$5.66 \cdot 10^{-2}$	$8.11 \cdot 10^{-1}$
12.9	$1.96 \cdot 10^3$	$3.87 \cdot 10^2$	$6.07 \cdot 10^{-3}$	$2.68 \cdot 10^{-2}$	$2.62 \cdot 10^{-1}$
17.9	$1.97 \cdot 10^3$	$3.51 \cdot 10^2$	$6.09 \cdot 10^{-3}$	$2.46 \cdot 10^{-2}$	$1.12 \cdot 10^{-1}$
22.9	$1.95 \cdot 10^3$	$3.45 \cdot 10^2$	$6.06 \cdot 10^{-3}$	$2.27 \cdot 10^{-2}$	$9.70 \cdot 10^{-2}$
27.9	$1.95 \cdot 10^3$	$3.06 \cdot 10^2$	$6.03 \cdot 10^{-3}$	$2.16 \cdot 10^{-2}$	$9.16 \cdot 10^{-2}$
32.9	$1.89 \cdot 10^3$	$2.97 \cdot 10^2$	$6.09 \cdot 10^{-3}$	$2.07 \cdot 10^{-2}$	$8.60 \cdot 10^{-2}$
37.9	$1.87 \cdot 10^3$	$3.04 \cdot 10^2$	$6.02 \cdot 10^{-3}$	$1.93 \cdot 10^{-2}$	$7.98 \cdot 10^{-2}$

Table 5.4: Average computing time (in seconds) measured in Scenario 2 of Section 5.2.3.

We describe below how each algorithm was applied to estimate \mathbf{C} .

1. **AAS**: Factors are computed by solving three disjoint subset of equations such as that of Example 4.21 and keeping the best solution. As the parameter vector $\boldsymbol{\theta}$ can only be estimated up to a complex scaling factor of the form $e^{j\frac{2\pi}{3}p}$, with $p \in \{0, 1, 2\}$ (as discussed in Section 4.2.3 and Example 4.21), it is necessary to suppress it by taking into account the fact that $\boldsymbol{\theta}$ is real. This is done by computing

$$\hat{p} = \operatorname{argmin}_{p \in \{0,1,2\}} \sum_{u=1}^U \min \left\{ \left[\operatorname{Arg}(\tilde{\theta}_u) - \operatorname{Arg} \left(e^{j\frac{2\pi}{3}p} \right) \right]^2, \left[\operatorname{Arg}(\tilde{\theta}_u) - \operatorname{Arg} \left(e^{j\left(\frac{2\pi}{3}p - \pi\right)} \right) \right]^2 \right\},$$

where $\operatorname{Arg} : \mathbb{C} \mapsto [-\pi, \pi[$ outputs the phase of its argument, and then estimating the generating vector as $\hat{\boldsymbol{\theta}} = \Re \left\{ e^{-j\frac{2\pi}{3}\hat{p}} \tilde{\boldsymbol{\theta}} \right\}$, with $\tilde{\boldsymbol{\theta}}$ denoting the output of AAS. This amounts to estimating the phase of the undesired complex scaling factor in the least-squares sense. Each term of the above sum is given by the minimum between two expressions because there are two possibilities for the phase of θ_u : it is either null (if $\theta_u \geq 0$) or it equals π (when $\theta_u < 0$).

2. **N_i -SCALS**: The SCALS algorithm described in Section 4.1.3 is applied. A multi-initialization scheme is again used with N_i random initial points.
3. **N_i -MSCALS**: Just like in the non-symmetric setting, we apply SCALS with approximate and then exact iterates, which is referred to as mixed symmetric constrained alternating least squares (MSCALS), and N_i different initializations.
4. **SBS**: As this method does not take symmetry into account, this is done *a posteriori*, by averaging all obtained factors, which are estimated exactly as described in Section 5.2.3.

We also evaluate the combined approaches 4) AAS-SCALS, 5) SBS-SCALS and 6) SBS-BFGS.

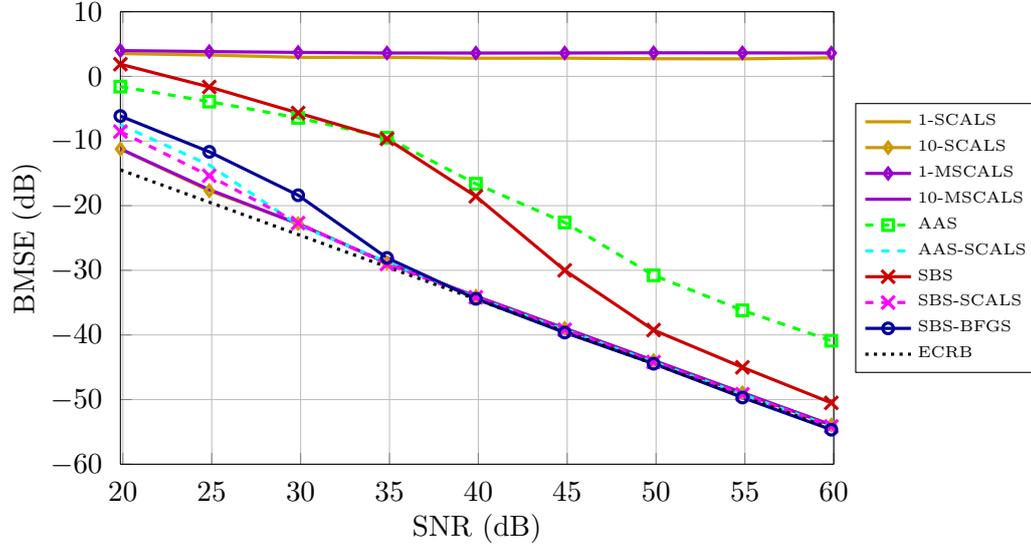


Figure 5.9: Symmetric setting of Section 5.2.4: estimated BMSE of several SCPD estimators and corresponding ECRB.

SNR	N_i -CALs	N_i -MSCALS	AAS		SBS		
	$N_i = 10$	$N_i = 10$	AAS	AAS-SCALS	SBS	SBS-SCALS	SBS-BFGS
19.9	$1.20 \cdot 10^3$	$8.40 \cdot 10^2$	$1.17 \cdot 10^{-3}$	$2.18 \cdot 10^{-2}$	$4.21 \cdot 10^{-3}$	$2.99 \cdot 10^{-2}$	$4.79 \cdot 10^{-2}$
29.9	$8.39 \cdot 10^2$	$4.99 \cdot 10^2$	$1.15 \cdot 10^{-3}$	$9.12 \cdot 10^{-3}$	$4.18 \cdot 10^{-3}$	$1.15 \cdot 10^{-2}$	$4.06 \cdot 10^{-2}$
39.9	$7.79 \cdot 10^2$	$4.52 \cdot 10^2$	$1.14 \cdot 10^{-3}$	$7.75 \cdot 10^{-3}$	$4.16 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	$3.58 \cdot 10^{-2}$
49.9	$7.06 \cdot 10^2$	$4.41 \cdot 10^2$	$1.13 \cdot 10^{-3}$	$6.72 \cdot 10^{-3}$	$4.13 \cdot 10^{-3}$	$8.82 \cdot 10^{-3}$	$3.24 \cdot 10^{-2}$
59.9	$6.95 \cdot 10^2$	$3.92 \cdot 10^2$	$1.13 \cdot 10^{-3}$	$5.78 \cdot 10^{-3}$	$4.15 \cdot 10^{-3}$	$7.82 \cdot 10^{-3}$	$3.00 \cdot 10^{-2}$

Table 5.5: Average computing time (in seconds) measured in the symmetric setting of Section 5.2.4.

The results are shown in Fig. 5.9. In comparison with the results of Section 5.2.3.1, we can see that the algorithms perform in general worse. In the case of SCALS, this is due to the imposition of symmetry. For both AAS and SBS, an additional stage is employed (in AAS, for fixing the scaling factor; in SBS, for computing a single factor estimate), which degrades performance. 1-SCALS and 1-MSICALS yield again quite unsatisfactory results, while all other iterative algorithms get quite close to the ECRB for $\text{SNR} \geq 35$ dB. Inspecting the average computing times on Table 5.5, one can conclude that AAS-SCALS and SBS-SCALS lead to the best compromise between statistical efficiency and computing cost.

5.2.5 Wiener-Hammerstein model identification

We now evaluate SCPD estimators when applied to identify the linear filters of Wiener-Hammerstein systems from their third-order Volterra kernels. Recall from Section 3.2 that

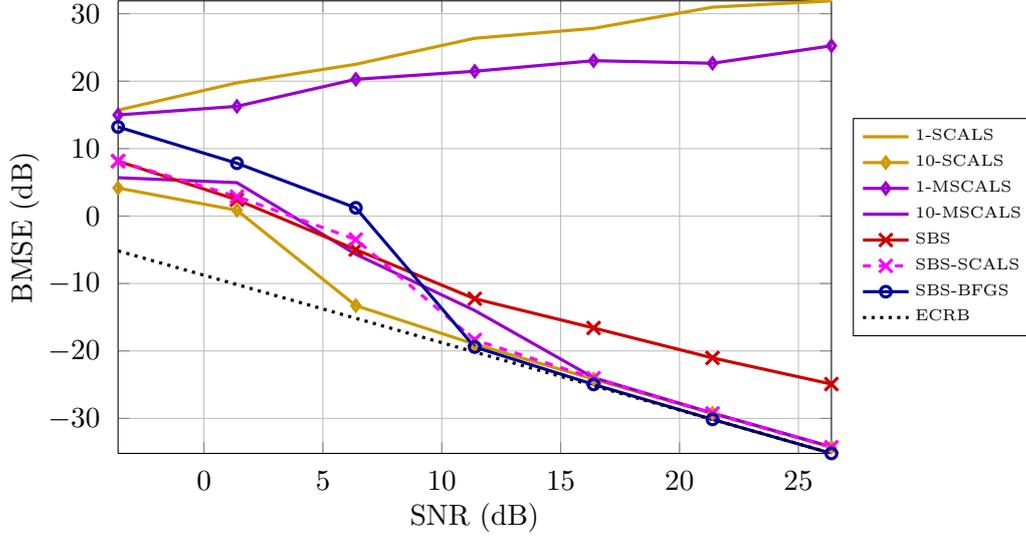


Figure 5.10: Wiener-Hammerstein identification scenario of Section 5.2.5: estimated BMSE of several SCPD estimators and corresponding ECRB.

this kernel admits a SCPD of the form $\mathbf{V}^{(p)} = \llbracket \boldsymbol{\lambda} ; \mathbf{A}, \mathbf{A}, \mathbf{A} \rrbracket$, where $\mathbf{A} \in \mathbb{R}^{N \times R}$ is a banded circulant matrix characterized by $U = N - R + 1$ parameters, which correspond to the coefficients of the linear filter $W(z)$ of Fig. 3.1. We set $U = 5$ and $R = 3$; thus, $N = 7$. To perform this experiment, $N_r = 500$ realizations of the parameters $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{\lambda}$ are generated by drawing each component $\bar{\theta}_u$ and λ_r uniformly over $[-1, 1]$. To ensure local identifiability, $\theta_1 = 1$ is imposed.

We perform the described procedure for a third-order tensor $\mathbf{Y} \in \mathcal{S} \left((\mathbb{R}^7)^{\otimes 3} \right)$, with $\mathbf{X} = \mathbf{V}^{(3)}$ built from the exact Volterra kernel $v^{(3)}$ generated as (3.10). The kernel estimation error is modeled by the (symmetric) noise tensor \mathcal{N} , which is generated exactly as in the previous scenario. The variance σ^2 of \mathcal{N} is again varied for simulating different SNR conditions.

The employed estimators are:

1. N_i -SCALS: SCALS is specialized to the particular structure of the Volterra kernel (3.12). This is done by estimating a single factor \mathbf{A} per iteration. A multi-initialization scheme with N_i initializations is also used, for improving performance.
2. SBS: The procedure is similar to that of Section 5.2.4, with the estimate of \mathbf{A} being obtained by averaging the two structured factors estimated by the algorithm. After that, it is suitably normalized, and then $\boldsymbol{\lambda}$ is estimated by employing (4.8).

Again, we apply SCALS and BFGS to refine the SBS solution.

The BMSE estimated at several SNR levels is shown in Fig. 5.10, with the corresponding time measurements reported in Table 5.6. It is seen that 1-SCALS and 1-MSCALS produce very poor results, due to the typical convergence problems encountered in practice. With 10

SNR	N_i -CALs	N_i -MSCALS	SBS		
	$N_i = 10$	$N_i = 10$	SBS	SBS-SCALS	SBS-BFGS
-3.6	$1.13 \cdot 10^3$	$3.85 \cdot 10^2$	$5.16 \cdot 10^{-3}$	$2.05 \cdot 10^{-2}$	$1.74 \cdot 10^{-1}$
1.4	$5.92 \cdot 10^2$	$1.57 \cdot 10^2$	$4.88 \cdot 10^{-3}$	$1.33 \cdot 10^{-2}$	$9.56 \cdot 10^{-2}$
6.4	$2.98 \cdot 10^2$	$9.92 \cdot 10^1$	$4.84 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$	$6.75 \cdot 10^{-2}$
11.4	$2.31 \cdot 10^2$	$8.20 \cdot 10^1$	$4.86 \cdot 10^{-3}$	$9.88 \cdot 10^{-3}$	$5.95 \cdot 10^{-2}$
16.4	$1.53 \cdot 10^2$	$7.09 \cdot 10^1$	$4.82 \cdot 10^{-3}$	$9.32 \cdot 10^{-3}$	$5.64 \cdot 10^{-2}$
21.4	$1.51 \cdot 10^2$	$6.53 \cdot 10^1$	$4.82 \cdot 10^{-3}$	$8.94 \cdot 10^{-3}$	$5.43 \cdot 10^{-2}$
26.4	$1.46 \cdot 10^2$	$6.15 \cdot 10^1$	$4.81 \cdot 10^{-3}$	$8.64 \cdot 10^{-3}$	$5.25 \cdot 10^{-2}$

Table 5.6: Average computing time (in seconds) measured in Scenario 1 of Section 5.2.4.

random initializations, this problem is overcome (for sufficiently high SNR), as it becomes more likely that at least one run will produce good estimates; however, the total computing cost is very high. In contrast, SBS's BMSE lies within moderate distance from the ECRB, but its computing cost is quite low. Taking advantage of its estimate, both SBS-SCALS and SBS-BFGS are able to reach quite close to the ECRB for SNR values greater than approximately 11 dB, with a slight advantage for the latter. Furthermore, SBS-SCALS outperforms SBS-BFGS from a computing cost perspective, thus offering the best compromise in this scenario.

5.3 Final remarks

The main conclusions we can draw from our numerical experiments are as follows:

1. When estimating the factors of a CCPD, AAS procedures are useful in that they deliver cheap initial estimates which often allow iterative algorithms to quickly reach an accurate solution.
2. Whenever an approximate initial solution is available (given by, *e.g.*, AAS or SBS), the CALS (or SCALS) algorithm is often able to converge within a relatively small computing time, and its BMSE approaches the ECRB for sufficiently high SNR levels.
3. When a cheap initial solution is *not* available (*e.g.*, when the structure of the factors cannot be handled neither by an AAS nor by SBS), then the MCALS algorithm which employs first approximate and then exact iterates seems to be a good option, as its probability of convergence to local minima is smaller than that of CALS, as seen in Section 5.1, while it requires less overall computing effort.

Part II

Low-rank tensor recovery

From compressive sensing to low-rank tensor recovery

In this chapter, we introduce the low-rank tensor recovery problem and review the main existing approaches which were devised to address it. Before that, it is opportune to briefly discuss the compressive sensing and low-rank matrix recovery problems, as they share fundamental aspects with our studied problem. This introductory material shall then serve as a basis for subsequent chapters.

Contents

6.1	Compressive sensing	107
6.2	Low-rank matrix recovery	110
6.3	Low-rank tensor recovery	113

6.1 Compressive sensing

The pioneering works of Candès, Romberg, Tao [31, 32, 30] and Donoho [74] sparked a flurry of interest in the topic of compressive sensing (CS).¹ Their remarkable results basically establish conditions under which a sparse vector can be *exactly* recovered from a set of linear measurements having much lower cardinality than its dimension. Moreover, it was shown that this is a tractable problem, as it can be solved by means of efficient algorithms [31, 74]. Importantly from a practical viewpoint, the body of CS results also comprises performance bounds for the approximate reconstruction of an approximately sparse vector (*e.g.*, having components with exponentially decaying magnitude) from noisy measurements [34, 166]. In view of the ubiquity of (approximately) parsimonious signals in nature, these ideas find application in many different domains, such as system identification [111, 12], biomedical signal processing [161, 138], radar imaging [101, 160] and astronomy [18].

¹The terminology “compressed sensing” or “compressive sampling” is also employed.

In the following, we briefly introduce the main aspects of CS. The interested reader can find an in-depth and thorough account of CS in [85].

6.1.1 Problem statement and main involved issues

Definition 6.1 (*s*-sparsity). A vector $\mathbf{x} \in \mathbb{R}^N$ is said to be *s*-sparse if $\text{card}(\mathbf{x}) \leq s < N$.

Problem 6.2 (Compressive sensing [31, 32, 30, 74]). Let $\mathbf{x}^* \in \mathbb{R}^N$ be an unknown signal of interest and $\mathbf{y} = \mathbf{A}\mathbf{x}^*$ be a *linear measurement* of \mathbf{x}^* obtained with a known *measurement matrix* $\mathbf{A} \in \mathbb{R}^{M \times N}$ which does *not* depend on \mathbf{x}^* , and such that $M \ll N$. Assuming that \mathbf{x}^* is *s*-sparse, with $s \in \langle M \rangle$, recover \mathbf{x}^* from \mathbf{y} .

Problem 6.2 is ill-posed without further assumptions, because its underdetermined system of linear equations has infinitely many solutions (provided $\mathbf{y} \in \text{colspace}(\mathbf{A})$). Nevertheless, it turns out that the *s*-sparsity of \mathbf{x}^* can be exploited to *uniquely* recover it from \mathbf{y} , as long as \mathbf{A} satisfies certain conditions. This is useful for many practical tasks, such as, *e.g.*, efficiently sensing real-world images which admit (approximately) sparse representations in the wavelet domain [33].

There are two central aspects involved in Problem 6.2 [85]:

- (i) *Encoding*: which measurement matrices (or classes or matrices) are appropriate for measuring arbitrary *s*-sparse vectors?

Since we do not know in advance which entries of \mathbf{x}^* are nonzero, some matrices are clearly inappropriate, such as those having canonical basis vectors as rows. To answer the above question, useful quality measures have been developed for characterizing suitable measurement matrices [31, 74, 33, 47]. They serve as important tools for establishing theoretical recovery guarantees of CS algorithms.

- (ii) *Decoding*: which recovery algorithms can be employed for efficient recovery?

The suitability of an algorithm hinges on two requirements. First, it must of course be computationally tractable. Second, it should successfully recover \mathbf{x}^* whenever \mathbf{A} is suitable (in terms of appropriate quality measures). Ideally, it should also be subject to performance bounds guaranteeing a certain degree of accuracy even if \mathbf{x}^* is only approximately sparse (which is called *stable* recovery) and if the measurements are noisy (which is referred to as *robust* recovery).

These two aspects are briefly discussed in the following sections.

6.1.2 Measurement matrices and the restricted isometry property

The very goal of CS is to take as few measurements as possible whilst still enabling recovery of arbitrary *s*-sparse vectors (at least with high probability). Ideally, we would like to

reconstruct \mathbf{x}^* from $\mathcal{O}(s)$ measurements, since s is its number of degrees of freedom (DOF).

Recoverability of an arbitrary \mathbf{x}^* clearly requires that the linear transformation associated with \mathbf{A} be *injective* in the set of s -sparse vectors $\mathcal{S}_s = \{\mathbf{x} \in \mathbb{R}^N : \text{card}(\mathbf{x}) \leq s\}$, that is, $\mathbf{A}\mathbf{v} - \mathbf{A}\mathbf{w} = \mathbf{A}(\mathbf{v} - \mathbf{w}) = \mathbf{0}$ must imply $\mathbf{v} = \mathbf{w}$ for all $\mathbf{v}, \mathbf{w} \in \mathcal{S}_s$. Since $\mathbf{v} - \mathbf{w}$ is $2s$ -sparse, this amounts to imposing $\text{nullspace}(\mathbf{A}) \cap \mathcal{S}_{2s} = \{\mathbf{0}\}$, which requires $\text{krank}(\mathbf{A}) \geq 2s$ [85, Theorem 2.13]. Hence, from Definition 2.40, it is clear that $M \geq 2s$ is a necessary condition. In principle, injectivity of \mathbf{A} on \mathcal{S}_s implies recovery can be achieved by solving

$$\min_{\mathbf{x} \in \mathbb{R}^N} \text{card}(\mathbf{x}) \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (6.1)$$

Yet, problem (6.1) is NP-hard in general [31, 85], requiring an exhaustive search among all possible submatrices of \mathbf{A} having s columns.

Hence, injectivity of \mathbf{A} on \mathcal{S}_s is necessary but not sufficient for establishing recovery guarantees of practical algorithms. To achieve this goal, other properties of measurement matrices have been exploited, such as coherence, the nullspace property and the restricted isometry property (RIP) [31, 33, 47]. The latter, in particular, stands on the rationale that an appropriate measurement matrix should approximately behave as an orthonormal system over \mathcal{S}_{2s} , *i.e.*, be such that $\|\mathbf{A}(\mathbf{v} - \mathbf{w})\|_2 \approx \|\mathbf{v} - \mathbf{w}\|_2$ for any $\mathbf{v}, \mathbf{w} \in \mathcal{S}_s$ [31]. In other words, \mathbf{A} must nearly preserve distances between vectors of \mathcal{S}_s , hence the expression “restricted isometry”. This idea was formalized in [31] as follows.

Definition 6.3 (Restricted isometry constants). Let $\mathbf{A} \in \mathbb{R}^{M \times N}$. The s -restricted isometry constant (RIC) of \mathbf{A} is the smallest positive real number δ_s such that

$$\forall \mathbf{x} \in \mathcal{S}_s, \quad (1 - \delta_s)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_s)\|\mathbf{x}\|_2^2. \quad (6.2)$$

If $\delta_s < 1$, then \mathbf{A} is said to have the RIP of order s .

Definition 6.3 allows deriving recovery guarantees (including stable and robust recovery) for many CS algorithms. However, the explicit construction of deterministic matrices having sufficiently small RICs at an optimal number of measurements is to date an open problem. A major breakthrough in CS theory is the discovery that certain random matrices (*e.g.*, having Gaussian or Rademacher elements) have small RICs with overwhelming probability as long as $M \geq \mathcal{O}(s \log(eN/s))$. This bound is in fact optimal, in the sense that it matches the number of measurements which is necessary for stable recovery [85, Theorem 11.7].

In applications, though, unstructured random matrices are often unsuitable due to physical constraints and also because the lack of structure prevents implementation of fast matrix-vector products. Fortunately, it has been shown that certain structured random matrices providing a number of measurements that scales nearly linearly in s , up to a logarithmic factor (larger than $\log(eN/s)$), also possess small enough RICs. This applies, in particular, to partial Fourier and partial noiselet measurements [166, 172], which can be efficiently computed by means of $\mathcal{O}(N \log(N))$ fast transform algorithms.

6.1.3 Recovery via ℓ_1 norm minimization

A standard CS approach consists in reformulating problem (6.1) by replacing the cardinality function with the ℓ_1 norm, which yields the basis pursuit problem:

$$\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (6.3)$$

Efficient algorithms exist for the above problem, which can be regarded as a convex relaxation of (6.1). Thus, contrarily to (6.1), every local minimum of (6.3) is a global one. In spite of such a relaxation, (robust and stable) recovery guarantees have been derived (see, *e.g.*, [34]). Therefore, under certain conditions on \mathbf{A} , (6.1) has the same solution as (6.3).

Variants of (6.3) have been formulated, such as the (convex) unconstrained penalized formulation known as basis pursuit denoising, written as

$$\min_{\mathbf{x} \in \mathbb{R}^N} \lambda \|\mathbf{x}\|_1 + \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2, \quad (6.4)$$

where $\lambda > 0$ is a regularization parameter controlling the compromise between the accuracy with which measurements are matched and the degree of parsimony of \mathbf{x} . Note that (6.4) is more reasonable than (6.3) when \mathbf{y} is noisy. A global minimum of (6.4) can be found by, *e.g.*, the alternating direction method of multipliers (ADMM) [22] or proximal methods [48].

Many other approaches exist, but are beyond the scope of this thesis. See [85] for a comprehensive review.

6.2 Low-rank matrix recovery

Motivated by the occurrence of matrices having (approximately) low rank in a myriad of applications, the pioneer work of Fazel [82] considered the problem of low-rank matrix recovery (LRMR) from a set of underdetermined linear measurements. She devised the nuclear norm minimization heuristic, which is very similar in spirit to the standard CS technique of ℓ_1 norm minimization.

Interestingly, the matrix rank can be thought of as an extension of cardinality to second-order tensors [169]. This connection is explained by the SVD: the cardinality of the diagonal matrix holding the singular values of a matrix is, by definition, its rank. In fact, both CS and LRMR are instances of the general problem of recovering a “parsimonious” element of a given set from a small number of linear measurements. This explains why some LRMR approaches are directly inspired by CS techniques.

6.2.1 Problem statement

Similarly to the CS setting, we consider here the set of matrices having rank bounded by R , defined as $\mathcal{L}_R \triangleq \{\mathbf{X} \in \mathbb{R}^{N_1 \times N_2} : \text{rank}(\mathbf{X}) \leq R\}$.

Problem 6.4 (Low-rank matrix recovery). Let $\mathbf{X}^* \in \mathbb{R}^{N_1 \times N_2}$ be some (unknown) matrix of interest and $\mathcal{A} : \mathbb{R}^{N_1 \times N_2} \mapsto \mathbb{R}^M$ be a linear measurement operator (MO), with $M < N_1 N_2$. Given $\mathbf{y} = \mathcal{A}(\mathbf{X}^*)$ and assuming $\mathbf{X}^* \in \mathcal{L}_R$, recover \mathbf{X}^* from \mathbf{y} .

The affine rank minimization problem considered in [82] has the form

$$\min_{\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}} \text{rank}(\mathbf{X}) \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}. \quad (6.5)$$

A very important particular case of LRMR is the matrix completion (MC) problem [28], where only a portion of the elements of a matrix is observed and the goal is to correctly infer the missing entries under a low-rank assumption. A well-known instance of this setting in the area of recommender systems is involved in the so-called ‘‘Netflix prize’’ [11]. Formally, in MC the operator \mathcal{A} assumes a particular form: each component y_m of $\mathbf{y} = \mathcal{A}(\mathbf{X}^*)$ satisfies

$$y_m = x_{n_1(m), n_2(m)}^* = \left\langle \mathbf{X}^*, \mathbf{e}_{n_1(m)}^{N_1} \otimes \mathbf{e}_{n_2(m)}^{N_2} \right\rangle = \mathbf{X}^* \times_1 \left(\mathbf{e}_{n_1(m)}^{N_1} \right)^T \times_2 \left(\mathbf{e}_{n_2(m)}^{N_2} \right)^T, \quad (6.6)$$

where \mathbf{e}_n^N denotes the n th canonical basis vector of \mathbb{R}^N . Such an \mathcal{A} is called a sampling operator (SO).

6.2.2 Suitable measurement operators

Just as in CS, we need useful ways of quantifying the suitability of MOs and we want to know how many measurements must be taken to ensure recovery. Recall that the SVD of a rank- R matrix is characterized by $R(N_1 + N_2 - R)$ DOF, which is precisely the dimension of the manifold of rank- R matrices [208]. It follows then that a necessary condition is $M \geq R(N_1 + N_2 - R)$ [35]. We thus hope to be capable of reconstructing arbitrary rank- R matrices from $\mathcal{O}(R(N_1 + N_2 - R))$ measurements with efficient algorithms.

In contrast to CS, where the best sampling bounds involve a logarithmic term, this hope is fulfilled in LRMR. For Gaussian measurements, this has been shown in [40, Proposition 3.11] by exploiting the convex geometry of the nuclear norm ball. Another way of deriving sampling bounds is by relying on an extension of the RIP to LRMR operators. The idea is defining the R -RIC of \mathcal{A} as the smallest positive real number δ_R such that

$$\forall \mathbf{X} \in \mathcal{L}_R, \quad (1 - \delta_R) \|\mathbf{X}\|_F^2 \leq \|\mathcal{A}(\mathbf{X})\|_2^2 \leq (1 + \delta_R) \|\mathbf{X}\|_F^2. \quad (6.7)$$

The authors of [35] demonstrate that certain random MOs (such as Gaussian ones) possess the matrix RIP for $M \geq cR \max\{N_1, N_2\}$, where c is a positive constant.

When it comes to MC, however, RIC-based results do not apply, due to the special form of \mathcal{A} . It is not hard to show that no SO can possess the RIP unless almost every entry is sampled. For instance, if $\mathbf{X}^* = [\mathbf{e}_1^{N_1} \ \dots \ \mathbf{e}_R^{N_1} \ \mathbf{0}_{N_1 \times N_2 - R}] \in \mathcal{L}_R$, then there exist SOs with $M \leq N_1 N_2 - R$ such that $\mathcal{A}(\mathbf{X}^*) = \mathbf{0}$, which thus cannot meet (6.7) for $\delta_R < 1$.

The reason of failure in the above example is the sparsity of \mathbf{X}^* , which makes a random sampling “miss” the relevant entries most of the time. This has motivated the formulation of the so-called *incoherence* conditions [28], which essentially state that vectors from the row and column spaces of the target low-rank matrix must be sufficiently uncorrelated with canonical basis vectors, *i.e.*, their energy must be sufficiently spread. This can be formalized by requiring that $\mu_0(\mathbf{X}^*) \triangleq \|\text{vec}(\mathbf{U}\mathbf{V}^T)\|_\infty$ be sufficiently small, where $\mathbf{X}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of \mathbf{X}^* , and also that the column and row spaces of \mathbf{X}^* have a small coherence measure, defined as follows.

Definition 6.5 (Coherence with respect to canonical basis [28]). Let \mathcal{U} be an R -dimensional subspace of \mathbb{R}^N . Then, the coherence of \mathcal{U} with respect to the canonical basis is defined as

$$\mu(\mathcal{U}) \triangleq \frac{N}{R} \max_{n \in \langle N \rangle} \|\mathcal{P}_{\mathcal{U}}(\mathbf{e}_n^N)\|_2^2, \quad (6.8)$$

where $\mathcal{P}_{\mathcal{U}}$ denotes the orthogonal projector onto \mathcal{U} .

Often, one assumes that the set of locations of the M sampled entries is uniformly drawn at random from $\{\mathcal{I} \in 2^{\langle N_1 \rangle \times \langle N_2 \rangle} : |\mathcal{I}| = M\}$. This is a reasonable model in applications where the occurrence of missing data follows no particular structure, but may be inadequate if it is more likely to be concentrated (*e.g.*, in space or time). In [29], the minimum value of M needed to guarantee reconstruction is established for this uniform sampling model, no matter which recovery algorithm is used. It is given by the number of DOF of the low-rank model, times a measure of coherence of the column and row spaces of \mathbf{X}^* , times a logarithmic factor in $\max\{N_1, N_2\}$. This logarithmic factor is due to the coupon-collector effect,² since each row and column has to be sampled at least once to enable reconstruction [28].

6.2.3 Recovery via nuclear norm minimization

A popular LRMR approach is that of nuclear norm minimization (NNM). As previously commented, a parallel can be drawn with CS, because $\|\mathbf{X}\|_* = \|\boldsymbol{\sigma}\|_1$ while $\text{rank}(\mathbf{X}) = \text{card}(\boldsymbol{\sigma})$, where $\|\cdot\|_*$ denotes the nuclear norm and $\boldsymbol{\sigma} \in \mathbb{R}^{\min\{N_1, N_2\}}$ contains the singular values of $\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}$. Formally, one poses the problem

$$\min_{\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}} \|\mathbf{X}\|_* \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}. \quad (6.9)$$

or variants such as $\min_{\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}} \lambda \|\mathbf{X}\|_* + \|\mathcal{A}(\mathbf{X}) - \mathbf{y}\|_2^2$ [35].

This approach can be well justified by geometric arguments [40]. Also, it can be seen as a convex relaxation of (the NP-hard) problem (6.5) where the rank function is replaced by its

²The terminology comes from the coupon collector’s problem in probability theory: how many draws (with replacement) are needed in average to collect at least once each one of the n different coupons placed inside an urn? This number is proportional to $n \log(n)$.

tightest convex relaxation over the unit spectral norm ball. This resulting convex problem can be efficiently tackled by means of semidefinite programming techniques or by **ADMM**. We refer the reader to [28, 35, 29, 169] for recovery results of **NNM**, which show it is capable of perfect recovery at an order-wise optimal number of measurements: $\mathcal{O}(R(N_1 + N_2 - R))$ in **LRMR** and $\mathcal{O}(\mu R(N_1 + N_2 - R) \log(\max\{N_1, N_2\}))$ in **MC** (with uniform sampling), where μ depends on $\mu_0(\mathbf{X}^*)$, $\mu(\mathcal{U})$ and $\mu(\mathcal{V})$, with $\mathcal{U} = \text{colspace}(\mathbf{X}^*)$ and $\mathcal{V} = \text{rowspace}(\mathbf{X}^*)$.

As in the case of **CS**, several other **LRMR** approaches exist but we shall not delve further into this topic.

6.3 Low-rank tensor recovery

When dealing with the recovery of tensors, rank is again a natural and relevant complexity measure. However, in passing from matrices to higher-order tensors, the notion of rank gains multiple non-equivalent meanings, as mentioned in **Chapter 2**. Correspondingly, multiple ways of approaching the low-rank tensor recovery (**LRTR**) problem exist. In the following, we state mathematical formulations for two of them.

6.3.1 Problem statement

Problem 6.6 (Low-rank tensor recovery). Let $\mathbf{X}^* \in \mathcal{T} \triangleq \bigotimes_{p=1}^P \mathbb{R}^{N_p}$ be some (unknown) tensor of interest and let $\mathcal{A} : \mathcal{T} \mapsto \mathbb{R}^M$ be a linear **MO**, with $M < \bar{N}$. Given $\mathbf{y} = \mathcal{A}(\mathbf{X}^*)$ and assuming \mathbf{X}^* belongs to a set of parsimonious tensors $\mathcal{S} \subset \mathcal{T}$, recover \mathbf{X}^* from \mathbf{y} .

As in the matrix setting, an important particularization of **Problem 6.6** is when \mathcal{A} is an **SO**, which is called the tensor completion (**TC**) problem. In this case, we have

$$y_m = x_{n_1(m), \dots, n_P(m)}^* = \mathbf{X}^* \times_{p=1}^P \left(\mathbf{e}_{n_p(m)}^{N_p} \right)^T, \quad (6.10)$$

which is a generalization of (6.6). The goal is thus reconstructing a tensor having missing entries. In this context, we shall employ the definition

$$\Omega \triangleq \{(n_1, \dots, n_P) \in \langle N_1 \rangle \times \dots \times \langle N_P \rangle : \exists m \in \langle M \rangle \text{ such that } \forall p \in \langle P \rangle, n_p = n_p(m)\},$$

i.e., Ω is the set of multi-indices whose corresponding entries are sampled. Associated with it is the orthogonal projection $(\cdot)_\Omega : \mathcal{T} \mapsto \mathcal{T}_\Omega$, where

$$\mathcal{T}_\Omega \triangleq \{\mathbf{X} \in \mathcal{T} : [\mathbf{X}]_{n_1, \dots, n_P} \neq 0 \text{ only if } (n_1, \dots, n_P) \in \Omega\}. \quad (6.11)$$

In particular, it is easy to show that for any tensor $\mathbf{X} \in \mathcal{T}$,

$$\left(\mathcal{A}^\dagger \mathcal{A} \right) (\mathbf{X}) = (\mathbf{X})_\Omega, \quad (6.12)$$

where \mathcal{A}^\dagger is the adjoint³ of \mathcal{A}

Essentially, in LRTR one wishes to exploit some joint low dimensionality along multiple modes of the tensor of interest in order to reconstruct it from a few measurements. We discuss in the sequel two possible choices of \mathcal{S} for achieving this goal, corresponding to the two notions of rank we introduced in Chapter 2, the tensor rank and the multilinear rank.

6.3.1.1 Formulation based on tensor rank

The set \mathcal{S} associated with the formulation based on tensor rank can be expressed as $\mathcal{S} = \mathcal{R}_R \triangleq \{\mathcal{X} \in \mathcal{T} : \text{rank}(\mathcal{X}) \leq R\}$. Note that this definition is equivalent to that given in Section 2.5.2.5. So, a tensor having low rank admits a CPD whose factors have a reduced number of columns, in comparison with the smallest typical rank (see Section 2.5.2.3). Therefore, the number of DOF of $\mathcal{X}^* \in \mathcal{S}$ grows as the number of its CPD parameters, *i.e.*, as $\mathcal{O}(R \sum_p N_p)$. This is largely inferior to $\bar{N} = \dim(\mathcal{T})$ for small values of R .

On the other hand, computational and analytical difficulties arise when dealing with the tensor rank, as discussed in Section 2.5.2. A formulation analogous to (6.5) seems here even harder, as merely computing the rank of a tensor is already intractable. Also, unlike the matrix case, the tensor nuclear norm introduced by Definition 2.18 is not a convex underestimator of the rank [133]. Nevertheless, in [40] it is argued that minimizing the nuclear norm still makes sense under the convex geometric framework they explore.

Another argument in favor of tensor nuclear norm minimization is given by [216], where it is shown that the convex formulation

$$\min_{\mathcal{X} \in \mathcal{T}} \|\mathcal{X}\|_* \quad \text{subject to} \quad \mathcal{A}(\mathcal{X}) = \mathbf{y} \quad (6.13)$$

leads to sampling bounds of the form⁴ $\mathcal{O}(\sum_p N_p \prod_{q \neq p} \text{rank}_q(\mathcal{X}^*) + \sqrt{\hat{R}\bar{N}} \text{polylog}(\sum_p N_p))$ in TC with uniform sampling, with $\hat{R} = \left(\sum_p N_p \prod_{q \neq p} \text{rank}_q(\mathcal{X}^*)\right)^{1/2} \left(\sum_p N_p\right)^{-1/2}$. This is suboptimal with respect to the DOF count above mentioned, but is still below the best bounds known for the formulation based on multilinear rank (which we will present next). For instance, when $R_p = R$ and $N_p = N$ for all p (which we will henceforth call “balanced model”), this yields $\mathcal{O}(PNR^{P-1} + N^{P/2}R^{(P-1)/4} \text{polylog}(PN))$.

As convincing as these arguments may sound, no efficient approach is currently known for solving (6.13). A considerable obstacle is the fact that the nuclear norm is hard to compute and even to approximate in a certain sense [87], but ways of sidestepping these difficulties are currently under investigation, as we shall discuss in Section 6.3.3.3. We mention also that the recent study [168] on convex relaxations for the tensor rank points at an interesting direction. Yet, further work seems to be needed for reducing the involved computing cost.

³Recall that the adjoint \mathcal{A}^\dagger of \mathcal{A} satisfies $\langle \mathcal{A}(x), \mathbf{y} \rangle = \langle x, \mathcal{A}^\dagger(\mathbf{y}) \rangle$ for all $x \in \mathbb{R}^{\bar{N}}$ and $\mathbf{y} \in \mathbb{R}^M$.

⁴The notation $\text{polylog}(n)$ stands for a polylogarithmic function $f(n) = \sum_{k=1}^K a_k \log^k(n)$ for some K .

6.3.1.2 Formulation based on multilinear rank

The approaches proposed in this thesis to handle [Problem 6.6](#) are based on the multilinear rank introduced in [Section 2.6](#). This amounts to taking $\mathcal{S} = \mathcal{L}_{\mathbf{r}}$, where $\mathcal{L}_{\mathbf{r}}$ is defined by [\(2.117\)](#) and $\mathbf{r} = (R_1, \dots, R_P) \in \langle N_1 \rangle \times \dots \times \langle N_P \rangle$ sets a component-wise bound for the mrank. Such a subspace-based approach to parsimonious modeling allows us to promptly resort to well-established and efficient computational matrix tools, unlike in the previously described approach. Furthermore, it is justifiable for tensors which can be well approximated by models of considerably low rank, meaning $\text{rank}(\mathcal{X}^*) < N_p$ for some⁵ $p \in \langle P \rangle$, due to the bound $\text{rank}_p(\mathcal{X}^*) \leq \text{rank}(\mathcal{X}^*)$ (see [Lemma 2.46](#)).

According to [Remark 2.51](#), a tensor \mathcal{X}^* is in $\mathcal{L}_{\mathbf{r}}$ if and only if it can be written as a Tucker model $\mathcal{X}^* = \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)}$ whose core \mathcal{G} is in $\bigotimes_{p=1}^P \mathbb{R}^{R_p}$ (and, evidently, $\mathbf{U}^{(p)} \in \mathbb{R}^{N_p \times R_p}$). Now, without loss of generality, one can constrain that model similarly to the [HOSVD](#) (see [Section 2.6.1.2](#)), by requiring each $\mathbf{U}^{(p)} \in \mathbb{R}^{N_p \times R_p}$ to have orthonormal columns and each mode- p unfolding of \mathcal{G} to have mutually orthogonal rows. $\mathbf{U}^{(p)}$ has thus $R_p N_p - R_p(R_p + 1)/2$ [DOF](#), where the subtracted term accounts for orthonormality of its columns. Similarly, \mathcal{G} has $\prod_p R_p - \sum_p R_p(R_p - 1)/2$ [DOF](#). Altogether, this yields

$$\Phi(\mathbf{r}) \triangleq \prod_p R_p + \sum_p R_p(N_p - R_p) = \mathcal{O}(\prod_p R_p + \sum_p R_p N_p). \quad (6.14)$$

In fact, $\Phi(\mathbf{r})$ corresponds precisely to the dimension of the manifold of tensors having mrank \mathbf{r} [\[123\]](#). Visibly, it is much smaller than $\bar{N} = \dim(\mathcal{T})$ for low enough values of R_p .

One recurrent formulation in this context, which we shall adopt here, is

$$\min_{\mathcal{X} \in \mathcal{L}_{\mathbf{r}}} \|\mathbf{y} - \mathcal{A}(\mathcal{X})\|_2^2. \quad (6.15)$$

Ideally, one would like to come up with a computationally efficient algorithm capable of recovering any $\mathcal{X}^* \in \mathcal{L}_{\mathbf{r}}$ from $M \approx \Phi(\mathbf{r})$ (sufficiently informative) measurements. Given an instance of problem [\(6.15\)](#), we shall refer to the ratio $\theta \triangleq \Phi(\mathbf{r})/M$ as its *regime*. In general, as θ approaches zero, successful recovery becomes more likely, and thus small values of θ correspond to favorable regimes. Conversely, the recovery performance of an algorithm typically degrades as $\theta \rightarrow 1$. Given M random measurements of a certain class (*e.g.*, Gaussian or Bernoulli), the interval $]0, \theta_0]$ in which perfect recovery is achieved with high probability using a given algorithm is called its *recovery regime* for M with respect to this class.

6.3.2 Suitable measurement operators

In [\[146\]](#), it is shown that a certain nonconvex formulation permits perfect recovery (assuming $\mathcal{X}^* \in \mathcal{L}_{\mathbf{r}}$) by taking the optimal amount of $\mathcal{O}(\bar{R} + \sum_p N_p R_p)$ Gaussian measurements. Yet,

⁵As discussed in [Section 2.5.2.2](#), the rank of a tensor can exceed all of its dimensions. For instance, the smallest typical rank of a $8 \times 8 \times 8$ real tensor is 24 [\[54\]](#); hence a low-rank tensor might still have high modal ranks if $8 < \text{rank}(\mathcal{X}^*) < 24$.

this result is valuable only as a theoretical reference, because that formulation is intractable. Mimicking the LRMR theory, analytical tools have been developed for characterizing suitable MOs in the analysis of tractable LRTR approaches, such as the following extension of RIP.

Definition 6.7 (RIP in the LRTR setting [165]). Let $\mathcal{A} : \mathcal{T} \mapsto \mathbb{R}^M$ and $\mathbf{r} \in \langle N_1 \rangle \times \cdots \times \langle N_P \rangle$. The **r-RIC** of \mathcal{A} is the smallest positive real number $\delta_{\mathbf{r}}$ such that

$$\forall \mathbf{X} \in \mathcal{L}_{\mathbf{r}}, \quad (1 - \delta_{\mathbf{r}}) \|\mathbf{X}\|_F^2 \leq \|\mathcal{A}(\mathbf{X})\|_2^2 \leq (1 + \delta_{\mathbf{r}}) \|\mathbf{X}\|_F^2. \quad (6.16)$$

If $\delta_{\mathbf{r}} < 1$, then \mathcal{A} is said to have the RIP of order \mathbf{r} .

Still in [165], a result allowing the derivation of sampling bounds has been given, stating that L -subgaussian MOs possess RICs satisfying $\delta_{\mathbf{r}} < \delta$ with probability at least $1 - \varepsilon$ if

$$M \geq C\delta^{-2} \max \{ (R^P + PNR) \log(P), \log(\varepsilon^{-1}) \}, \quad (6.17)$$

where $N = \max_p N_p$, $R = \max_p R_p$ and C depends only on L . This class of MOs comprehends, *e.g.*, Gaussian and Rademacher measurement ensembles. The same result is also stated in their follow-up work [167], where it is claimed that the term PNR in (6.17) can be refined to $\sum_p N_p R_p$. When $R_p \approx R$ for all p , this number of measurements is close to the optimum, since for fixed δ and ε one has the bound $M \geq M_{\min} \approx \mathcal{O}(\Phi(\mathbf{r}))$. More interestingly in practice, [167, Theorem 4] shows also that partial Fourier measurements (combined with random sign flips) satisfy $\delta_{\mathbf{r}} < \delta$ with probability greater than $1 - 2e^{-\eta}$ provided

$$M \geq C\delta^{-1} (1 + \eta) \log^2(N^P) \max \{ (R^P + PNR) \log(P), \delta^{-1} (1 + \eta) \log^2(N^P) \}. \quad (6.18)$$

For the particular setting of TC, (in)coherence conditions have been proposed in [106] for establishing the success of their approach. They can be seen as an extension of those proposed in [28], requiring that there exists a mode $p \in \langle P \rangle$ and a constant μ such that

$$\mu \left(\mathcal{U}^{(p)} \right) \leq \mu \frac{R_p}{N_p}, \quad \mu \left(\mathcal{V}^{(p)} \right) \leq \mu \frac{R_p}{\bar{N}_p}, \quad \left\| \mathbf{U}^{(p)} \mathbf{V}^{(p)T} \right\|_{H,\infty} \leq \sqrt{\mu \frac{R_p}{\bar{N}}} \quad (6.19)$$

and

$$\frac{1}{P} \|\mathbf{X}^*\|_{H,\infty} \leq \sqrt{\mu R_p (\min\{N_p, \bar{N}_p\})^{-1}}, \quad (6.20)$$

where $\mathcal{U}^{(p)} = \text{colspace}(\mathbf{X}_{(p)})$, $\mathcal{V}^{(p)} = \text{rowspace}(\mathbf{X}_{(p)})$, and $\mathbf{U}^{(p)} \mathbf{\Sigma}^{(p)} \mathbf{V}^{(p)T}$ is the SVD of $\mathbf{X}_{(p)}$. As in LRMR, the motivation is guaranteeing that any set of entries sampled uniformly at random be sufficiently informative, which cannot work for sparse tensors.

6.3.3 Review of main approaches

In the following, we summarize the main existing LRTR techniques, except for the iterative hard thresholding approach, which is the subject of Chapter 7.

6.3.3.1 Solutions based on minimizing nuclear norms of matrix unfoldings

The first and perhaps most popular approach to date consists of minimizing or bounding a weighted sum of nuclear norms (SNN) of matrix unfoldings. This idea, originally proposed in [88, 135, 196], was motivated by the effectiveness of nuclear norm minimization in LRMR and by the fact that it yields convex formulations. For these reasons, it was later employed many times, as in [134, 181, 197].

Many variants of this approach have been devised. A commonly employed formulation is

$$\min_{\mathbf{x} \in \mathcal{T}} \sum_{p=1}^P \gamma_p \|\mathbf{X}_{\langle p \rangle}\|_* \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}, \quad (6.21)$$

where the positive constants γ_p are weighting parameters. The approach proposed in [196, 195] considers the (more general) penalized version

$$\min_{\mathbf{x} \in \mathcal{T}} \frac{1}{2\lambda} \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2 + \sum_{q=1}^Q \gamma_q \|\mathbf{X}_{\langle p_q \rangle}\|_* \quad (6.22)$$

in two cases:

- (i) $Q = 1$ and $p_1 = p \in \langle P \rangle$. This actually amounts to solving an LRMR problem having the mode- p unfolding as its target matrix.
- (ii) $Q = P$ and $p_q = q$ for all $q \in \langle P \rangle$, which means that the nuclear norm of every flat unfolding is included in the sum, just as in (6.21).

An ADMM algorithm is proposed in [195] for formulation (6.22). Its derivation starts by considering the equivalent convex problem

$$\min_{\substack{\mathbf{x} \in \mathcal{T} \\ \mathbf{z}^{(q)} \in \mathbb{R}^{N_{p_q} \times \bar{N}_{p_q}}}} \underbrace{\frac{1}{2\lambda} \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2 + \sum_{q=1}^Q \gamma_q \|\mathbf{z}^{(q)}\|_*}_{\triangleq J(\mathbf{x}, \{\mathbf{z}^{(q)}\}_{q=1}^Q)} \quad \text{subject to} \quad \forall q, \mathbf{X}_{\langle p_q \rangle} = \mathbf{z}^{(q)}, \quad (6.23)$$

whose augmented Lagrangian is given by

$$\begin{aligned} L_\eta \left(\mathbf{X}, \{\mathbf{z}^{(q)}\}_{q=1}^Q, \{\mathbf{A}^{(q)}\}_{q=1}^Q \right) &= J \left(\mathbf{X}, \{\mathbf{z}^{(q)}\}_{q=1}^Q \right) \\ &\quad + \sum_{q=1}^Q \left(\left\langle \mathbf{A}^{(q)}, \mathbf{X}_{\langle p_q \rangle} - \mathbf{z}^{(q)} \right\rangle + \frac{\eta}{2} \left\| \mathbf{X}_{\langle p_q \rangle} - \mathbf{z}^{(q)} \right\|_F^2 \right), \end{aligned} \quad (6.24)$$

where η is called penalty parameter and the matrices $\mathbf{A}^{(q)} \in \mathbb{R}^{N_{p_q} \times \bar{N}_{p_q}}$ are estimates of the Lagrangian multipliers [22, 152]. Note that L_η corresponds to the Lagrangian of the cost function of (6.23) with an added quadratic term which “enforces” the constraint (or

Algorithm 6.1 ADMM algorithm for LRTR based on SNN minimization approach [195].

Inputs: Initial estimate $\mathbf{x}_0 = \text{vec}(\mathbf{X}_0)$, vector of measurements \mathbf{y} , matrix $\mathbf{A} \in \mathbb{R}^{M \times \bar{N}}$ such that $\mathcal{A}(\mathbf{X}) = \mathbf{A} \text{vec}(\mathbf{X})$ and parameters γ_q , λ and η

Output: Estimate of \mathbf{X}^*

```

1:  $k \leftarrow 0$ 
2: repeat
3:    $\mathbf{x}_{k+1} = (\mathbf{A}^T \mathbf{A} + Q\lambda\eta \mathbf{I}_{\bar{N}})^{-1} \left( \mathbf{A}^T \mathbf{y} + \lambda\eta \sum_{q=1}^Q \mathbf{\Pi}_{p_q}^T \left( \mathbf{z}_k^{(q)} - \frac{1}{\eta} \mathbf{a}_k^{(q)} \right) \right)$ 
4:   for  $q = 1, \dots, Q$  do
5:      $\mathbf{z}_{k+1}^{(q)} = \text{vec} \left[ \text{prox}_{\gamma_q/\eta}^{\|\cdot\|_*} \left( \text{unvec}_{N_{p_q} \times \bar{N}_{p_q}} \left( \mathbf{\Pi}_{p_q} \mathbf{x}_{k+1} + \frac{1}{\eta} \mathbf{a}_k^{(q)} \right) \right) \right]$ 
6:   for  $q = 1, \dots, Q$  do
7:      $\mathbf{a}_{k+1}^{(q)} = \mathbf{a}_k^{(q)} + \eta \left( \mathbf{\Pi}_{p_q} \mathbf{x}_{k+1} - \mathbf{z}_{k+1}^{(q)} \right)$ 
8:    $k \leftarrow k + 1$ 
9: until the stopping criterion is fulfilled
10: return  $\hat{\mathbf{X}}^* = \text{unvec}_{N_1 \times \dots \times N_P}(\mathbf{x}_k)$ 

```

“strengthens” convexity). The idea is then to solve the optimization problem by searching for a saddle point of L_η , which characterizes the solution of (6.23). To this end, one must simultaneously minimize L_η with respect to \mathbf{X} and $\mathbf{Z}^{(q)}$ and maximize it with respect to $\mathbf{A}^{(q)}$. In ADMM, this is done iteratively in an alternating fashion, via the scheme

$$\begin{aligned}
\text{(i)} \quad & \mathbf{X}_{k+1} = \arg \min_{\mathbf{X} \in \mathcal{T}} L_\eta \left(\mathbf{X}, \{\mathbf{Z}_k^{(q)}\}_{q=1}^Q, \{\mathbf{A}_k^{(q)}\}_{q=1}^Q \right), \\
\text{(ii)} \quad & \mathbf{Z}_{k+1}^{(q)} = \arg \min_{\mathbf{Z}^{(q)} \in \mathbb{R}^{N_{p_q} \times \bar{N}_{p_q}}} L_\eta \left(\mathbf{X}_{k+1}, \{\mathbf{Z}^{(q)}\}_{q=1}^Q, \{\mathbf{A}_k^{(q)}\}_{q=1}^Q \right), \quad q = 1, \dots, Q, \\
\text{(iii)} \quad & \mathbf{A}_{k+1}^{(q)} = \mathbf{A}_k^{(q)} + \eta \left((\mathbf{X}_{k+1})_{\langle p_q \rangle} - \mathbf{Z}_{k+1}^{(q)} \right), \quad q = 1, \dots, Q.
\end{aligned}$$

Note that the third equation can be interpreted as a gradient ascent step of size η .

The resulting scheme is given in Algorithm 6.1. For concreteness, it is presented in terms of vectorizations of \mathbf{X} , $\mathbf{Z}^{(q)}$ and $\mathbf{A}^{(q)}$, denoted by \mathbf{x} , $\mathbf{z}^{(q)}$ and $\mathbf{a}^{(q)}$, respectively. The linear operator \mathcal{A} is represented by a matrix $\mathbf{A} \in \mathbb{R}^{M \times \bar{N}}$ such that $\mathcal{A}(\mathbf{X}) = \mathbf{A}\mathbf{x}$. We employ the symbol $\text{prox}_\beta^{\|\cdot\|_*}$ to denote the *proximity operator* of the matrix nuclear norm, defined as [48]

$$\text{prox}_\beta^{\|\cdot\|_*}(\mathbf{W}) \triangleq \arg \min_{\mathbf{T}} \beta \|\mathbf{T}\|_* + \frac{1}{2} \|\mathbf{W} - \mathbf{T}\|_F^2, \quad (6.25)$$

which arises at step (ii). It can be shown that the solution of (6.25) is given by the soft singular value thresholding operator $\text{prox}_\beta^{\|\cdot\|_*}(\mathbf{W}) = \mathbf{U}(\mathbf{\Sigma} - \beta\mathbf{I})_+ \mathbf{V}^T$, where $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of \mathbf{W} and $(\cdot)_+$ projects each component of its argument onto \mathbb{R}_+ . Finally, $\mathbf{\Pi}_p \in \mathbb{R}^{\bar{N} \times \bar{N}}$ denotes the permutation matrix which satisfies for all $p \in \langle P \rangle$

$$\mathbf{\Pi}_p \mathbf{x} = \mathbf{\Pi}_p \text{vec}(\mathbf{X}) = \mathbf{\Pi}_p \text{vec}(\mathbf{X}_{(1)}) = \text{vec}(\mathbf{X}_{(p)}) \Leftrightarrow \mathbf{\Pi}_p^T \text{vec}(\mathbf{X}_{(p)}) = \mathbf{x}. \quad (6.26)$$

It is evident that the products $\mathbf{A}^T \mathbf{A}$ and $\mathbf{A}^T \mathbf{y}$ can be precomputed in order to reduce the overall load. However, this comes at the price of storing a matrix having \bar{N}^2 entries, and still demands a large number of operations to initialize the algorithm. Thus, for unstructured operators, [Algorithm 6.1](#) quickly becomes too costly as \bar{N} grows. When tackling a **TC** problem, one can of course reduce this burden by rewriting [line 3](#) as⁶

$$[\mathbf{x}_{k+1}]_n = \begin{cases} ([\mathbf{x}^*]_n + \lambda\eta[\mathbf{v}_{k+1}]_n) / (1 + Q\eta\lambda), & n \in \Omega \\ \frac{1}{Q}[\mathbf{v}_{k+1}]_n, & n \notin \Omega \end{cases} \quad (6.27)$$

where $\mathbf{x}^* = \text{vec}(\mathcal{X}^*)$ and

$$\mathbf{v}_{k+1} \triangleq \sum_{q=1}^Q \mathbf{\Pi}_{p_q}^T \left(\mathbf{z}_k^{(q)} - \frac{1}{\eta} \mathbf{a}_k^{(q)} \right). \quad (6.28)$$

In [\[195\]](#), special attention is paid to the case where $\lambda \rightarrow 0$, since then [\(6.22\)](#) becomes equivalent to [\(6.21\)](#) (with $P = Q$ and $p_q = q$). In that scenario, [\(6.27\)](#) can be simplified to

$$[\mathbf{x}_{k+1}]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ \frac{1}{Q}[\mathbf{v}_{k+1}]_n, & n \notin \Omega. \end{cases} \quad (6.29)$$

Let us describe some theoretical results pertaining to this approach. Recovery guarantees were given in [\[197\]](#) considering formulation [\(6.22\)](#). For fixed P , they certify success (in the noiseless case) as soon as $\mathcal{O}(\max_p R_p \bar{N}_p)$ Gaussian measurements are taken (assuming all unfoldings are involved in the **SNN** term). For instance, for the balanced model, this yields $\mathcal{O}(RN^{P-1})$. More recently, [\[146\]](#) has shown that such a number of Gaussian measurements is actually *necessary* for recovery via this approach.

In an attempt to reduce the gap with respect to the optimal rate of $\mathcal{O}(\Phi(\mathbf{r}))$ measurements, [\[146\]](#) proposed minimizing the nuclear norm of a single matrix unfolding having “more balanced” dimensions (*i.e.*, as close as possible to being square). With respect to the balanced model, for example, one should then choose $\mathbf{X}_{[q]}$ with $q = \lfloor P/2 \rfloor$, which allows reaching sampling bounds growing as $\mathcal{O}(R^{\lfloor \frac{P}{2} \rfloor} N^{\lceil \frac{P}{2} \rceil})$ for Gaussian measurements. Despite the progress, this bound still grows much faster than $\Phi(\mathbf{r})$ and only brings improvement for $P > 3$. Also, the optimal choice of unfolding generally depends not only on the dimensions N_p , but also on the mrank components R_p , which are generally not known beforehand.

Still in the realm of convex **SNN**-based approaches, robust principal component analysis (PCA) techniques are extended to a **TC** setting in [\[106\]](#), relying on an underlying model which consists of a sum of a low-mrank tensor plus a sparse one. With this approach, [\[106\]](#) states the first recovery guarantees for **TC**, which apply to the balanced model with $\mathcal{O}(\mu RN^{P-1} P^2 \log^2(N^{P-1}))$ measurements, where μ is the same constant of [\(6.19\)](#)–[\(6.20\)](#).

⁶Here, with some abuse of notation, we also denote by $\Omega \subset \langle \bar{N} \rangle$ the set of indices associated with the multi-indices (n_1, \dots, n_P) of the entries sampled by \mathcal{A} , according to the ordering defined by the $\text{vec}(\cdot)$ operator (see [Section 2.2](#)).

6.3.3.2 Modal unfolding factorization approach

An approach based on matrix factorization is proposed in [213] for TC, extending the method of [212] employed in the context of matrix recovery. Its formulation is written as

$$\min_{\substack{\mathbf{X} \in \mathcal{T} \\ \mathbf{A}^{(p)} \in \mathbb{R}^{N_p \times R_p} \\ \mathbf{B}^{(p)} \in \mathbb{R}^{\tilde{N}_p \times R_p}}} \sum_{p=1}^P \frac{\alpha_p}{2} \left\| \mathbf{A}^{(p)} \mathbf{B}^{(p)T} - \mathbf{X}_{\langle p \rangle} \right\|_F^2 \quad \text{subject to} \quad \mathcal{A}(\mathbf{X}) = \mathbf{y}, \quad (6.30)$$

where $\sum_p \alpha_p = 1$ and the modal ranks R_1, \dots, R_P are fixed *a priori*. One thus performs a simultaneous low-rank factorization of the estimate's mode- p unfoldings. To tackle problem (6.30), the algorithm given in [213], named tensor completion by parallel matrix factorization (TMac), relies on a simple alternating minimization procedure which cyclically updates $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}, \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(P)}$ and \mathbf{X} via conditional least-squares estimates.

An important aspect of formulation (6.30) is how to choose adequate values for the modal ranks R_p . While too low values might not allow accurate reconstruction of a given tensor, setting these ranks too high with respect to the number of available measurements also degrades estimation performance. Yet, in general one does not know in advance which values sit in between these two extremes. To deal with this issue, [213] employs a rank adaptation scheme which either (i) starts off with an underestimated value for each R_p , increasing each of them throughout the iterations when slow progress is detected with respect to its associated mode or (ii) starts from overestimated values for the ranks and decreases each one of them if a significant gap is found between two consecutive eigenvalues of $(\hat{\mathbf{A}}_k^{(p)})^T \hat{\mathbf{A}}_k^{(p)}$, where $\hat{\mathbf{A}}_k^{(p)}$ is the estimate of $\mathbf{A}^{(p)}$ produced at iteration k . See [213] for details on this heuristic.

In addition to modal ranks, the weights α_p in (6.30) should ideally be set based on knowledge of the relative parsimony of each modal unfolding, which often is not available. Hence, an adaptive scheme for estimating these weights is also proposed in [213]. The idea is to set them according to the error contained in each factorization, essentially by assigning larger values to the modes whose unfoldings best fit their corresponding factorizations.

No recovery results are provided in [213]; only a convergence proof (to a local minimum) is developed.

6.3.3.3 Algorithms based on tensor rank

Despite the predominance of the mrank-based approach of Section 6.3.1.2, some (few) works have been devoted to attacking Problem 6.6 from a CPD-oriented perspective.

Recall from Section 3.3.1.1 that a standard method for computing the CPD consists in estimating the best factors in the least-squares sense. A simple way of addressing the TC problem thus consists in formulating a weighted least-squares criterion where the weight of each particular entry is either unitary or null, depending on whether it is observed or not.

This approach is suggested in [24], and a second-order optimization algorithm based on the Levenberg-Marquardt method [143] is proposed on its basis in [194]. Later, [1] followed the same idea but derived a first-order algorithm, aiming at completing large-scale tensors. Note that this approach requires choosing the rank R , and is subject to facing problems related to the non-existence of a best rank- R approximation (see Section 2.5.2.5).

A block coordinate descent algorithm is proposed in [214] to deal with the case in which the factor matrices are nonnegative. Interestingly, a best rank- R approximation always exists under this constraint [132]. The resulting algorithm draws upon the alternating proximal gradient method and resorts also to an extrapolation technique in order to accelerate convergence. A proof of its convergence is provided in [214].

Another method proposed in [194] employs a single imputation technique combined with the ALS algorithm. The idea is fairly simple: at each iteration, one updates the factors after filling the missing data with the values provided by the most recent model estimate (*i.e.*, in light of the current factor estimates). This can be seen as a special case of the celebrated expectation minimization method [118, 71].

Quite recently, [215, 43] have adopted a different strategy, resorting to the conditional gradient (CG) algorithm⁷ [86] and to the generalized CG algorithm [99], respectively. The work in [215] addresses the TC problem via the formulation

$$\min_{\mathbf{X} \in \mathcal{T}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2 \quad \text{subject to} \quad \|\mathbf{X}\|_* \leq \beta, \quad (6.31)$$

where $\beta > 0$. At each k th iteration of their algorithm, one first solves

$$\mathbf{S}_k = \arg \min_{\|\mathbf{X}\|_* \leq \beta} \langle \nabla J(\mathbf{X}_k), \mathbf{X} \rangle = -\beta \arg \max_{\|\mathbf{X}\|_* \leq 1} \langle \nabla J(\mathbf{X}_k), \mathbf{X} \rangle. \quad (6.32)$$

As the functional maximized in the above expression is, by definition, the dual of the nuclear norm, *i.e.*, the tensor spectral norm (see Section 2.3.2), we can write

$$\mathbf{S}_k = -\beta \bigotimes_{p=1}^P \mathbf{u}^{(p)}, \quad \text{where} \quad \left(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(P)} \right) = \arg \max_{\|\mathbf{u}^{(p)}\|_2 \leq 1} \left\langle \nabla J(\mathbf{X}_k), \bigotimes_{p=1}^P \mathbf{u}^{(p)} \right\rangle. \quad (6.33)$$

Hence, the difficulty of dealing with the nuclear norm is circumvented by rewriting subproblem (6.32) in terms of a rank-one approximation problem, for which efficient approximation methods exist (see, *e.g.*, [182] and references therein). Once \mathbf{S}_k is obtained, the update $\mathbf{X}_{k+1} = (1 - \alpha_k)\mathbf{X}_k + \alpha_k\mathbf{S}_k$ is calculated, where α_k is some chosen step size. Though [215] empirically shows the effectiveness of this scheme, no recovery results are derived.

The approach of [43], in its turn, relies on the penalized formulation

$$\min_{\mathbf{X} \in \mathcal{T}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2 + \lambda \|\mathbf{X}\|_*, \quad (6.34)$$

⁷Also known as Frank-Wolfe algorithm.

where $\lambda > 0$ is a regularization constant. Following the generalized CG method, one computes at each iteration

$$\mathbf{S}_k \in \arg \max_{\|\mathbf{z}\|_* \leq 1} \langle \nabla J(\mathbf{X}_k), \mathbf{z} \rangle \quad (6.35)$$

and then calculates the iterate $\mathbf{X}_{k+1} = (1 - \alpha_k)\mathbf{X}_k + \alpha_k\beta_k\mathbf{S}_k$, where $0 < \alpha_k \leq 1$ and β_k is a scaling factor computed in an optimal fashion. Similarly to the above described algorithm, this allows leveraging rank-one approximation methods in order to compute \mathbf{S}_k . Specifically, [43] uses a method which approximates the spectral norm of a tensor by the maximum spectral norm possessed by one of its matrix slices, *i.e.*, subtensors in which only two indices vary and all others are fixed. Sampling bounds for Gaussian measurements are then derived on the basis of a result obtained by [146]. Yet, these sampling bounds involve a factor of the form \bar{N}_p (which yields N^{P-1} in the balanced model), and thus are quite disappointing when compared with the number of DOF of the underlying CPD model, which is $\mathcal{O}(R \sum_p N_p)$.

6.3.3.4 Other approaches

Among other works which have been devoted to LRTR, we can mention the efforts to handle TC via Riemannian optimization techniques, by exploiting the smooth manifold structure of sets of low-mrank tensors. Namely, both [112, 123] develop nonlinear conjugate gradient algorithms for reconstructing low-mrank tensors with missing data, essentially by relying on the Tucker model. A detailed explanation of these approaches is out of the scope of this thesis, as it requires introducing Riemannian geometry concepts.

Recent works have also exploited the so-called hierarchical tensor (HT) representation⁸ [95, Chapter 11]. Specifically, [55] addresses the TC problem by profiting from the smooth manifold structure of the set of tensors having fixed rank with respect to that representation [205]. The TC problem is also tackled in [92] by means of the tensor train model, which is a particular case of the HT representation. When the representation rank of that model has components bounded by R , its number of DOF grows (with respect to the balanced model) as $\mathcal{O}(PR^2N)$, making it attractive for large P and small R . Indeed, the use of HT representations is generally advocated on the basis that it scales better with P than Tucker-based approaches, which involve a P th-order core, whilst still avoiding typical conceptual and computational problems associated with CPD-based ones [95].

⁸The expression “hierarchical Tucker representation” is also used [205].

Iterative hard thresholding for low-rank tensor recovery

Iterative hard thresholding (**IHT**) has been proven an effective approach for parsimonious signal recovery. In this chapter, we first review the theoretical underpinnings of this technique and its application to **CS** and **LRMR**. Then, we propose an **IHT** algorithm for **LRTR** which employs sequential modal **SVD** truncation as its thresholding operator. This operator is more computationally efficient than currently used alternatives and allows deriving recovery guarantees by relying solely on restricted isometry constants. A step size selection heuristic is then developed for accelerating convergence. When recovering realistic data, further acceleration and robustness with respect to mrank overestimation are obtained by a continuation technique that estimates a sequence of increasingly complex models. Comprehensive numerical experiments are presented, corroborating the relevance of our contributions.

Contents

7.1	Iterative hard thresholding (IHT)	123
7.2	An IHT algorithm based on sequentially optimal modal projections	127
7.3	Step size selection	134
7.4	Gradual rank increase heuristic	137
7.5	Simulation results	142
7.6	Concluding remarks	146

7.1 Iterative hard thresholding (**IHT**)

IHT is a simple and effective technique for the recovery of parsimonious signals from undercomplete measurements, having been successfully applied in **CS**, **LRMR** and **LRTR** [16, 109, 165, 192, 217]. Its rationale is as follows. In an arbitrary finite-dimensional inner product space \mathcal{H} endowed with a scalar product $\langle \cdot, \cdot \rangle$, one poses

$$\min_{x \in \mathcal{S}} J(x) = \min_{x \in \mathcal{S}} \|\mathbf{y} - \mathcal{A}(x)\|_2^2, \quad (7.1)$$

where $\mathcal{A} : \mathcal{H} \mapsto \mathbb{R}^M$ is a linear operator, $\|x\|_2^2 \triangleq \langle x, x \rangle$ and the set $\mathcal{S} \subset \mathcal{H}$ contains the parsimonious elements of interest. This set is typically nonconvex, closed and nonempty. The basic idea of IHT is then to generate iterates of the form

$$x_k \in \mathcal{P}_{\mathcal{S}} \left(x_{k-1} - \frac{\mu_k}{2} \nabla J(x_{k-1}) \right), \quad \text{with} \quad \nabla J(x) = -2 \mathcal{A}^\dagger (\mathbf{y} - \mathcal{A}(x)), \quad (7.2)$$

where $\mu_k > 0$ is some chosen step size and $\mathcal{P}_{\mathcal{S}}$ denotes¹ the (orthogonal) projector onto \mathcal{S} . Because \mathcal{S} is possibly nonconvex, $\mathcal{P}_{\mathcal{S}}(x) = \arg \min_{z \in \mathcal{S}} \|x - z\|_2^2$ generally yields a set (which is nonempty by the extreme value theorem, since \mathcal{S} is closed and nonempty). In practice, whichever the chosen x_k satisfying (7.2) is, convergence and recovery guarantee results usually remain the same.

The iterates in (7.2) resemble the projected gradient (or projected Landweber) algorithm, which is a convex optimization method [48]. Interestingly, it turns out that they apply to (7.1) even for nonconvex \mathcal{S} , due to the form of $J(x)$. The explanation relies on the majorization-minimization technique [17], which consists in minimizing at iteration k the functional

$$J_k(x) = \mu_k J(x) + \|x - x_{k-1}\|_2^2 - \mu_k \|\mathcal{A}(x - x_{k-1})\|_2^2 \quad (7.3)$$

over \mathcal{S} for some value of μ_k such that $J_k(x) > \mu_k J(x)$ for all $x \neq x_{k-1}$. Such a μ_k always exists: as \mathcal{H} is finite-dimensional and thus $\|\mathcal{A}\|$ is bounded,² one can take $\mu_k < \|\mathcal{A}\|^{-1}$. Clearly, if $x_k \in \arg \min_{x \in \mathcal{S}} J_k(x)$ and $x_k \neq x_{k-1}$, then $\mu_k J(x_k) < J_k(x_k) \leq J_k(x_{k-1}) = \mu_k J(x_{k-1})$, thus achieving objective function reduction.

So, the question is how to compute such a x_k . Expanding $J(x)$ in (7.3), we have

$$J_k(x) = \|x - x_{k-1}\|_2^2 - 2\mu_k \langle \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(x_{k-1})), x \rangle - \mu_k \|\mathcal{A}(x_{k-1})\|_2^2 + \mu_k \|\mathbf{y}\|_2^2. \quad (7.4)$$

The expression in (7.4) is strictly convex, and hence its (unique) unconstrained minimum is straightforwardly obtained by solving $J'_k(x) = 0$, which gives

$$x_k^* \triangleq \arg \min_{x \in \mathcal{H}} J_k(x) = x_{k-1} + \mu_k \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(x_{k-1})) = x_{k-1} - \frac{\mu_k}{2} \nabla J(x_{k-1}). \quad (7.5)$$

The crucial point is that, because the quadratic term in x of $J(x)$ is canceled out in $J_k(x)$, the latter has circular level curves, and thus $\arg \min_{x \in \mathcal{S}} J_k(x) = \mathcal{P}_{\mathcal{S}}(x_k^*)$ for any nonempty closed set \mathcal{S} . This result is shown below, generalizing the one we have given in [66].

Proposition 7.1. Let $\mathcal{S} \subset \mathcal{H}$ be a closed nonempty set. Then, $\mathcal{P}_{\mathcal{S}}(x_k^*)$ is the set of minimizers of $J_k(x)$ over \mathcal{S} , where x_k^* is given by (7.5).

Proof. Since \mathcal{S} is closed and nonempty, J_k is continuous and $J_k(x) \rightarrow \infty$ for $\|x\|_2 \rightarrow \infty$, J_k admits at least one minimum in \mathcal{S} . Also, for any $x \in \mathcal{S}$, we can write $x = x_k^* + z$ for some

¹This notation will be repeatedly used throughout this chapter.

²The symbol $\|\cdot\|$ denotes here the operator norm of \mathcal{A} .

$z \in \mathcal{H}$ and then express J_k as

$$J_k(x_k^* + z) = J_k(x_k^*) + \|z\|_2^2 + 2\langle z, x_k^* - x_{k-1} \rangle - 2\mu_k \langle \mathcal{A}(z), \mathbf{y} - \mathcal{A}(x_{k-1}) \rangle \quad (7.6)$$

$$= J_k(x_k^*) + \|z\|_2^2 + 2\langle z, x_k^* - x_{k-1} \rangle + 2 \left\langle z, \frac{\mu_k}{2} \nabla J(x_{k-1}) \right\rangle \quad (7.7)$$

$$= J_k(x_k^*) + \|z\|_2^2, \quad (7.8)$$

where the last equality follows directly from (7.5). Hence, as $z = x - x_k^*$, we have

$$\arg \min_{x \in \mathcal{S}} J_k(x) = \arg \min_{x \in \mathcal{S}} \|x - x_k^*\|_2^2 = \mathcal{P}_{\mathcal{S}}(x_k^*). \quad (7.9)$$

□

This simplicity is precisely the benefit of iteratively minimizing $J_k(x)$ rather than $J(x)$.

7.1.1 Application to compressive sensing and low-rank matrix recovery

Formulation (7.1) applies to CS with $\mathcal{H} = \mathbb{R}^N$ and $\mathcal{S} = \mathcal{S}_s$, as defined in Section 6.1.2. Note that \mathcal{S}_s is not convex, since $\mathbf{u}, \mathbf{v} \in \mathcal{S}_s$ generally implies $\alpha \mathbf{u} + (1 - \alpha) \mathbf{v} \in \mathcal{S}_{2s}$ for $\alpha \in (0, 1)$. The iterates thus have the form

$$\mathbf{x}_{k+1} = \mathcal{H}_s(\mathbf{x}_k + \mu_k \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_k)), \quad (7.10)$$

where $\mathcal{H}_s : \mathbb{R}^N \mapsto \mathcal{S}_s$ projects a vector onto its best s -sparse approximation by zeroing all but its components of largest magnitude. As multiple best approximations may exist (for instance, $|\mathcal{P}_{\mathcal{S}_s}(\mathbf{1})| = \binom{N}{s}$), an arbitrary $\mathcal{H}_s(\mathbf{x}) \in \mathcal{P}_{\mathcal{S}_s}(\mathbf{x})$ is picked. \mathcal{H}_s is called a *hard thresholding operator*.

By imposing certain RIC bounds on \mathbf{A} , recovery results such as the following one can be derived. In particular, it implies linear convergence to the sought solution in the ideal case where \mathbf{x}^* is s -sparse and there is no measurement error.

Theorem 7.2 (Performance bound for IHT [85, Theorem 6.18]). Suppose $\mathbf{A} \in \mathbb{R}^{M \times N}$ has a $3s$ -RIC bounded as $\delta_{3s} < 1/\sqrt{3} \approx 0.5773$. Then, if $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \mathbf{e}$, the iterates (7.10) satisfy

$$\|\mathbf{x}_k - \mathbf{x}_s^*\|_2 \leq \gamma^k \|\mathbf{x}_0 - \mathbf{x}_s^*\|_2 + \tau \|\mathbf{A}(\mathbf{x}^* - \mathbf{x}_s^*) + \mathbf{e}\|_2 \quad (7.11)$$

for $\mu_k = 1$, where $\gamma = \sqrt{3} \delta_{3s} < 1$, $\tau \leq 2.18/(1 - \gamma)$ and $\mathbf{x}_s^* = \mathcal{H}_s(\mathbf{x}^*)$.

In analogy with (7.10), IHT can be applied to LRMR with $\mathcal{H} = \mathbb{R}^{N_1 \times N_2}$ and $\mathcal{S} = \mathcal{L}_R$ by iteratively computing

$$\mathbf{X}_{k+1} = \mathcal{H}_R(\mathbf{X}_k + \mu_k \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(\mathbf{X}_k))), \quad (7.12)$$

where $\mathcal{H}_R : \mathbb{R}^{N_1 \times N_2} \mapsto \mathcal{L}_R$ delivers a best rank- R approximation of a matrix, \mathcal{L}_R being defined as in Section 6.2.1. From Eckart-Young's theorem [76], it can be computed through

$$\mathcal{H}_R(\mathbf{X}) = \sum_{r=1}^R \sigma_r \mathbf{u}_r \mathbf{v}_r^T, \quad \text{where} \quad \mathbf{X} = \sum_{n=1}^{\min\{N_1, N_2\}} \sigma_n \mathbf{u}_n \mathbf{v}_n^T \quad (7.13)$$

is the SVD of \mathbf{X} , with $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min\{N_1, N_2\}}$. If $\sigma_R = \dots = \sigma_{R+d}$, with $d \in \langle \min\{N_1, N_2\} - R \rangle$, then \mathcal{H}_R delivers one of the multiple best approximations of \mathbf{X} . Similarly to (7.11), performance bounds of the form

$$\|\mathbf{X}^* - \mathbf{X}_k\|_F \leq \xi^k \|\mathbf{X}^* - \mathbf{X}_0\|_F + \frac{\gamma}{1-\xi} \|\mathbf{e}\|_2 \quad (7.14)$$

are derived in [125] under RIP assumptions, with both ξ and γ depending on δ_{2R} and δ_{3R} . Unfortunately, though, no recovery guarantees exist for IHT in the matrix completion setting.

7.1.2 Application to tensor recovery based on multilinear rank

Consider now $\mathcal{H} = \mathcal{T} \triangleq \bigotimes_{p=1}^P \mathbb{R}^{N_p}$ and $\mathcal{S} = \mathcal{L}_{\mathbf{r}}$ (as defined by (2.117)). Recall from Section 2.6.2 that, though computing projections onto $\mathcal{L}_{\mathbf{r}}$ is NP-hard, efficient approximate methods exist. The truncated HOSVD described in Section 2.6.2.1, in particular, is quasi-optimal (in the sense of (2.120)) and has moderate computing cost. This has motivated its use in the tensor iterative hard thresholding (TIHT) algorithm [165], whose iterates read

$$\mathbf{x}_{k+1} = \mathcal{H}_{\mathbf{r}}(\mathbf{v}_k), \quad \mathbf{v}_k \triangleq \mathbf{x}_k + \mu_k \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(\mathbf{x}_k)), \quad (7.15)$$

where $\mathcal{H}_{\mathbf{r}} : \mathcal{T} \mapsto \mathcal{L}_{\mathbf{r}}$ outputs the HOSVD of its argument truncated at $\text{mrnk } \mathbf{r}$ and

$$\mu_k = \frac{\|\nabla J(\mathbf{x}_k)\|_F^2}{\|\mathcal{A}(\nabla J(\mathbf{x}_k))\|_F^2}. \quad (7.16)$$

Later on, the same authors have proposed in [167] the normalized tensor iterative hard thresholding (NTIHT) algorithm whose step size is given by

$$\mu_k = \frac{\|\mathcal{G}_k\|_F^2}{\|\mathcal{A}(\mathcal{G}_k)\|_F^2}, \quad \mathcal{G}_k = \nabla J(\mathbf{x}_k) \times_{p=1}^P \mathbf{U}_k^{(p)} \mathbf{U}_k^{(p)T} \quad (7.17)$$

where the orthogonal matrices $\mathbf{U}_k^{(p)}$ are bases for the modal subspaces of \mathbf{x}_k which are obtained in the application of THOSVD at iteration $k-1$.

Although the effectiveness of TIHT and NTIHT was experimentally shown, recovery results based solely on typical RIP conditions are still lacking. The best known one in this sense is as follows.

Theorem 7.3 (Performance bound of NTIHT [167, Theorem 1]). Put $a \in (0, 1)$ and let \mathcal{A} be an MO possessing a $3\mathbf{r}$ -RIC satisfying $\delta_{3\mathbf{r}} < a/(a+8)$, where $3\mathbf{r} = (3R_1, \dots, 3R_P)$. Let $\mathbf{x}^* \in \mathcal{L}_{\mathbf{r}}$. Then, given measurements $\mathbf{y} = \mathcal{A}(\mathbf{x}^*) + \mathbf{e}$, if

$$\|\mathbf{x}_k - \mathbf{v}_k\|_F \leq (1 + \varepsilon(a)) \|\mathbf{x}^* - \mathbf{v}_k\|_F, \quad (7.18)$$

where $\varepsilon(a) = a^2(1 - \delta_{3\mathbf{r}})^2(17(1 - \delta_{3\mathbf{r}} + \sqrt{1 + \delta_{2\mathbf{r}}}\|\mathcal{A}\|_2))^{-2}$, then for all k we have

$$\|\mathbf{x}^* - \mathbf{x}_{k+1}\|_F \leq a^k \|\mathbf{x}^* - \mathbf{x}_0\|_F + \frac{b(a)}{1-a} \|\mathbf{e}\|_2, \quad (7.19)$$

where $b(a) = 2\frac{\sqrt{1+\delta_{3\mathbf{r}}}}{1-\delta_{3\mathbf{r}}} + \sqrt{4\varepsilon(a) + 2\varepsilon(a)^2} \frac{1}{1-\delta_{3\mathbf{r}}} \|\mathcal{A}\|_2$.

Though a heuristic justification is given in [167] for condition (7.18), it cannot be guaranteed in general because the THOSVD is quasi-optimal by a factor \sqrt{P} , whereas $\varepsilon(a) \approx 0$.

A similar scheme called minimum n -rank approximation (MnRA) is proposed in [217]. However, as it uses a convex combination of truncated SVDs in lieu of the hard thresholding operator, no projection onto \mathcal{L}_r is performed. Given nonnegative weights w_p satisfying $\sum_{p \in \langle P \rangle} w_p = 1$, this operator, denoted here by \mathcal{C}_r , is defined as

$$\mathcal{C}_r(\mathbf{X}) = \sum_{p=1}^P w_p \mathbf{Z}_p \quad \text{such that} \quad (\mathbf{Z}_p)_{\langle p \rangle} = \mathcal{H}_{R_p}(\mathbf{X}_{\langle p \rangle}), \quad (7.20)$$

in which \mathcal{H}_{R_p} is the same of (7.13). The step size of MnRA is fixed, thus yielding iterates

$$\mathbf{x}_{k+1} = \mathcal{C}_r\left(\mathbf{x}_k + \mu \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(\mathbf{x}_k))\right). \quad (7.21)$$

Although $\mathbf{x}_k \notin \mathcal{L}_r$ in general, convergence to the true estimate in the ideal case has been shown in [217] under RIP conditions. For convenience, this result is reproduced below.

Theorem 7.4 (Performance bound of MnRA [217, Theorem 4.2]). Let \mathcal{A} be an MO with RICs $\delta_{\bar{r}_p} < 1$ for all $p \in \langle P \rangle$, where $\bar{\mathbf{r}}_p \triangleq (N_1, \dots, N_{p-1}, 3R_p, N_{p+1}, \dots, N_P)$. Let also $\mathbf{x}^* \in \mathcal{L}_r$ and $\mathbf{y} = \mathcal{A}(\mathbf{x}^*) + \mathbf{e}$ and assume $3/4 < \mu < 5/4$. If $\max_p \delta_{\bar{r}_p} < \tau$, then MnRA satisfies

$$\forall k, \quad \|\mathbf{x}^* - \mathbf{x}_k\|_F \leq 2^{-k} \|\mathbf{x}^* - \mathbf{x}_0\|_F + 2C \|\mathbf{e}\|_2, \quad (7.22)$$

where $C = 2\mu\sqrt{1+\tau}$ and $\tau = \frac{1/4 - |1-\mu|}{\mu(1 + \lceil \max_p N_p/R_p \rceil)}$.

Unlike Theorem 7.3, this result does not involve a restrictive assumption such as (7.18). Yet, it is not satisfying from a sampling efficiency standpoint, because \mathcal{A} can only have a RIC $\delta_{\bar{r}_p} < 1$ if $M \geq M_{\min} = \mathcal{O}(R_p \bar{N}_p)$ (e.g., $\mathcal{O}(RN^{P-1})$ for the balanced model having $R_p = R$ and $N_p = N$ for all p), which grows way more quickly than $\Phi(\mathbf{r})$ (see (6.14)).

7.2 An IHT algorithm based on sequentially optimal modal projections

Recall from Section 2.6.2.2 that the SeMP operator, denoted by \mathcal{S}_r , is an alternative to THOSVD which consists in computing a chain of sequentially optimal modal projections of the argument. We propose here to employ it in the context of the IHT method, which yields

$$\mathbf{x}_k = \mathcal{S}_r\left(\mathbf{x}_{k-1} + \mu_k \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(\mathbf{x}_{k-1}))\right). \quad (7.23)$$

In this equation, \mathcal{S}_r is applied with the modes ordered as $\pi = (p_1, p_2, \dots, p_P)$, where π is some permutation of $(1, \dots, P)$, referred to as the modal projection ordering (MPO). The above scheme gives rise to the SeMP-based iterative hard thresholding (SeMPIHT) algorithm, which is laid out in Algorithm 7.1.

Algorithm 7.1 SeMPIHT algorithm: $\text{sempiht}(\mathbf{X}_0, \mathbf{y}, \mathcal{A}, \mathbf{r}, K_{\max}, \epsilon)$.

Inputs: Initial solution \mathbf{X}_0 , measurement vector \mathbf{y} , measurement operator \mathcal{A} , target mrank \mathbf{r} , maximum number of iterations K_{\max} , tolerance ϵ

Outputs: Estimated tensor $\hat{\mathbf{X}}^*$ and used number of iterations K

- 1: **for** $k = 1, \dots, K_{\max}$ **do**
 - 2: $\mathcal{G}_k \leftarrow \mathcal{A}^\dagger(\mathbf{y} - \mathcal{A}(\mathbf{X}_{k-1}))$
 - 3: compute step size μ_k using either ISS (see Section 7.3.2) or formula (7.17)
 - 4: compute $\mathbf{X}_k \leftarrow \mathcal{S}_{\mathbf{r}}(\mathbf{X}_{k-1} + \mu_k \mathcal{G}_k)$ using Algorithm 2.1
 - 5: **if** criterion (7.48) is satisfied **then**
 - 6: **break**
 - 7: **return** $\hat{\mathbf{X}}^* \leftarrow \mathbf{X}_k$ and $K \leftarrow k$
-

7.2.1 Comparison with previous approaches

Clearly enough, the hard thresholding operator employed in an IHT algorithm has a major impact on its convergence speed, computing cost and recovery effectiveness. We thus compare the operators of SeMPIHT, TIHT and MnRA according to the following criteria.

1) Approximation accuracy. As seen in Sections 2.6.2.1 and 2.6.2.2, both $\mathcal{H}_{\mathbf{r}}$ and $\mathcal{S}_{\mathbf{r}}$ are quasi-optimal by a factor \sqrt{P} . In fact, our practical experience is consistent with the observations reported in [209], in that $\|\mathbf{X} - \mathcal{S}_{\mathbf{r}}(\mathbf{X})\|_F < \|\mathbf{X} - \mathcal{H}_{\mathbf{r}}(\mathbf{X})\|_F$ holds in most observed cases. MnRA's operator $\mathcal{C}_{\mathbf{r}}$, in its turn, satisfies

$$\begin{aligned} \|\mathbf{X} - \mathcal{C}_{\mathbf{r}}(\mathbf{X})\|_F &= \left\| \sum_{p=1}^P w_p (\mathbf{X} - \mathbf{Z}_p) \right\|_F \leq \sum_{p=1}^P w_p \|\mathbf{X} - \mathbf{Z}_p\|_F \\ &\leq \sum_{p=1}^P w_p \|\mathbf{X} - \mathbf{X}_{\mathbf{r}}\|_F = \|\mathbf{X} - \mathbf{X}_{\mathbf{r}}\|_F \end{aligned}$$

for all $\mathbf{X}_{\mathbf{r}} \in \mathcal{P}_{\mathcal{L}_{\mathbf{r}}}(\mathbf{X})$, where the second inequality comes from the fact that $(\mathbf{Z}_p)_{\langle p \rangle}$ is the best rank- R_p approximation of $\mathbf{X}_{\langle p \rangle}$. This perhaps surprising result is explained by the fact that $\mathcal{C}_{\mathbf{r}}$ is not really a projection onto $\mathcal{L}_{\mathbf{r}}$, due to the (weighted) sum of projections which are low-rank only with respect to one mode.

2) Computing cost. Both $\mathcal{H}_{\mathbf{r}}$ and $\mathcal{C}_{\mathbf{r}}$ require computing the SVD of each unfolding $\mathbf{X}_{\langle p \rangle}$. Assuming it takes $\mathcal{O}(N_1 N_2 \min\{N_1, N_2\})$ operations to compute the SVD of a $N_1 \times N_2$ matrix,³ applying $\mathcal{H}_{\mathbf{r}}$ requires

$$\mathcal{O} \left(\sum_{p=1}^P N_p \bar{N}_p \min\{N_p, \bar{N}_p\} \right) + \mathcal{O} \left(\sum_{p=1}^P R_1 \dots R_p N_p \dots N_P \right) + \mathcal{O} \left(\sum_{p=1}^P N_1 \dots N_p R_p \dots R_P \right) \quad (7.24)$$

³In principle, the first R terms can be computed with $\mathcal{O}(RN_1N_2)$ operations [97]. Yet, in our experience, optimized classical algorithms delivering the whole SVD, such as that of LAPACK, are usually faster.

operations, where the first sum is the cost of the SVDs and the second one comes from the projection onto the dominant modal subspaces (see (2.124)). The latter is broken down into two terms because it is faster to first compute the $R_1 \times \dots \times R_P$ core of the THOSVD, and then reconstruct the full tensor. Overall, the cost is dominated by the first sum of (7.24). Similarly, applying \mathcal{C}_r demands

$$\mathcal{O}\left(\sum_p N_p \bar{N}_p \min\{N_p, \bar{N}_p\}\right) + \mathcal{O}(P\bar{N}) \quad (7.25)$$

operations, where the first term is related to the SVDs, and the second one to the convex combination of (7.20). Though (7.24) and (7.25) are asymptotically equivalent, \mathcal{C}_r is less costly in practice due to the difference between the second terms of these expressions.

As seen in Section 2.6.2.2, the cost of applying SeMP is

$$\mathcal{O}\left(\sum_{p=1}^P N_p L_p \min\{N_p, L_p\}\right) + \mathcal{O}\left(\sum_{p=1}^P R_1 \dots R_p N_p \dots N_P\right) + \mathcal{O}\left(\sum_{p=1}^P N_1 \dots N_p R_p \dots R_P\right). \quad (7.26)$$

Comparing the first term of (7.26) with those of (7.24) and (7.25), it is seen that \mathcal{S}_r is less costly than \mathcal{H}_r and \mathcal{C}_r , which is due to the dimensionality reduction performed at each step of Algorithm 2.1.

3) Analytical tractability. Theorem 7.3 states a partial recovery result which applies to TIHT. Unfortunately, it relies upon a condition which cannot be assured *a priori*. MnRA, in its turn, enjoys the RIC-based performance bound of Theorem 7.4, despite the fact that in general $\mathcal{C}_r(\mathbf{X}) \notin \mathcal{L}_r$. This result, however, leads to suboptimal sampling bounds. At this point, it is not clear whether a similar (suboptimal) result based only on RIC assumptions can be derived for TIHT. In particular, note that

$$\forall p, \quad (\mathcal{H}_r(\mathbf{X}))_{\langle p \rangle} \notin \arg \min_{\text{rank}(\mathbf{W}) \leq R_p} \|\mathbf{W} - \mathbf{X}_{\langle p \rangle}\|_F. \quad (7.27)$$

In contrast, the sequential optimality of the modal projections performed by SeMP allows establishing RIC-based performance bounds, as we will show next.

7.2.2 Performance bound

This section establishes recovery results for SeMPIHT, under the standard assumption that \mathcal{A} has sufficiently low RICs. We first state two important lemmata, and then proceed to our main result, which is inspired by (but simpler than) that of [217].

Lemma 7.5. Let $\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{R}^{N_p})$, $p \in \langle P \rangle$, and define

$$\mathcal{U} = \left\{ \mathbf{X} : \mathbf{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \dots \times_P \mathbf{U}^{(P)} \text{ for some } \mathcal{G} \in \mathbb{R}^{R_1 \times \dots \times R_P} \right\} \subset \mathcal{L}_r, \quad (7.28)$$

with $\mathbf{r} = (R_1, \dots, R_P)$. Denote $\mathcal{A}_U = \mathcal{A}\mathcal{P}_U$, where \mathcal{P}_U is the orthogonal projector onto \mathcal{U} , and assume \mathcal{A} has a RIC $\delta_{\mathbf{r}} < 1$. Then, $\|\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{I}\| \leq \delta_{\mathbf{r}}$, where \mathcal{I} is the identity over \mathcal{T} .

Proof. Our proof is an extension of the argument supporting [85, Eq. (6.2)] (given in the context of CS). Consider $\mathbf{x} \in \mathcal{U}$, for which $\mathcal{A}_U(\mathbf{x}) = \mathcal{A}(\mathbf{x})$. By definition of $\delta_{\mathbf{r}}$, we deduce $\|\mathcal{A}_U(\mathbf{x})\|_F^2 - \|\mathbf{x}\|_F^2 \leq \delta_{\mathbf{r}}\|\mathbf{x}\|_F^2$. Rewriting the left-hand side of this inequality, we obtain

$$\langle \mathcal{A}_U(\mathbf{x}), \mathcal{A}_U(\mathbf{x}) \rangle - \langle \mathbf{x}, \mathbf{x} \rangle = \langle (\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{I})(\mathbf{x}), \mathbf{x} \rangle \leq \delta_{\mathbf{r}}\|\mathbf{x}\|_F^2. \quad (7.29)$$

Assuming that $\|\mathbf{x}\|_F \neq 0$, dividing by $\|\mathbf{x}\|_F^2$ and taking the maximum with respect to $\mathbf{x} \in \mathcal{U} \setminus \{0\}$ yields

$$\max_{\mathbf{x} \in \mathcal{U} \setminus \{0\}} \frac{\|\mathcal{A}_U(\mathbf{x})\|_F^2}{\|\mathbf{x}\|_F^2} - 1 = \max_{\mathbf{x} \in \mathcal{U} \setminus \{0\}} \frac{\langle (\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{I})(\mathbf{x}), \mathbf{x} \rangle}{\|\mathbf{x}\|_F^2} \leq \delta_{\mathbf{r}}. \quad (7.30)$$

Now, note that for any $\mathbf{z} \in \mathcal{T}$, $\|\mathcal{A}_U(\mathbf{z})\|_F^2 = \|\mathcal{A}_U(\mathcal{P}_U(\mathbf{z}))\|_F^2$ and $\|\mathbf{z}\|_F^2 \geq \|\mathcal{P}_U(\mathbf{z})\|_F^2$. Consequently, the maximum must be the same over the whole space, because

$$\max_{\mathbf{z} \neq 0} \frac{\|\mathcal{A}_U(\mathbf{z})\|_F^2}{\|\mathbf{z}\|_F^2} \leq \max_{\mathbf{z} \neq 0} \frac{\|\mathcal{A}_U(\mathcal{P}_U(\mathbf{z}))\|_F^2}{\|\mathcal{P}_U(\mathbf{z})\|_F^2} = \max_{\mathbf{x} \in \mathcal{U} \setminus \{0\}} \frac{\|\mathcal{A}_U(\mathbf{x})\|_F^2}{\|\mathbf{x}\|_F^2},$$

and therefore (7.30) implies

$$\max_{\mathbf{z} \neq 0} \frac{\langle (\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{I})(\mathbf{z}), \mathbf{z} \rangle}{\|\mathbf{z}\|_F^2} \leq \delta_{\mathbf{r}}. \quad (7.31)$$

Finally, since $\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{I}$ is self adjoint, the left-hand side of the above expression is precisely the definition of its operator norm, and thus the proof is complete. \square

The next lemma is an extension of [85, Lemma 6.20] (which also applies to CS).

Lemma 7.6. If $\mathcal{U} \subseteq \mathcal{L}_{\mathbf{r}}$ and \mathcal{A} has a RIC $\delta_{\mathbf{r}} < 1$, then for all $\mathbf{e} \in \mathbb{R}^M$ we have

$$\|\mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F \leq \sqrt{1 + \delta_{\mathbf{r}}}\|\mathbf{e}\|_2. \quad (7.32)$$

Proof. We assume $\|\mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F \neq 0$ (otherwise the result is trivial) and start by deriving

$$\|\mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F^2 = \langle \mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e}), \mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e}) \rangle = \langle \mathbf{e}, \mathcal{A} \mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e}) \rangle \leq \|\mathbf{e}\|_2 \|\mathcal{A} \mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_2. \quad (7.33)$$

Now, by definition of $\delta_{\mathbf{r}}$, $\|\mathcal{A} \mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F \leq \sqrt{1 + \delta_{\mathbf{r}}}\|\mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F$. Combining this inequality with (7.33) and dividing both sides by $\|\mathcal{P}_U \mathcal{A}^\dagger(\mathbf{e})\|_F$ yields the desired result. \square

Theorem 7.7 (Performance bounds for SeMP). Let $\mathbf{x}^* \in \mathcal{T}$ and $\mathbf{y} = \mathcal{A}(\mathbf{x}^*) + \mathbf{e}$. If \mathcal{A} has a RIC $\delta_{\bar{\mathbf{r}}_p} < 2^{-P}$, where $\bar{\mathbf{r}}_p = (N_1, \dots, N_{p-1}, 3R_p, N_{p+1}, \dots, N_P)$, then the scheme (7.23) applied with fixed step size $\mu_k = 1$ and MPO $\pi = (p, p_2, \dots, p_P)$ satisfies after k iterations:

$$\|\mathbf{x}_{\mathbf{r}}^* - \mathbf{x}_k\|_F \leq \xi^k \|\mathbf{x}_{\mathbf{r}}^* - \mathbf{x}_0\|_F + \frac{2^P \sqrt{1 + \delta_{\bar{\mathbf{r}}_p}}}{1 - \xi} \|\mathcal{A}(\mathbf{x}^* - \mathbf{x}_{\mathbf{r}}^*) + \mathbf{e}\|_2, \quad (7.34)$$

where $\xi = 2^P \delta_{\bar{r}_p} < 1$ and $\mathbf{X}_r^* \in \mathcal{P}_{\mathcal{L}_r}(\mathbf{X}^*) = \arg \min_{\mathbf{Z} \in \mathcal{L}_r} \|\mathbf{X}^* - \mathbf{Z}\|_F$, with $\mathbf{r} = (R_1, \dots, R_P)$. If the step size formula (7.17) is used, then (7.34) holds with $\delta_{\bar{r}_p} < 1/(2^{P+1} + 1)$ and $\xi = \sup_k 2^P (|1 - \mu_k| + \mu_k \delta_{\bar{r}_p}) < 1$.

Proof. For simplicity, we assume, without loss of generality, $\pi = (p, p_2, \dots, p_P) = (1, 2, \dots, P)$. To describe the computation of \mathcal{S}_r at each iteration, we use the notation:

$$\mathbf{V}_0 = \mathbf{X}_{k-1} + \mu_k \mathcal{A}^\dagger (\mathbf{y} - \mathcal{A}(\mathbf{X}_{k-1})) \quad (7.35)$$

$$= \mathbf{X}_{k-1} + \mu_k \mathcal{A}^\dagger \mathcal{A}(\mathbf{X}_r^* - \mathbf{X}_{k-1}) + \mu_k \mathcal{A}^\dagger (\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e}), \quad (7.36)$$

$(\mathbf{V}_p)_{\langle p \rangle} = \mathcal{H}_{R_p} \left((\mathbf{V}_{p-1})_{\langle p \rangle} \right)$, where \mathcal{H}_{R_p} is as defined in (7.13), and $\mathbf{X}_k = \mathcal{S}_r(\mathbf{V}_0) = \mathbf{V}_P$. The result is then obtained by bounding the errors of the approximations $\mathbf{V}_1, \dots, \mathbf{V}_P$. First, note that

$$(\mathbf{V}_p)_{\langle p \rangle} \in \arg \min_{\text{rank}(\mathbf{Z}) \leq R_p} \|\mathbf{Z} - (\mathbf{V}_{p-1})_{\langle p \rangle}\|_F \implies \forall \mathbf{Z} \in \mathcal{L}_r, \quad \|\mathbf{V}_p - \mathbf{V}_{p-1}\|_F \leq \|\mathbf{Z} - \mathbf{V}_{p-1}\|_F,$$

which, together with $\mathbf{X}_r^* \in \mathcal{L}_r$, implies

$$\|\mathbf{X}_r^* - \mathbf{V}_p\|_F \leq \|\mathbf{X}_r^* - \mathbf{V}_{p-1}\|_F + \|\mathbf{V}_p - \mathbf{V}_{p-1}\|_F \leq 2 \|\mathbf{X}_r^* - \mathbf{V}_{p-1}\|_F. \quad (7.37)$$

Therefore, as $\mathbf{X}_k = \mathbf{V}_P$, iterating over this inequality for $p = 2, \dots, P$, we deduce

$$\|\mathbf{X}_r^* - \mathbf{X}_k\|_F \leq 2^{P-1} \|\mathbf{X}_r^* - \mathbf{V}_1\|_F. \quad (7.38)$$

Now, to bound $\|\mathbf{X}_r^* - \mathbf{V}_1\|_F$, we employ the same reasoning as in [89, Lemma 4.1]. Let

$$\mathcal{U} = \left\{ \mathbf{Z} : \text{colspace}(\mathbf{Z}_{\langle 1 \rangle}) \subset \text{colspace} \left((\mathbf{X}_r^*)_{\langle 1 \rangle} \right) \cup \text{colspace} \left((\mathbf{V}_1)_{\langle 1 \rangle} \right) \cup \text{colspace} \left((\mathbf{X}_{k-1})_{\langle 1 \rangle} \right) \right\},$$

so that $\mathbf{X}_r^*, \mathbf{V}_1, \mathbf{X}_{k-1} \in \mathcal{U} \subset \mathcal{L}_{\bar{r}_1}$. We thus have

$$\|\mathbf{V}_1 - \mathbf{V}_0\|_F^2 = \|\mathcal{P}_{\mathcal{U}}(\mathbf{V}_1 - \mathbf{V}_0)\|_F^2 + \|\mathcal{P}_{\mathcal{U}^\perp}(\mathbf{V}_1 - \mathbf{V}_0)\|_F^2 \quad (7.39)$$

$$= \|\mathcal{P}_{\mathcal{U}}(\mathbf{V}_1 - \mathbf{V}_0)\|_F^2 + \|\mathcal{P}_{\mathcal{U}^\perp}(\mathbf{V}_0)\|_F^2 \quad (7.40)$$

and also

$$\|\mathbf{V}_1 - \mathbf{V}_0\|_F^2 \leq \|\mathbf{X}_r^* - \mathbf{V}_0\|_F^2 = \|\mathcal{P}_{\mathcal{U}}(\mathbf{X}_r^* - \mathbf{V}_0)\|_F^2 + \|\mathcal{P}_{\mathcal{U}^\perp}(\mathbf{V}_0)\|_F^2, \quad (7.41)$$

which follows from $(\mathbf{V}_1)_{\langle 1 \rangle} = \mathcal{H}_{R_1} \left((\mathbf{V}_0)_{\langle 1 \rangle} \right)$ and $\mathbf{X}_r^* \in \mathcal{L}_r \cap \mathcal{U}$. Combining (7.40) and (7.41), we obtain

$$\|\mathbf{V}_1 - \mathcal{P}_{\mathcal{U}}(\mathbf{V}_0)\|_F = \|\mathcal{P}_{\mathcal{U}}(\mathbf{V}_1 - \mathbf{V}_0)\|_F \leq \|\mathcal{P}_{\mathcal{U}}(\mathbf{X}_r^* - \mathbf{V}_0)\|_F = \|\mathbf{X}_r^* - \mathcal{P}_{\mathcal{U}}(\mathbf{V}_0)\|_F. \quad (7.42)$$

Hence, using the above equation and (7.35), we have

$$\begin{aligned}
\|\mathbf{X}_r^* - \mathbf{V}_1\|_F &\leq \|\mathbf{X}_r^* - \mathcal{P}_U(\mathbf{V}_0)\|_F + \|\mathbf{V}_1 - \mathcal{P}_U(\mathbf{V}_0)\|_F \\
&\leq 2\|\mathbf{X}_r^* - \mathcal{P}_U(\mathbf{V}_0)\|_F = 2\|\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{V}_0)\|_F \\
&= 2\left\|\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{X}_{k-1}) - \mu_k \mathcal{P}_U \mathcal{A}^\dagger \mathcal{A}(\mathbf{X}_r^* - \mathbf{X}_{k-1}) - \mu_k \mathcal{P}_U \mathcal{A}^\dagger (\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e})\right\|_F \\
&= 2\left\|(1 - \mu_k)\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{X}_{k-1}) - \mu_k \mathcal{P}_U (\mathcal{A}^\dagger \mathcal{A} - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1})\right. \\
&\quad \left. - \mu_k \mathcal{P}_U \mathcal{A}^\dagger (\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e})\right\|_F \\
&\leq 2|1 - \mu_k| \|\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{X}_{k-1})\|_F + 2\mu_k \left\|\mathcal{P}_U(\mathcal{A}^\dagger \mathcal{A} - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1})\right\|_F \\
&\quad + 2\mu_k \left\|\mathcal{P}_U \mathcal{A}^\dagger (\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e})\right\|_F. \tag{7.43}
\end{aligned}$$

It follows from the nonexpansiveness of \mathcal{P}_U that $\|\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{X}_{k-1})\|_F \leq \|\mathbf{X}_r^* - \mathbf{X}_{k-1}\|_F$. By noting that $\mathbf{X}_r^*, \mathbf{X}_{k-1} \in \mathcal{U}$ and using the notation $\mathcal{A}_U = \mathcal{A}\mathcal{P}_U$, we have also

$$\mathcal{P}_U(\mathcal{A}^\dagger \mathcal{A} - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1}) = \mathcal{P}_U(\mathcal{A}^\dagger \mathcal{A} - \mathcal{J})\mathcal{P}_U(\mathbf{X}_r^* - \mathbf{X}_{k-1}) = (\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1}).$$

Thus, from Lemma 7.5 and the fact that $\mathcal{U} \subset \mathcal{L}_{\bar{r}_1}$ we derive the bound

$$\left\|\mathcal{P}_U(\mathcal{A}^\dagger \mathcal{A} - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1})\right\|_F = \left\|(\mathcal{A}_U^\dagger \mathcal{A}_U - \mathcal{J})(\mathbf{X}_r^* - \mathbf{X}_{k-1})\right\|_F \leq \delta_{\bar{r}_1} \|\mathbf{X}_r^* - \mathbf{X}_{k-1}\|_F.$$

Finally, resorting to Lemma 7.6, the last term of (7.43) can be bounded as

$$\left\|\mathcal{P}_U \mathcal{A}^\dagger (\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e})\right\|_F \leq \sqrt{1 + \delta_{\bar{r}_1}} \|\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e}\|_2. \tag{7.44}$$

The above inequalities, combined with (7.38), yield

$$\|\mathbf{X}_r^* - \mathbf{X}_k\|_F \leq \xi \|\mathbf{X}_r^* - \mathbf{X}_{k-1}\|_F + 2^P \mu_k \sqrt{1 + \delta_{\bar{r}_1}} \|\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e}\|_2,$$

where $\xi \triangleq 2^P(|1 - \mu_k| + \mu_k \delta_{\bar{r}_1})$. We consider two choices of step size:

- For $\mu_k = 1$, the assumption $\delta_{\bar{r}_1} < 2^{-P}$ implies $\xi_k = 2^P \delta_{\bar{r}_1} < 1$.
- If (7.17) is employed, it follows from the definition of the RIC that $(1 + \delta_{\bar{r}_1})^{-1} \leq \mu_k \leq (1 - \delta_{\bar{r}_1})^{-1}$. We then have two cases: (i) if $\mu_k > 1$, then $|1 - \mu_k| + \mu_k \delta_{\bar{r}_1} = \mu_k(1 + \delta_{\bar{r}_1}) - 1 \leq 2\delta_{\bar{r}_1}(1 - \delta_{\bar{r}_1})^{-1}$; (ii) similarly, if $\mu_k \leq 1$ then $|1 - \mu_k| + \mu_k \delta_{\bar{r}_1} = \mu_k(\delta_{\bar{r}_1} - 1) + 1 \leq 2\delta_{\bar{r}_1}(1 + \delta_{\bar{r}_1})^{-1} \leq 2\delta_{\bar{r}_1}(1 - \delta_{\bar{r}_1})^{-1}$. It can be checked that the condition $\delta_{\bar{r}_1} < 1/(2^{P+1} + 1)$ implies $2\delta_{\bar{r}_1}(1 - \delta_{\bar{r}_1})^{-1} < 2^{-P}$, thus yielding $\xi_k < 1$ in both cases.

Defining $\xi \triangleq \sup_k \xi_k < 1$, it follows that

$$\|\mathbf{X}_r^* - \mathbf{X}_k\|_F \leq \xi^k \|\mathbf{X}_r^* - \mathbf{X}_0\|_F + \left(\sum_{l=0}^{k-1} \xi^l\right) 2^P \sqrt{1 + \delta_{\bar{r}_1}} \|\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e}\|_2 \tag{7.45}$$

$$\leq \xi^k \|\mathbf{X}_r^* - \mathbf{X}_0\|_F + 2^P \frac{\sqrt{1 + \delta_{\bar{r}_1}}}{1 - \xi} \|\mathcal{A}(\mathbf{X}^* - \mathbf{X}_r^*) + \mathbf{e}\|_2, \tag{7.46}$$

as claimed. To conclude, note that the same reasoning holds for any other MPO $\pi = (p, p_2, \dots, p_P)$, in which case the role of $\delta_{\bar{\mathbf{r}}_1}$ is played more generally by $\delta_{\bar{\mathbf{r}}_p}$. \square

Corollary 7.8. Let $\mathbf{X}^* \in \mathcal{L}_{\mathbf{r}}$ and $\mathbf{y} = \mathcal{A}(\mathbf{X}^*)$. If \mathcal{A} has a RIC $\delta_{\bar{\mathbf{r}}_p} < 2^{-P}$, then the scheme (7.23) with fixed step size $\mu_k = 1$ and MPO $\pi = (p, p_2, \dots, p_P)$ converges to \mathbf{X}^* . If the step size formula (7.17) is used, then the same result holds with $\delta_{\bar{\mathbf{r}}_p} < 1/(2^{P+1} + 1)$.

Proof. Follows from taking $k \rightarrow \infty$ in (7.34) with $\mathbf{X}^* = \mathbf{X}_{\mathbf{r}}^*$ and $\mathbf{e} = \mathbf{0}$. \square

We would like to highlight the similarity of Theorem 7.7 with standard IHT results in CS, such as Theorem 7.2. In particular, the estimated rate of convergence depends on R_p . Moreover, if $\mathbf{X}^* \notin \mathcal{L}_{\mathbf{r}}$ and $\mathbf{e} = \mathbf{0}$, the algorithm approaches an estimate whose distance to \mathbf{X}^* is proportional to $\|\mathbf{X}^* - \mathbf{X}_{\mathbf{r}}^*\|_F$. Also, the gap between the bound and this optimal error grows with R_p , since then $\delta_{\bar{\mathbf{r}}_p}$ and ξ also grow.

It is important to bear in mind that, since the above results are RIC-based, they do not apply to TC, because sampling operators cannot possess small RICs (cf. discussion in Section 6.2.2).

7.2.2.1 Implied sampling bound

Ideally, performance bounds for mrank-based recovery should assume a small RIC of order (dR_1, \dots, dR_P) for some constant d . But, just as in Theorem 7.4, our results rely instead on a RIC having order $(N_1, \dots, N_{p-1}, 3R_p, N_{p+1}, \dots, N_P)$. Consequently, they unfortunately do not improve upon currently known sampling bounds. Indeed, applying (6.17) with $\delta = 2^{-P}$, the RIC condition in Theorem 7.7 is met with high probability provided that one takes⁴

$$M \geq M_{\min} = \mathcal{O}(4^P(R_p \bar{N}_p + R_p N_p + \sum_{q \neq p} N_q^2)) \quad (7.47)$$

subgaussian measurements, which grows much faster than the model complexity $\Phi(\mathbf{r})$ (see (6.14)). Formally demonstrating (near-)optimality of our approach seems presently out of reach. Nevertheless, our numerical simulations of Section 7.5.1 will show that in practice $M_{\min} = \mathcal{O}(\Phi(\mathbf{r})) = \mathcal{O}(\prod_p R_p + \sum_p N_p R_p)$ Gaussian measurements are sufficient for achieving recovery with SeMPIHT. We note that the same is observed also for both TIHT and MnRA.

7.2.3 Computing cost per iteration

The computing effort involved in the use of $\mathcal{S}_{\mathbf{r}}$ is given by (7.26). Calculation of the argument of $\mathcal{S}_{\mathbf{r}}$ can be split into three stages: (i) computing the gradient of J , (ii) calculating the step

⁴[167, Th. 2] states that $\delta_{\mathbf{r}} \leq \delta$ if a bound of the form $M \geq \mathcal{O}(\delta^{-2}(R^P + PNR) \log(P))$ is met, where $\mathbf{r} = (R_1, \dots, R_P)$, $R = \max_p R_p$ and $N = \max_p N_p$. Nonetheless, an inspection of its proof reveals that this bound can be refined as $M \geq \mathcal{O}(\delta^{-2}(\prod_p R_p + \sum_p N_p R_p) \log(P))$, which for fixed P and $\mathbf{r} = \bar{\mathbf{r}}_p = (N_1, \dots, N_{p-1}, 3R_p, N_{p+1}, \dots, N_P)$ implies (7.47). The refinement of the term PNR is mentioned in [167].

size μ_k and (iii) calculating the sum $\mathbf{v}_0 = \mathbf{x}_{k-1} - \frac{\mu_k}{2} \nabla J(\mathbf{x}_{k-1})$. Stage (i) requires $\mathcal{O}(M\bar{N})$ operations for unstructured (*e.g.*, Gaussian) operators, which can be alleviated by working with structured MOs. For instance, it requires $\mathcal{O}(M)$ in TC, while a cost of $\mathcal{O}(\bar{N} \log(\bar{N}))$ is achieved when random partial Fourier or noiselet measurements are taken by means of fast transform algorithms (see, *e.g.*, [125, 172]). The cost of stage (ii) depends on the step size selection strategy, and thus we postpone its discussion to Section 7.3. Finally, (iii) generally takes $\mathcal{O}(\bar{N})$ operations. In TC, this cost drops to $\mathcal{O}(M)$ because the gradient is sparse (due to the form of the SO).

7.2.4 Stopping criteria

In a practical implementation of (7.23), one needs to design appropriate criteria for stopping the algorithm when convergence is nearly attained, or when it fails to converge. To this end, at each iteration k we check whether the condition

$$\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_F \leq \epsilon \|\mathbf{x}_{k-1}\|_F \quad (7.48)$$

is satisfied for two consecutive estimates. If so, convergence is declared and the algorithm stops. Otherwise, it keeps running until a maximum number of iterations K_{\max} is met.

Although (7.48) does not necessarily imply proximity to a (local) minimum, it is our experience that this criterion works well in practice, provided an appropriate value is chosen for ϵ . When $\mathbf{y} = \mathcal{A}(\mathbf{x}^*)$ and $\mathbf{x}^* \in \mathcal{L}_{\mathbf{r}}$, then any reasonable value (*i.e.*, sufficiently small but above machine precision) is suitable, because as $k \rightarrow \infty$, $\|\mathbf{x}_{k-1}\|_F \rightarrow \|\mathbf{x}^*\|_F$ and $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_F \rightarrow 0$; hence, (7.48) will eventually be satisfied. However, in the more realistic and practically relevant setting where $\mathbf{x}^* \approx \mathbf{x}_{\mathbf{r}}^* \in \mathcal{L}_{\mathbf{r}}$, and/or $\mathbf{e} \neq \mathbf{0}$, choosing an appropriate ϵ becomes a subtler issue. In practice, one typically must tune ϵ empirically.

7.3 Step size selection

The issue of step size selection is of great importance in IHT [192]. On the one hand, μ_k should be sufficiently large to accelerate convergence. In particular, the requirement $J_k(x) > \mu_k J(x)$ for all $x \neq x_{k-1}$ can be relaxed, since it is sufficient but not necessary for objective function decrease. On the other hand, too large steps may cause the algorithm to diverge. In addition, invariance with respect to the scaling of the MO is desirable, which is not possible with a fixed step size. To pursue these goals, some adaptive strategies have been developed in the literature.

The normalized iterative hard thresholding (NIHT) for CS was proposed in [16], motivated by the poor performance of IHT with unit step size, which succeeds only in highly favorable regimes, and by its lack of robustness *vis-à-vis* the scaling of \mathbf{A} . The idea is to

calculate $\mu_k = \|\nabla J(\mathbf{x}_k)\|_2^2 \|\mathbf{A} \nabla J(\mathbf{x}_k)\|_2^{-2}$, the corresponding next estimate \mathbf{x}_k and then keep the latter if $\text{supp}(\mathbf{x}_k) = \text{supp}(\mathbf{x}_{k-1})$, in which case the step is optimal. Otherwise, μ_k is kept only if

$$\mu_k < (1 - c) \frac{\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2^2}{\|\mathbf{A}(\mathbf{x}_k - \mathbf{x}_{k-1})\|_2^2} \quad (7.49)$$

for a given positive constant $c < 1$. If (7.49) does not hold, then μ_k is repeatedly decreased through division by some constant $\kappa > 1$ and new candidate estimates \mathbf{x}_k are computed until (7.49) is met. As can be checked from (7.3), (7.49) implies $J(\mathbf{x}_k) < J(\mathbf{x}_{k-1})$ for $\mathbf{x}_k \neq \mathbf{x}_{k-1}$. Furthermore, it guarantees convergence of the algorithm and, as empirically verified, markedly improves its recovery performance [16].

Similar ideas were applied to LRMR in [192]. In this case, the role of vector support is played by row and column spaces of the low-rank estimate. As the algorithm progresses, two consecutive estimates are increasingly likely to have approximately the same column and row spaces (though they are never really identical, because subspaces vary continuously). Thus, the algorithm of [192] simply uses

$$\mu_k = \frac{\|\mathcal{P}_{k-1}(\nabla J(\mathbf{X}_{k-1}))\|_F^2}{\|\mathcal{A}\mathcal{P}_{k-1}(\nabla J(\mathbf{X}_{k-1}))\|_2^2}, \quad (7.50)$$

where \mathcal{P}_{k-1} denotes the orthogonal projector onto the column space of \mathbf{X}_{k-1} . Notwithstanding the lack of a verification akin to (7.49), this algorithm provably achieves perfect recovery (in the ideal case) when the MO has a 3R-RIC $\delta_{3R} < 1/5$ [192].

7.3.1 Step size selection in low-rank tensor recovery

The step sizes generated by TIHT via (7.16) are often too conservative, which considerably slows down convergence. In particular, when tackling a TC problem, property (6.12) implies that (7.16) delivers a constant unit step, because then $\nabla J(\mathbf{X}_k) = 2((\mathbf{X}_k)_\Omega - (\mathbf{X}^*)_\Omega)$, and hence

$$\mu_k = \frac{\|(\mathbf{X}_k)_\Omega - (\mathbf{X}^*)_\Omega\|_2^2}{\|\mathcal{A}((\mathbf{X}_k)_\Omega - (\mathbf{X}^*)_\Omega)\|_2^2} = 1. \quad (7.51)$$

This causes inefficient behavior especially for small ρ , *i.e.*, when few measurements are available.

To address this shortcoming, the proponents of TIHT have formulated the NTIHT algorithm, where formula (7.50) is extended to the higher-order setting by projecting the gradient onto the tensor space formed by the most recently estimated modal subspaces. This is the idea underlying (7.17). It is thus optimal only if the modal subspaces do not change from one iteration to the next, which evidently happens only approximately. Nonetheless, it usually delivers a quite satisfying convergence speed, at a cost of

$$\mathcal{O}\left(\sum_{p=1}^P R_1 \dots R_p N_p \dots N_P\right) + \mathcal{O}\left(\sum_{p=1}^P N_1 \dots N_p R_p \dots R_P\right) + c_{\mathcal{A}} + \mathcal{O}(M) + \mathcal{O}(\bar{N}) \quad (7.52)$$

operations per iteration, where the sum is associated with the multilinear transformations of (7.17), $c_{\mathcal{A}}$ denotes the cost of applying the MO \mathcal{A} (as discussed in Section 7.2.3) and the last two terms are related with the calculation of the norms.

7.3.2 A heuristic for improving step size selection

In [66], we have proposed a step size selection heuristic which relies upon an extension of condition (7.49) to the tensor setting. Originally, this heuristic, which we call improved step size (ISS), was employed together with the THOSVD operator $\mathcal{H}_{\mathbf{r}}$, yielding the ISS-TIHT algorithm. In what follows, its rationale is described.

In the IHT framework described in Section 7.1, it clearly suffices to guarantee that the inequality $\mu_k J(x_k) < J_k(x_k)$ holds at x_k for deriving $J(x_k) < J(x_{k-1})$, instead of requiring that $J_k(x)$ majorizes $\mu_k J(x)$ for all x . Specifically, in the tensor setting one can check whether

$$\mu_k < \omega(\mu_k) = \frac{\|\mathbf{X}_k - \mathbf{X}_{k-1}\|_F^2}{\|\mathcal{A}(\mathbf{X}_k - \mathbf{X}_{k-1})\|_2^2}, \quad (7.53)$$

since then $\mu_k \|\mathcal{A}(\mathbf{X}_k - \mathbf{X}_{k-1})\|_2^2 < \|\mathbf{X}_k - \mathbf{X}_{k-1}\|_F^2$, which together with (7.3) implies $\mu_k J(\mathbf{X}_k) < J_k(\mathbf{X}_k)$. The notation $\omega(\mu_k)$ emphasizes that the bound on μ_k depends on μ_k itself. We have observed that TIHT's formula (7.16) usually does not violate it. However, (7.16) often yields $\mu_k \ll \omega(\mu_k)$, while empirical evidence suggests the optimal step mostly lies closer to $\omega(\mu_k)$.

Our proposed step selection heuristic is rather simple: given a fixed α such that $0 \ll \alpha < 1$, one checks whether the candidate μ_k satisfies

$$\alpha\omega(\mu_k) \leq \mu_k < \omega(\mu_k), \quad (7.54)$$

keeping its associated estimate \mathbf{X}_k when it does. Otherwise, we simply set $\mu_k = \beta\omega(\mu_k)$ for some $\beta \in (\alpha, 1)$, compute a new \mathbf{X}_k and repeat the process. The initial candidate step size can be computed, *e.g.*, via (7.16). As there is no guarantee of finding a step fulfilling (7.54) with this procedure, we establish a maximum number of trials L , after which we keep the largest generated step size satisfying the upper bound in (7.54). If none of them does, we take the smallest candidate step and proceed as in NIHT [16], reducing it via division by a factor $\kappa > 1$ until the upper bound is verified.

We point out that the idea of choosing a new candidate for the step size as $\beta\omega(\mu_k)$ is already suggested in [16] with $\beta = 1$, but only to enforce the upper bound $\omega(\mu_k)$. In the case of TIHT, enforcing also the lower bound of (7.54) substantially accelerates convergence (which also holds true for SeMPIHT). This is illustrated by Fig. 7.1, where we display the resulting $J(\mathbf{X}_k)$ for a range of values of μ_k . The step sizes given by formulas (7.16) and (7.17) are indicated, as well as the bounds relative to the latter. These values were observed in one iteration of an actual run of ISS-TIHT with $\rho = 0.3$, $\alpha = 0.5$ and $\beta = 0.7$. As we can see, TIHT's choice μ_{T} is far below the optimal step size. NTIHT's formula, in its turn, yields

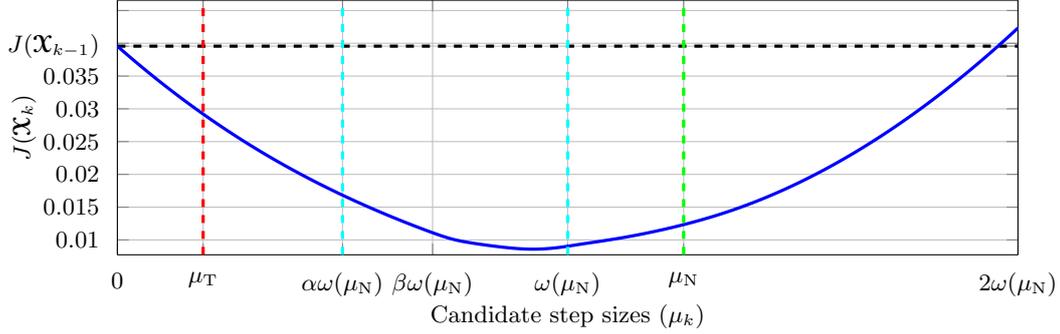


Figure 7.1: Cost function value $J(\mathbf{x}_k)$ resulting from different choices of step size in one iteration of **TIHT**: μ_T denotes the step given by (7.16); while μ_N denotes the step given by (7.17). In this example, $\alpha = 0.5$ and $\beta = 0.7$. A similar behavior is observed for **SeMPIHT**.

a step μ_N which violates (7.54), and thus is rejected. Note that the next candidate step $\beta\omega(\mu_N)$ is better than both μ_T and μ_N (also, it satisfies its own bounds). Alike situations also take place when running **SeMPIHT**.

7.4 Gradual rank increase heuristic

More often than not, tensors measured in applications possess modal singular spectra which decay steadily, instead of having an exactly low mrank. In that case, gradually increasing the mrank of the estimated model along iterations can improve recovery [123]. We pursue this idea here, proposing a continuation technique, called gradual rank increase (**GRI**), which starts off with a small mrank and conducts the algorithm through increasingly complex estimates.

There are several ways in which one can implement a **GRI** scheme. A fairly simple one starts with a chosen mrank \mathbf{r}_1 and then runs **Algorithm 7.1** for a maximum of $K'_{\max} < K_{\max}$ iterations or until (7.48) is satisfied. The outcome $\hat{\mathbf{x}}_{\mathbf{r}_1}^*$ is then used to initialize a subsequent run in which the mrank components are set as $[\mathbf{r}_2]_p = \min\{[\mathbf{r}_{\max}]_p, [\mathbf{r}_1 + \mathbf{i}]_p\}$ for all p , where $\mathbf{i} \in \mathbb{N}^P$ is a prescribed increment and \mathbf{r}_{\max} is the (final) target mrank. This process is repeated until reaching \mathbf{r}_{\max} , at which point normal operation is resumed, as depicted in **Fig. 7.2**. Note that a sequence of increasingly complex estimates $\hat{\mathbf{x}}_{\mathbf{r}_t}^*$, $t = 1, 2, \dots$, is produced before outputting $\hat{\mathbf{x}}^* = \hat{\mathbf{x}}_{\mathbf{r}_{\max}}^*$.

A disadvantage of the above scheme is that one cannot control the increments of each mrank component separately. If, *e.g.*, $\mathbf{i} = \mathbf{1}$ and $[\mathbf{r}_{\max}]_p \ll [\mathbf{r}_{\max}]_q$, then the algorithm reaches $[\mathbf{r}_{\max}]_p$ many iterations before reaching $[\mathbf{r}_{\max}]_q$. But, we would rather assign to each component a growth rate proportional to its magnitude. To this end, we can check the convergence of each modal subspace basis matrix $\mathbf{U}^{(p)}$ separately. An even simpler

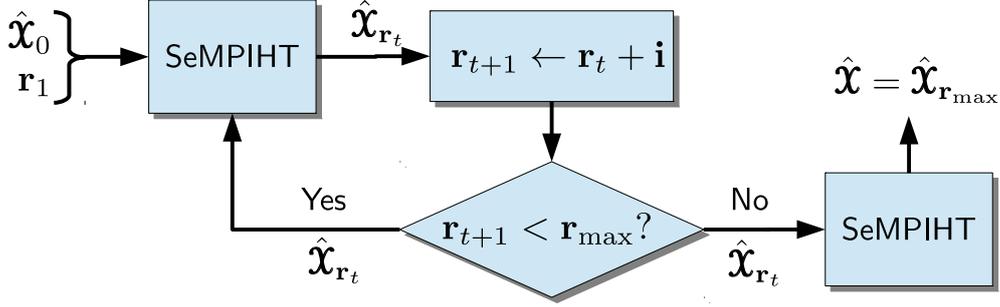


Figure 7.2: Diagram of SeMPIHT algorithm with GRI heuristic.

strategy is to predefine modal rank profiles specifying values for the mrank components at each iteration, until attaining the target mrank. From that point, normal operation is resumed. For instance, if $\mathbf{r}_{\max} = (R, 2R, 10R)$, then one can increment $[\mathbf{r}]_p$ by one unit at every $10R/[\mathbf{r}]_p$ iterations, so that \mathbf{r}_{\max} is attained at iteration $k = 10R$.

7.4.1 Effect of gradual rank increase

Though GRI strategies have been already employed in LRTR (see, *e.g.*, [92, 123]), no detailed discussion has been offered (to the best of our knowledge) on why it works in the first place. In the following, we attempt to fill this gap by drawing upon experimental results.

7.4.1.1 Tensor models

Two types of tensor models are considered in these experiments:

- A *type-1 (T1)* tensor consists of a Tucker model having a $R_1 \times \cdots \times R_P$ core and $N_p \times R_p$ factors. Thus, by construction, it belongs to $\mathcal{L}_{\mathbf{r}}$. All factors and the core have standard Gaussian entries. As a consequence, T1 tensors have highly concentrated nonzero modal singular values.
- A *type-2 (T2)* tensor generally has full mrank but exhibits exponentially decaying modal spectra. To impose this property, we adopt the Tucker model used in [213, Sec. 2.3], which has a $N_1 \times \cdots \times N_P$ Gaussian core and matrix factors $\mathbf{A}_p = \mathbf{Q}_p \mathbf{S}_p \in \mathbb{R}^{N_p \times N_p}$, where \mathbf{Q}_p is orthogonal and $\mathbf{S}_p = \text{Diag}(1, 2^{-\varphi}, \dots, N_p^{-\varphi})$, with $\varphi > 0$.

The typical spectral characteristics of T1 and T2 tensors are illustrated in Fig. 7.3(a). Specifically, it shows the average modal spectra of 500 realizations of $20 \times 20 \times 20$ T1 ($\mathbf{r} = (10, 10, 10)$) and T2 ($\varphi = 3$) tensors, which are normalized to have unit Frobenius norm. Numerically, the T2 tensors have full mrank. Consistently with the shown behavior, the mean squared error of the best mrank- (R, R, R) approximation as a function of R displays an abrupt variation for the T1 tensors; that of the T2 tensors decays steadily and smoothly.

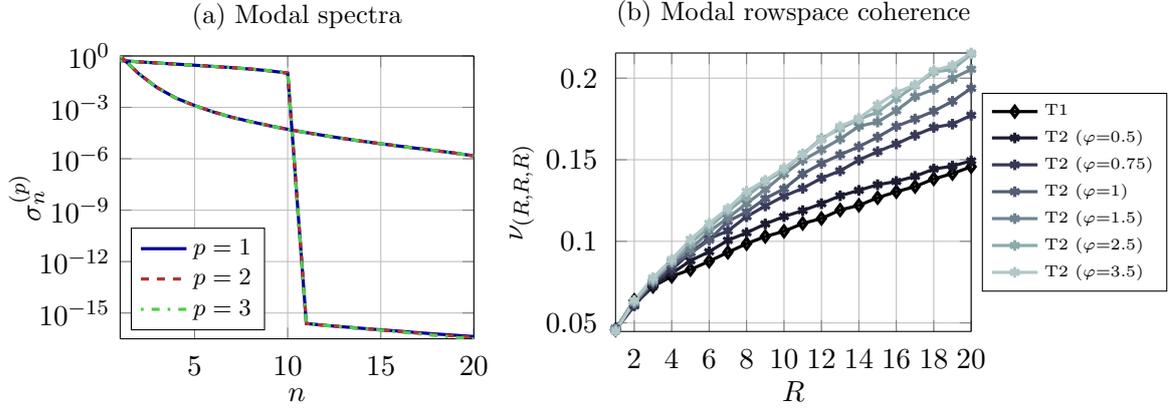


Figure 7.3: Typical behavior of the considered random tensor models: (a) modal spectra; (b) row space coherence (see (7.55)).

Another relevant property of these tensor models is highlighted in Fig. 7.3(b). Namely, we generated 500 realizations of T1 tensors of varying mrank $\mathbf{r} = (R, R, R)$ and T2 tensors with varying φ , all having dimensions $20 \times 20 \times 20$, and then plotted the average modal row space coherence of their approximate projections onto $\mathcal{L}_{\mathbf{r}}$, *i.e.*,

$$\nu_{\mathbf{r}}(\mathcal{X}) = \min_p \frac{R_p}{N_p} \mu(\mathcal{W}_p), \quad \text{where } \begin{cases} \mathcal{W}_p = \text{rowspace}((\mathcal{X}_{\mathbf{r}})_{(p)}), \\ \mathcal{X}_{\mathbf{r}} = \mathcal{S}_{\mathbf{r}}(\mathcal{X}) \end{cases} \quad (7.55)$$

and $\mu(\cdot)$ is as defined by (6.8). (For T2 tensors, note that $\mathcal{X}_{\mathbf{r}} \neq \mathcal{X}$). The (approximate) projection is performed because we are ultimately interested in the properties of the best mrank- \mathbf{r} approximation of \mathcal{X} , since it is this approximation which is sought by SeMPIHT, the difference $\mathcal{X} - \mathcal{X}_{\mathbf{r}}$ being regarded as modeling error (cf. Theorem 7.7). Fig. 7.3(b) indicates that the modal row space coherence of the (approximately) projected T2 tensors grows with φ . Also, the gap among the curves grows with R . As we shall see ahead, this has important negative implications when trying to complete T2 tensors sampled uniformly at random.

We would like to draw attention to the fact that, although the modal spectra of T2 tensors are more akin to those of most real-world tensors, to date most published works have exclusively considered T1 (or similar) tensors in computer experiments with synthetic data.

7.4.1.2 Experimental results

Our experiments consist of Monte Carlo simulations involving the recovery of $20 \times 20 \times 20$ tensors by employing Algorithm 7.1 with ISS.

We first employ Gaussian MOs. For each value of $\rho = M/20^3 \in \{0.10, 0.25, 0.40\}$, $N_r = 100$ realizations of an MO \mathcal{A} are generated by drawing the entries of its associated matrix $\mathbf{A} \in \mathbb{R}^{M \times 20^3}$ (which satisfies $\mathcal{A}(\mathcal{X}) = \mathbf{A} \text{vec}(\mathcal{X})$) from a zero-mean Gaussian distribution of

variance $1/M$. Each **MO** is then used to sense T1 tensors having $\text{mrank}(R, R, R)$, with $R \in \langle 15 \rangle$, and T2 tensors with spectral decay factors $\varphi \in \{\frac{3}{2}, \frac{5}{2}, \frac{7}{2}\}$. When recovering T1 tensors, the target mrank always matches $\text{mrank}(\mathcal{X}^*)$ and we set $K_{\max} = 1000$ and $\epsilon = 10^{-10}$. The algorithm is run once initialized with the null tensor (initialization I) and then three more times with random initializations.

In the recovery of T2 tensors, we vary the target $\text{mrank}(R, R, R)$ and run the algorithm twice for each R : once initialized with the null tensor (initialization I) and once using the solution obtained with $\text{mrank}(R-1, R-1, R-1)$ to initialize the run in which $\mathbf{r} = (R, R, R)$ (initialization II). Note that the latter initialization strategy is closely related to our **GRI** heuristic. Again, $K_{\max} = 1000$, but a specific ϵ was chosen for each combination of φ and ρ by a trial and error procedure.

At the end of each run, we compute the squared error of the resulting estimate $\hat{\mathcal{X}}_l^*$ with respect to the generated tensor \mathcal{X}_l^* , where l denotes the realization. The normalized mean squared error (**NMSE**), defined as

$$\text{NMSE} = \frac{1}{N_r} \sum_{l=1}^{N_r} \frac{\|\mathcal{X}_l^* - \hat{\mathcal{X}}_l^*\|_F^2}{\|\mathcal{X}_l^*\|_F^2}, \quad (7.56)$$

is shown in **Fig. 7.4**. In the case of T1 tensors, only the best outcome among the runs with initialization II is kept for computing (7.56). For T2 tensors, we plot also the **NMSE** of $\mathcal{S}_{\mathbf{r}}(\mathcal{X}_l^*)$, which gives an approximate lower bound. **Fig. 7.4(a)** displays a sharp transition from success to failure in the recovery of T1 tensors, which is a typical behavior in parsimonious signal recovery problems. Concerning T2 tensors, **Fig. 7.4(b)–(d)** show that the **NMSE** gets quite close to the lower bound when inside the region of successful recovery of T1 tensors (cf. **Fig. 7.4(a)**), regardless of the initialization. Beyond that region, a gap appears: results obtained with initialization I rapidly degrade, while those for initialization II degrade (or even improve) only slightly before stabilizing. The rate of deviation from the lower bound depends on ρ and φ , in conformity with (7.34).

The results of a similar experiment performed with (uniformly) random **SOs** are shown in **Fig. 7.5**. As we can see, transition into failure happens in **Fig. 7.5(a)** for $\rho = 0.10$ as soon as $R = 2$, against $R = 6$ in the Gaussian case. Also, the results are very poor for T2 tensors with initialization I, even in a favorable regime (*i.e.*, where recovery of T1 tensors succeeds). Moreover, the performance worsens as the singular values decay rate φ grows, which is explained by the behavior shown in **Fig. 7.3(b)**, as the recoverability of $\mathcal{X}_{\mathbf{r}}^*$ depends on \mathbf{r} and on some measure of coherence. The use of initialization II does a remarkable job in avoiding such a degradation. Indeed, the results for $\rho = 0.25$ and $\rho = 0.40$ are similar to those obtained with Gaussian sensing. For $\rho = 0.10$, not enough measurements seem to be available for achieving comparable results.

Let us now interpret these outcomes in light of the results of **Section 7.2.2**. Since $\mathcal{X}^* \in \mathcal{L}_{\mathbf{r}}$

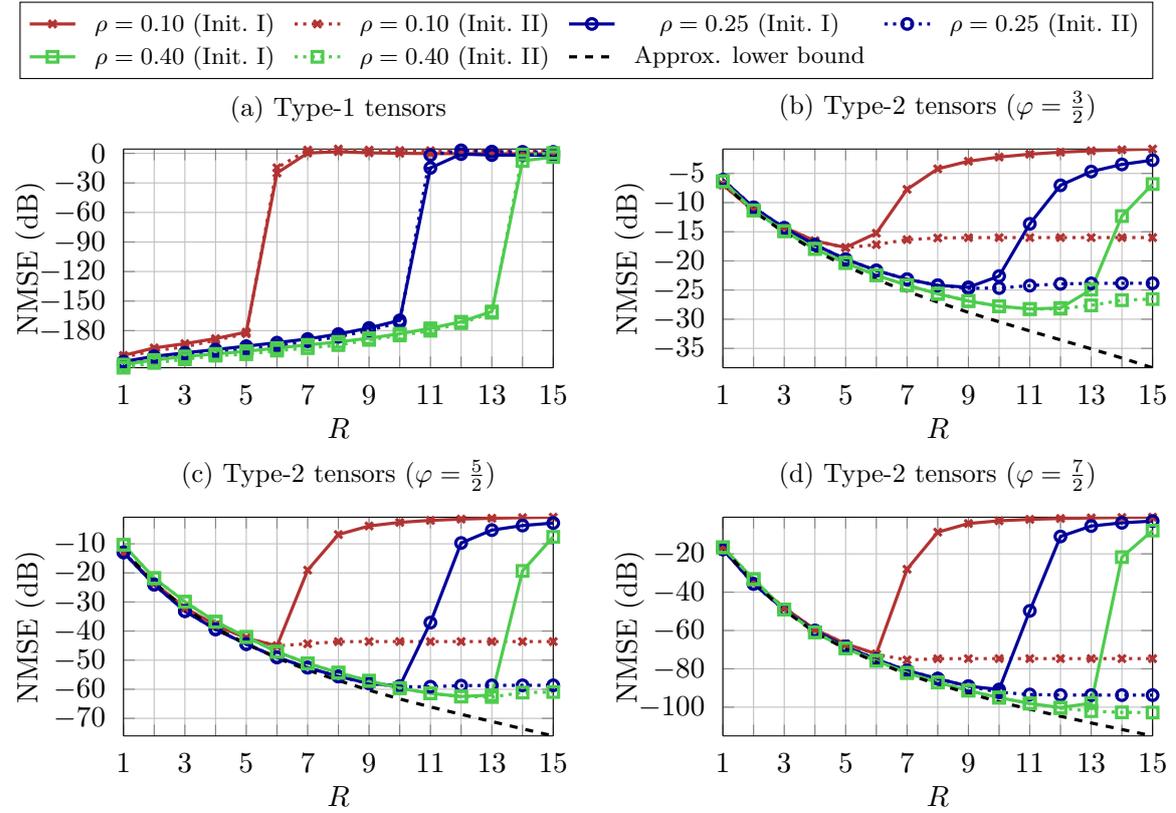


Figure 7.4: Recovery performance of SeMPIHT with Gaussian sensing. The approximate lower bound in (b)–(d) is the average of $\text{NSE}(\mathcal{S}_R(\mathcal{X}^*); \mathcal{X}^*)$.

in our experiment with T1 tensors, Corollary 7.8 guarantees convergence to the global minimizer \mathcal{X}^* whenever \mathcal{A} satisfies the stated RIC condition, regardless of the initialization (and despite the nonconvexity of (7.1)). Hence, when using Gaussian sensing, initialization plays no role in the recovery regime (with high probability), which is corroborated by Fig. 7.4(a). In the phase transition region, the influence of initialization comes into play, as apparently the insufficiency of measurements *vis-à-vis* the number of DOF causes convergence to local minima, with rapidly increasing probability as R grows. Similar remarks hold for T2 tensors, in that the iterates approach a ball centered at a best approximation $\mathcal{X}_r^* \in \mathcal{L}_r$ of \mathcal{X}^* regardless of the initialization for appropriate \mathcal{A} (cf. Theorem 7.7), which explains Fig. 7.4(b)–(d).

Now, for T2 tensors, gradually increasing the mrank can stabilize the approximation error when a too high mrank (with respect to ρ) is chosen. This apparently happens because, once the phase transition region is reached, the lack of sufficient information causes convergence to a local minimum not far from the initial point. In particular, when completing T2 tensors this continuation strategy delivers good results despite their non-ideal coherence properties. It also brings computational advantages, because the convergence rate ξ is faster and the cost of \mathcal{S}_r is reduced when \mathbf{r} has small components.

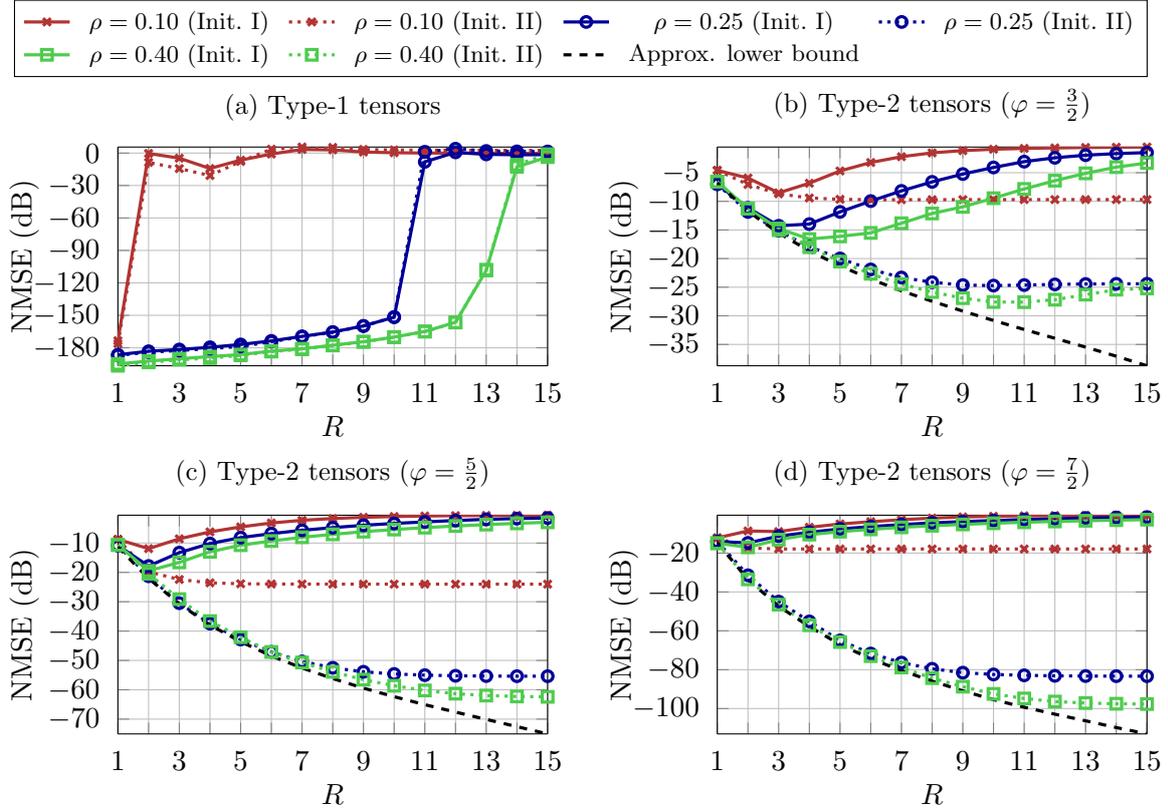


Figure 7.5: Recovery performance of SeMPIHT with sampling operators (TC). The approximate lower bound in (b)–(d) is given by $\text{NMSE}(\mathcal{S}_{\mathbf{r}}(\mathcal{X}^*); \mathcal{X}^*)$.

7.5 Simulation results

In the following experiments, $P = 3$, for simplicity. Also, ISS is always employed with parameters $L = 3$, $\alpha = 0.5$, $\beta = 0.7$ and, unlike [66], with initial candidate step size $\mu_k = 1$.

7.5.1 Estimation of sampling bounds

First, we numerically estimate how many measurements are necessary for recovering a model with a given complexity. More precisely, the idea is to find, for several values of ρ , the maximum normalized number of DOF $\bar{\Phi}(\mathbf{r}) = \Phi(\mathbf{r})/\bar{N}$ up to which recovery is highly likely. For simplicity, we take $N_1 = N_2 = N_3 = N$, and sort all possible values of $\bar{\Phi}(\mathbf{r})$ by considering every mrank $\mathbf{r} = (R_1, R_2, R_3)$ such that (i) $R_1 \leq R_2 \leq R_3$ and (ii) $R_3 \leq R_1 R_2$. This entails no loss of generality, as constraint (i) avoids redundant tuples, while constraint (ii) eliminates those which are not feasible.⁵ Then, for each $\rho \in \{0.05, 0.10, \dots, 1\}$, we start from the simplest model, $\mathbf{r} = (1, 1, 1)$, and generate 15 joint realizations of an MO \mathcal{A} and a T1 tensor

⁵Note that $\text{mrank}(\mathcal{X}) = \mathbf{r}$ is equivalent to the existence of a Tucker model constrained as discussed in Section 6.3.1.2, whose core can only have a mode-3 unfolding with orthogonal rows if $R_3 \leq R_1 R_2$.

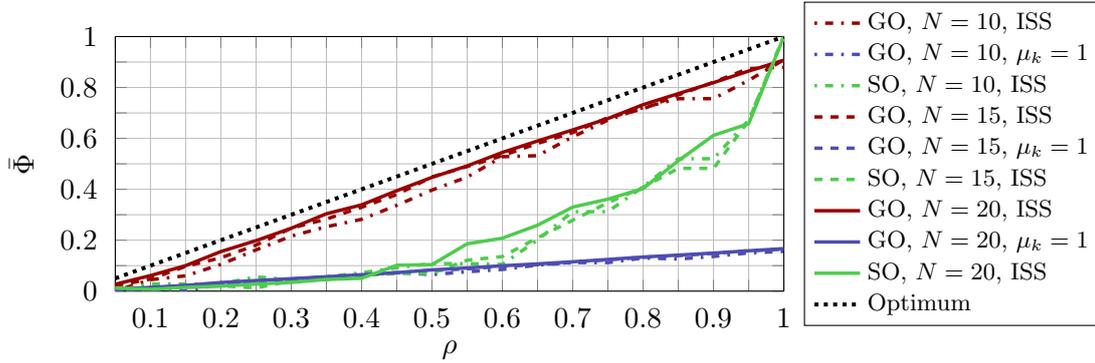


Figure 7.6: Estimated (normalized) number of DOF which can be recovered by SeMPIHT for each level of ρ , using Gaussian MOs (GO) and SOs. Recovery was successful in 15 consecutive realizations for values of $\bar{\Phi}$ below or over the curve.

$\mathbf{X}^* \in \mathcal{L}_{\mathbf{r}}$. Recovery of \mathbf{X}^* from $\mathbf{y} = \mathcal{A}(\mathbf{X}^*)$ is declared successful when $\text{NSE}(\hat{\mathbf{X}}^*; \mathbf{X}^*) \leq -90$ dB. If all 15 runs are successful, then the process is repeated with the next model of higher complexity (in terms of $\Phi(\mathbf{r})$). When failure occurs for some \mathbf{r}' , then the value $\Phi(\mathbf{r})/\bar{N}$ of the immediately less complex model is declared to be frontier of the recovery region. To reduce computing time, instead of starting from $\mathbf{r} = (1, 1, 1)$ for every level of ρ , we start from the mrank tuple associated with the frontier obtained for the immediately preceding undersampling rate (*i.e.*, for $\rho - 0.05$). The stopping criteria are set as $\epsilon = 10^{-8}$ and $K_{\max} = 1500$. Gaussian MOs and SOs are generated as described in Section 7.4.1.2.

The results obtained for $N \in \{10, 15, 20\}$ are shown in Fig. 7.6. When using Gaussian operators (GO), the maximum $\bar{\Phi}(\mathbf{r})$ clearly grows approximately linearly with ρ for all N . Moreover, the improvement due to ISS is visible, as the slope becomes much higher (about 0.9) than with fixed step size (about 0.17). Hence, $M \geq M_{\min} = \mathcal{O}(\Phi(\mathbf{r}))$ Gaussian measurements (are highly likely to) suffice for recovery, with $M_{\min} \approx \frac{1}{0.9}\Phi(\mathbf{r}) = 1.11\Phi(\mathbf{r})$ when using ISS and $M_{\min} \approx \frac{1}{0.17}\Phi(\mathbf{r}) = 5.88\Phi(\mathbf{r})$ when $\mu_k = 1$. So, despite the quite loose sampling bounds implied by Theorem 7.7, in practice SeMPIHT with ISS succeeds for a quasi-optimal number of Gaussian measurements. On the other hand, the relation between $\bar{\Phi}(\mathbf{r})$ and ρ is no longer linear in TC.

For the sake of comparison, the same procedure is applied with $N = 20$ to ISS-TIHT [66], MnRA [217], geomCG [123], TMac [213] and the ADMM scheme based on SNN minimization (SNNM) of Algorithm 6.1. In the latter, the penalty parameter η is adapted along the iterations to accelerate convergence, as discussed in [22, Sec. 3.4.1], and observations are taken as constraints ($\lambda \rightarrow 0$). Having been devised specifically for TC, the performances of

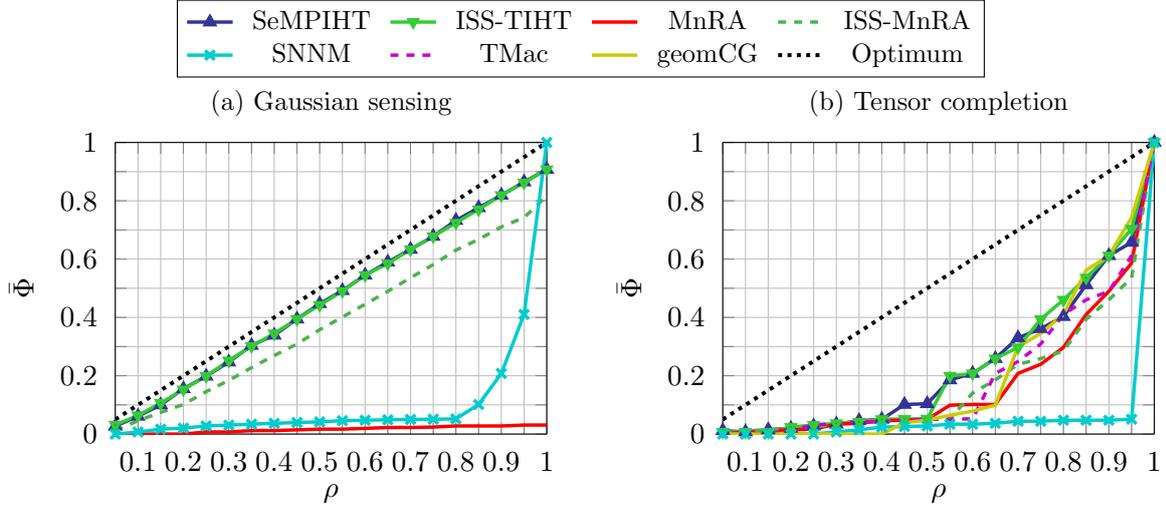


Figure 7.7: Estimated (normalized) number of DOF which can be recovered for each level of ρ , using (a) Gaussian MOs and (b) SOs. The target tensors are $20 \times 20 \times 20$. Recovery was successful in 15 consecutive realizations for values of $\bar{\Phi}$ below or over the curve.

geomCG and TMac are only evaluated⁶ with SOs. For a fair comparison, a variant of MnRA using ISS is also included. All methods are initialized with the null tensor and cannot exceed $K_{\max} = 1500$ iterations.

This comparison is shown on Fig. 7.7. In the Gaussian sensing setting of Fig. 7.7(a), the sampling requirements of SeMPIHT and ISS-TIHT are almost identical, while those of ISS-MnRA are a little stricter. Though MnRA with fixed step size $\mu_k = 1$ displays quite a poor performance, $\bar{\Phi}$ still grows roughly linearly with ρ . In its turn, the behavior of the SNNM approach is markedly different, abruptly improving in the region $\rho > 0.8$. Such a nonlinear relation is expected, as discussed in Section 6.3.3.1. In the TC scenario of Fig. 7.7(b), SeMPIHT and ISS-TIHT have generally the least strict sampling requirements, with geomCG competing closely for $\rho \geq 0.7$. TMac’s performance is less satisfying, but slightly better than that of MnRA for $\rho \geq 0.65$. Here, ISS does not improve MnRA’s sampling requirements. Finally, the SNNM approach displays an outstandingly poor performance in comparison with the others.

7.5.2 Convergence and computational cost

In order to evaluate the studied algorithms with respect to their convergence speed and computational cost, they are applied to recover 60 realizations of $N \times N \times N$ T1 and T2 tensors sensed by Gaussian and SOs. At each iteration, we measure the quadratic error

⁶We employ the implementations provided by their authors, obtained from <http://anchp.epfl.ch/geomCG> and <http://www.math.ucla.edu/~wotaoyin/papers/tmac.html>. Yet, we have replaced geomCG’s MEX routines by Matlab code, which turns out to be much faster in our setting (as suggested by [55]).

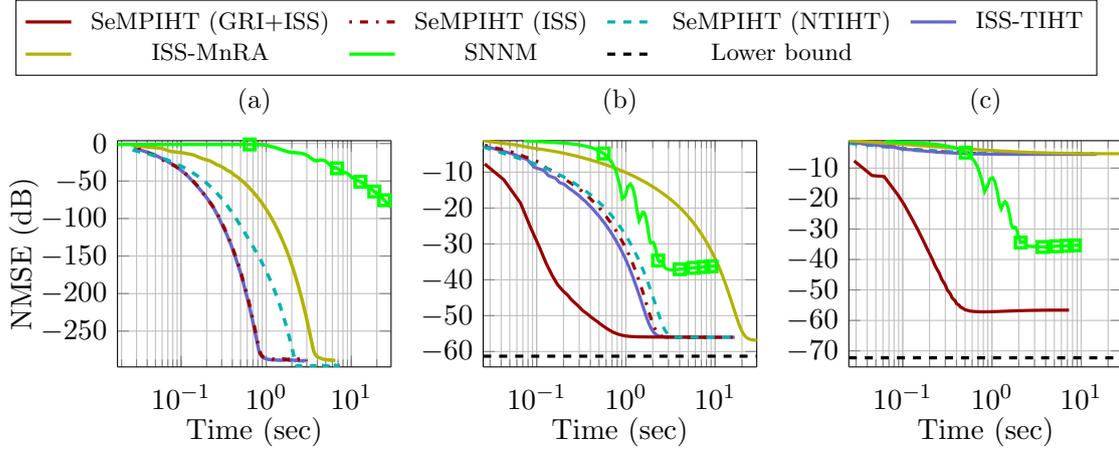


Figure 7.8: Convergence of several algorithms in a Gaussian sensing scenario where $\rho = 0.25$ and $N_1 = N_2 = N_3 = 20$: (a) T1 tensors of mrank $\mathbf{r} = (3, 3, 3)$; (b) T2 tensors ($\varphi = 2.5$) modeled with mrank $\mathbf{r} = (9, 9, 9)$; (c) T2 tensors ($\varphi = 2.5$) modeled with mrank $\mathbf{r} = (13, 13, 13)$.

of the current solution with respect to \mathcal{X}^* and also the time spent. Results concerning T2 tensors are displayed along with an average (approximate) lower bound calculated as in Section 7.4.1.2.

SeMPIHT is run both with the ISS heuristic and with the NTIHT step size selection rule (7.17). When (and only when) T2 tensors are recovered, SeMPIHT is also run with GRI (in which case the ISS heuristic is used). The tolerance parameter used in geomCG’s rank increase condition (cf. [123, Eq. 4.2]) is set as $\delta = 0.1$. TMac’s adaptive weight heuristic is used, starting with weights $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$ [213]. Algorithm 6.1 for SNNM is again run with $\lambda \rightarrow 0$ and penalty parameter adapted as described by [22, Sec. 3.4.1].

We start by considering Gaussian operators. In this case, $N = 20$, $\rho = 0.25$, T1 tensors have mrank $\mathbf{r} = (3, 3, 3)$, and T2 tensors have decay parameter $\varphi = 2.5$. The results for T1 tensors are shown in Fig. 7.8(a). In this scenario, both SeMPIHT (with ISS) and ISS-TIHT outperform the other algorithms, having practically indistinguishable performances. This happens because the cost of applying the Gaussian MO dominates that of the projection. Fig. 7.8(b) displays the results obtained for T2 tensors modeled with the mrank $\mathbf{r} = (9, 9, 9)$. The GRI used in SeMPIHT follows the first procedure described in Section 7.4, with $K'_{\max} = 1$, $\mathbf{i} = (1, 1, 1)$ and $\mathbf{r}_1 = (1, 1, 1)$. From Fig. 7.4(c), one can verify that $\Phi(\mathbf{r})/(\rho\bar{N})$ falls inside the recovery regime of SeMPIHT, and thus it reaches quite close to the bound, as well as the other algorithms. Among them, SeMPIHT with GRI is clearly the fastest to converge. Now, in Fig. 7.8(c), the model mrank is set as $\mathbf{r} = (13, 13, 13)$, which yields too high a value of $\Phi(\mathbf{r})$ for $\rho = 0.25$. In this case, we have set $K'_{\max} = 2$. Note that the GRI technique prevents the degradation brought by mrank overestimation, while the performances of the other IHT

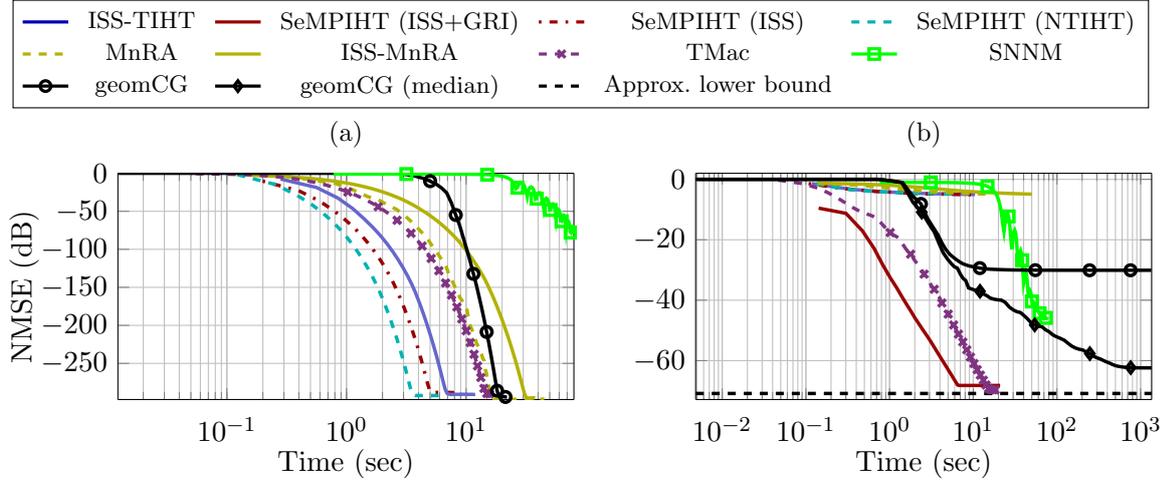


Figure 7.9: Convergence of several algorithms in a TC scenario where $\rho = 0.2$ and $N_1 = N_2 = N_3 = 100$: (a) T1 tensors of mrank $\mathbf{r} = (10, 10, 10)$; (b) T2 tensors ($\varphi = 2$) modeled with mrank $\mathbf{r} = (30, 30, 30)$.

algorithms are severely deteriorated. This robustness with respect to mrank overestimation is valuable, since in practice one generally does not know which mrank values fall inside the recovery region for a given M .

Fig. 7.9 displays the results obtained for TC, with $N = 100$ and $\rho = 0.2$. The T1 tensors and T2 tensors are generated with, respectively, $\mathbf{r} = (10, 10, 10)$ $\varphi = 2$. Upon inspection of Fig. 7.9(a), it is clear that both variants of SeMPIHT are more efficient than all other algorithms in recovering T1 tensors. The gap between SeMPIHT with ISS and ISS-TIHT is due to the reduced cost of the thresholding operator. The NTIHT variant is even faster in this scenario. For the recovery of T2 tensors, the mrank is set as $\mathbf{r} = (30, 30, 30)$, and we choose $K'_{\max} = 1$. Both geomCG and TMac are run with their mrank increase heuristics [123, 213], with initial mrank $\mathbf{r}_1 = (1, 1, 1)$ and unit increments. SeMPIHT uses the same settings. Fig. 7.9(b) show that the IHT algorithms without GRI clearly fail, which is due to the non-ideal coherence properties of the T2 tensors. Among the others, SeMPIHT with GRI provides the best performance, followed by TMac. Unlike the other methods, geomCG's results have large variance, and so we also plot in Fig. 7.9(b) its median results, which are much more reasonable in terms of final NMSE, but at the expense of a large computing cost.

7.6 Concluding remarks

In this chapter, we have proposed an IHT algorithm for the LRTR problem, relying on SeMP for hard thresholding. It has smaller cost per iteration compared to other tensor IHT algorithms, while delivering better or comparable convergence speed. Also, the sequential

optimality of the modal projections in **SeMP** allows deriving performance bounds based solely on **RICs**, which is still an open problem for **THOSVD**. Though the exploited **RICs** imply largely loose sampling bounds for certain random (*e.g.*, Gaussian) measurement ensembles, our simulations show perfect recovery is achieved by **SeMPIHT** (and also by **TIHT** and **MnRA**, for that matter) with a number of Gaussian measurements which scales linearly with the intrinsic complexity of the model.

Step size and gradual rank increase heuristics were also proposed. The former accelerates convergence by imposing upper and lower bounds to candidate step sizes, while the latter prevents degradation due to mrank overestimation and further accelerates the algorithm when the sought tensor has fast decaying modal spectra. **GRI** is especially important in **TC**, as it can avoid degradation due to non-ideal coherence properties of the measured tensors.

Feedback-controlled imputation schemes for tensor completion

In this chapter, we develop an iterative TC scheme combining a single imputation technique with soft thresholding of the HOSVD core. This thresholding operation arises as an approximate solution of a ℓ_1 -regularized least-squares problem which is well-suited when the target tensor admits a compressible orthogonal Tucker model. We show the link between this property and a fast decay of its modal singular spectra. A connection between the ADMM algorithm and iterative single imputation schemes is also elaborated, with an interpretation in terms of a feedback-controlled system. Then, we propose an improved version of our algorithm incorporating such a feedback control mechanism. Finally, simulation results are presented, validating the potential of the proposed method.

Contents

8.1	Single imputation schemes for tensor completion	148
8.2	Single imputation with ℓ_1-regularized least squares	150
8.3	Imputation schemes with feedback	155
8.4	Experimental validation	160
8.5	Conclusions	166

8.1 Single imputation schemes for tensor completion

In Section 6.3.3.3, we mentioned that the approach of [194] couples a single imputation technique with the ALS algorithm in order to fill in the missing entries of a tensor $\mathcal{X}^* \in \mathcal{T} = \bigotimes_{p=1}^P \mathbb{R}^{N_p}$ under the assumption of low tensor rank. It can be summarized along the following lines:

1) At iteration k , one forms the estimate

$$[\mathcal{Z}_k]_{n_1, \dots, n_P} = \begin{cases} [\mathcal{X}^*]_{n_1, \dots, n_P}, & (n_1, \dots, n_P) \in \Omega, \\ [\mathcal{X}_k]_{n_1, \dots, n_P}, & (n_1, \dots, n_P) \notin \Omega, \end{cases} \quad (8.1)$$

where \mathbf{X}_k is given by the tensor model estimated at iteration $k - 1$. This means that the components of \mathbf{Z}_k corresponding to the missing entries of \mathbf{X}^* are imputed (only once, hence the term “single imputation” which is borrowed from the statistics literature) according to the most recent model estimate.

2) Then, the following least-squares problem is posed:

$$\min_{\mathbf{A}^{(p)} \in \mathbb{R}^{N_p \times R}} \left\| \mathbf{Z}_k - \left[\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)} \right] \right\|_F^2, \quad (8.2)$$

for some chosen R . In order to approximately solve it, [194] employs a single iteration of the ALS algorithm described in Section 3.3.1.1.

3) Subsequently, one computes the reconstructed tensor

$$\mathbf{X}_{k+1} = \left[\hat{\mathbf{A}}_{k+1}^{(1)}, \dots, \hat{\mathbf{A}}_{k+1}^{(P)} \right], \quad (8.3)$$

where the factors $\hat{\mathbf{A}}_{k+1}^{(p)}$ are the approximate minimizers of (8.2) obtained at step 2, and then goes back to step 1 until some stopping criterion is met.

An interpretation of the above described algorithm as an instance of the expectation minimization (EM) method [71] is discussed in [194]. It holds when the residuals are zero-mean Gaussian i.i.d. variables, which implies that maximizing the conditional expectation of the log-likelihood function is equivalent to minimizing the least-squares cost function of (8.2). Formulating that problem thus amounts to the E-step, while solving it corresponds to the M-step.

A similar algorithm is devised in [121], where instead of a CPD model, a Tucker model is employed. Its first step is identical. In step 2, the least-squares problem has the form:

$$\min_{\substack{\mathcal{G} \in \bigotimes_{p=1}^P \mathbb{R}^{R_p} \\ \mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{R}^{N_p})}} \left\| \mathbf{Z}_k - \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)} \right\|_F^2 = \max_{\mathbf{U}^{(p)} \in \mathcal{V}_{R_p}(\mathbb{R}^{N_p})} \left\| \mathbf{Z}_k \times_{p=1}^P \left(\mathbf{U}^{(p)} \right)^T \right\|_F^2, \quad (8.4)$$

where the modal ranks R_p are fixed *a priori* and the equality is due to Proposition 2.54. Finally, given an approximate solution $(\hat{\mathbf{U}}_{k+1}^{(1)}, \dots, \hat{\mathbf{U}}_{k+1}^{(P)})$ of (8.4), step 3 is carried out as

$$\mathbf{X}_{k+1} = \mathbf{Z}_k \times_{p=1}^P \left[\hat{\mathbf{U}}_{k+1}^{(p)} \left(\hat{\mathbf{U}}_{k+1}^{(p)} \right)^T \right]. \quad (8.5)$$

Again, this can be interpreted as application of the EM scheme, provided the residuals satisfy the same assumptions mentioned above. To obtain an approximate solution of (the NP-hard problem) (8.4), [121] resorts to the standard THOSVD defined in Section 2.6.2.1.

We point out that, in the TC setting, the TIHT algorithm is equivalent to this THOSVD-based single imputation scheme. The equivalence is due to property (6.12), which entails $\mu_k = 1$ (see (7.51)), implying that the tensor \mathbf{V}_k appearing in (7.15) is given by $\mathbf{V}_k = \mathbf{X}_k + \mu_k \mathcal{A}^\dagger (\mathbf{y} - \mathcal{A}(\mathbf{X}_k)) = \mathbf{X}_k + (\mathbf{X}^*)_\Omega - (\mathbf{X}_k)_\Omega$. This is clearly equivalent to the definition of \mathbf{Z}_k in (8.1).

8.2 Single imputation with ℓ_1 -regularized least squares

The rationale of our proposed approach shares common ground with the methods we have just described. Namely, it is also an iterative procedure coupling a single imputation technique with the approximate estimation of a parsimonious model, which we now describe.

8.2.1 Sparse and compressible orthogonal Tucker models

Recall from Section 2.6.1.1 that an OTD of a tensor \mathbf{X}^* can be written as

$$\mathbf{X}^* = \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)} = \sum_{n_1=1}^{N_1} \cdots \sum_{n_P=1}^{N_P} g_{n_1, \dots, n_P} \bigotimes_{p=1}^P \mathbf{u}_{n_p}^{(p)}, \quad (8.6)$$

where, in general, $\mathcal{G} \in \mathcal{T}$ and $\mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{R}^{N_p})$ for all $p \in \langle P \rangle$. Since this expression can be seen as the representation of \mathbf{X}^* under orthonormal bases $\mathbf{U}^{(p)}$, we can think of the cardinality of the core \mathcal{G} (i.e., its number of nonzero elements) as a measure of complexity of \mathbf{X}^* . When $\text{card}(\text{vec}(\mathcal{G})) \ll \bar{N}$, we say that (8.6) is a sparse OTD model of \mathbf{X}^* .

Clearly, if $\text{rank}_p(\mathbf{X}^*) \ll N_p$ for some $p \in \langle P \rangle$, then \mathcal{G} will be sparse. Thus, this model obviously applies to low-rank tensors. However, we aim here at the more realistic situation in which \mathcal{G} is not exactly sparse, but only *compressible* in some sense. In the following, this notion is formalized and connected to the definition of T2 tensors of Chapter 7.

8.2.1.1 Compressibility of vectors in the ℓ_1 sense

Recall that

$$\forall \mathbf{x} \in \mathbb{R}^N, \quad \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1 \leq \sqrt{N} \|\mathbf{x}\|_2. \quad (8.7)$$

The lower bound is attained if there is a unique $n \in \langle N \rangle$ such that x_n is nonzero, since in that case $\|\mathbf{x}\|_2 = \sqrt{x_n^2} = |x_n| = \|\mathbf{x}\|_1$. Therefore, it is attained when \mathbf{x} has its energy *maximally concentrated*. When all components of \mathbf{x} have equal magnitude, i.e., $|x_n| = x_0 \geq 0$ for all $n \in \langle N \rangle$, then $\|\mathbf{x}\|_1 = N|x_0| = \sqrt{N} \sqrt{N|x_0|^2} = \sqrt{N} \|\mathbf{x}\|_2$, and thus the upper bound is attained. In other words, the energy of \mathbf{x} is *minimally concentrated* in that case.

The above observations motivate the following definitions.

Definition 8.1. The *compressibility measure* $\Gamma : \mathbb{R}^N \setminus \{\mathbf{0}\} \mapsto [1, \sqrt{N}]$ of a nonzero vector $\mathbf{x} \in \mathbb{R}^N$ is given by

$$\Gamma(\mathbf{x}) = \frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2}. \quad (8.8)$$

Definition 8.2 ([85]). For any $s \in \langle N \rangle$, we define the best s -sparse approximation error of $\mathbf{x} \in \mathbb{R}^N$ in the ℓ_2 -norm sense as

$$\mu_s(\mathbf{x}) \triangleq \inf \{ \|\mathbf{x}_s - \mathbf{x}\|_2 : \mathbf{x}_s \in \mathbb{R}^N \text{ is } s\text{-sparse} \}. \quad (8.9)$$

Similarly, we can define a normalized version of $\mu_s(\mathbf{x})$ as

$$\bar{\mu}_s(\mathbf{x}) \triangleq \frac{\mu_s(\mathbf{x})}{\|\mathbf{x}\|_2}. \quad (8.10)$$

As $\bar{\mu}_s$ is scale-invariant, it is easier to interpret and more useful for comparison purposes than μ_s . The utility of the above definitions is justified by the following result, which links the best s -sparse approximation error of a vector to its ℓ_1 norm.

Theorem 8.3 ([85, Theorem 2.5]). Every $\mathbf{x} \in \mathbb{R}^N$ satisfies $\mu_s(\mathbf{x}) \leq \frac{1}{2\sqrt{s}}\|\mathbf{x}\|_1$.

Specifically, it implies that vectors with small ℓ_1 norm can be well approximated by sparse representations. Using now the definition of Γ , we have the following corollary.

Corollary 8.4. Any $\mathbf{x} \in \mathbb{R}^N$ satisfies

$$\bar{\mu}_s(\mathbf{x}) \leq \frac{1}{2\sqrt{s}} \frac{\|\mathbf{x}\|_1}{\|\mathbf{x}\|_2} = \frac{1}{2\sqrt{s}} \Gamma(\mathbf{x}) \leq \frac{1}{2} \sqrt{\frac{N}{s}}. \quad (8.11)$$

As a consequence of this bound, Γ provides an index of compressibility for comparing different vectors of a same space.

8.2.1.2 Compressibility of the HOSVD core

In Chapter 7, we have adopted T2 tensors, whose modal unfoldings have rapidly decaying singular values, for modeling tensors found in real-world applications. In terms of the above introduced definitions, this amounts to saying that T2 tensors have compressible modal spectra. We now state new results connecting the compressibility of the HOSVD core of a tensor to the compressibility of its modal spectra.

Theorem 8.5. The core \mathcal{S} of the HOSVD of a P th-order tensor $\mathcal{X} \in \mathcal{T}$ satisfies

$$\max_{p \in \langle P \rangle} \left\| \boldsymbol{\sigma}^{(p)} \right\|_1 \leq \|\mathcal{S}\|_{H,1} \leq \min_{p \in \langle P \rangle} \sqrt{\bar{N}_p} \left\| \boldsymbol{\sigma}^{(p)} \right\|_1, \quad (8.12)$$

where $\boldsymbol{\sigma}^{(p)} \in \mathbb{R}^{\bar{N}_p}$ contains the singular values of $\mathbf{X}_{(p)}$.

Proof. Let the HOSVD of \mathcal{X} be given by $\mathcal{X} = \mathcal{S} \times_{p=1}^P \mathbf{U}^{(p)}$, with $\mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{R}^{N_p})$ and $\mathcal{S} \in \mathcal{T}$. From Proposition 2.52, we have that

$$\mathbf{S}_{(p)} = \boldsymbol{\Sigma}^{(p)} \mathbf{W}^{(p)T} \in \mathbb{R}^{N_p \times \bar{N}_p}, \quad (8.13)$$

where $\mathbf{W}^{(p)} \triangleq (\mathbf{U}^{(P)} \otimes \dots \otimes \mathbf{U}^{(p+1)} \otimes \mathbf{U}^{(p-1)} \otimes \dots \otimes \mathbf{U}^{(1)})^T \mathbf{V}^{(p)}$ and $\mathbf{X}_{(p)} = \mathbf{U}^{(p)} \boldsymbol{\Sigma}^{(p)} \mathbf{V}^{(p)T}$ is the SVD of the mode- p unfolding of \mathcal{X} . Now, noting that $\mathbf{W}^{(p)} \in \mathcal{V}_{\bar{N}_p}(\mathbb{R}^{\bar{N}_p})$, denoting the

n th row of $\mathbf{S}_{\langle p \rangle}$ by \mathbf{s}_n^T and the n th column of $\mathbf{W}^{(p)}$ by $\mathbf{w}_n^{(p)}$, we compute

$$\|\mathbf{S}\|_{H,1} = \sum_{n=1}^{N_p} \|\mathbf{s}_n^T\|_1 = \sum_{n=1}^{N_p} \left\| \sigma_n^{(p)} \mathbf{w}_n^{(p)} \right\|_1 = \sum_{n=1}^{N_p} \sigma_n^{(p)} \left\| \mathbf{w}_n^{(p)} \right\|_1 \quad (8.14)$$

$$\leq \sum_{n=1}^{N_p} \sigma_n^{(p)} \sqrt{\bar{N}_p} \left\| \mathbf{w}_n^{(p)} \right\|_2 = \sqrt{\bar{N}_p} \sum_{n=1}^{N_p} \sigma_n^{(p)} \quad (8.15)$$

$$= \sqrt{\bar{N}_p} \left\| \boldsymbol{\sigma}^{(p)} \right\|_1, \quad (8.16)$$

where the inequality follows from (8.7). Similarly,

$$\|\mathbf{S}\|_{H,1} = \sum_{n=1}^{N_p} \sigma_n^{(p)} \left\| \mathbf{w}_n^{(p)} \right\|_1 \geq \sum_{n=1}^{N_p} \sigma_n^{(p)} \left\| \mathbf{w}_n^{(p)} \right\|_2 = \left\| \boldsymbol{\sigma}^{(p)} \right\|_1. \quad (8.17)$$

Since these results are valid for all $p \in \langle P \rangle$, (8.12) follows. \square

Corollary 8.6. The core tensor \mathbf{S} of the HOSVD of a nonzero P th-order tensor $\mathbf{X} \in \mathcal{T}$ satisfies

$$\max_{p \in \langle P \rangle} \Gamma \left(\boldsymbol{\sigma}^{(p)} \right) \leq \Gamma(\text{vec}(\mathbf{S})) \leq \min_{p \in \langle P \rangle} \sqrt{\bar{N}_p} \Gamma \left(\boldsymbol{\sigma}^{(p)} \right). \quad (8.18)$$

Proof. Follows from $\|\text{vec}(\mathbf{S})\|_2 = \|\mathbf{S}\|_F = \|\mathbf{X}\|_F = \|\mathbf{X}_{\langle p \rangle}\|_F = \|\boldsymbol{\sigma}^{(p)}\|_2$ and Definition 8.1. \square

As the matrix factors in (8.6) are orthogonal, $\|\mathbf{X}^*\|_F = \|\mathbf{G}\|_F$. Therefore, approximating \mathbf{G} by a sparse core (via simple truncation of the sum in (8.6)) yields quadratic error (with respect to \mathbf{X}^*) which equals the sum of the squares of the neglected core elements. If the core is compressible, then a significant number of terms can be neglected while still achieving a reasonable approximation. Now, the above results show that the HOSVD cores of many parsimonious tensors found in applications are compressible as a consequence of their modal spectral behavior. Of course, the HOSVD is just one of the infinitely many possible OTDs of a tensor; conceivably, it could admit OTD models having even more compressible cores.

8.2.1.3 Some empirical evidence and discussion

We now illustrate the above discussion by means of some numerical examples. To this end, we generate 500 realizations of T2 tensors from $\mathbb{R}^{N \times N \times N}$, with $N = 20$, using exactly the same procedure as that of Section 7.4.1.1. The spectral decay rate φ assumes values from $\{1.0, 1.2, \dots, 2.4\}$. For every generated tensor, the core of its HOSVD, \mathbf{S} , is computed. In Fig. 8.1(a), we show the ensemble average of the elements of $\bar{\mathbf{s}} = \text{sortabs}(\text{vec}(\mathbf{S})) \in \mathbb{R}^{\bar{N}}$, where the operator `sortabs` arranges the absolute values of the elements of its argument in non-increasing order. Observe that the magnitude of the elements of the core decay more quickly for larger values of φ , in consonance with our results. Fig. 8.1(b) shows the cumulative distribution function (c.d.f.) of $\Gamma(\text{vec}(\mathbf{S}))$. As we can see, increasing φ induces a

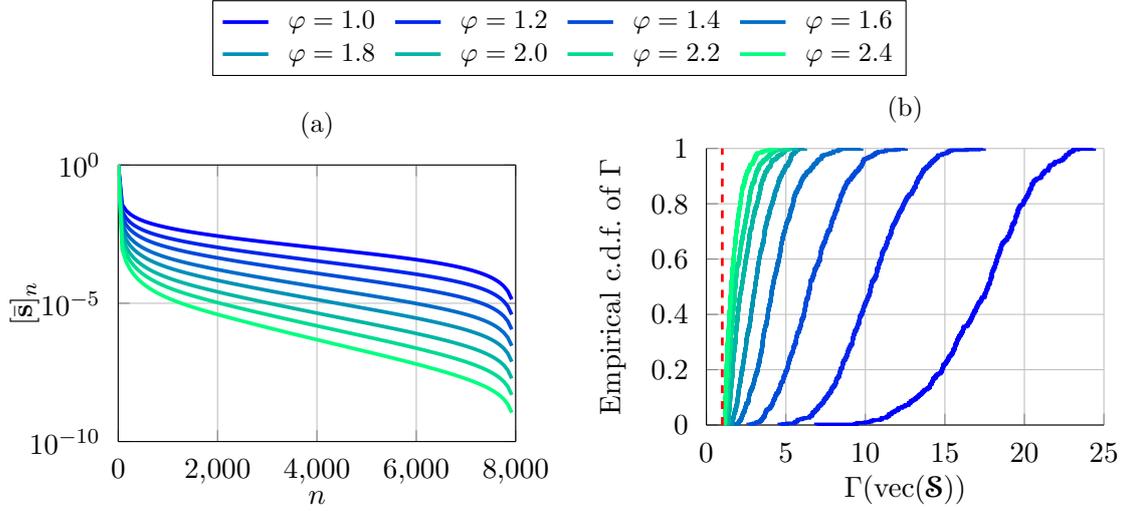


Figure 8.1: Compressibility of the HOSVD core of T2 tensors for different spectral decay rates φ : (a) average sorted absolute values of its components; (b) empirical c.d.f. of $\Gamma(\text{vec}(\mathcal{S}))$. The minimum value of 1 is indicated by a red dashed line in (b).

sharper concentration of the distribution around its mean, which tends to decrease towards the minimum value of 1.

In a second experiment, we generate 500 realizations of tensors \mathcal{X}_p , $p \in \{1, 2\}$, having fast decaying singular values only with respect to the first p modes. The decay rate employed for \mathcal{X}_1 is $\varphi = 2.5$, while it is set as $\varphi = 2.2$ for \mathcal{X}_2 . On Fig. 8.2(a)–(b) we plot the same quantities as in Fig. 8.1, while on Fig. 8.2(c)–(e) we plot the average modal spectra of the generated tensors. Fig. 8.2(a)–(b) indicate that the core of \mathcal{X}_2 is in average significantly more compressible than that of \mathcal{X}_1 . However, as Fig. 8.2(c)–(e) shows, the mode-1 spectrum $\sigma^{(1)}$ of \mathcal{X}_1 is more compressible than all modal spectra of \mathcal{X}_2 : indeed, on average we have $\Gamma(\sigma^{(1)}(\mathcal{X}_1)) = 0.0903$, $\Gamma(\sigma^{(1)}(\mathcal{X}_2)) = 0.1226$ and $\Gamma(\sigma^{(2)}(\mathcal{X}_2)) = 0.1248$. This suggests that the bound $\Gamma(\text{vec}(\mathcal{S})) \leq \min_{p \in \langle P \rangle} \sqrt{N_p} \Gamma(\sigma^{(p)})$ is pessimistic, as in practice the most compressible modal spectrum it is not the only one that contributes to the compressibility of the HOSVD core. The latter seems instead to be affected by the behavior of all modal spectra, thus jointly capturing the spectral behavior of multiple modes.

8.2.2 Imputation scheme with ℓ_1 -regularized least squares

We now formulate our approach on the basis of the above discussion. Similarly to the algorithms discussed in Section 8.1, it proceeds iteratively as follows:

- 1) First, \mathcal{Z}_k is constructed exactly as in (8.1), with \mathcal{X}_k denoting the tensor reconstructed from the most recently estimated model.

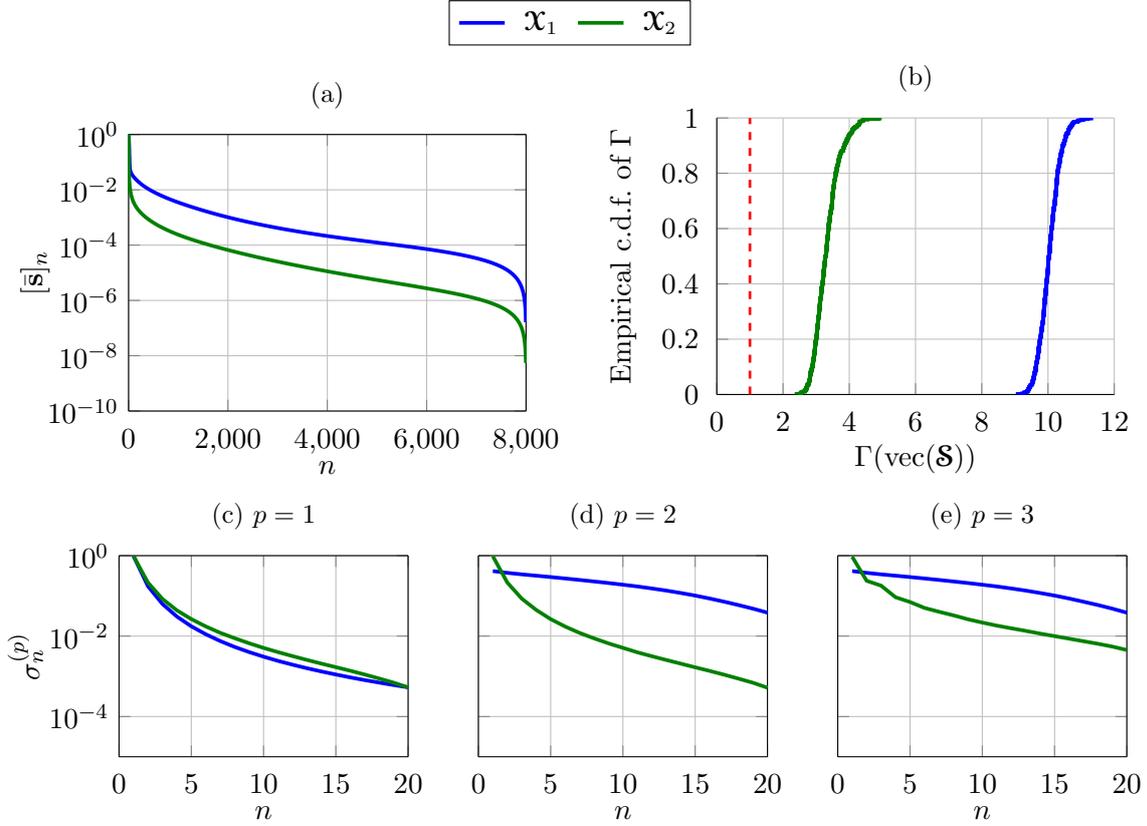


Figure 8.2: Compressibility of the HOSVD core of tensors with varying number of modes having fast decaying singular spectra: (a) average sorted absolute values of its components; (b) empirical c.d.f. of $\Gamma(\text{vec}(\mathbf{S}))$; (c)–(e): average modal spectra of generated tensors. The minimum value of 1 is indicated by a red dashed line in (b).

2) In order to estimate an OTD model having a compressible core, we consider the ℓ_1 -regularized least-squares formulation

$$\min_{\substack{\mathcal{G} \in \mathcal{T} \\ \mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{R}^{N_p})}} \frac{1}{2} \left\| \mathbf{z}_k - \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)} \right\|_F^2 + \tau \|\mathcal{G}\|_{H,1}, \quad (8.19)$$

which, due to the orthogonality of the factors $\mathbf{U}^{(p)}$, is equivalent to

$$\min_{\substack{\mathcal{G} \in \mathcal{T} \\ \mathbf{U}^{(p)} \in \mathcal{V}_{N_p}(\mathbb{R}^{N_p})}} \frac{1}{2} \left\| \mathbf{z}_k \times_{p=1}^P \left(\mathbf{U}^{(p)} \right)^T - \mathcal{G} \right\|_F^2 + \tau \|\mathcal{G}\|_{H,1}. \quad (8.20)$$

Observe that parsimony is imposed by regularization of the OTD core. But, just as in the schemes based on low-rank CPD and low-mrank Tucker models, an exact solution of (8.20) is hard to obtain. One can nonetheless obtain an approximate one by relying on the following reasoning.

First, observe that for $\tau \rightarrow 0$, any OTD of \mathbf{Z}_k serves as a solution of (8.19)–(8.20). Now, as τ is increased, we expect the global minimizers to vary continuously, due to continuity of the norms and of the OTD model with respect to its components. We thus propose to take the factors of an OTD of \mathbf{Z}_k as an approximation of the optimal factors $\mathbf{U}^{(p)}$, and then find the corresponding optimal \mathcal{G} . In order to attain a sufficiently low value of the objective function, this OTD should have a core as compressible as possible. Due to Corollary 8.6, we choose to employ the HOSVD. Letting $\mathbf{Z}_k = \mathbf{S}_k \times_{p=1}^P \mathbf{U}_k^{(p)}$ denote the¹ HOSVD of \mathbf{Z}_k , we have $\mathbf{Z}_k \times_{p=1}^P \left(\mathbf{U}_k^{(p)}\right)^T = \mathbf{S}_k$. Therefore, from (8.20) we derive the approximate solution

$$\arg \min_{\mathcal{G} \in \mathcal{T}} \frac{1}{2} \|\mathbf{S}_k - \mathcal{G}\|_F^2 + \tau \|\text{vec}(\mathcal{G})\|_{H,1} = \text{prox}_\tau^{\|\cdot\|_{H,1}}(\mathbf{S}_k), \quad (8.21)$$

where $\text{prox}_\tau^{\|\cdot\|_{H,1}}$ is defined analogously to the ℓ_1 proximity operator, *i.e.*,

$$\left[\text{prox}_\tau^{\|\cdot\|_{H,1}}(\mathbf{S}_k) \right]_{n_1, \dots, n_P} = \begin{cases} [\mathbf{S}_k]_{n_1, \dots, n_P} - \text{sign}([\mathbf{S}_k]_{n_1, \dots, n_P})\tau, & \text{if } |[\mathbf{S}_k]_{n_1, \dots, n_P}| > \tau, \\ 0, & \text{otherwise.} \end{cases}$$

3) In the third step, we compute the new reconstructed estimate

$$\mathbf{X}_{k+1} = \left(\text{prox}_\tau^{\|\cdot\|_{H,1}}(\mathbf{S}_k) \right) \times_{p=1}^P \mathbf{U}_k^{(p)} \quad (8.22)$$

and then return to step 1 until some stopping criterion is satisfied. A simple stopping criterion consists in bounding the relative error between two consecutive estimates by some specified tolerance level $\epsilon > 0$, as in

$$\frac{\|\mathbf{X}_k - \mathbf{X}_{k+1}\|_F}{\|\mathbf{X}_k\|_F} < \epsilon. \quad (8.23)$$

We finish this section by noting that [210] has relied on similar ideas for computing an approximation of a tensor having low (tensor) rank: a CPD model of the form $[[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \dots, \mathbf{A}^{(P)}]]$ is estimated by minimizing a least-squares criterion regularized by a term of the form $\|\boldsymbol{\lambda}\|_1$. The motivation is promoting a sparse (or compressible) vector of weights $\boldsymbol{\lambda}$, thereby reducing the complexity of the model. Though not discussed in [210], it seems that an extension of their method to the TC problem is possible.

8.3 Imputation schemes with feedback

8.3.1 Nuclear norm minimization of a single unfolding

We now go back to the ADMM scheme for LRTR described in Algorithm 6.1, and consider the case where $Q = 1$, with $\lambda \rightarrow 0$. In other words, the measurements are taken as exact

¹Since the SVD is not unique if it has repeated singular values, the HOSVD of a tensor is also possibly not unique. At any rate, the result in Corollary 8.6 applies to *all* possible HOSVDs.

constraints, and only one modal unfolding is supposed to be of low rank. Note that in this case, the superscripts (q) are unnecessary, as well as the permutation matrices $\mathbf{\Pi}^{(q)}$.

Under these assumptions, and considering the TC scenario, (6.28) and (6.29) yield

$$[\mathbf{x}_{k+1}]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ [\mathbf{z}_k - \frac{1}{\eta} \mathbf{a}_k]_n, & n \notin \Omega. \end{cases} \quad (8.24)$$

The update equation of \mathbf{z}_{k+1} is

$$\mathbf{z}_{k+1} = \text{vec} \left(\text{prox}_{1/\eta}^{\|\cdot\|_*} \left(\text{unvec}_{N_p \times \bar{N}_p} \left(\mathbf{x}_{k+1} + \frac{1}{\eta} \mathbf{a}_k \right) \right) \right), \quad (8.25)$$

while that of \mathbf{a}_{k+1} can be written as

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta(\mathbf{x}_{k+1} - \mathbf{z}_{k+1}). \quad (8.26)$$

Though the ADMM algorithm is in principle quite robust with regard to the choice of the penalty parameter η , being guaranteed to converge (for convex problems) under mild conditions for any $\eta > 0$, in practice this parameter has a strong impact on convergence speed [77]. For this reason, a standard heuristic consists in varying η along the iterations for accelerating convergence [22]. A straightforward possibility is to start off with some sufficiently small value η_0 and then monotonically increase it, as in the simple rule [134]

$$\eta_k = \frac{1}{\psi} \eta_{k-1}, \quad (8.27)$$

where $0 < \psi < 1$.

Let us thus insert this adaptive η rule into the above equations. With the definition

$$\bar{\mathbf{z}}_{k+1} \triangleq \mathbf{x}_{k+1} + \frac{1}{\eta_k} \mathbf{a}_k, \quad (8.28)$$

equations (8.24)–(8.26) become

$$[\mathbf{x}_{k+1}]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ [\mathbf{z}_k - \frac{1}{\eta_k} \mathbf{a}_k]_n, & n \notin \Omega, \end{cases} \quad (8.29)$$

$$\mathbf{z}_{k+1} = \text{prox}_{1/\eta_k}^{\|\cdot\|_*} \left(\text{unvec}_{N_p \times \bar{N}_p} (\bar{\mathbf{z}}_{k+1}) \right), \quad (8.30)$$

$$\mathbf{a}_{k+1} = \mathbf{a}_k + \eta_k(\mathbf{x}_{k+1} - \mathbf{z}_{k+1}). \quad (8.31)$$

In particular, the third one implies

$$\frac{1}{\eta_k} \mathbf{a}_k = \frac{1}{\eta_k} \mathbf{a}_{k-1} + \frac{\eta_{k-1}}{\eta_k} (\mathbf{x}_k - \mathbf{z}_k) = \frac{1}{\eta_k} \mathbf{a}_{k-1} + \psi (\mathbf{x}_k - \mathbf{z}_k). \quad (8.32)$$

Together with (8.28), the above identity yields

$$\frac{1}{\eta_k} \mathbf{a}_k = \frac{1}{\eta_k} \mathbf{a}_{k-1} + \psi \left(\bar{\mathbf{z}}_k - \frac{1}{\eta_{k-1}} \mathbf{a}_{k-1} - \mathbf{z}_k \right) = \psi (\bar{\mathbf{z}}_k - \mathbf{z}_k). \quad (8.33)$$

By using (8.28) and (8.29), we thus derive

$$\bar{\mathbf{z}}_{k+1} = \begin{cases} [\mathbf{x}^* + \frac{1}{\eta_k} \mathbf{a}_k]_n = [\mathbf{x}^* + \psi(\bar{\mathbf{z}}_k - \mathbf{z}_k)], & n \in \Omega \\ [\mathbf{z}_k - \frac{1}{\eta_k} \mathbf{a}_k + \frac{1}{\eta_k} \mathbf{a}_k]_n = [\mathbf{z}_k]_n, & n \notin \Omega. \end{cases} \quad (8.34)$$

Now, to rewrite the expression given in (8.29) for $n \notin \Omega$, we deduce from (8.33)–(8.34)

$$\left[\frac{1}{\eta_k} \mathbf{a}_k \right]_n = [\psi(\bar{\mathbf{z}}_k - \mathbf{z}_k)]_n = [\psi(\mathbf{z}_{k-1} - \mathbf{z}_k)]_n, \quad (8.35)$$

from which it follows that

$$[\mathbf{x}_{k+1}]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ [\mathbf{z}_k + \psi(\mathbf{z}_k - \mathbf{z}_{k-1})]_n, & n \notin \Omega. \end{cases} \quad (8.36)$$

The resulting scheme is summarized in [Algorithm 8.1](#). Two interesting observations can be drawn from its final form:

- 1) The algorithm is simplified, because the iterates no longer depend on \mathbf{x}_k , and hence we update only two variables per iteration, $\bar{\mathbf{z}}_{k+1}$ and \mathbf{z}_{k+1} , rather than three. It can also be noted that, since \mathbf{z}_k and \mathbf{z}_{k-1} should be close near convergence, the reconstructed entries of the estimated tensor are approximately given by $[\mathbf{z}_K]_n$, for $n \notin \Omega$.
- 2) There exists a strong similarity between [Algorithm 8.1](#) and the single imputation schemes discussed in [Section 8.1](#), in the sense that they all consist in iteratively (i) taking the most recent estimate, (ii) replacing the known elements by their observed values, with possibly an added correction term, and then (iii) mapping the result into a new estimate by means of some parsimony-inducing operator. In particular, the operator of [Algorithm 8.1](#) performs a soft thresholding of the singular values of the mode- p unfolding of its argument.

8.3.2 Interpretation as feedback-controlled system

In the following, we continue the analysis of [Algorithm 8.1](#). By repeatedly plugging (8.31) into (8.32), we can derive

$$\frac{1}{\eta_k} \mathbf{a}_k = \frac{1}{\eta_k} \mathbf{a}_{k-2} + \psi^2(\mathbf{x}_{k-1} - \mathbf{z}_{k-1}) + \psi(\mathbf{x}_k - \mathbf{z}_k) \quad (8.37)$$

$$= \frac{1}{\eta_k} \mathbf{a}_{k-3} + \psi^3(\mathbf{x}_{k-2} - \mathbf{z}_{k-2}) + \psi^2(\mathbf{x}_{k-1} - \mathbf{z}_{k-1}) + \psi(\mathbf{x}_k - \mathbf{z}_k) \quad (8.38)$$

and so on, ultimately obtaining

$$\frac{1}{\eta_k} \mathbf{a}_k = \frac{1}{\eta_k} \mathbf{a}_0 + \sum_{j=1}^k \psi^{k+1-j}(\mathbf{x}_j - \mathbf{z}_j). \quad (8.39)$$

Algorithm 8.1 ADMM algorithm with varying penalty parameter for TC based on NNM of a single unfolding.

Inputs: Initial estimate $\mathbf{X}_0 \in \mathcal{T}$, vector of measurements \mathbf{y} and parameters η_0 and ψ

Outputs: Estimate $\hat{\mathbf{X}}^*$ of \mathbf{X}^*

```

1:  $k \leftarrow 0$ 
2: repeat
3:    $[\bar{\mathbf{z}}_{k+1}]_n = \begin{cases} [\mathbf{y}]_n + \psi([\bar{\mathbf{z}}_k]_n - [\mathbf{z}_k]_n), & n \in \Omega \\ [\mathbf{z}_k]_n, & n \notin \Omega. \end{cases}$ 
4:    $\mathbf{z}_{k+1} = \text{vec} \left( \text{prox}_{1/\eta_k}^{\|\cdot\|_*} (\text{unvec}_{N_p \times N_p}(\bar{\mathbf{z}}_{k+1})) \right)$ 
5:    $\eta_{k+1} = \psi^{-1} \eta_k$ 
6:    $k \leftarrow k + 1$ 
7: until the stopping criterion is fulfilled
8: return  $[\hat{\mathbf{x}}^*]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ [\mathbf{z}_k]_n + \psi([\mathbf{z}_k]_n - [\mathbf{z}_{k-1}]_n), & n \notin \Omega. \end{cases}$ 

```

Hence, for $n \in \Omega$, inserting (8.39) into (8.34) and using $[\mathbf{x}_k]_n = [\mathbf{x}^*]_n$, we compute

$$[\bar{\mathbf{z}}_{k+1}]_n = \left[\mathbf{x}^* + \frac{1}{\eta_k} \mathbf{a}_0 + \sum_{j=1}^k \psi^{k+1-j} (\mathbf{x}^* - \mathbf{z}_j) \right]_n. \quad (8.40)$$

This shows that the correction term in the update of $[\bar{\mathbf{z}}_{k+1}]_n$ for $n \in \Omega$ is nothing but an exponentially weighted sum of all previous errors with respect to the corresponding (observed) entries of \mathbf{x}^* , plus a term which satisfies $\frac{1}{\eta_k} \mathbf{a}_0 \rightarrow \mathbf{0}$ for $k \rightarrow \infty$ when $\psi < 1$ (and which disappears if $\mathbf{a}_0 = \mathbf{0}$). We thus can see the operation of this algorithm as that of a feedback control mechanism which drives the outputs of the proximity operator (the controlled system) corresponding to the known elements (*i.e.*, having indices $n \in \Omega$) towards their observed values. This can be easily seen by rewriting (8.34) with the use of a unit delay operator q^{-1} , which yields

$$(1 - \psi q^{-1}) [\bar{\mathbf{z}}_{k+1}]_n = q^{-1} ([\mathbf{x}^*]_n - \psi [\mathbf{z}_{k+1}]_n), \quad (8.41)$$

implying

$$\forall n \in \Omega, \quad [\bar{\mathbf{z}}_{k+1}]_n = \frac{q^{-1}}{1 - \psi q^{-1}} ([\mathbf{x}^*]_n - \psi [\mathbf{z}_{k+1}]_n). \quad (8.42)$$

Hence, the \bar{N} inputs of the controlled system (*i.e.*, the components of $\bar{\mathbf{z}}_{k+1}$) can be clearly partitioned into two sets, as depicted in Fig. 8.3: (i) components having indices in Ω (denoted by $[\bar{\mathbf{z}}_{k+1}]_\Omega$) are manipulated by a first-order feedback control mechanism with time constant ψ and setpoint given by the observed values of \mathbf{x}^* ; (ii) the other ones (denoted by $[\bar{\mathbf{z}}_{k+1}]_{\bar{\Omega}}$, where $\bar{\Omega}$ stands for the complement set with respect to Ω) are *not* manipulated, being given simply by the corresponding outputs of the controlled system.

Interestingly, ψ can also be seen as a *forgetting factor*, because the smaller it is, the faster the factors ψ^{k+1-j} in (8.40) decay, so that deviations from \mathbf{x}^* “in a far past” contribute less

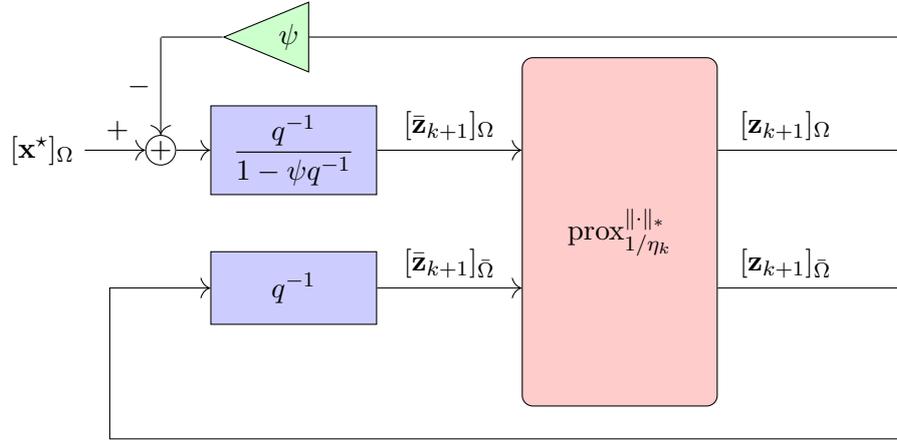


Figure 8.3: Interpretation of [Algorithm 8.1](#) (with null initial conditions) as a feedback control system.

and less to the correction produced by feedback control mechanism. In the limit case $\psi \rightarrow 0$, these components of $\bar{\mathbf{z}}_k$ approach the corresponding observed values of \mathbf{x}^* . At the other extreme, $\psi = 1$ yields a scheme in which all previous deviations of with respect to \mathbf{x}^* have equal weight. In the latter case, the first-order control system of (8.42) becomes a pure discrete-time integrator, and hence unstable behavior may be observed.

A control system interpretation of [ADMM](#) has already been (briefly) discussed in [157]. However, it differs from ours with respect to the involved variables and to the goal of the control strategy, because of the special case we consider here and of our rewriting of the update equations. Specifically, in their case three variables are iteratively updated and the feedback control mechanism is meant to gradually suppress the error between two variables in order to satisfy an equality constraint. Also, our interpretation is further developed and easier to grasp, thanks to the simplified update equations of [Algorithm 8.1](#).

8.3.3 Generalized imputation scheme for tensor completion

Having elaborated on the connection between the [ADMM](#) scheme with varying penalty parameter of [Algorithm 8.1](#) and the imputation schemes studied in [Section 8.1](#), we now propose a general imputation scheme for [TC](#) having a feedback first-order control mechanism. A concrete description of it is given in [Algorithm 8.2](#), where $\mathcal{P} : \mathbb{R}^{\bar{N}} \times \mathbb{R} \mapsto \mathbb{R}^{\bar{N}}$ represents a parsimony-inducing operator chosen according to the adopted underlying model.

In particular, \mathcal{P} can perform the operation described by (8.22). We refer to this possibility as imputation scheme with feedback and HOSVD soft thresholding ([IFHST](#)). Another alternative, which we call [IFHHT](#), is to assimilate \mathcal{P} with the hard thresholding operator $\mathcal{H}_{\mathbf{r}}$, the target mrank \mathbf{r} being chosen *a priori*. The incorporation of these approaches into the

framework of [Algorithm 8.2](#) allows profiting from the first-order feedback control mechanism, which accelerates convergence speed, as empirically illustrated in the next section.

Algorithm 8.2 Imputation scheme with feedback and forgetting factor for TC.

Inputs: Initial estimate $\mathbf{X}_0 \in \mathcal{T}$, vector of measurements \mathbf{y} and parameters η_0 and ψ

Outputs: Estimate $\hat{\mathbf{X}}^*$ of \mathbf{X}^*

```

1:  $k \leftarrow 0$ 
2: repeat
3:    $[\bar{\mathbf{z}}_{k+1}]_n = \begin{cases} [\mathbf{y}]_n + \psi([\bar{\mathbf{z}}_k]_n - [\mathbf{z}_k]_n), & n \in \Omega \\ [\mathbf{z}_k]_n, & n \notin \Omega. \end{cases}$ 
4:    $\mathbf{z}_{k+1} = \mathcal{P}(\bar{\mathbf{z}}_{k+1}, 1/\eta_k)$ 
5:    $\eta_{k+1} = \psi^{-1}\eta_k$ 
6:    $k \leftarrow k + 1$ 
7: until the stopping criterion is fulfilled
8: return  $[\hat{\mathbf{x}}^*]_n = \begin{cases} [\mathbf{x}^*]_n, & n \in \Omega \\ [\mathbf{z}_k]_n + \psi([\mathbf{z}_k]_n - [\mathbf{z}_{k-1}]_n), & n \notin \Omega. \end{cases}$ 
    
```

In the specific case of [IFHST](#), initializing the algorithm with a small enough value of η_0 and choosing an appropriate forgetting factor $\psi < 1$ leads to an interesting behavior, namely:

- the first iterates have very sparse cores, because $1/\eta_k$ is sufficiently large so that nearly all core components are zeroed by the soft thresholding operator for small k ;
- since $1/\eta_k$ decays exponentially as the algorithm progresses, the sparsity of the core is gradually reduced and thus more and more rank-one terms associated with nonzero core elements are added to the model.

In some sense, this leads to a sequential estimation of increasingly complex models, similarly to the effect produced by the use of the [GRI](#) technique in [SeMPIHT](#).

8.4 Experimental validation

8.4.1 Improvement of imputation scheme due to first-order feedback control mechanism

We start by highlighting the improvement brought by the feedback control mechanism. Initially, we run a Monte Carlo simulation where several realizations of a T1 $20 \times 20 \times 20$ tensor having $\text{mrank}(4, 4, 4)$ are generated as $\mathbf{X}_0 = \mathcal{G} \times_{p=1}^P \mathbf{U}^{(p)}$, where $\mathcal{G} \in \mathbb{R}^4 \otimes \mathbb{R}^4 \otimes \mathbb{R}^4$ has standard Gaussian entries and unit Frobenius norm and $\mathbf{U}^{(p)} \in \mathcal{V}_4(\mathbb{R}^{20})$. Each one of these tensors is reconstructed from 15% randomly sampled elements. Before sampling, we add a rescaled standard Gaussian term \mathcal{N} (also of unit Frobenius norm), yielding

$$\mathbf{X}^* = \mathbf{X}_0 + 10^{-5} \mathcal{N}. \quad (8.43)$$

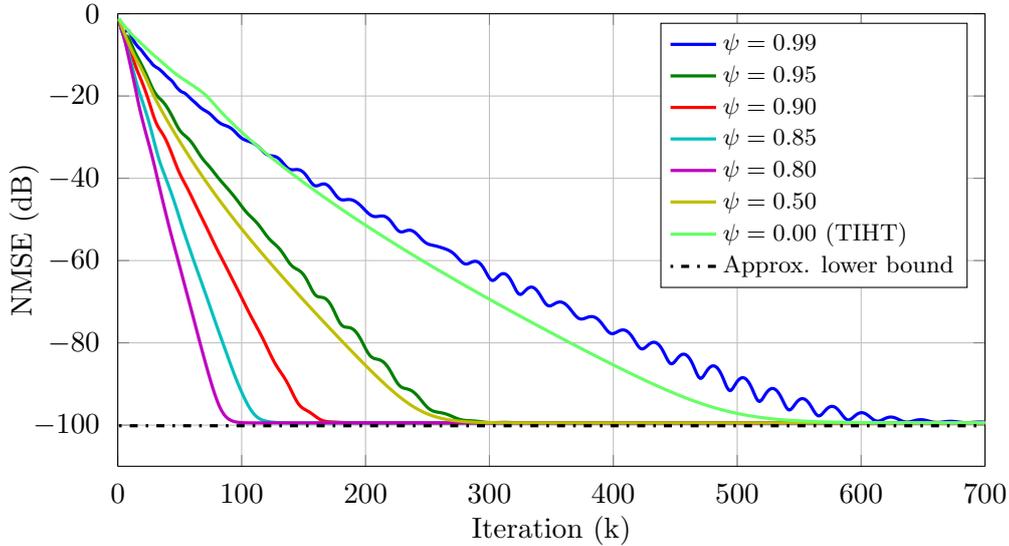


Figure 8.4: Acceleration of convergence of IFHHT due to first-order feedback control mechanism with time constant ψ .

The IFHHT algorithm is then applied for several values of ψ in the interval $(0, 0.99)$. Note that, when $\psi = 0$ (*i.e.*, without feedback control), IFHHT is essentially equivalent to TIHT.

Fig. 8.4 shows the average NSE (see (3.19)) achieved after k iterations, for $k \in \langle 700 \rangle$ and 20 realizations. Observe that for $\psi = 0.99$, there is no gain in comparison with TIHT, as the convergence is even slower. However, as we reduce ψ , convergence is remarkably accelerated, up to a certain value beyond which it starts to slow down again. In particular, with $\psi = 0.8$ IFHHT attains convergence at least five times faster than TIHT.

Pronounced oscillations are seen in some curves, especially that associated with $\psi = 0.99$. This phenomenon is related to our control system interpretation: $\psi = 0.99$ corresponds to an “aggressive” tuning of the first-order control mechanism where a pole is introduced near the unit circle. In fact, when we used $\psi = 1$ in our experiments, the algorithm often diverged. As an illustration, we show in Fig. 8.5 the value of $[\mathbf{z}]_n$ attained along the first 160 iterations of one realization for a given $n \in \Omega$. The corresponding observed value (*i.e.*, $[\mathbf{x}^*]_n$) is also shown for reference. Clearly, the contribution of past errors in (8.40) induces large oscillations for $\psi = 0.99$. When $\psi = 0$, no past errors are taken into account, and thus a smooth curve is observed, but it takes long to converge. Visibly, the choice $\psi = 0.8$ attains a reasonable compromise.

Our second experiment focus on the recovery of T2 tensors generated as in Section 7.4.1.1, with a spectral decaying parameter of $\varphi = 3$. They are completed from 40% randomly sampled elements by the IFHST algorithm. The average results of 20 realizations is shown in Fig. 8.6 for various values of ψ and of η_0 . When $\psi = 0$, line 5 of Algorithm 8.2 is not performed; instead, we set $\eta_k = \eta_0$ for all k . Visibly, the choice of η_0 then controls the

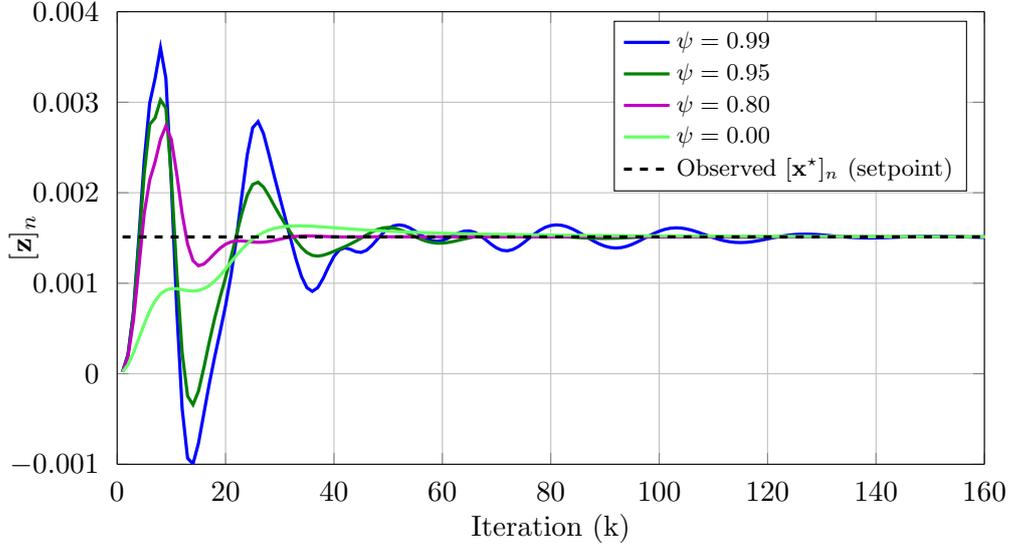


Figure 8.5: Evolution of a reconstructed component of the target tensor during a run of IFHHT, for several values of ψ .

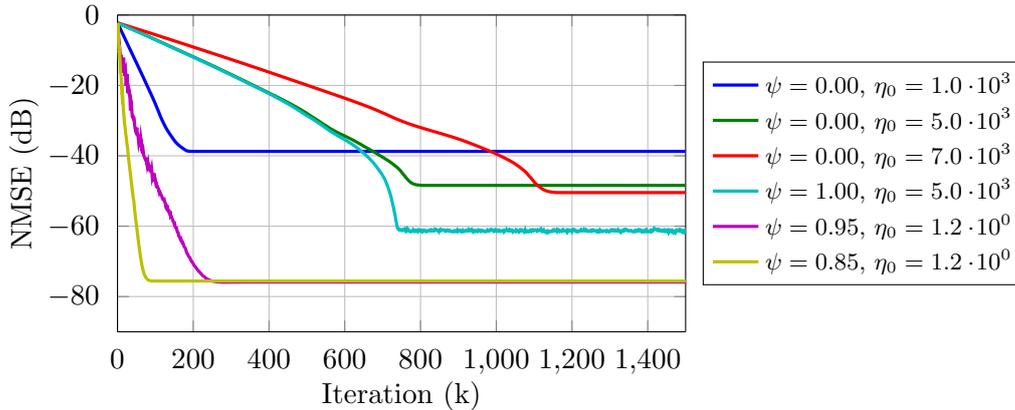


Figure 8.6: Acceleration of convergence of IFHST due to feedback mechanism with forgetting factor ψ .

compromise between convergence speed and final error: $\eta_0 = 10^3$ (dark blue curve) leads to quick convergence but a somewhat high NSE, while augmenting η_0 (dark green and red curves) allows decreasing the final error, but slows down convergence significantly. The curve for $\psi = 1$ was generated with the same η_0 as the second one, which causes them to be superposed until around $k = 600$. From that point on, the integral control mechanism yields a marked improvement of the final NSE (*i.e.*, it reduces steady-state error), but convergence is still slow. Now, for $\psi = 0.95$ and $\psi = 0.85$, there is a remarkable gain with respect to both convergence speed and final NSE.

When we take a look again at the evolution of $[z]_n$ for some $n \in \Omega$ along the iterations

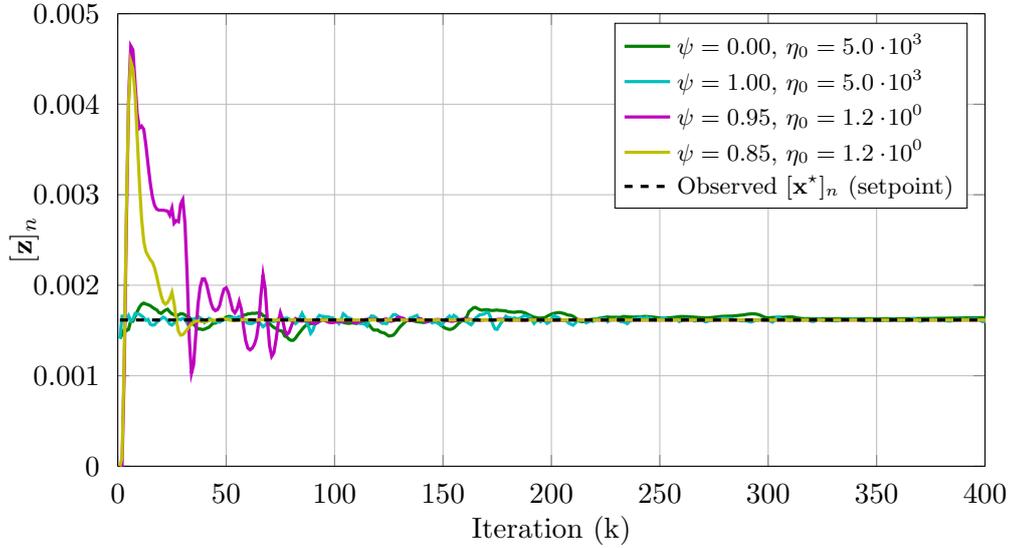


Figure 8.7: Evolution of a reconstructed component of the target tensor during a run of IFHST, for several values of ψ .

of a single run, the benefit of employing the feedback control mechanism with a well-tuned forgetting factor is evident, as seen in Fig. 8.7. Note that, when $\psi = 0.85$ and $\psi = 0.95$, the oscillations die down much more quickly. The large oscillations in the beginning happen because $[\mathbf{z}]_n = 0$ for small k due to the small value of η_k , which implies a large threshold.

8.4.2 Evolution of Tucker core sparsity along iterations

As we mentioned in Section 8.3.3, in IFHST the sparsity of the thresholded HOSVD core of the iterates typically decays as k grows. This is illustrated by Fig. 8.8, which was generated with the same experimental procedure used for Fig. 8.6, but now only with $\psi = 0.85$, $\eta_0 = 1.2$ and for a single realization. Incidentally, it shows that a highly sparse core can accurately model \mathcal{X}^* : for $k = 60$, the sparsity level is 98.7%, while the NSE is around -73.5 dB.

8.4.3 Empirical sampling bounds

Proceeding exactly as in Section 7.5.1, we estimate the recovery regime of IFHST for $\rho \in \{0.05, 0.10, \dots, 1\}$. This allows us to compare it with several other algorithms, as done in Fig. 8.9. It can be seen that, unlike the solution based on minimizing the SNN of matrix unfoldings, IFHST (which was run with $\psi = 0.95$ and $\eta_0 = 0.02$) produces reasonable results, behaving similarly to the other algorithms, despite its narrower recovery regime. Also, we should note that IFHST is not fed with *a priori* knowledge on the mrank of the solution, contrarily to all other algorithms except SNN.

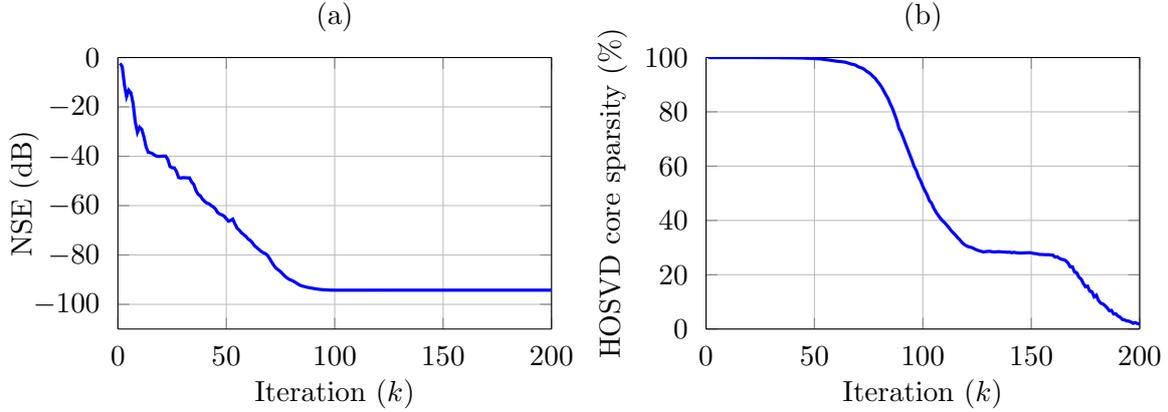


Figure 8.8: Evolution of (a) NSE and (b) sparsity of the thresholded HOSVD core along a run of the IFHST algorithm with $\psi = 0.85$ and $\eta_0 = 1.2$.

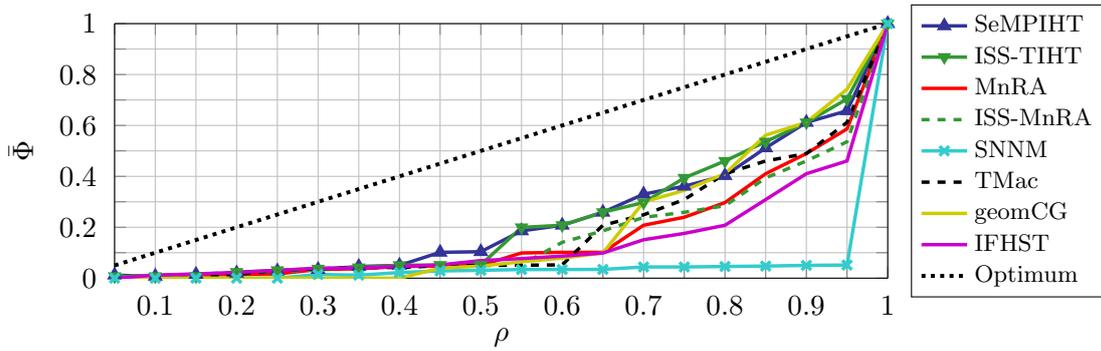


Figure 8.9: Estimated (normalized) number of DOF which can be recovered by several algorithms for each level of ρ , using sampling MOs. The measured third-order tensors have dimensions $N_p = 20$ for all p . Recovery was successful in all 15 realizations for values of $\bar{\Phi}$ below or over the curve.

8.4.4 Convergence and computing cost

To conclude our experimental validation, we now compare IFHST and IFHHT with other algorithms in terms of convergence speed and computing cost, as done in Section 7.5.2. In the first scenario, we recover approximately $(10, 10, 10)$ -mrnk $100 \times 100 \times 100$ tensors with an undersampling rate of $\rho = 0.2$. Some modeling error is introduced by generating these tensors exactly as described by (8.43). For a fair comparison between IFHST and SNN, the latter was also run with an exponentially varying penalty parameter updated as in line 5 of Algorithm 8.2. Also, TMac employs its adaptive weight scheme, as described in [213].

The algorithmic parameters are set as follows: $\eta = 1.2$ and $\psi = 0.7$ for IFHST; $\psi = 0.5$ for IFHHT; $\eta = 5$ and $\psi = 0.85$ for SNN; $\alpha = 0.5$, $\beta = 0.7$, $\kappa = 1.2$ and $L = 3$ (ISS settings) for SeMPIHT, TIHT and MnRA; $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$ (initial weights) for TMac.

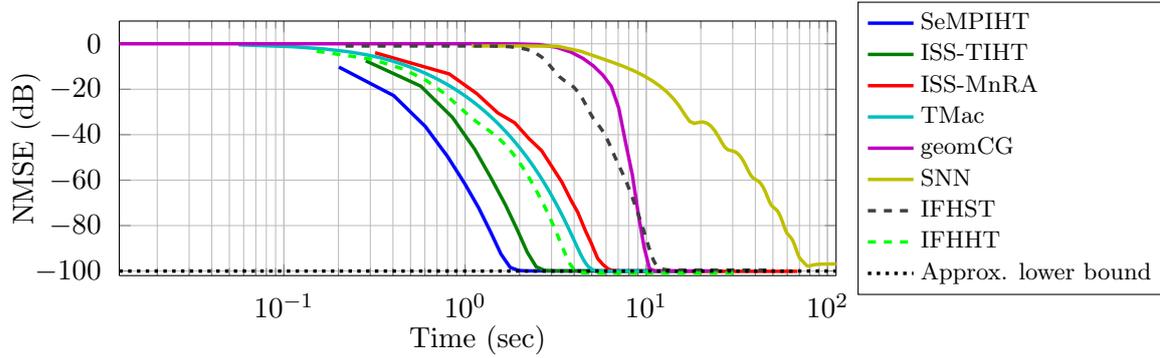


Figure 8.10: Evolution of **NMSE** as a function of time for several **TC** algorithms when applied to recover approximately $(10, 10, 10)$ -mrank $100 \times 100 \times 100$ tensors, with $\rho = 0.2$.

In Fig. 8.10, we display the evolution of the **NMSE** of 60 realizations. The approximate lower bound is estimated by applying **SeMP** to each generated tensor. It can be seen that all algorithms which directly impose a low-mrank constraint do a better job than **IFHST**, as they converge more quickly to the lower bound (with only a slight advantage in the case of **geomCG**). Among them, **IFHHT** is outperformed only by **ISS-TIHT** and **SeMPIHT**. In comparison with **SNN**, **IFHST** converges much faster.

In the second scenario, **T2** tensors with a decay parameter of $\varphi = 3/2$ are randomly generated and reconstructed by **TC** algorithms after being sampled with $\rho = 0.15$. Now, only **SeMPIHT** is kept as the representative of **IHT** algorithms, because it is the only one incorporating a **GRI** scheme, which is necessary for satisfactory performance in this case. For the same reason, imputation scheme with feedback and **HOSVD** hard thresholding (**IFHHT**) is not included. Both **TMac** and **geomCG** are run with their rank-increasing strategies starting from $\mathbf{r}_0 = (1, 1, 1)$. The target mrank of **SeMPIHT**, **TMac** and **geomCG** is $\mathbf{r} = (40, 40, 40)$, and unit increments are applied to each mrank component in all three algorithms. The tolerance which controls mrank increase is set as 0.01 for both **TMac** and **geomCG**. In **SeMPIHT**, mrank components are incremented every two iterations. **IFHST** is run with $\psi = 0.85$ and $\eta_0 = 12$.

As we can see from the curves in Fig. 8.11, the chosen \mathbf{r} allows, in principle, reaching an **NMSE** of about -49 dB. Yet, the **IHT** algorithms are only able to reach a level of approximately -46 dB, which is due to the lack of enough measurements. **IFHST**, in its turn, is not limited by that bound, for it does not impose an exact low mrank upon its model. Coincidentally, however, it approximately attains the bound. It also outperforms the other algorithms with regard to convergence speed, thus being the preferable solution in this scenario.

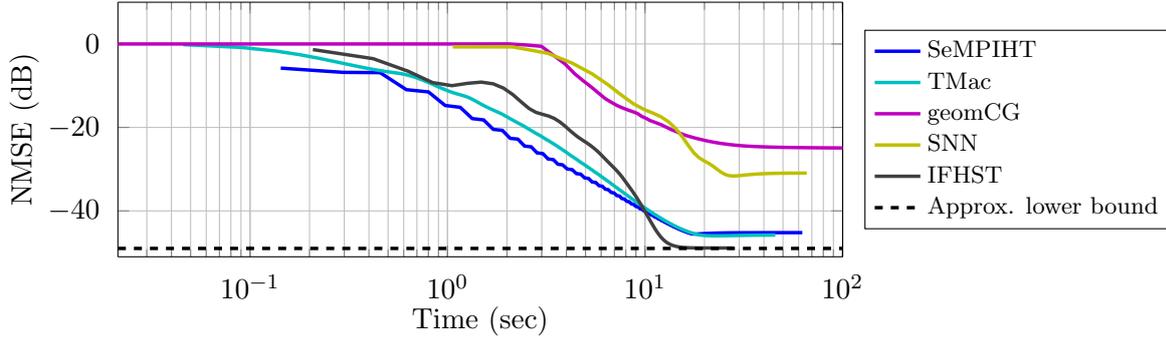


Figure 8.11: Evolution of **NMSE** as a function of time for several **TC** algorithms when applied to recover $100 \times 100 \times 100$ **T2** tensors having a decay parameter of $\varphi = 3/2$, with $\rho = 0.15$.

8.5 Conclusions

Let us highlight the main contributions which were developed along this chapter.

1. The first one is the idea of formulating a single imputation algorithm based on soft **HOSVD** core thresholding aiming to promote a parsimonious (compressible) **OTD** core. We have shown that this can be justified on the grounds of the relation between the compressibility of the **HOSVD** core and that of the modal singular spectra of a tensor.
2. We have explored an interpretation of algorithms based on single imputation with feedback correction as control systems which attempt to drive a certain parsimony-inducing operator into producing outcomes matching the observed tensor entries. This connection sheds light upon the behavior of those algorithms, which may present oscillations, depending on the tuning of the parameters.
3. Our experimental results clearly show the benefits of employing such a feedback correction mechanism in terms of both convergence speed and final reconstruction error, which is true both for **IFHHT** and **IFHST**.
4. It was observed that the recovery regime of **IFHST** varies with ρ in a similar fashion as other state-of-the-art **TC** algorithms, such as **ISS-TIHT** and **TMac**. This is in clear contrast with the performance of **SNN**, which is less efficient.
5. Finally, the results of [Section 8.4.4](#) show that, although **IFHST** is less efficient (in terms of convergence speed and computing cost) when dealing with **T1** tensors, it is well adapted when it comes to the recovery of **T2** tensors, which approximate more closely the characteristics of real-world tensors arising in practical contexts.

The last point will be further corroborated by the experimental results presented in the subsequent chapter.

Traffic data reconstruction via tensor completion

This chapter presents experimental results which demonstrate the usefulness of IFHST in the reconstruction of traffic data. Before presenting our simulation results, we give a concise description of the application context, the relevant performance measures and the employed data.

Contents

9.1	Introduction	167
9.2	Grenoble Traffic Lab	168
9.3	Traffic data employed in our case study	170
9.4	Simulation of missing data	173
9.5	Completion of speeds data	174
9.6	Final remarks	182

9.1 Introduction

For over three decades, several intelligent transportation systems (ITS) have emerged world-wide as a means of improving mobility of people and goods, and particularly as a response to the growing and pervasive problem of traffic congestion, which impinges especially on large urban areas [129]. Currently, there exists a large literature dedicated to the development of models, systems and technology for: (1) enhancing economic and social benefits brought by an efficient transportation system, (2) reducing the occurrence of accidents and (3) mitigating the negative effects caused by traffic congestion, which include time losses, waste of energetic resources and heavy pollution.

A central piece in an ITS is the timely and reliable collection of traffic data, which is fed into models and application systems employed by transportation users, companies, relevant authorities and researchers for information, management, decision-making and prediction.

These data, which are obtained via a wide range of devices and techniques, can comprise vehicle counts, speeds and lengths, among others. The resulting datasets are typically spatio-temporal, featuring daily profiles acquired by multiple sensors. In practice, however, frequent failure of sensing devices, of network links or of processing units causes the occurrence of missing data. As the effectiveness of subsequent processing tasks (such as, *e.g.*, the prediction of travel times [129]) is impaired by the absence of measurements, multiple techniques have been proposed in order to fill in the missing entries.

A strong correlation is typically observed among daily traffic profiles, not only in a spatial sense (*i.e.*, among different sensors), but also in a temporal one, due to the occurrence of quasi-periodic patterns (*i.e.*, very similar behavior among different days or weeks). Consequently, road traffic datasets can often be well approximated by low-rank or low-mrank tensors. Hence, some recent works have focused on the problem of traffic data reconstruction by resorting to TC techniques [191, 163, 164]. In this section, we shall pursue this approach for reconstructing real data gathered in the context of the Grenoble Traffic Lab project [70] by resorting to the IFHST algorithm.

9.2 Grenoble Traffic Lab

Grenoble traffic lab (GTL) is a collaborative project undertaken by the NeCS team,¹ which involves researchers of the GIPSA-lab and of INRIA and is supported by local traffic authorities. A thorough description of this initiative is found in [70]. Its main goal is providing an experimental platform for testing, validating and comparing traffic management/monitoring algorithms, relying upon data collected by a dense network of sensors.

To date, 130 magnetometers have been deployed along the south ring of Grenoble, which is a peri-urban corridor of about 10.35 km connecting two highways (see Fig. 9.1). These devices are capable of detecting the passage of a vehicle, thus serving for counting purposes, for estimating inter-vehicle time intervals, vehicle lengths and speeds (by exploiting detections performed by a pair of sensors). Overall, there are 68 sensors,² 22 of which are placed under on- and off-ramps. The relevance of the collected data stems from the relatively high traffic load often observed on this ring.

One important feature of GTL is the estimation of average travel times for traversing the entire ring. This traffic index is approximated by the formula [129, Section 2.3]

$$\Delta_t = \sum_{j=1}^N \frac{\Delta_{s_j}}{v_j(\tau_j)}, \quad \text{with} \quad \tau_n = t_0 + \sum_{j=1}^{n-1} \frac{\Delta_{s_j}}{v_j(\tau_j)}, \quad (9.1)$$

where the position differences Δ_{s_n} originate from a partition of the corridor length $s_B - s_A$

¹Networked Controlled Systems. Website: <http://www.inria.fr/en/teams/necs>.

²For simplicity, we refer to a pair of magnetometers which yield a single reading as a single sensor.

overestimation of Δ_t due to the use of a too small $v_j(\tau_j)$ in formula (9.1), we use instead $v_j(\tau_j + lT_s)$, where l is the smallest natural number such that $v_j(\tau_j + lT_s) \geq v_{\min}$. Accordingly, a “penalty time” of lT_s is added to Δ_t . Note that this means that vehicles are (virtually) stopped at the time period $\mathcal{I}_k \cup \mathcal{I}_{k+1} \cup \dots \cup \mathcal{I}_{k+l-1}$ due to congestion, and then resume flowing at sampling interval \mathcal{I}_{k+l} . In our simulations, we have employed $v_{\min} = 5$ km/h.

Due to the importance of the above described calculation, which is in particular exploited for future time-travel forecasting [70], we shall evaluate reconstruction methods in terms of their capability of delivering speeds data allowing an accurate approximation of the travel times calculated from the original (reference) data.

9.3 Traffic data employed in our case study

We had access to data spanning the 91-days period between April 1st 2015 and June 30th 2015. This dataset includes time-mean vehicle speeds (measured in km/h) and vehicle counts gathered at all 68 measurement points. The sampling period is $T_s = 15$ seconds, so that there are $24 \times 60 \times 4 = 5760$ daily samples. Hence, the available data comprises two tensors $\tilde{\mathbf{X}}_0, \tilde{\mathbf{W}}_0 \in \mathbb{R}^{5760 \times 68 \times 91}$, whose modes are associated with, respectively, time samples, sensors and days. In our notation, $\tilde{\mathbf{X}}_0$ contains speeds data, while $\tilde{\mathbf{W}}_0$ contains vehicle counts data.

Because $\tilde{\mathbf{X}}_0$ contains a high proportion of missing entries (around 53%), we use as references two other data tensors, $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{W}}$, which coincide with $\tilde{\mathbf{X}}_0$ and $\tilde{\mathbf{W}}_0$ on the effectively measured data but contain imputed values in the place of missing entries. The imputation has been done by GTL’s real-time imputation routine, which we describe ahead.

9.3.1 Preprocessing

Fig. 9.2(a) shows, for each pair $(s, d) \in \langle 68 \rangle \times \langle 91 \rangle$, the proportion of elements $t \in \langle 5760 \rangle$ for which $[\tilde{\mathbf{X}}_0]_{t,s,d}$ is missing; and similarly for $\tilde{\mathbf{W}}_0$ in Fig. 9.2(b). Clearly, there are more missing speed entries than missing counts, because measuring the former requires two successful detections, while only one suffices for the latter. Since speeds measured on access ramps are irrelevant in the above described computation of travel times, we discard data from ramp sensors having average daily missing data ratio larger than 0.7. This allows us to ignore portions of $\tilde{\mathbf{X}}$ containing too many imputed entries. Data coming from other ramp sensors are kept, as they potentially provide useful information for the reconstruction. Similarly, we have discarded all data from Sundays, as they contain a disproportionate amount of imputed entries which do not correlate well with other daily profiles. So, overall we remove data from 12 sensors and 13 days.

As a second preprocessing step, all components of $\tilde{\mathbf{X}}$ are thresholded at the speed limit of 90 km/h to avoid estimating travel times incompatibly with the legal limit. Then, the

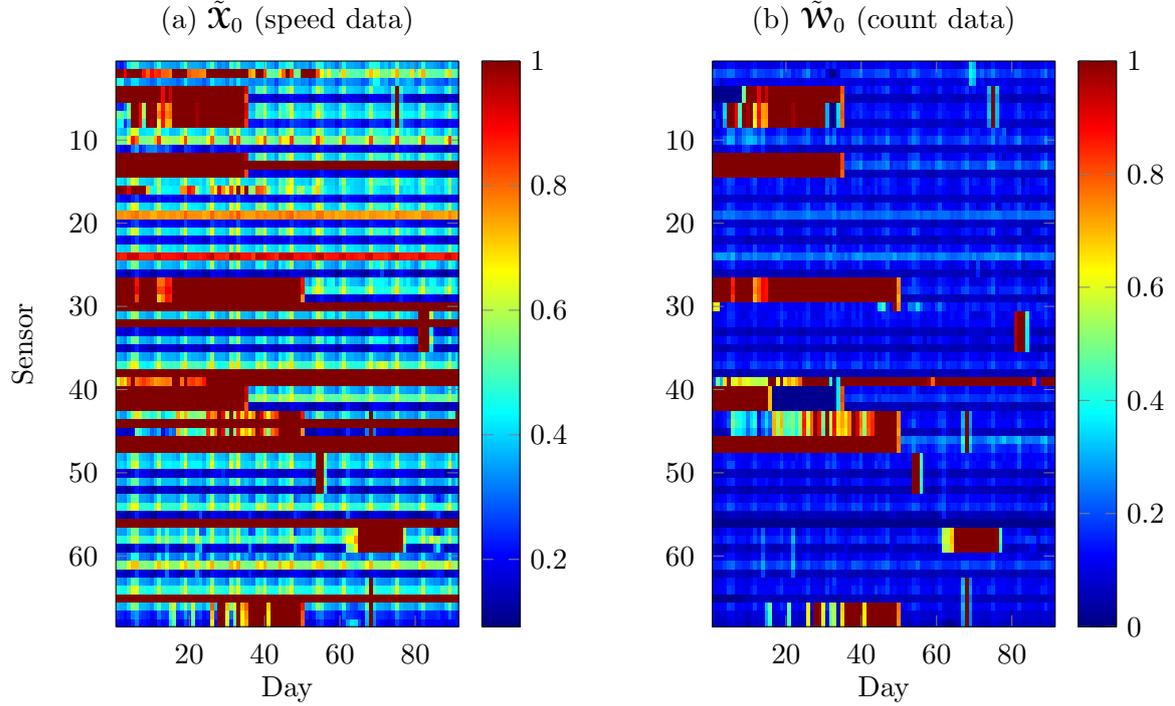


Figure 9.2: Proportion of daily missing data for each sensor: (a) time-mean speeds; (b) vehicle counts.

data are downsampled by a factor of four by computing per-minute average speeds. Also, we discard data measured outside our window of interest, between 6:00 and 20:00, where most traffic jams and slowdowns take place. The resulting downsampled data tensors are denoted by $\mathcal{X}, \mathcal{W} \in \mathbb{R}^{840 \times 56 \times 78}$.

After all these preprocessing steps, the final missing data ratio of \mathcal{X} is 0.38, while that of \mathcal{W} is 0.20. For illustration, in Fig. 9.3 we show the daily profiles measured by three different sensors over 12th June 2015, (a) before and (b) after preprocessing. One can clearly see the degree of correlation among different sensors on this day.

Fig. 9.4 displays for all 56 sensors (a) the measured speeds tensor \mathcal{X}_0 (thresholded at 90 km/h), with missing entries denoted by black dots, and (b) the reference tensor \mathcal{X} , where these missing entries have been imputed. Since the axis of sensors is ordered according to their physical placement along the road, we can clearly visualize spatial correlations among nearby sensors, represented by horizontally spread regions having close values. Fig. 9.4(a) shows that, though some missing elements are scattered over the data, they are to a great extent concentrated, indicating that certain sensors suffer from consistent failure during time periods which can last from some minutes to many hours. As observed in Fig. 9.4(b), the imputed data incurs some loss of information concerning spatial correlations, as seen from the “discontinuities” arising inside approximately homogeneous horizontal regions.

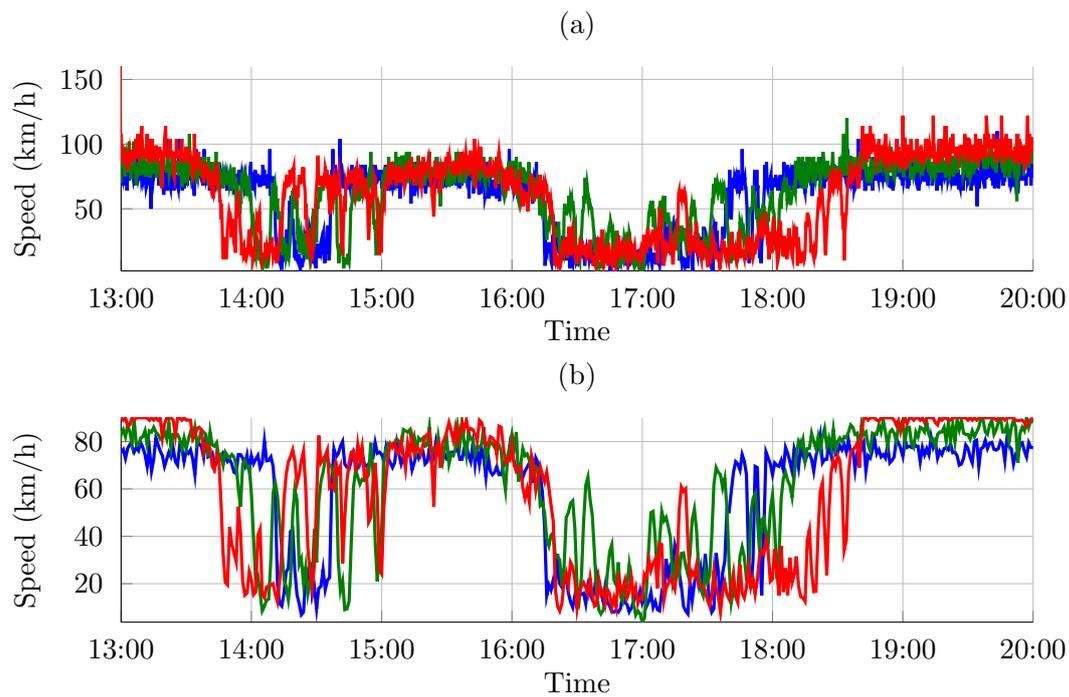


Figure 9.3: Portion of daily profiles measured by three different sensors on 12th June 2015: (a) before preprocessing; (b) after preprocessing.

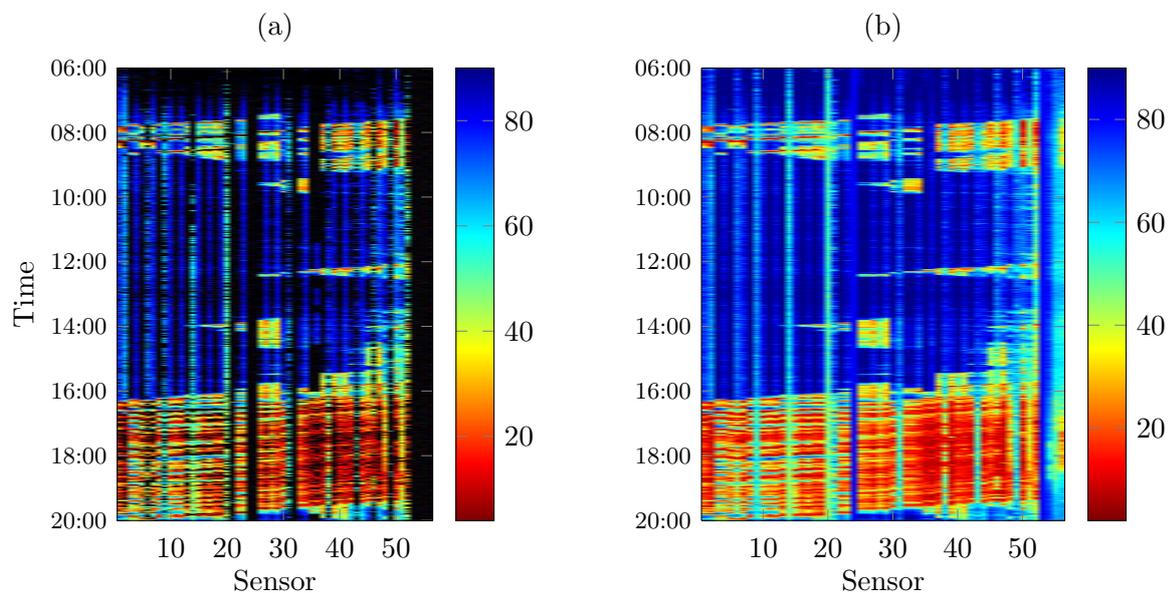


Figure 9.4: Speeds measured by all sensors on 7th May 2015: (a) original data, (b) imputed data.

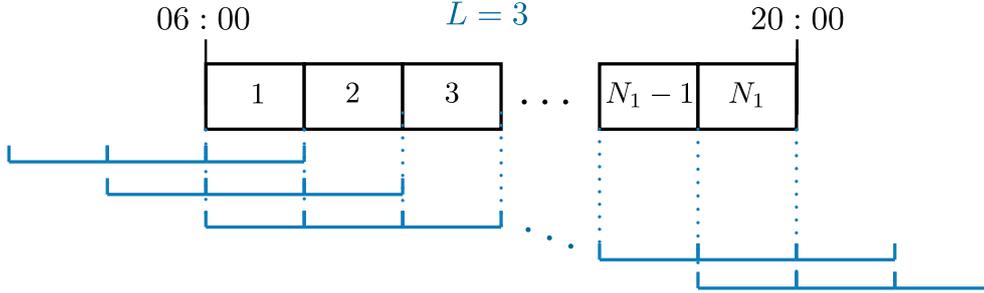


Figure 9.5: Possible placements of a missing data window of length $L = 3$ (represented by the blue brackets) within a daily profile. The squares denote the samples along a given mode-1 fiber (thus associated with time).

9.4 Simulation of missing data

We turn now to the design of sampling operators for our experiments. In order to simulate the systematic loss of data which happens in practice (cf. Fig. 9.4), it is desirable to randomly distribute missing data windows of some given length along the time dimension (*i.e.*, along the mode-1 fibers) of the speeds tensor. We thus proceed as follows:

1. A subsampling ratio $\rho \in (0, 1)$ and a missing window length L (in minutes) are specified.
2. Then, we compute the total number of missing data windows as $N_w = \lfloor (\bar{N} - M)/L \rfloor$, where $M = \rho\bar{N}$ and $\bar{N} = 3,669,120$.
3. Each one of these windows is randomly assigned to one daily profile of one sensor, with uniform probability. Its location inside that daily profile is also randomly drawn from a uniform distribution which includes the possibilities of starting at the last sample or of ending at the first sample. This is illustrated in Fig. 9.5, where the squares represent all samples of a daily profile (of length $N_1 = 840$) and the blue brackets indicate all possible placements of a missing data window having length $L = 3$. Note that the effective length of the placed window can thus be smaller than L .
4. After distributing all N_w windows, the effective number of missing data entries, N_m , can be smaller than $\bar{N} - M$, because of the border effect mentioned in the previous item and also due to the superposition of multiple windows. Hence, in order to attain the prescribed value of ρ , we randomly distribute $\bar{N} - M - N_m$ single missing entries uniformly over the samples which are not covered by any window.

By tuning the window length L , we can adjust the sampling scheme to reflect a more or less systematic failure of sensors. In practice, if ρ is sufficiently small, there is a significant probability of window overlapping and of window placement on the borders of a daily profile, and thus a fair amount of uniform sampling also takes place. This approximates the behavior

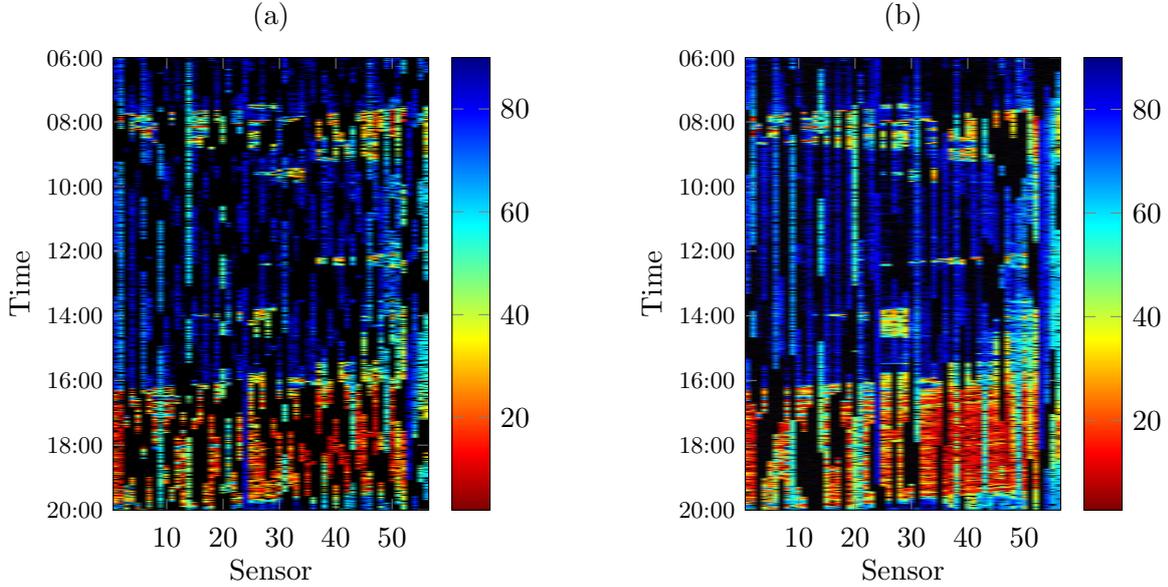


Figure 9.6: Examples of missing data patterns generated with the procedure of Section 9.4: (a) $\rho = 0.4$ and $L = 60$; (b) $\rho = 0.6$ and $L = 120$. The black dots indicate missing entries.

seen in Fig. 9.4, where, in addition to the long missing data windows, many single missing entries are also scattered around.

Fig. 9.6 shows two examples of missing data patterns generated as described above: (a) with $\rho = 0.4$ and $L = 60$; and (b) with $\rho = 0.6$ and $L = 120$. These plots refer to the same day of Fig. 9.4. It can be seen that Fig. 9.6(a) contains more horizontally connected regions of missing data than Fig. 9.6(b), due to the larger values of ρ and L used in the latter. One also observes many single missing entries scattered on both plots.

9.5 Completion of speeds data

This section presents the completion experiments performed on speeds data collected by GTL. After describing the evaluated algorithms, we outline the experimental procedure, and then display the simulation results.

9.5.1 Description of evaluated algorithms

The compared reconstruction algorithms are as follows:

1. Temporal interpolation (Interp): As a baseline of comparison, we employ a very simple linear interpolation scheme along the temporal (the first) mode. It goes as follows:
 - Whenever the samples $x_{k,s,d}, \dots, x_{k+I-1,s,d}$ are not observed, but $x_{k-1,s,d}$ and $x_{k+I,s,d}$ are, with $k-1 \geq 1$ and $k+I \leq N_1$, then these missing entries are

estimated as

$$\hat{x}_{k+i-1,s,d} = x_{k-1,s,d} + \frac{(x_{k+I,s,d} - x_{k-1,s,d})}{I+1} i, \quad \text{for } i \in \langle I \rangle. \quad (9.3)$$

- If, on the other hand, $k = 1$ or $k + I - 1 = N_1$, then the estimates $\hat{x}_{k,s,d}, \dots, \hat{x}_{k+I-1,s,d}$ are set to the nominal value $s_{\max} = 90$ km/h. Note that this is a reasonable rule for small I , since the traffic is typically in free-flow condition at the boundaries of the daily period of interest (between 6:00 and 20:00).

2. **GTL's algorithm:** This imputation method also exploits temporal correlations. But, since the missing samples are estimated immediately after failure is detected, their reconstruction cannot depend on future information. Instead, only a limited number of (temporally) preceding samples are averaged. Spatial correlations are similarly exploited. The procedure is as follows. At the (discrete) time instant k of day d :

- (i) The algorithm first tries to impute each missing sample by averaging its four temporally preceding ones (at the same day and sensor), via the formula

$$\hat{x}_{k,s,d} = \frac{1}{L_t} \sum_{l=1}^{L_t} x_{k-l,s,d}, \quad \text{where } L_t \triangleq \min\{4, k-1\} \quad (9.4)$$

denotes the effective length of the temporal averaging window.

- (ii) If at least one among the samples in the sum of (9.4) was also not observed,⁵ then step (i) fails. So, the algorithm tries to impute each remaining missing entry by averaging values read by neighboring sensors (*i.e.*, those who immediately precede and follow sensor s on the road), via the formula

$$\hat{x}_{k,s,d} = \frac{1}{L_n} \sum_{l=1}^{L_n} x_{k,s_l,d}, \quad (9.5)$$

where $L_n \in \langle 2 \rangle$ is the number of available neighbors, whose indices are s_l , $l \in \langle L_n \rangle$. Most sensors have two neighbors, but those on ramps and on the first and last collection points have only one.

- (iii) Again, step (ii) fails when neighbors' samples are also missing at instant k . In that case, the algorithm draws from historical information, attempting to compute

$$\hat{x}_{k,s,d} = \frac{1}{L_h} \sum_{l=1}^{L_h} x_{k,s,d_l}, \quad (9.6)$$

where $\{d_1, \dots, d_{L_h}\}$ is the set of L_h preceding days of the same class of d —*i.e.*, they are either weekdays or Saturdays,⁶ according to d . In practice, a large value

⁵Even if one or more of the samples in the summand of (9.4) were themselves imputed, these imputed values are *not* employed when computing that of $\hat{x}_{k,s,d}$, in order to avoid biasing the results.

⁶Recall that Sundays have been excluded from the dataset.

is assigned to the length L_h (such as 120 for weekdays and 24 for Saturdays). However, given the limited length of our dataset, here we compute the average among all available preceding days of the same class.

- (iv) As expected, the imputation attempt of step (iii) also fails when one of the samples x_{k,s,d_l} in the historical data was not measured. If so, the algorithm simply sets, as a last resort, $\hat{x}_{k,s,d} = s_{\max}$, which means assumption of a free-flow condition.
3. Fixed-point continuation algorithm (FPCA) with approximate SVD [139]: This algorithm consists of a fixed-point continuation scheme which solves the matrix completion problem (6.9) by considering a sequence of subproblems of the form

$$\min_{\mathbf{X} \in \mathbb{R}^{N_1 \times N_2}} \tau \|\mathbf{X}\|_* + \frac{1}{2} \|\mathbf{y} - \mathcal{A}(\mathbf{X})\|_2^2, \quad (9.7)$$

with decreasing values of the regularization constant τ . We apply it to complete the mode-1 unfolding $\mathbf{X}_{(1)} \in \mathbb{R}^{840 \times 4368}$. This allows exploiting correlation among daily profiles. We have employed the Matlab code provided by its authors,⁷ which contains a routine for automatic adjustment of the parameters. In our case, it yields the parameter values $\mu = 10^{-4}$, $\eta = 0.25$, $\tau = 2$ and a maximum number of inner iterations (for the subproblems) of 10. For details, see [139].

4. SNN: The popular TC approach based on SNN minimization (cf. Algorithm 6.1) has been employed for traffic data reconstruction in [163, 164]. We therefore include it in our evaluation with the measurements as constraints (*i.e.*, with $\lambda \rightarrow 0$), which is essentially the same method as that used in [163, 164]. As in [195], we set $\eta = \eta_0 / \sigma_{\mathbf{y}}$, choosing $\eta_0 = 50$, where $\sigma_{\mathbf{y}}$ is the standard deviation of the observed entries of \mathcal{X} . Also, the algorithm is allowed to run for a maximum number of $K_{\max} = 150$ iterations.
5. Block coordinate descent (BCD) algorithm for nonnegative CPD: Because the speed of a vehicle is a nonnegative quantity, taking that property into account is desirable. We thus include in our evaluation the BCD algorithm proposed in [214] which estimates nonnegative factors of a CPD model. This also allows assessing how a technique based on tensor rank compares with others based on mrank. In the context of traffic data completion, the use of a CPD-based approach has been considered in, *e.g.*, [191]. The maximum number of iterations for this algorithm was set at $K_{\max} = 500$. We chose the rank empirically, by trying to find a sufficiently high rank value for which no significant improvement was observed when it was further increased. This led to the values of $R = 300$ for $\rho = 0.4$, $R = 700$ for $\rho = 0.6$ and $R = 800$ for $\rho = 0.8$.
6. SeMPIHT: The following ISS settings are applied: $\alpha = 0.5$, $\beta = 0.7$, $L = 3$ and initial candidate step size $\mu_k = 1$. The target mrank \mathbf{r} was also empirically adjusted in an

⁷The code is available at <http://ww1.se.cuhk.edu.hk/~sqma/FPCA.html>.

attempt to achieve the best possible performance. It was set as $\mathbf{r} = (100, 20, 25)$ for $\rho = 0.4$, $\mathbf{r} = (170, 25, 35)$ for $\rho = 0.6$ and $\mathbf{r} = (350, 30, 40)$ for $\rho = 0.8$. The second GRI scheme discussed in Section 7.4 (in which each mrank component has a growth rate proportional to its final value) is employed, being tuned so that the target mrank is attained at iteration $[\mathbf{r}]_1$. The maximum number of iterations is $K_{\max} = [\mathbf{r}]_1 + 150$.

7. **IFHST**: The parameters of **IFHST** are set as $\psi = 0.85$, $\eta_0 = 0.01$ and $K_{\max} = 150$.

While the maximum number of iterations was set for each algorithm according to its observed computing cost and convergence speed, their tolerances are all given by $\epsilon = 10^{-6}$. In particular, **BCD** was generally stopped without satisfying this tolerance, due to its slow convergence. Yet, the obtained **NSE** curves suggest no significant improvements would be achieved by performing additional iterations. **SeMPIHT** also did not perform well in this regard, being usually stopped only after reaching its maximum number of iterations.

9.5.2 Experimental procedure

We have considered 9 scenarios, corresponding to all pairs $(\rho, L) \in \{0.2, 0.4, 0.6\} \times \{30, 60, 120\}$. In each scenario, a random **SO** generated as described in Section 9.4 is applied to sample \mathbf{X} . The output $\hat{\mathbf{X}}$ of each completion algorithm is then filtered by the first-order lowpass Butterworth filter

$$H(z) = \frac{0.1602z + 0.1602z^{-1}}{1 - 0.6796z^{-1}}, \quad (9.8)$$

whose normalized cutoff frequency is of 0.03π rad/sample. This filtering process takes place along the first mode, so that all daily profiles are smoothed out by $H(z)$.

To guarantee that all entries of the filtered estimate lie in $[0, 90]$, we threshold them at 90 km/h and replace negative samples by employing an interpolation scheme akin to **Interp**. The processed estimate is then denoted by $\hat{\mathbf{X}}_f$. Of course, \mathbf{X} is also transformed in the same way, yielding the filtered reference tensor \mathbf{X}_f . At that point, we measure the normalized quadratic error

$$\text{NSE}_{\text{speed}} \triangleq \frac{\|(\mathbf{X}_f)_{\bar{\Omega}} - (\hat{\mathbf{X}}_f)_{\bar{\Omega}}\|_F^2}{\|(\mathbf{X}_f)_{\bar{\Omega}}\|_F^2}, \quad (9.9)$$

for each reconstruction method, where $\bar{\Omega}$ is the complement of the set of sampled multi-indices, Ω . Similarly, we compute the root mean squared error (**RMSE**)

$$\text{RMSE}_{\text{speed}} \triangleq \sqrt{\frac{1}{|\bar{\Omega}|} \sum_{(n_1, n_2, n_3) \in \bar{\Omega}} \left([\mathbf{X}_f]_{n_1, n_2, n_3} - [\hat{\mathbf{X}}_f]_{n_1, n_2, n_3} \right)^2} \quad (9.10)$$

and also the mean absolute percentage error (**MAPE**), which is defined as

$$\text{MAPE}_{\text{speed}} \triangleq \frac{100}{|\bar{\Omega}|} \sum_{(n_1, n_2, n_3) \in \bar{\Omega}} \left| \frac{[\mathbf{X}_f]_{n_1, n_2, n_3} - [\hat{\mathbf{X}}_f]_{n_1, n_2, n_3}}{[\mathbf{X}_f]_{n_1, n_2, n_3}} \right|. \quad (9.11)$$

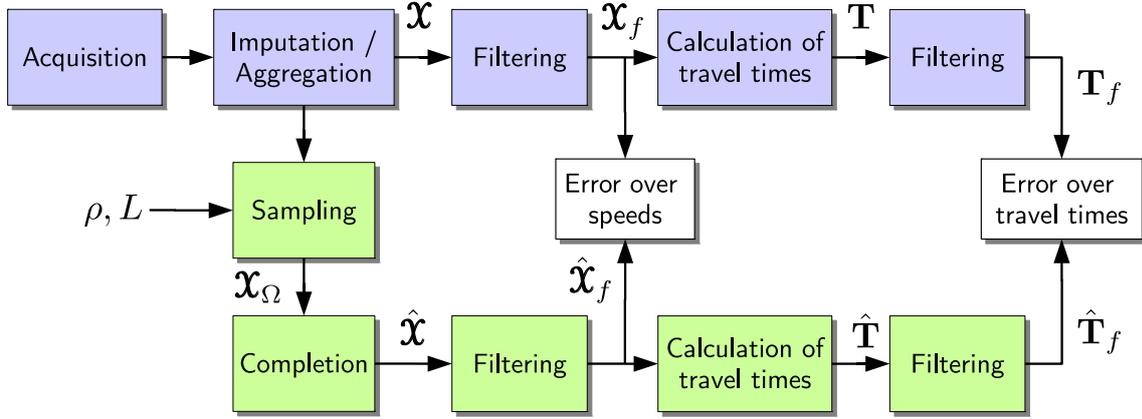


Figure 9.7: Experimental procedure: blue boxes indicate the usual stages of GTL, while the green ones represent the stages involved in our experiment.

Subsequently, we proceed to the computation of travel times for each day of the considered time horizon. This is done by applying the calculation procedure described in Section 9.2 once for each $t_0 \in \{6:00, 6:01, \dots, 19:29\}$. This set has cardinality 810. Hence, a matrix $\hat{\mathbf{T}} \in \mathbb{R}^{810 \times 78}$ containing all computed travel times is constructed for each algorithm. Accordingly, a reference matrix of same dimension, \mathbf{T} , is formed by travel times calculated from \mathbf{X}_f . Due to measurement imperfections and to approximations involved in the calculation, the resulting daily travel time profiles contain strong high-frequency oscillations. Yet, they convey the evolution of the traffic congestion state along one day, which should be smooth. Thus, we filter each column of the matrices $\hat{\mathbf{T}}$ and \mathbf{T} by (9.8), obtaining matrices $\hat{\mathbf{T}}_f$ and \mathbf{T}_f , respectively, which comprise more physically reasonable travel time profiles. Then, we compute for each algorithm the performance measures NSE_{tt} , RMSE_{tt} and MAPE_{tt} , which are defined analogously to (9.9)–(9.11).

For clarity, this experimental procedure is portrayed in Fig. 9.7. The blue boxes represent operations pertaining to (a simplified version of) GTL’s usual flow, while the green ones stand for the stages involved in our experiment.

9.5.3 Simulation results

Table 9.1 presents the results, with the best performance indices per scenario underlined. Overall, BCD and IFHST have the best performance, generally with a large advantage over the other methods. The accuracy of Interp, in particular, degrades significantly when L grows, as expected. The same rate of degradation is not observed for other algorithms. In the case of GTL, this is explained by the fixed window size of its temporal interpolation stage. Yet, its estimates are quite inaccurate, due to the poor exploitation of correlations.

When $\rho = 0.6$, IFHST achieves the best reconstruction of speeds, but BCD scores the best

	$\rho = 0.4, L = 30$										$\rho = 0.4, L = 60$										$\rho = 0.4, L = 120$																					
	BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN			BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN		
	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}						
Speeds	4.49	7.72	3.98	7.92	17.87	6.23	6.32	4.67	8.52	4.18	9.79	18.04	6.39	6.59	4.72	9.22	4.74	12.44	17.98	6.57	6.72	8.13	13.15	25.03	6.70	6.70	8.13	13.15	25.03	6.70	6.70	8.13	13.15	25.03	6.70	6.70	8.13					
Travel times	0.51	1.23	0.53	0.51	2.67	2.36	1.12	0.56	1.38	0.50	0.75	2.69	2.76	1.15	0.53	1.72	0.49	1.20	2.71	2.38	1.16	1.16	3.31	3.47	1.16	1.16	3.31	3.47	1.16	1.16	3.31	3.47	1.16	1.16	3.31	3.47						
Exec. time (sec)	312.36	84.46	127.75	7.85	21.26	618.23	937.69	293.13	81.68	119.66	8.79	22.05	566.31	857.84	313.62	81.12	130.36	10.02	21.23	585.12	960.98	293.13	81.68	119.66	8.79	22.05	566.31	857.84	313.62	81.12	130.36	10.02	21.23	585.12	960.98	293.13	81.68	119.66	8.79	22.05		

	$\rho = 0.6, L = 30$										$\rho = 0.6, L = 60$										$\rho = 0.6, L = 120$																					
	BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN			BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN		
	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}						
Speeds	3.82	7.66	3.56	7.59	16.59	6.10	5.77	4.06	8.76	3.75	9.53	17.06	6.30	6.16	4.11	9.24	3.79	12.34	16.84	6.45	6.26	7.55	13.01	21.86	6.21	7.40	4.37	10.65	4.11	13.01	21.86	6.21	7.40	4.37	10.65	4.11	13.01	21.86	6.21	7.40		
Travel times	0.32	0.93	0.44	0.38	1.51	1.21	0.69	0.30	1.01	0.46	0.54	1.47	1.13	0.70	0.32	1.00	0.46	0.83	1.49	1.20	0.77	1.15	1.20	0.77	1.15	1.20	0.77	1.15	1.20	0.77	1.15	1.20	0.77	1.15	1.20	0.77	1.15	1.20	0.77			
Exec. time (sec)	903.83	94.15	128.63	6.15	14.32	445.86	657.27	797.25	84.61	118.79	6.34	13.65	423.63	700.13	912.98	92.36	124.09	6.79	14.37	431.63	623.28	797.25	84.61	118.79	6.34	13.65	423.63	700.13	912.98	92.36	124.09	6.79	14.37	431.63	623.28	797.25	84.61	118.79	6.34	13.65		

	$\rho = 0.8, L = 30$										$\rho = 0.8, L = 60$										$\rho = 0.8, L = 120$																					
	BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN			BCD			FPCA			IFHST			Interp			GTL			SeMPIHT			SNN		
	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}	NSE _{speed} (dB)	RMSE _{speed}	MAPE _{speed}	NSE _{tt} (dB)	RMSE _{tt}	MAPE _{tt}						
Speeds	3.39	8.01	3.24	7.53	15.49	6.32	5.35	3.65	8.53	3.44	9.50	15.68	7.14	5.70	3.79	9.76	3.52	12.02	15.80	7.02	5.82	7.30	12.97	19.59	6.73	7.30	12.97	19.59	6.73	7.30	12.97	19.59	6.73	7.30	12.97	19.59	6.73	7.30				
Travel times	0.15	0.47	0.16	0.24	0.59	0.78	0.35	0.17	0.52	0.20	0.33	0.52	0.63	0.37	0.20	0.64	0.31	0.50	0.61	1.05	0.44	0.44	1.32	1.51	1.49	1.11	1.49	1.11	1.49	1.11	1.49	1.11	1.49	1.11	1.49	1.11	1.49	1.11				
Exec. time (sec)	921.50	93.94	114.75	2.67	7.23	771.67	507.41	919.23	98.21	121.58	3.09	7.85	858.00	541.99	923.17	95.92	114.49	3.31	7.15	791.16	465.09	923.17	95.92	114.49	3.31	7.15	791.16	465.09	923.17	95.92	114.49	3.31	7.15	791.16	465.09	923.17	95.92	114.49				

Table 9.1: Simulation results on the reconstruction of speeds data and subsequent estimation of travel times.

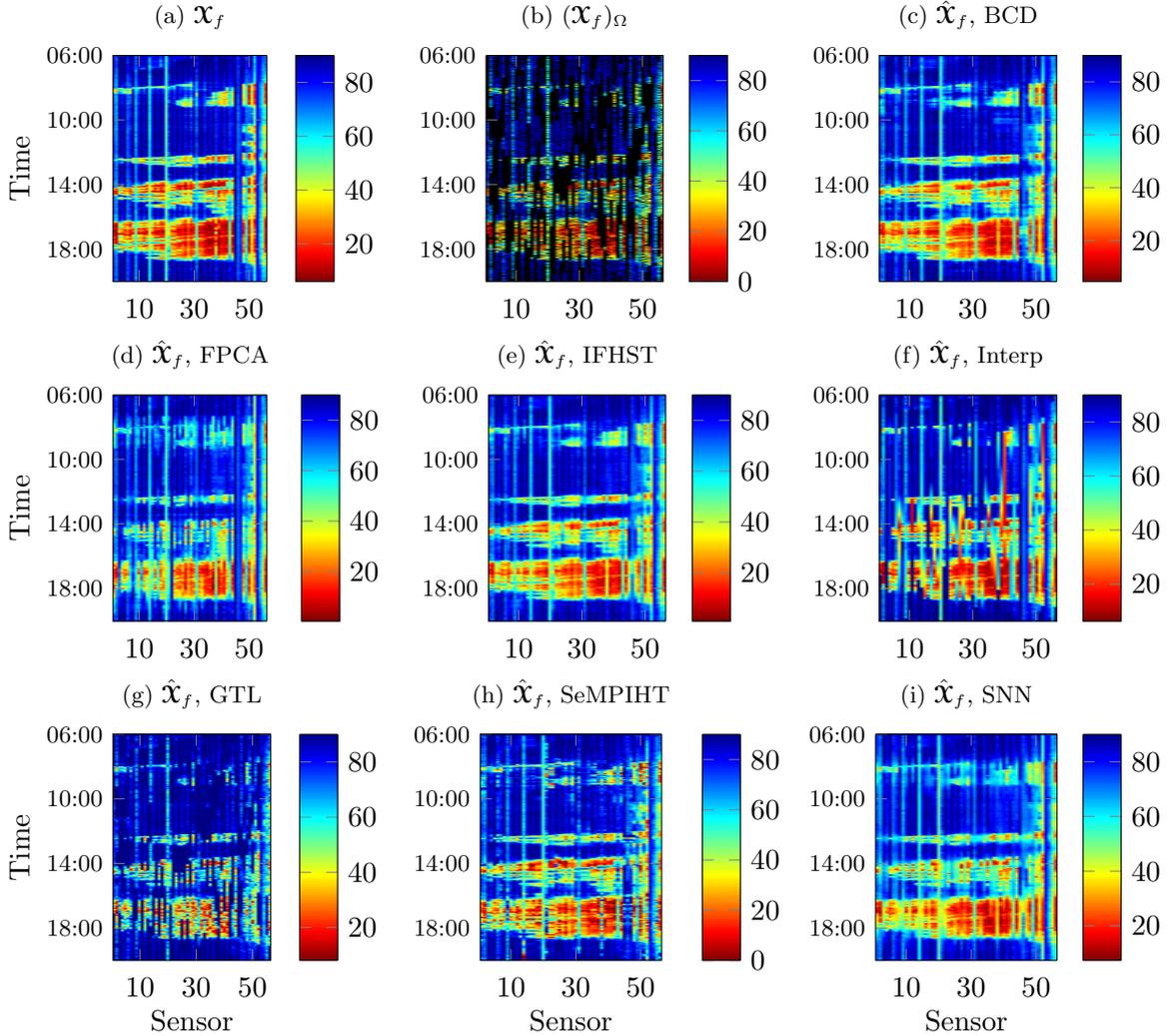


Figure 9.8: Reconstruction of speeds on June 12th 2015 achieved by all algorithms for $\rho = 0.4$ and $L = 120$: (a) reference data; (b) pattern of missing entries (in black color); (c)–(i) outputs of reconstruction algorithms.

NSE_{tt} and $RMSE_{tt}$. This apparently contradictory behavior happens because the computed travel times are somewhat sensitive to the occurrence of underestimated speeds, as it is the inverse of space-mean speeds which appears on formula (9.1). It was observed that IFHST is more prone to producing underestimated speeds than BCD, which can yield a few “outliers” in $\hat{\mathbf{T}}_f$ that largely deviate from their counterparts of \mathbf{T}_f . Of course, this impacts the quadratic performance indices more severely than $MAPE_{tt}$.

At any rate, the estimates produced by IFHST are often the most accurate ones, or at least among the two most accurate ones when it is outperformed by BCD. Now, when we consider the execution times of these two algorithms, it becomes clear that IFHST provides the best compromise between computing cost and overall accuracy.

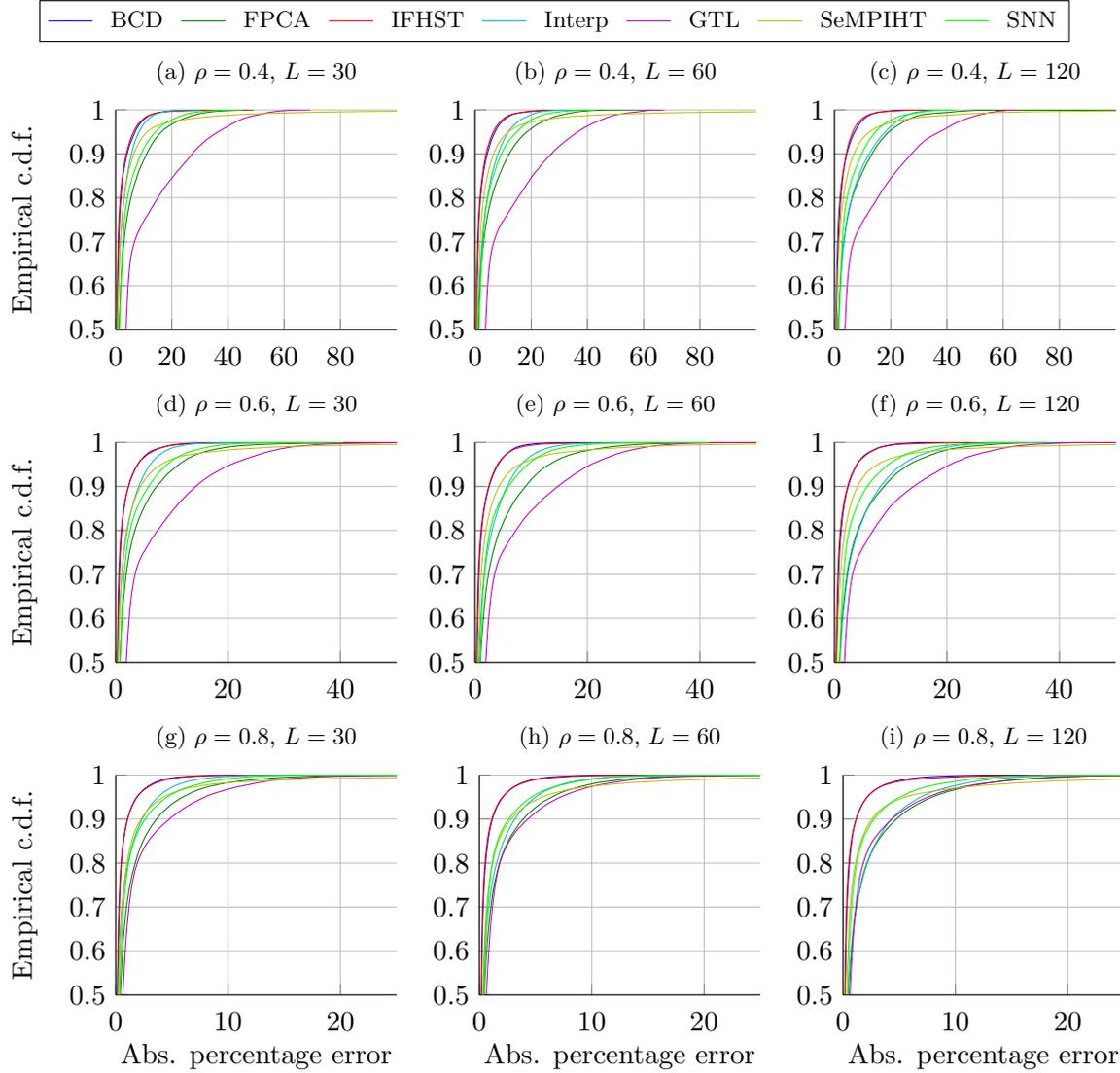


Figure 9.9: Empirical c.d.f. of the absolute percentage errors committed by each algorithm with respect to travel times.

In Fig. 9.8 we compare the results of all algorithms over June 12th 2015 in terms of speeds reconstruction, considering the scenario $\rho = 0.4$, $L = 120$. A quick inspection shows the clear superiority of **BCD** and **IFHST** over the other approaches. **Interp** and **GTL** arguably deliver the worst results in terms of overall preservation of data features. In particular, **Interp** ignores spatial correlations, thus destroying the horizontal coherence of the data to a great extent. This effect is also observed for **GTL**, which imputes too many samples with the nominal speed of 90 km/h. **FPCA** performs poorly, especially within the time window from 12:00 to 15:00. This seems to come from the loss of structure caused by the “matricization,” due to which a low-rank solution is obtained at the expense of losing day-specific details. **SNN** and **SeMPIHT** perform better, but their results are also perceptibly degraded.

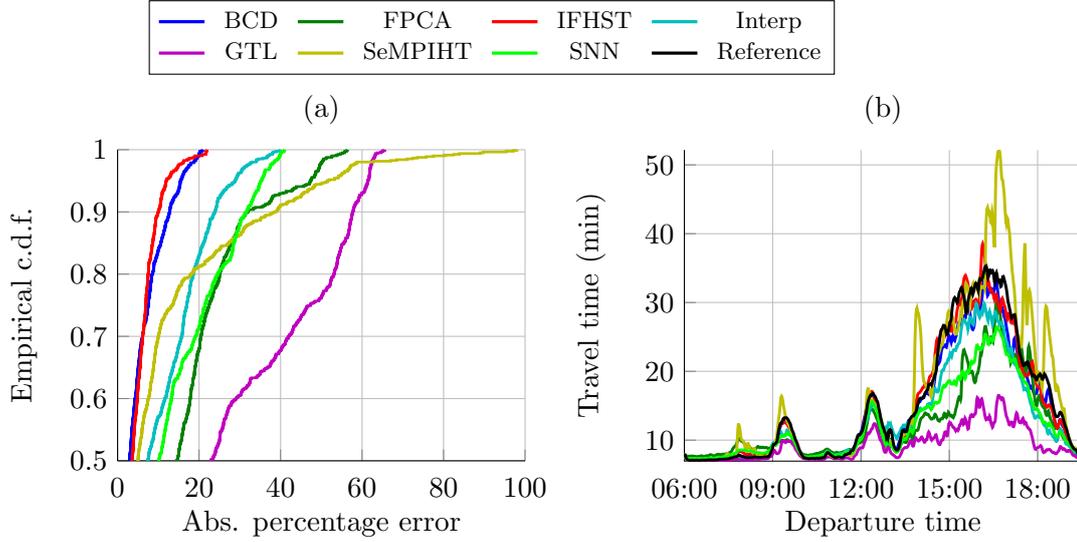


Figure 9.10: Estimation of travel times over the 26th June 2015: (a) c.d.f. of absolute percentage errors; (b) estimated times and reference values.

Let us now turn our attention to the accuracy of estimated travel times. First, we plot in Fig. 9.9 the c.d.f. of the absolute percentage errors

$$[\mathbf{e}]_n = 100 \times \left| \frac{[\text{vec}(\mathbf{T}_f)]_n - [\text{vec}(\hat{\mathbf{T}}_f)]_n}{[\text{vec}(\mathbf{T}_f)]_n} \right| \quad (9.12)$$

produced by each method. These curves show the clear superiority of BCD and IFHST, whose errors are smaller than 20% for over 99.5% computed travel times in every scenario. In Fig. 9.10(a), we plot the same c.d.f. but focusing on 26th June 2015. This date was chosen because a substantial congestion took place on it. Fig. 9.10(b) shows the reference travel times on that day, along with the estimates produced by each algorithm. IFHST and BCD clearly deliver the best performance, while SeMPIHT gives the worst one, due to its frequent underestimation of vehicle speeds. GTL, on the other hand, is prone to overestimate the speeds, as it imputes missing values with the nominal speed (*i.e.*, the speed limit) too often.

Lastly, convergence plots of all iterative algorithms are plotted on Fig. 9.11 for two different scenarios: $\rho = 0.4$, $L = 120$ and $\rho = 0.8$, $L = 60$. (In these figures, the NSE measured at each iteration is over the whole tensor, and not only over $\bar{\Omega}$.) The curves exhibit the unambiguous superiority of IFHST over other approaches, which was seen to hold true in all scenarios.

9.6 Final remarks

Our computer experiments ratify the usefulness of IFHST for the reconstruction of real-world tensor data. More specifically, the performance achieved by this algorithm in the completion

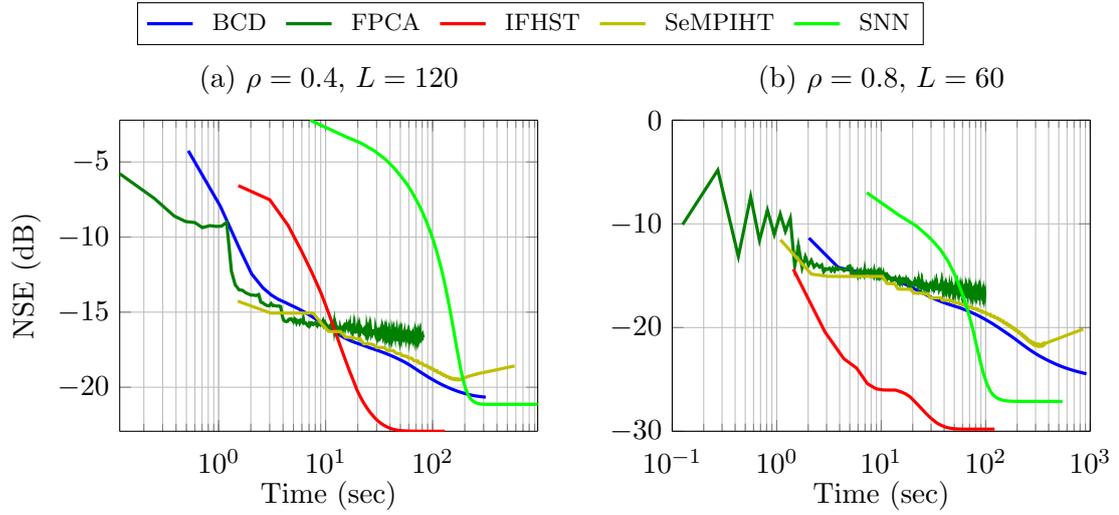


Figure 9.11: Convergence plots of iterative algorithms for two selected scenarios.

of tensors comprising vehicle speeds data was quite satisfactory and better than those of other approaches, either in terms of reconstruction accuracy or of computing time (or both), even in the presence of relatively long portions of missing data. This robustness with respect to consistent failure of sensing devices is a desirable feature in the context of traffic data acquisition, which thus makes **IFHST** a viable alternative in practice.

Conclusions and perspectives

This thesis has offered contributions to the problems of structured canonical polyadic decomposition estimation and of low-rank tensor recovery. We now present our overall conclusions and point out some research perspectives worth future investigation.

10.1 Structured canonical polyadic decomposition

Chapter 4 has developed two approaches to the problem of SCPD which are quite different in nature. In addition to the simplicity and generality of CALS, the approximate iterates we have formulated add to its versatility, allowing mixed strategies for the reduction of its computing cost. The algebraic solution for circulant-constrained CPDs, in its turn, is directed towards a particular case, but is able to completely exploit the structure of the target tensor in a non-iterative manner. In particular, it provides an exact solution in the absence of noise.

The numerical studies of Chapter 5 make clear the utility of the mixed CALS variant which employs first approximate and then exact iterates, as they show it achieves cost reduction and even avoids local minima when the initialization is far from a global optimum. Another major conclusion coming from that chapter is the quite satisfying statistical performance of SCPD estimators which combine a non-iterative method providing an approximate initial solution with an iterative algorithm for refinement. In particular, combinations of simple *ad-hoc* methods based on our algebraic solution or on the SBS method with CALS were capable of approaching the ECRB in our simulations, while keeping a smaller cost in comparison with other evaluated alternatives.

10.1.1 Perspectives

We list below some research topics concerning structured tensor decompositions which deserve further investigation.

- *Generalization of algebraic solution.* It would be valuable to investigate whether the algebraic solution devised on Chapter 4 can be extended to more general situations in which non-circulant matrix factors are also involved. A motivating example is the Wiener-Hammerstein identification problem described in Section 3.2, where one of

the factors is a circulant matrix post-multiplied by a diagonal one. Generalizations to handle block-Topelitz or block-circulant factors seem also desirable, due to their importance in some applications [57, 185].

- *Nonnegativity constraints.* Nonnegative tensor decompositions are useful in many problems, such as data mining, chemometrics, hyperspectral imaging, computer vision, biomedical engineering and audio source separation. Such a wide applicability has led several researchers to study this topic [177, 46, 214, 162]. The introduction of structural constraints of the form we have considered could be of interest in some of these problems. For example, a structured nonnegative CPD arises in [154].
- *Theoretical properties of SCPD.* Given the practical relevance of the SCPD, a better understanding of its theoretical properties is desirable. In particular, an important question is whether the ill-posedness of the best rank- R approximation problem discussed in Section 2.5.2.5 can be circumvented by imposing structured factors, such as banded circulant matrices. Also, the derivation of other uniqueness results for particular structures, in addition to existing ones [53, 185, 184], can be considered.
- *Other structured tensor models.* More generally, the estimation of structured versions of other tensor models can also be envisaged. For instance, a structured tensor decomposition in rank- $(1, L, L)$ terms, which is more general than the CPD, arises in the problems of convolutive blind signal separation and blind deconvolution [62, 185]. Computing hierarchical tensor models [95] having structured components seems also an interesting problem, although we are not aware of applications in this case.

10.2 Low-rank tensor recovery

The SeMPIHT algorithm we have proposed in Chapter 7 takes advantage of the attractive features of the approximate best mrank- r approximation method via sequentially optimal modal projections, offering an efficient alternative to the LRTR problem. We have presented systematic numerical simulations which suggest that the existing tensor IHT algorithms, including SeMPIHT, have order-optimal sampling requirements for Gaussian measurement operators. Unfortunately, though, our derived theoretical results fall short of the expected RIC conditions for ensuring recovery of low-mrank tensors. Our simulation results also give convincing evidence on the remarkable improvement brought by our step size selection and gradual rank increase heuristics.

Despite the merits of SeMPIHT, it fails to deliver satisfactory results when the low-mrank assumption is violated to a significant extent, meaning the mrank of the target tensor is full (as typically happens in applications) and its modal singular spectra do not decay fast

enough. To address this more adversarial scenario in the case of tensor completion, we have proposed in Chapter 8 the IFHST algorithm, which couples a single imputation scheme with a soft thresholding operation whose purpose is promoting a solution having a compressible HOSVD core. This property was shown to be tied to a fast decay of the modal singular spectra of the target tensor.

Chapter 9 has illustrated the effectiveness of the IFHST strategy when applied to real-world data arising in a traffic data reconstruction context. The employed sampling scheme simulates systematic failure of some data acquisition devices, resulting in concentrated portions of missing data in the observed tensor. Even so, IFHST yields an accurate reconstruction of these unknown entries. Though a CPD-based approach with nonnegative factors had a quite close performance with respect to reconstruction accuracy, it was seen to demand more computing effort, thus making IFHST the most cost-effective solution.

10.2.1 Perspectives

Let us now list some foreseeable extensions of our contributions to LRTR.

- *Derivation of recovery results.* An improvement of the performance bound (7.34) and of the exploited RIC condition is worthwhile. This probably requires employing other proof techniques for taking advantage of the complete low-rank structure. An even larger theoretical gap exists in the case of tensor completion, where RIC-based results do not apply. The derivation of recovery guarantees for SeMPIHT in this setting is likely to be quite hard, given the absence of similar results for IHT methods in LRMR. It is unclear whether a proof based on coherence conditions could be pursued for IFHST.
- *Improvements in computational efficiency.* The computing cost per iteration of our developed algorithms is still a limiting factor when we consider application to very large tensors. In the case of SeMPIHT, the randomized techniques presented in [97] for approximate computation of a truncated SVD can alleviate the required effort, but possibly at the expense of a slower convergence. As for IFHST, one could try to come up with a way of exploiting the sparsity of the thresholded HOSVD core at early iterations in order to improve efficiency.
- *Extension of conjugate gradient-based approach.* IHT algorithms employing conjugate gradient techniques in order to accelerate convergence have been proposed in [14] for CS and MC. It is reasonable to suppose that this strategy could also be beneficial in the LRTR case. In particular, a SeMPIHT variant employing a conjugate gradient scheme seems worth pursuing.
- *Recovery of low-rank and sparse components.* A formulation modeling the target tensor

as a sum of low-rank and sparse components has already been proposed in [106]. However, it relies on the minimization of **SNN** approach, which is suboptimal [146]. Hence, an interesting research direction points at the extension of our proposed algorithms in order to deal with this problem. First, this can yield a robust recovery method with respect to outliers. Second, it is of interest in some applications such as the separation of foreground and background elements in video sequences. Indeed, the video data used in the simulations of [211] is unfolded (matricized) for treatment with their **MC** algorithm; thus, a tensor approach preserving the structure of the data could lead to better results.

- *Comparison with other methods based on hierarchical tensor models and on tensor rank.* In our numerical simulations, we have not compared our Tucker-based approaches with algorithms based on hierarchical tensor models, such as those of [55, 92]. Comparison experiments involving the recovery of real-world data tensors are therefore desirable in order to study which parsimony notion is better suited for each class of application. It would be equally interesting to perform a comparison between **IFHST** and a **TC** algorithm extending the method proposed in [210], given the similarity we have mentioned in Chapter 8.
- *Application to Volterra model identification.* The encouraging results obtained in [26] with the application of the Volterra-CPD model [79] to kernels derived from a loudspeaker model indicate that low-rank approximations are useful to model physical systems. Since the outputs of a homogeneous Volterra system can be seen as measurements of a Volterra kernel taken with a measurement operator induced by the applied input signal, an **LRTR**-based identification approach merits investigation. In particular, the identification of non-homogeneous Volterra models with this strategy involves significant difficulties, as it requires a joint recovery of low-rank tensors of different orders. A study of this problem could thus lead to a generalization of **LRTR**.

Conclusions et perspectives

Cette thèse a offert des contributions aux problèmes de calcul d'une décomposition canonique polyadique structurée et de récupération de tenseurs de rang faible. On présente dans ce chapitre nos conclusions globales et on souligne quelques perspectives de recherche.

Décomposition canonique polyadique structurée

Le Chapitre 4 a développé deux approches pour le problème de calcul d'une SCPD qui sont assez différentes. En plus de la simplicité et de la généralité de CALS, les itérées approchées que l'on a formulées contribuent à sa versatilité, permettant l'utilisation de stratégies mixtes pour réduire son coût de calcul. La solution algébrique pour les CPDs avec contrainte circulante vise un cas particulier, mais est capable d'exploiter complètement la structure du tenseur d'intérêt de façon non-itérative. En particulier, elle fournit une solution exacte en l'absence de bruit.

Les études numériques du Chapitre 5 attestent clairement de l'utilité de la variante CALS mixte qui emploie des itérées d'abord approchées puis exactes, en montrant que cette approche réduit le coût et évite des minima locaux lorsque l'initialisation est loin de l'optimum global. L'autre conclusion majeure de ce chapitre est la performance statistique assez satisfaisante de certains estimateurs SCPD qui combinent une méthode non-itérative fournissant une solution initiale approchée avec un algorithme itératif pour la raffiner. En particulier, des combinaisons de méthodes simples *ad-hoc* basées sur notre solution algébrique ou sur la méthode SBS avec CALS ont été capables de se rapprocher de la ECRB dans nos simulations, tout en gardant un coût plus faible par rapport à d'autres alternatives évaluées.

Perspectives

On énumère ci-dessous quelques thèmes de recherche concernant les décompositions tensorielles structurées qui méritent d'être étudiés davantage.

- *Généralisation de la solution algébrique.* Il serait intéressant de déterminer si la solution algébrique développée dans le Chapitre 4 peut être étendue à d'autres situations plus générales où des facteurs matriciels non-circulants sont aussi impliqués. Un exemple qui motive cette investigation est le problème d'identification Wiener-Hammerstein décrit dans le Chapitre 3.2, où l'un des facteurs est une matrice circulante multipliée à droite par une matrice diagonale. Des généralisations pour traiter des facteurs Toeplitz ou circulants en bloc paraissent aussi souhaitables, dû à leur importance dans certaines applications [57, 185].

- *Contraintes de non-négativité.* Les décompositions tensorielles non-négatives sont utiles dans beaucoup de problèmes, comme dans la fouille de données, la chimiométrie, l'imagerie hyperspectrale, la vision par ordinateur, l'ingénierie biomédicale et la séparation de sources audio. Cette large applicabilité a conduit plusieurs chercheurs à étudier ce sujet [177, 46, 214, 162]. L'introduction de contraintes structurelles de la forme considérée pourrait être d'intérêt pour certains de ces problèmes. Par exemple, on retrouve une CPD structurée non-négative dans [154].
- *Propriétés théoriques de la SCPD.* Compte tenu de l'intérêt pratique de la SCPD, une meilleure compréhension de ses propriétés théoriques est souhaitable. En particulier, une question importante est de déterminer si la recherche de la meilleure approximation de rang R d'un tenseur, qui est un problème mal posé dans le cas général, peut avoir une solution garantie lorsque l'on impose de la structure dans les facteurs, tels que des facteurs circulants en bande. Par ailleurs, la dérivation d'autres résultats d'unicité pour des structures particulières, en plus de ceux qui existent [53, 185, 184], peut être considérée.
- *D'autres modèles tensoriels structurés.* De façon plus générale, l'estimation de versions structurés d'autres modèles tensoriels peut être également envisagée. Par exemple, une décomposition tensorielle structurée dans des termes de rang- $(1, L, L)$, qui est plus générale que la CPD, se produit dans les problèmes de séparation de sources aveugle convolutive et de déconvolution aveugle [62, 185]. Le calcul de modèles tensoriels hiérarchiques [95] ayant des composants structurés semble aussi intéressant, bien qu'on ne puisse pas en donner des applications.

Récupération de tenseurs de rang faible

L'algorithme SeMPIHT que l'on a proposé dans le Chapitre 7 exploite les propriétés de la méthode approchée de meilleure approximation de rang multilinéaire (mrang) \mathbf{r} via des projections modales séquentiellement optimales, fournissant une alternative efficace au problème LRTR. On a présenté des simulations numériques systématiques qui suggèrent que les algorithmes IHT existants, y compris SeMPIHT, ont besoin d'un nombre optimal (en ordre de grandeur) de mesures Gaussiennes. Nous avons aussi dérivé des résultats théoriques qui garantissent la récupération de tenseurs de mrang faible. Cependant, malheureusement ces résultats s'appuient sur des conditions RIC largement sous-optimales. Nos résultats de simulations ont mis en évidence l'amélioration remarquable apportée par nos heuristiques de choix du pas et d'augmentation graduelle du mrang.

En dépit des mérites de SeMPIHT, il n'arrive pas à produire des résultats satisfaisants lorsque l'hypothèse de mrang faible est considérablement violée, c'est-à-dire lorsque le mrang

du tenseur d'intérêt est plein (ce qui arrive typiquement dans les applications) et ses spectres singuliers modaux ne décroissent pas assez rapidement. Afin d'aborder ce scénario moins favorable dans le cas de la reconstruction tensorielle, on a proposé dans le Chapitre 8 l'algorithme IFHST, qui couple un schéma d'imputation simple à une opération de seuillage doux dont le but est de promouvoir une solution qui ait une HOSVD avec cœur compressible. On a montré que cette propriété est liée à une décroissance rapide des spectres singuliers modaux du tenseur d'intérêt.

Le Chapitre 9 a illustré l'efficacité de la stratégie IFHST appliquée à des données réelles issues d'un contexte de reconstruction de données de trafic. Le schéma d'échantillonnage simule la défaillance systématique de quelques capteurs, ce qui cause l'occurrence de portions concentrées de données manquantes dans le tenseur observé. Malgré cela, IFHST produit une reconstruction précise des composantes inconnues. Bien qu'une approche basée sur la CPD non-négative ait eu une performance assez proche en ce qui concerne la qualité de reconstruction, elle a demandé un effort de calcul supérieur, ce qui montre que IFHST atteint un compromis plus favorable entre coût de calcul et précision.

Perspectives

Nous décrivons ci-après une liste de possibles extensions de nos contributions au problème LRTR.

- *Dérivation de résultats de récupération.* Une amélioration de la borne de performance (7.34) et de la condition RIC exploitée est d'intérêt. Ceci exige probablement l'emploi d'autres techniques de preuve pour profiter de la structure de rang faible complètement. Un écart théorique encore plus large existe dans le cas de la reconstruction tensorielle, où les résultats basés sur RIC ne s'appliquent pas. Le développement de garanties de récupération pour SeMPIHT dans ce cadre est probablement assez difficile, compte tenu de l'absence de résultats similaires pour des méthodes IHT dans le cadre LRMR. Ce n'est pas clair si une démonstration basée sur des conditions de cohérence pourrait être envisagée pour IFHST.
- *Amélioration de l'efficacité de calcul.* Le coût de calcul par itération de nos algorithmes est encore un facteur limitant quand on considère leur application à des tenseurs de très grandes dimensions. Dans le cas de SeMPIHT, les techniques randomisées présentées dans [97] pour le calcul approché d'une SVD tronquée peuvent réduire l'effort de calcul, mais possiblement au détriment de la vitesse de convergence. Quant à IFHST, on pourrait concevoir une manière d'exploiter la sparsité du cœur seuillé de la HOSVD lors des premières itérations pour améliorer l'efficacité.
- *Extension de l'approche basée sur le gradient conjugué.* Des algorithmes IHT employ-

ant les techniques du gradient conjugué afin d'accélérer la convergence ont été proposés dans [14] pour CS et MC. Il est alors raisonnable de supposer que cette stratégie pourrait aussi être bénéfique dans le cas LRTR. En particulier, une variante de SeMPIHT employant un schéma du gradient conjugué mériterait d'être étudiée.

- *Récupération conjointe de composants de rang faible et parcimonieuse.* Une formulation modélisant le tenseur d'intérêt comme une somme d'un tenseur de rang faible et d'un tenseur parcimonieux a été déjà proposée dans [106]. Cependant, elle s'appuie sur l'approche de minimisation de la somme des normes nucléaires, ce qui est sous-optimal [146]. Ainsi, une piste de recherche intéressante consiste à éteindre nos algorithmes afin de traiter ce problème. D'abord, cela peut fournir une méthode de récupération robuste par rapport aux données aberrantes (*outliers*). Puis, il s'agit d'une problématique intéressante pour certaines applications comme la séparation d'éléments de premier plan et d'arrière-plan d'une vidéo. En effet, le tenseur de données de vidéo utilisé dans les simulations de [211] est déplié pour que la méthode soit applicable; une approche tensorielle qui préserve la structure des données pourrait aboutir à de meilleurs résultats.
- *Comparaison à d'autres méthodes basées sur les modèles tensoriels hiérarchiques et sur le rang tensoriel.* Dans nos simulations numériques, nous n'avons pas comparé nos approches basées sur le modèle de Tucker aux algorithmes basés sur les modèles tensoriels hiérarchiques, tels que ceux de [55, 92]. Des comparaisons portant sur la récupération de tenseurs de données réelles sont alors souhaitables pour qu'on étudie quelle notion de parcimonie est mieux adaptée à chaque classe d'applications. Il serait également intéressant de réaliser une comparaison entre IFHST et un algorithme de TC fondé sur la méthode de [210], compte tenu de la similarité que l'on a évoquée dans le Chapitre 8.
- *Application à la modélisation de Volterra.* Les résultats encourageants obtenus dans [26] avec l'application du modèle Volterra-CPD [79] à des noyaux calculés à partir du modèle d'un haut-parleur indiquent que les approximations de rang faible sont utiles pour modéliser les systèmes physiques. Puisque les sorties d'un modèle de Volterra homogène peuvent être regardées comme des mesures d'un noyau de Volterra acquises par un opérateur de mesure induit par le signal d'entrée, une approche d'identification basée sur LRTR mérite d'être étudiée. En particulier, l'identification de modèles de Volterra non-homogènes à l'aide de cette stratégie entraîne des difficultés significatives, car elle exige une récupération conjointe de tenseurs de rang faible de différents ordres. Une étude de ce problème pourrait alors mener à une généralisation de LRTR.

Bibliography

- [1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations with missing data.” in *Proceedings of the International Conference on Data Mining*. SIAM, 2010, pp. 701–712. (Cited on page 121.)
- [2] E. Acar, T. G. Kolda, and D. M. Dunlavy, “An optimization approach for fitting canonical tensor decompositions,” *Sandia National Laboratories, Tech. Rep. SAND2009-0857*, 2009. (Cited on pages 2 and 8.)
- [3] E. Acar and B. Yener, “Unsupervised multiway data analysis: A literature survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 1, pp. 6–20, 2009. (Cited on pages 33 and 42.)
- [4] C. J. Appellof and E. R. Davidson, “Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents,” *Analytical Chemistry*, vol. 53, no. 13, pp. 2053–2056, 1981. (Cited on pages 1 and 7.)
- [5] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009. (Cited on page 25.)
- [6] B. W. Bader, M. W. Berry, and M. Browne, “Discussion tracking in enron email using PARAFAC,” in *Survey of Text Mining II*. Springer, 2008, pp. 147–163. (Cited on pages 1 and 7.)
- [7] R. Ballester-Ripoll, S. K. Suter, and R. Pajarola, “Analysis of tensor approximation for compression-domain volume visualization,” *Computers & Graphics*, vol. 47, pp. 34–47, 2015. (Cited on pages 1 and 7.)
- [8] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C.-G. Bénar, and I. Merlet, “EEG extended source localization: tensor-based vs. conventional methods,” *NeuroImage*, vol. 96, pp. 143–157, 2014. (Cited on pages 1 and 7.)
- [9] A. Belouchrani and B. Derras, “An efficient fourth order system identification (FOSI) algorithm utilizing the joint diagonalization procedure,” in *Proceedings of the IEEE Workshop on Statistical Signal and Array Processing*, 2000, pp. 621–625. (Cited on pages 2 and 8.)
- [10] Z. Ben-Haim and Y. C. Eldar, “A lower bound on the Bayesian MSE based on the optimal bias function,” *IEEE Transactions on Information Theory*, vol. 55, no. 11, pp. 5179–5196, 2009. (Cited on page 70.)

- [11] J. Bennett and S. Lanning, “The Netflix prize,” in *Proceedings of KDD cup and workshop*, vol. 2007, 2007, p. 35. (Cited on page 111.)
- [12] C. R. Berger, Z. Wang, J. Huang, and S. Zhou, “Application of compressive sensing to sparse channel estimation,” *IEEE Communications Magazine*, vol. 48, no. 11, pp. 164–174, 2010. (Cited on page 107.)
- [13] J. O. Berger, *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013. (Cited on page 70.)
- [14] J. D. Blanchard, J. Tanner, and K. Wei, “CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion,” *Information and Inference*, vol. 4, no. 4, pp. 289–327, 2015. (Cited on pages 186 and 191.)
- [15] G. Blekherman and Z. Teitler, “On maximum, typical and generic ranks,” *Mathematische Annalen*, vol. 362, no. 3-4, pp. 1021–1031, 2015. (Cited on pages 1 and 7.)
- [16] T. Blumensath and M. E. Davies, “Normalized iterative hard thresholding: Guaranteed stability and performance,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010. (Cited on pages 123, 134, 135 and 136.)
- [17] —, “Iterative thresholding for sparse approximations,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, 2008. (Cited on page 124.)
- [18] J. Bobin, J.-L. Starck, and R. Ottensamer, “Compressed sensing in astronomy,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 5, pp. 718–726, 2008. (Cited on page 107.)
- [19] M. Boizard, R. Boyer, G. Favier, and P. Comon, “Performance estimation for tensor CP decomposition with structured factors,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015. (Cited on pages xiii, 4, 10, 54 and 63.)
- [20] M. Boizard, J. H. de Morais Goulart, R. Boyer, G. Favier, and P. Comon, “Performance asymptotique d’estimateurs de modèles CP avec facteurs structurés,” in *XXVème colloque GRETSI (GRETSI 2015)*, Lyon, France, Sep. 2015. (Cited on pages 4, 5, 10 and 11.)
- [21] R. M. Bowen and C. C. Wang, *Introduction to Vectors and Tensors: Linear and multilinear algebra*. New York, NY: Plenum Press, 1976. (Cited on page 50.)
- [22] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations*

- and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011. (Cited on pages 110, 117, 143, 145 and 156.)
- [23] R. Boyer, “Deterministic asymptotic Cramér–Rao bound for the multidimensional harmonic model,” *Signal Processing*, vol. 88, no. 12, pp. 2869–2877, 2008. (Cited on page 96.)
- [24] R. Bro, “Multi-way analysis in the food industry: models, algorithms, and applications,” Ph.D. dissertation, Royal Veterinary and Agricultural University, Denmark, 1998. (Cited on pages 1, 7 and 121.)
- [25] —, “PARAFAC. tutorial and applications,” *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997. (Cited on pages 1, 7 and 33.)
- [26] P. M. S. Burt and J. H. de Morais Goulart, “Evaluating the potential of Volterra-PARAFAC IIR models,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 5745–5749. (Cited on pages 1, 7, 187 and 191.)
- [27] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995. (Cited on pages 97 and 99.)
- [28] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009. (Cited on pages 111, 112, 113 and 116.)
- [29] E. J. Candès and T. Tao, “The power of convex relaxation: Near-optimal matrix completion,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010. (Cited on pages 112 and 113.)
- [30] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006. (Cited on pages 107 and 108.)
- [31] E. J. Candès and T. Tao, “Decoding by linear programming,” *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005. (Cited on pages 107, 108 and 109.)
- [32] —, “Near-optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006. (Cited on pages 107 and 108.)

- [33] E. Candès and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007. (Cited on pages 108 and 109.)
- [34] E. J. Candès, “The restricted isometry property and its implications for compressed sensing,” *Comptes Rendus Mathématique*, vol. 346, no. 9–10, pp. 589–592, 2008. (Cited on pages 107 and 110.)
- [35] E. J. Candès and Y. Plan, “Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements,” *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011. (Cited on pages 111, 112 and 113.)
- [36] J.-F. Cardoso, “Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1990, pp. 2655–2658. (Cited on pages 1 and 7.)
- [37] ———, “Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1991, pp. 3109–3112. (Cited on pages 1 and 7.)
- [38] J.-F. Cardoso and P. Comon, “Tensor-based independent component analysis,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. IEEE, 1990, pp. 673–676. (Cited on pages 1 and 7.)
- [39] J. D. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart-Young” decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970. (Cited on pages 1, 2, 7, 8 and 32.)
- [40] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, “The convex geometry of linear inverse problems,” *Foundations of Computational mathematics*, vol. 12, no. 6, pp. 805–849, 2012. (Cited on pages 111, 112 and 114.)
- [41] Y. Chen, D. Han, and L. Qi, “New als methods with extrapolating search directions and optimal step size for complex-valued tensor decompositions,” *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5888–5898, 2011. (Cited on pages 2 and 8.)
- [42] E. Cheng, Y. Pang, Y. Zhu, J. Yu, and H. Ling, “Curvilinear structure tracking by low rank tensor approximation with model propagation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3057–3064. (Cited on pages 1 and 7.)

- [43] H. Cheng, Y. Yu, X. Zhang, E. Xing, and D. Schuurmans, “Scalable and sound low-rank tensor learning,” in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 1114–1123. (Cited on pages 1, 7, 121 and 122.)
- [44] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven, “On generic identifiability of symmetric tensors of subgeneric rank,” *arXiv preprint arXiv:1504.00547*, 2015. (Cited on pages 1 and 7.)
- [45] A. Cichocki, D. Mandic, L. De 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 Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015. (Cited on pages 1, 7, 33 and 42.)
- [46] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009. (Cited on pages 185 and 189.)
- [47] A. Cohen, W. Dahmen, and R. Devore, “Compressed sensing and best k -term approximation,” *Journal of the American Mathematical Society*, pp. 211–231, 2009. (Cited on pages 108 and 109.)
- [48] P. L. Combettes and J.-C. Pesquet, “Proximal splitting methods in signal processing,” in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212. (Cited on pages 110, 118 and 124.)
- [49] P. Comon, “Tensors: A brief introduction,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 44–53, 2014. (Cited on pages 15, 36, 37, 38 and 50.)
- [50] —, “Independent component analysis, a new concept?” *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994. (Cited on pages 1 and 7.)
- [51] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, “Symmetric tensors and symmetric tensor rank,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1254–1279, 2008. (Cited on page 35.)
- [52] P. Comon, X. Luciani, and A. L. F. De Almeida, “Tensor decompositions, alternating least squares and other tales,” *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 393–405, 2009. (Cited on pages 37 and 74.)
- [53] P. Comon and M. Sørensen, “Decomposing tensors with structured matrix factors reduces to rank-1 approximations,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 3858–3861. (Cited on pages 185 and 189.)

- [54] P. Comon, J. M. F. ten Berge, L. De Lathauwer, and J. Castaing, “Generic and typical ranks of multi-way arrays,” *Linear Algebra and its Applications*, vol. 430, no. 11-12, pp. 2997–3007, 2009. (Cited on pages 37 and 115.)
- [55] C. Da Silva and F. J. Herrmann, “Optimization on the hierarchical Tucker manifold—applications to tensor completion,” *Linear Algebra and its Applications*, vol. 481, pp. 131–173, 2015. (Cited on pages 122, 144, 187 and 191.)
- [56] J. Dauwels, K. Srinivasan, M. Ramasubba Reddy, and A. Cichocki, “Multi-channel eeg compression based on matrix and tensor decompositions,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 629–632. (Cited on pages 1, 7 and 34.)
- [57] A. L. de Almeida, G. Favier, and J. C. M. Mota, “PARAFAC-based unified tensor modeling for wireless communication systems with application to blind multiuser equalization,” *Signal Processing*, vol. 87, no. 2, pp. 337–351, 2007. (Cited on pages 1, 2, 7, 8, 185 and 188.)
- [58] L. De Lathauwer, B. De Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000. (Cited on page 43.)
- [59] —, “On the best rank-1 and rank- (R_1, R_2, \dots, R_n) approximation of higher-order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000. (Cited on pages 44, 45, 47 and 62.)
- [60] L. De Lathauwer, “Signal processing based on multilinear algebra,” Ph.D. dissertation, 1997. (Cited on pages 1 and 7.)
- [61] L. De Lathauwer, J. Castaing, and J.-F. Cardoso, “Fourth-order cumulant-based blind identification of underdetermined mixtures,” *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2965–2973, 2007. (Cited on page 33.)
- [62] L. De Lathauwer and A. de Baynast, “Blind deconvolution of DS-CDMA signals by means of decomposition in rank- $(1, L, L)$ terms,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1562–1571, 2008. (Cited on pages 185 and 189.)
- [63] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, “Statistical efficiency of structured CPD estimation applied to Wiener-Hammerstein modeling,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 948–952. (Cited on pages 4, 5, 10, 11 and 63.)

- [64] J. H. de Morais Goulart and G. Favier, “An algebraic solution for the Candecom/PARAFAC decomposition with circulant factors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 4, pp. 1543–1562, 2014. (Cited on pages 3, 4, 10, 11, 71, 74, 75 and 76.)
- [65] J. H. de Morais Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, “Tensor CP decomposition with structured factor matrices: Algorithms and performance,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 757–769, 2016. (Cited on pages xiii, 3, 4, 5, 10, 11, 54, 63, 67, 68, 70, 71, 72, 73 and 96.)
- [66] J. H. de Morais Goulart and G. Favier, “An iterative hard thresholding algorithm with improved convergence for low-rank tensor recovery,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. Nice, France: IEEE, Sep. 2015, pp. 1701–1705. (Cited on pages 4, 5, 10, 11, 124, 136, 142 and 143.)
- [67] —, “Iterative hard thresholding based algorithms for low-rank tensor recovery,” Symposium on Tensors for Signals and Systems at 20th Conference of the International Linear Algebra Society (ILAS), Leuven, Belgium, Jul. 2016. (Cited on pages 4, 5, 10 and 11.)
- [68] —, “Low-rank tensor recovery using sequentially optimal modal projections in iterative hard thresholding (SeMPIHT),” *SIAM Journal on Scientific Computing*, 2016, (submitted). (Cited on pages 4, 5, 10 and 12.)
- [69] V. De Silva and L.-H. Lim, “Tensor rank and the ill-posedness of the best low-rank approximation problem,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008. (Cited on pages 31, 38 and 58.)
- [70] C. C. De Wit, F. Morbidi, L. L. Ojeda, A. Y. Kibangou, I. Bellicot, and P. Bellemain, “Grenoble Traffic Lab: An experimental platform for advanced traffic monitoring and forecasting,” *IEEE Control Systems*, vol. 35, no. 3, pp. 23–39, 2015. (Cited on pages 168 and 170.)
- [71] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977. (Cited on pages 121 and 149.)
- [72] I. Domanov and L. De Lathauwer, “On the uniqueness of the canonical polyadic decomposition of third-order tensors—part i: Basic results and uniqueness of one factor matrix,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 855–875, 2013. (Cited on pages 1, 7 and 36.)

- [73] —, “On the uniqueness of the canonical polyadic decomposition of third-order tensors—part ii: Uniqueness of the overall decomposition,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 876–903, 2013. (Cited on pages 1, 7 and 36.)
- [74] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006. (Cited on pages 107 and 108.)
- [75] D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*. NJ, USA: Prentice Hall, 1984. (Cited on page 84.)
- [76] C. Eckart and G. Young, “The approximation of one matrix by another of lower rank,” *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936. (Cited on page 125.)
- [77] J. Eckstein, “Parallel alternating direction multiplier decomposition of convex programs,” *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 39–62, 1994. (Cited on page 156.)
- [78] G. Favier and A. Y. Kibangou, “Tensor-based methods for system identification. Part 2: Three examples of tensor-based system identification methods,” *International Journal on Sciences and Techniques of Automatic Control & Computer Engineering (IJ-STA)*, vol. 3, no. 1, 2009. (Cited on pages 1, 2, 7, 8, 55 and 56.)
- [79] G. Favier, A. Y. Kibangou, and T. Bouilloc, “Nonlinear system modeling and identification using Volterra-PARAFAC models,” *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 1, pp. 30–53, 2012. (Cited on pages 1, 7, 34, 74, 187 and 191.)
- [80] G. Favier and A. L. de Almeida, “Overview of constrained PARAFAC models,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–25, 2014. (Cited on page 50.)
- [81] —, “Tensor space-time-frequency coding with semi-blind receivers for MIMO wireless communication systems,” *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5987–6002, 2014. (Cited on pages 1 and 7.)
- [82] M. Fazel, “Matrix rank minimization with applications,” Ph.D. dissertation, Department of Electrical Engineering, Stanford University, 2002. (Cited on pages 110 and 111.)
- [83] C. A. R. Fernandes, G. Favier, and J. C. M. Mota, “PARAFAC-based channel estimation and data recovery in nonlinear MIMO spread spectrum communication

- systems,” *Signal Processing*, vol. 91, no. 2, pp. 311–322, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168410003105> (Cited on pages 1, 7 and 33.)
- [84] C. E. R. Fernandes, G. Favier, and J. C. M. Mota, “Blind channel identification algorithms based on the Parafac decomposition of cumulant tensors: the single and multiuser cases,” *Signal Processing*, vol. 88, no. 6, pp. 1382–1401, 2008. (Cited on pages 1, 2, 7, 8, 33, 59 and 74.)
- [85] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, ser. Applied and Numerical Harmonic Analysis. Birkhauser Verlag GmbH, 2013. (Cited on pages 108, 109, 110, 125, 130, 150 and 151.)
- [86] M. Frank and P. Wolfe, “An algorithm for quadratic programming,” *Naval research logistics quarterly*, vol. 3, no. 1-2, pp. 95–110, 1956. (Cited on page 121.)
- [87] S. Friedland and L.-H. Lim, “Nuclear norm of higher-order tensors,” *arXiv preprint arXiv:1410.6072v3*, 2014. (Cited on pages 1, 7, 25 and 114.)
- [88] S. Gandy, B. Recht, and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” *Inv. Prob.*, vol. 27, no. 2, p. 025010, 2011. (Cited on pages 3, 9 and 117.)
- [89] D. Goldfarb and S. Ma, “Convergence of fixed-point continuation algorithms for matrix rank minimization,” *Foundations of Computational Mathematics*, vol. 11, no. 2, pp. 183–210, 2011. (Cited on page 131.)
- [90] G. H. Golub and C. F. Van Loan, *Matrix Computations*, ser. Matrix Computations. Johns Hopkins University Press, 2012. (Cited on pages 45, 58 and 73.)
- [91] L. Grasedyck, D. Kressner, and C. Tobler, “A literature survey of low-rank tensor approximation techniques,” *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013. (Cited on page 47.)
- [92] L. Grasedyck, M. Kluge, and S. Kramer, “Variants of alternating least squares tensor completion in the tensor train format,” *SIAM Journal on Scientific Computing*, vol. 37, no. 5, pp. A2424–A2450, 2015. (Cited on pages 122, 138, 187 and 191.)
- [93] R. M. Gray, “Toeplitz and circulant matrices: A review,” *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 3, pp. 155–239, 2006. (Cited on page 77.)

- [94] R. Haber and L. Keviczky, *Nonlinear system identification – Input-Output Modeling Approach*, ser. Mathematical Modelling: Theory and Applications. Kluwer Academic Publishers, 1999, vol. 1. (Cited on page 55.)
- [95] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, ser. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2012. (Cited on pages 1, 7, 14, 15, 16, 17, 24, 31, 36, 37, 38, 39, 44, 50, 122, 185 and 189.)
- [96] W. Hackbusch, D. Kressner, and A. Uschmajew, “Perturbation of higher-order singular values,” Jul. 2016, iNS Preprint No. 1616. (Cited on pages 1 and 7.)
- [97] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011. (Cited on pages 128, 186 and 190.)
- [98] F. L. Hall, “Traffic stream characteristics,” in *Traffic flow theory*, N. Gartner, C. J. Messer, and A. K. Rathi, Eds. US Federal Highway Administration, 1992, ch. 2. (Cited on page 169.)
- [99] Z. Harchaoui, A. Juditsky, and A. Nemirovski, “Conditional gradient algorithms for norm-regularized smooth convex optimization,” *Mathematical Programming*, vol. 152, no. 1-2, pp. 75–112, 2015. (Cited on page 121.)
- [100] R. A. Harshman, “Foundations of the PARAFAC procedure: models and conditions for an “explanatory” multi-modal factor analysis,” *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970. (Cited on pages 1, 2, 7, 8 and 32.)
- [101] M. A. Herman and T. Strohmer, “High-resolution radar via compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009. (Cited on page 107.)
- [102] C. J. Hillar and L.-H. Lim, “Most tensor problems are NP-hard,” *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 45:1–45:39, 2013. (Cited on pages 2, 8, 25, 38 and 45.)
- [103] F. L. Hitchcock, “Multiple invariants and generalized rank of a p-way matrix or tensor,” *Journal of Mathematics and Physics*, vol. 7, no. 1, pp. 39–79, 1928. (Cited on pages 39 and 40.)
- [104] F. L. Hitchcock, “The expression of a tensor or a polyadic as a sum of products,” *Journal of Mathematics and Physics*, vol. 6, no. 1–4, pp. 164–189, 1927. (Cited on pages 2, 8, 31, 32 and 39.)

- [105] J. E. Hopcroft and L. R. Kerr, “On minimizing the number of multiplications necessary for matrix multiplication,” *SIAM Journal on Applied Mathematics*, vol. 20, no. 1, pp. 30–36, 1971. (Cited on page 34.)
- [106] B. Huang, C. Mu, D. Goldfarb, and J. Wright, “Provable low-rank tensor recovery,” *Optimization-Online*, vol. 4252, 2014. (Cited on pages 116, 119, 187 and 191.)
- [107] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, “Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme,” *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 1, pp. 115–135, 2011. (Cited on pages 45 and 47.)
- [108] M. Ishteva, “Numerical methods for the best low multilinear rank approximation of higher-order tensors,” Ph.D. dissertation, Department of Electrical Engineering, Katholieke Universiteit Leuven, 2009. (Cited on page 45.)
- [109] P. Jain, R. Meka, and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” in *Advances in Neural Information Processing Systems*, 2010, pp. 937–945. (Cited on page 123.)
- [110] B. Jiang, F. Yang, and S. Zhang, “Tensor and its Tucker core: the invariance relationships,” *arXiv preprint arXiv:1601.01469*, 2016. (Cited on pages 42 and 43.)
- [111] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, “Adaptive algorithms for sparse system identification,” *Signal Processing*, vol. 91, no. 8, pp. 1910–1919, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168411000697> (Cited on page 107.)
- [112] H. Kasai and B. Mishra, “Riemannian preconditioning for tensor completion,” *arXiv preprint arXiv:1506.02159*, 2015. (Cited on page 122.)
- [113] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, ser. Fundamentals of Statistical Signal Processing. PTR Prentice-Hall, 1993. (Cited on pages 63, 64 and 97.)
- [114] A. Y. Kibangou and G. Favier, “Wiener-Hammerstein systems modeling using diagonal Volterra kernels coefficients,” *IEEE Signal Processing Letters*, vol. 13, no. 6, pp. 381–384, 2006. (Cited on pages 1, 7 and 55.)
- [115] —, “Identification of fifth-order Volterra systems using i.i.d. inputs,” *IET signal processing*, vol. 4, no. 1, pp. 30–44, 2010. (Cited on page 56.)

- [116] —, “Non-iterative solution for parafac with a toeplitz matrix factor,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. IEEE, 2009, pp. 691–695. (Cited on pages [2](#), [8](#), [59](#), [60](#) and [61](#).)
- [117] —, “Tensor analysis-based model structure determination and parameter estimation for block-oriented nonlinear systems,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 3, pp. 514–525, 2010. (Cited on page [56](#).)
- [118] H. A. Kiers, “Weighted least squares fitting using ordinary least squares algorithms,” *Psychometrika*, vol. 62, no. 2, pp. 251–266, 1997. (Cited on page [121](#).)
- [119] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. (Cited on pages [22](#), [36](#) and [50](#).)
- [120] T. G. Kolda and J. Sun, “Scalable tensor decompositions for multi-aspect data mining,” in *Proceedings of the 8th IEEE international conference on data mining*, 2008, pp. 363–372. (Cited on pages [1](#) and [7](#).)
- [121] N. Kreimer and M. D. Sacchi, “A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation,” *Geophysics*, vol. 77, no. 3, pp. V113–V122, 2012. (Cited on page [149](#).)
- [122] N. Kreimer, A. Stanton, and M. D. Sacchi, “Tensor completion based on nuclear norm minimization for 5d seismic data reconstruction,” *Geophysics*, vol. 78, no. 6, pp. V273–V284, 2013. (Cited on pages [3](#) and [9](#).)
- [123] D. Kressner, M. Steinlechner, and B. Vandereycken, “Low-rank tensor completion by Riemannian optimization,” *BIT Numerical Mathematics*, pp. 1–22, 2013. (Cited on pages [115](#), [122](#), [137](#), [138](#), [143](#), [145](#) and [146](#).)
- [124] J. B. Kruskal, “Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics,” *Linear Algebra and its Applications*, vol. 18, no. 2, pp. 95–138, 1977. (Cited on pages [35](#) and [36](#).)
- [125] A. Kyrillidis and V. Cevher, “Matrix recipes for hard thresholding methods,” *Journal of mathematical imaging and vision*, vol. 48, no. 2, pp. 235–265, 2014. (Cited on pages [126](#) and [134](#).)
- [126] A. J. Laub, *Matrix analysis for scientists and engineers*. Siam, 2005. (Cited on page [26](#).)
- [127] J. Lebrun, “Normal forms in statistical signal processing,” in *Gröbner Bases in Control Theory and Signal Processing*. de Gruyter, 2007, pp. 107–126. (Cited on page [84](#).)

- [128] E. L. Lehmann and G. Casella, “Theory of point estimation,” 1998. (Cited on page 64.)
- [129] L. Leon Ojeda, “Short-term multi-step ahead traffic forecasting,” Ph.D. dissertation, GIPSA-lab, Département d’Automatique, Université de Grenoble, Grenoble, France, 2014. (Cited on pages 167 and 168.)
- [130] S. Leurgans and R. T. Ross, “Multilinear models: applications in spectroscopy,” *Statistical Science*, pp. 289–310, 1992. (Cited on pages 1 and 7.)
- [131] L.-H. Lim, “Tensors and hypermatrices,” in *Handbook of Linear Algebra*, 2nd ed. Boca Raton, FL: CRC Press, 2013, pp. 231–260. (Cited on pages 14, 17, 18, 20, 24, 25, 32, 38 and 50.)
- [132] L.-H. Lim and P. Comon, “Nonnegative approximations of nonnegative tensors,” *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 432–441, 2009. (Cited on page 121.)
- [133] ———, “Blind multilinear identification,” *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1260–1280, 2014. (Cited on pages 25, 31, 38, 50, 58 and 114.)
- [134] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013. (Cited on pages 3, 9, 117 and 156.)
- [135] ———, “Tensor completion for estimating missing values in visual data,” in *Proceedings of the IEEE International Conference on Computer Vision*, Sept 2009, pp. 2114–2121. (Cited on pages 1, 7 and 117.)
- [136] X. Liu and N. D. Sidiropoulos, “Cramér-Rao lower bounds for low-rank decomposition of multidimensional arrays,” *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 2074–2086, 2001. (Cited on pages 57, 96 and 97.)
- [137] L. S. Lorente, J. M. Vega, and A. Velazquez, “Compression of aerodynamic databases using high-order singular value decomposition,” *Aerospace Science and Technology*, vol. 14, no. 3, pp. 168–177, 2010. (Cited on page 42.)
- [138] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, “Compressed sensing MRI,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, 2008. (Cited on page 107.)
- [139] S. Ma, D. Goldfarb, and L. Chen, “Fixed point and Bregman iterative methods for matrix rank minimization,” *Mathematical Programming*, vol. 128, no. 1-2, pp. 321–353, 2011. (Cited on page 176.)

- [140] A. G. Mahyari, D. Zoltowski, E. Bernat, and S. Aviyente, “A tensor decomposition based approach for detecting dynamic network states from EEG,” *IEEE Transactions on Biomedical Engineering*, 2016. (Cited on pages 1 and 7.)
- [141] V. Z. Marmarelis, *Nonlinear dynamic modeling of physiological systems*. John Wiley & Sons, 2004, vol. 10. (Cited on page 42.)
- [142] V. Z. Marmarelis, D. C. Shin, D. Song, R. E. Hampson, S. A. Deadwyler, and T. W. Berger, “Nonlinear modeling of dynamic interactions within neuronal ensembles using principal dynamic modes,” *Journal of computational neuroscience*, vol. 34, no. 1, pp. 73–87, 2013. (Cited on page 42.)
- [143] D. W. Marquardt, “An algorithm for least squares estimation of non-linear parameters,” *SIAM Journal*, vol. 11, no. 2, pp. 431–441, 1963. (Cited on page 121.)
- [144] B. C. Mitchell and D. S. Burdick, “Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies,” *Journal of Chemometrics*, vol. 8, no. 2, pp. 155–168, 1994. (Cited on pages 2, 8 and 59.)
- [145] M. Mørup, “Applications of tensor (multiway array) factorizations and decompositions in data mining,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 24–40, 2011. (Cited on pages 1, 7, 33 and 42.)
- [146] C. Mu, B. Huang, J. Wright, and D. Goldfarb, “Square deal: Lower bounds and improved relaxations for tensor recovery,” in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014, pp. 73–81. (Cited on pages 115, 119, 122, 187 and 191.)
- [147] D. Muti and S. Bourenmane, “Multidimensional filtering based on a tensor approach,” *Signal Processing*, vol. 85, no. 12, pp. 2338–2353, 2005. (Cited on pages 1 and 7.)
- [148] H. Neudecker, “Some theorems on matrix differentiation with special reference to kronecker matrix products,” *Journal of the American Statistical Association*, vol. 64, no. 327, pp. 953–963, 1969. (Cited on page 26.)
- [149] Q. Nguyen, A. Gautier, and M. Hein, “A flexible tensor block coordinate ascent scheme for hypergraph matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5270–5278. (Cited on pages 1 and 7.)
- [150] D. Nion and L. D. Lathauwer, “A block component model-based blind DS-CDMA receiver,” *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5567–5579, 2008. (Cited on page 59.)

- [151] D. Nion, K. N. Mokios, N. D. Sidiropoulos, and A. Potamianos, “Batch and adaptive parafac-based blind separation of convolutive speech mixtures,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1193–1207, 2010. (Cited on page 33.)
- [152] J. Nocedal and S. J. Wright, *Numerical Optimization*, ser. Springer series in operations research and financial engineering. Springer, 1999. (Cited on pages 97 and 117.)
- [153] L. Oeding and G. Ottaviani, “Eigenvectors of tensors and algorithms for waring decomposition,” *Journal of Symbolic Computation*, vol. 54, pp. 9–35, 2013. (Cited on pages 1 and 7.)
- [154] A. Ozerov, C. Févotte, R. Blouet, and J.-L. Durrieu, “Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 257–260. (Cited on pages 1, 7, 185 and 189.)
- [155] P. Paatero, “Construction and analysis of degenerate PARAFAC models,” *Journal of Chemometrics*, vol. 14, no. 3, pp. 285–299, 2000. (Cited on pages 2, 8, 58 and 59.)
- [156] E. E. Papalexakis, T. M. Mitchell, N. D. Sidiropoulos, C. Faloutsos, P. P. Talukdar, and B. Murphy, “Turbo-SMT: Parallel coupled sparse matrix-tensor factorizations and applications,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 9, no. 4, pp. 269–290, 2016. (Cited on pages 1 and 7.)
- [157] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 123–231, 2013. (Cited on page 159.)
- [158] A. H. Phan and A. Cichocki, “Tensor decompositions for feature extraction and classification of high dimensional datasets,” *IEICE Nonlinear theory and its applications*, vol. 1, no. 1, pp. 37–68, 2010. (Cited on page 42.)
- [159] —, “PARAFAC algorithms for large-scale problems,” *Neurocomputing*, vol. 74, no. 11, pp. 1970–1984, 2011. (Cited on page 57.)
- [160] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, “Sparsity and compressed sensing in radar imaging,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1006–1020, 2010. (Cited on page 107.)
- [161] J. Provost and F. Lesage, “The application of compressed sensing for photo-acoustic tomography,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 4, pp. 585–594, 2009. (Cited on page 107.)

- [162] Y. Qi, P. Comon, and L.-H. Lim, “Uniqueness of nonnegative tensor approximations,” *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 2170–2183, 2016. (Cited on pages 1, 7, 185 and 189.)
- [163] B. Ran, H. Tan, J. Feng, Y. Liu, and W. Wang, “Traffic speed data imputation method based on tensor completion,” *Computational intelligence and neuroscience*, vol. 2015, p. 22, 2015. (Cited on pages 3, 9, 168 and 176.)
- [164] B. Ran, H. Tan, Y. Wu, and P. J. Jin, “Tensor based missing traffic data completion with spatial–temporal correlation,” *Physica A: Statistical Mechanics and its Applications*, vol. 446, pp. 54–63, 2016. (Cited on pages 3, 9, 168 and 176.)
- [165] H. Rauhut, R. Schneider, and Z. Stojanac, “Low rank tensor recovery via iterative hard thresholding,” in *Proceedings of the 10th International Conference on Sampling Theory and Applications*, 2013. (Cited on pages 116, 123 and 126.)
- [166] H. Rauhut, “Compressive sensing and structured random matrices,” in *Theoretical foundations and numerical methods for sparse recovery*, ser. Radon Series Comp. Appl. Math., M. Fornasier, Ed. de Gruyter, 2010, vol. 9, pp. 1–92. (Cited on pages 107 and 109.)
- [167] H. Rauhut, R. Schneider, and Z. Stojanac, “Low rank tensor recovery via iterative hard thresholding,” *CoRR*, vol. abs/1602.05217, 2016. [Online]. Available: <http://arxiv.org/abs/1602.05217> (Cited on pages 1, 7, 116, 126, 127 and 133.)
- [168] H. Rauhut and Ž. Stojanac, “Tensor theta norms and low rank recovery,” *arXiv preprint arXiv:1505.05175*, 2015. (Cited on page 114.)
- [169] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010. (Cited on pages 110 and 113.)
- [170] L. N. Ribeiro, A. L. De Almeida, and V. Zarzoso, “Enhanced block term decomposition for atrial activity extraction in atrial fibrillation ECG,” in *Proceedings of the 9th IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2016. (Cited on pages 1 and 7.)
- [171] S. F. Roohi, D. Zonoobi, A. A. Kassim, and J. L. Jaremko, “Dynamic MRI reconstruction using low rank plus sparse tensor decomposition,” in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1769–1773. (Cited on pages 1 and 7.)

- [172] M. Rudelson and R. Vershynin, “On sparse reconstruction from Fourier and Gaussian measurements,” *Communications on Pure and Applied Mathematics*, vol. 61, no. 8, pp. 1025–1045, 2008. (Cited on pages 109 and 134.)
- [173] S. Sahnoun and P. Comon, “Tensor polyadic decomposition for antenna array processing,” in *Proceedings of the International Conference on Computational Statistics (Compstat)*, Geneva, Switzerland, 2014, pp. 233–240. (Cited on page 96.)
- [174] —, “Joint source estimation and localization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 10, pp. 2485–2495, 2015. (Cited on pages 1 and 7.)
- [175] B. Savas and L. Eldén, “Handwritten digit classification using higher order singular value decomposition,” *Pattern recognition*, vol. 40, no. 3, pp. 993–1003, 2007. (Cited on page 42.)
- [176] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley & Sons, 1980. (Cited on page 34.)
- [177] A. Shashua and T. Hazan, “Non-negative tensor factorization with applications to statistics and computer vision,” in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 792–799. (Cited on pages 1, 7, 185 and 189.)
- [178] N. D. Sidiropoulos and R. Bro, “On the uniqueness of multilinear decomposition of N-way arrays,” *Journal of Chemometrics*, vol. 14, no. 3, pp. 229–239, 2000. (Cited on page 36.)
- [179] N. D. Sidiropoulos, G. B. Giannakis, and R. Bro, “Blind PARAFAC receivers for DS-CDMA systems,” *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 810–823, 2000. (Cited on pages 1, 7 and 33.)
- [180] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, “Parallel factor analysis in sensor array processing,” *IEEE transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000. (Cited on pages 1 and 7.)
- [181] M. Signoretto, R. Van de Plas, B. De Moor, and J. A. K. Suykens, “Tensor versus matrix completion: A comparison with application to spectral data,” *IEEE Signal Processing Letters*, vol. 18, no. 7, pp. 403–406, 2011. (Cited on pages 3, 9 and 117.)
- [182] A. D. Silva, P. Comon, and A. de Almeida, “A finite algorithm to compute rank-1 tensor approximations,” *IEEE Signal Processing Letters*, vol. 23, no. 7, pp. 959–963, 2016. (Cited on page 121.)

- [183] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*, ser. Wiley InterScience online books. Wiley, 2004. (Cited on page 33.)
- [184] M. Sorensen and P. Comon, “Tensor decompositions with banded matrix factors,” *Linear Algebra and its Applications*, vol. 438, no. 2, pp. 919–941, 2013. (Cited on pages 2, 8, 52, 61, 62, 63, 185 and 189.)
- [185] M. Sørensen and L. De Lathauwer, “Tensor decompositions with block-Toeplitz structure and applications in signal processing,” in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2011, pp. 454–458. (Cited on pages 185, 188 and 189.)
- [186] —, “Tensor decompositions with vandermonde factor and applications in signal processing,” in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2012, pp. 890–894. (Cited on pages 2 and 8.)
- [187] —, “Blind signal separation via tensor decomposition with vandermonde factor: Canonical polyadic decomposition,” *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5507–5519, 2013. (Cited on pages 2 and 8.)
- [188] A. Stegeman, “Low-rank approximation of generic $p \times q \times 2$ arrays and diverging components in the Candecomp/Parafac model,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 988–1007, 2008. (Cited on page 58.)
- [189] P. Stoica and R. L. Moses, *Spectral analysis of signals*. Pearson/Prentice Hall Upper Saddle River, NJ, 2005. (Cited on page 64.)
- [190] V. Strassen, “Gaussian elimination is not optimal,” *Numerische Mathematik*, vol. 13, no. 4, pp. 354–356, 1969. (Cited on page 34.)
- [191] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, “A tensor-based method for missing traffic data completion,” *Transportation Research Part C: Emerging Technologies*, vol. 28, pp. 15–27, 2013. (Cited on pages 3, 9, 168 and 176.)
- [192] J. Tanner and K. Wei, “Normalized iterative hard thresholding for matrix completion,” *SIAM Journal on Scientific Computing*, vol. 35, no. 5, pp. S104–S125, 2013. (Cited on pages 123, 134 and 135.)
- [193] G. Tomasi and R. Bro, “A comparison of algorithms for fitting the PARAFAC model,” *Computational Statistics and Data Analysis*, vol. 50, no. 7, pp. 1700–1734, 2006. (Cited on pages 2, 8 and 57.)

- [194] —, “PARAFAC and missing values,” *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 163–180, 2005. (Cited on pages 121, 148 and 149.)
- [195] R. Tomioka, K. Hayashi, and H. Kashima, “Estimation of low-rank tensors via convex optimization,” *arXiv:1010.0789*, 2010. (Cited on pages 117, 118, 119 and 176.)
- [196] —, “On the extension of trace norm to tensors,” in *NIPS Workshop on Tensors, Kernels, and Machine Learning*, 2010, p. 7. (Cited on page 117.)
- [197] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima, “Statistical performance of convex tensor decomposition,” in *Advances in Neural Information Processing Systems*, Dec. 2011, pp. 972–980. (Cited on pages 117 and 119.)
- [198] L. Tong, “Identification of multichannel MA parameters using higher-order statistics,” *Signal Processing*, vol. 53, no. 2, pp. 195–209, 1996. (Cited on pages 2 and 8.)
- [199] C.-H. Tseng and E. J. Powers, “Identification of cubic systems using higher order moments of IID signals,” *IEEE Transactions on Signal Processing*, vol. 43, no. 7, pp. 1733–1735, 1995. (Cited on page 56.)
- [200] L. R. Tucker, “Implications of factor analysis of three-way matrices for measurement of change,” in *Problems in measuring change.*, C. W. Harris, Ed. Madison WI: University of Wisconsin Press, 1963, pp. 122–137. (Cited on pages 1 and 7.)
- [201] —, “The extension of factor analysis to three-dimensional matrices,” in *Contributions to mathematical psychology.*, H. Gulliksen and N. Frederiksen, Eds. New York: Holt, Rinehart and Winston, 1964, pp. 110–127. (Cited on pages 1 and 7.)
- [202] —, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966. (Cited on pages 1, 7 and 40.)
- [203] A. Uschmajew, “Local convergence of the alternating least squares algorithm for canonical tensor approximation,” *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 2, pp. 639–652, 2012. (Cited on pages 57 and 58.)
- [204] —, “A new convergence proof for the higher-order power method and generalizations,” *arXiv preprint arXiv:1407.4586*, 2014. (Cited on pages 57 and 58.)
- [205] A. Uschmajew and B. Vandereycken, “The geometry of algorithms using hierarchical tensors,” *Linear Algebra and its Applications*, vol. 439, no. 1, pp. 133–166, 2013. (Cited on page 122.)

- [206] A.-J. Van Der Veen and A. Paulraj, “An analytical constant modulus algorithm,” *IEEE Transactions on Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996. (Cited on pages 2 and 8.)
- [207] H. L. Van Trees and K. L. Bell, *Bayesian Bounds for Parameter Estimation and Non-linear Filtering/Tracking*. Wiley, 2007. (Cited on page 70.)
- [208] B. Vandereycken, “Low-rank matrix completion by Riemannian optimization,” *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013. (Cited on page 111.)
- [209] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, “A new truncation strategy for the higher-order singular value decomposition,” *SIAM Journal on Scientific Computing*, vol. 34, no. 2, pp. A1027–A1052, 2012. (Cited on pages 47, 48, 49 and 128.)
- [210] X. Wang and C. Navasca, “Low rank approximation of tensors via sparse optimization,” *arXiv preprint arXiv:1504.05273*, 2015. (Cited on pages 155, 187 and 191.)
- [211] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk, “SpaRCS: Recovering low-rank and sparse matrices from compressive measurements,” in *Advances in neural information processing systems*, 2011, pp. 1089–1097. (Cited on pages 187 and 191.)
- [212] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012. (Cited on page 120.)
- [213] Y. Xu, R. Hao, W. Yin, and Z. Su, “Parallel matrix factorization for low-rank tensor completion,” *Inverse Problems & Imaging*, vol. 9, no. 2, pp. 601–624, 2015. (Cited on pages 3, 9, 120, 138, 143, 145, 146 and 164.)
- [214] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013. (Cited on pages 121, 176, 185 and 189.)
- [215] Y. Yang, Y. Feng, and J. Suykens, “A rank-one tensor updating algorithm for tensor completion,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1633–1637, 2015. (Cited on page 121.)
- [216] M. Yuan and C.-H. Zhang, “On tensor completion via nuclear norm minimization,” *Foundations of Computational Mathematics*, pp. 1–38, 2014. (Cited on page 114.)
- [217] M. Zhang, L. Yang, and Z.-H. Huang, “Minimum n -rank approximation via iterative hard thresholding,” *Applied Mathematics and Computation*, vol. 256, pp. 860 – 875, 2015. (Cited on pages 123, 127, 129 and 143.)

- [218] T. Zhang and G. H. Golub, “Rank-one approximation to high order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 2, pp. 534–550, 2001. (Cited on pages 57, 58 and 62.)
- [219] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, “L-BFGS-B: FORTRAN subroutines for large scale bound constrained optimization,” EECS Department, Northwestern University, Tech. Rep. NAM-11, 1994. (Cited on pages 97 and 99.)